Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Badji Mokhtar - Annaba
Faculté des Sciences de l'Ingénieur
Département d'Informatique

COSI'2009

Annaba 25 - 27 Mai 2009

PROCEEDINGS

6 ème Colloque sur l'Optimisation et les Systèmes d'Information

Avec la collaboration des laboratoires:

Laboratoire Analyse Numérique, Optimisation et Statistique (LANOS) - Laboratoire Etude et Recherche en Instrumentation et Communication (LEURICA) - Laboratoire génie civil - Laboratoire mécanique industrielle - Laboratoire mécanique des matériaux et maintenance industrielle - Laboratoire physique des rayonnements - Laboratoire architecture et urbanisme - Laboratoire ressources naturelles et aménagement.

-

Actes du Sixième Colloque sur l'Optimisation et les Systèmes d'Information - COSI'2009

25-27 Mai 2009, Annaba, Algérie

Université Badji Mokhtar, Annaba
Faculté des sciences de l'ingénieur
Département d'Informatique

Table des matières

Préface

Ce volume contient les actes de la sixième édition du Colloque sur l'Optimisation et les Systèmes d'Information (COSI'2009), qui a eu lieu à Annaba, Algérie, du 25 au 27 Mai 2009. Il inclut les résumés de 4 conférences plénières ainsi que 44 articles sélectionnés parmi 203 soumissions (soit un taux d'acceptation inférieur à 22%). L'évaluation n'étant pas une science exacte, le comité de programme a probablement rejeté des articles qu'il aurait dû accepter. Nous espérons que les auteurs trouveront dans les rapports des évaluateurs matière à améliorer leurs travaux.

COSI'2009 perpétue la tradition des éditions précédentes en se voulant une manifestation scientifique de qualité et un lieu de rencontre et d'échange à la fois scientifique et humain. Cette année, le programme scientifique comporte 4 conférences plénières qui couvrent des thématiques au cœur de COSI, à savoir : la création et la gestion des ontologies, l'algorithmique d'énumération, l'optimisation parallèle et les mathématiques financières et la modélisation des procédés logiciels.

De la même manière qu'en 2007, l'édition COSI'2009 sera précédée d'une école d'été de deux jours portant sur la thématique générale de l'aide à la décision. La qualité des cours dispensés ainsi que l'engouement des jeunes chercheurs pour cette école sont des preuves indéniables de son intérêt.

Nous souhaitons remercier très vivement les nombreux artisans de cet événement. Nous remercions tout d'abord les auteurs des soumissions ainsi que les membres du comité de programme qui ont fait un travail remarquable. L'organisation d'un tel événement nécessite un effort considérable. Nous sommes très reconnaissant au Recteur de l'Université Badji Mokhtar d'Annaba, le Professeur Mohamed Tayeb Laskri, d'avoir accepté et soutenu l'organisation de ce colloque au sein de son Université. Nos remerciements les plus vifs vont aux membres du comité d'organisation, présidé par Djamel Essolh Amrane et Dr. Bornia Tighiouart pour leur disponibilité, leur constante bonne humeur, et la grande qualité de leur travail.

N'oublions pas que l'organisation de ce colloque n'aurait pu se faire sans l'engagement de la faculté des sciences de l'ingénieur et la collaboration précieuse de plusieurs laboratoires de recherches de l'université Badji Mokhtar. Qu'ils reçoivent ici notre profonde gratitude.

Enfin, nous ne pouvons terminer cette préface sans un remerciement particulier aux membres du comité de pilotage de COSI qui veillent dans l'ombre à la pérennité de ce colloque.

Pr. Farouk Toumani

Président du comité de programme

Organisation

Université Badji Mokhtar, Annaba, Algérie

Président d'honneur

Professeur Mohamed Tayeb Laskri
Recteur de Université Badji Mokhtar, Annaba, Algérie

Comité d'Organisation

Présidents

Djamel Essolh Amrane, Université Badji Mokhtar, Annaba
Bornia Tighiouart, Université Badji Mokhtar, Annaba

Membres

H. Baaziz, Université Badji Mokhtar, Annaba
T. Bensbaa, Université Badji Mokhtar, Annaba
A. Benouareth, Université Badji Mokhtar, Annaba
Y. Djemmam, Université Badji Mokhtar, Annaba
A. Kermi, Université Badji Mokhtar, Annaba
T. Sari, Université Badji Mokhtar, Annaba

Comité de Pilotage

Mohamed Aidene, Université Mouloud Mammeri de Tizi-Ouzou, Algérie
Nacéra Benamrane, Université des Sciences et Technologie d'Oran, Algérie
Abdelhafidh Berrachedi, Université des Sciences et Technologie Houari Boumédienne, Alger, Algérie
Mohand-Saïd Hacid, Université de Lyon I, France
Lhouari Nourine, Université de Clermont-Ferrand II, France
Brahim Oukacha, Université de Tizi-Ouzou, Algérie
Jean Marc Petit, INSA de Lyon, France
Bachir Sadi, Université de Tizi-Ouzou, Algérie
Lakhdar Saïs, CRIL - CNRS, Université d'Artois, France
Kamel Tari, Université Abderahmane Mira de Bejaia, Algérie

Comité de Programme

Président

Farouk Toumani, LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand

Co-présidents

Meziane Aider, USTHB, Alger
Lakhdar Saïs , CRIL - CNRS, Université d'Artois, France

Membres

Agier Marie LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Ahmed-Ouamer Rachid UMMTO, Tiszi-Ouzou
Aidene Mohamed Université de Tizi-Ouzou
Aider M. Université des Sciences et de la Technologie HOUARI BOUMEDIENE, Alger
Ait haddadene H. Université des Sciences et de la Technologie HOUARI BOUMEDIENE, Alger
Akaichi Jalel ISG, Tunis
Aussem Alexandre Univ Lyon 1
Baina Karim ENSIAS, Rabat, Maroc
Baiou M. LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Barra Vincent LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Belaissaoui Mustapha Université Mohammed V (Maroc)
Belbachir H. USTO (Oran)
Ben Yahia Sadok Faculté des Sciences de Tunis
Benamrane Nacéra USTO (Oran)
Benatallah Boualem University of New South Wals (Australie)
Benbernou Salima Université de Lyon I (France)
Benhamou Belaid LSIS, Marseille
Bensebaa Tahar Université Badji Mokhtar, Annaba
Berrachedi Abdelhafid Université des Sciences et de la Technologie HOUARI BOUMEDIENE, Alger
Bibi M.O. Université de Béjaia
Bouchemakh Isma Université des Sciences et de la Technologie HOUARI BOUMEDIENE, Alger
Boufaida Mahmoud Université Mentouri, Constantine
Boughanem Mohand IRIT, Toulouse
Bouzeghoub Mokrane Université de Versailles (Paris)
Chitour Yacine LSS, Université Pari-sud 11
Chmeiss Assef CRIL, Université d'Artois
Engelbert Mephu CRIL - Université d'Artois
Farah Riadh ENSI,Tunis
Farah Nadir Université Badji Mokhtar, Annaba
Fauvet Marie-Christine Université Joseph Fourier, Grenoble
Gourvès Laurent Lamsade, France
Guoinaud Christophe LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Habib Michel Université de Paris VII (France)
Hacid Mohand-Said LIRIS, Université de Lyon I (France)
Hadjali Allel ENSSAT, Lannion
Hamadi Youssef Microsoft Research Cambridge
Hamid SERIDI Université 8 mai 1945, Guelma
Hassina Seridi Université Badji Mokhtar, Annaba
Haytham Elghazel Université Claude Bernard Lyon 1
Jaudoin Hélène ENSSAT, Lannion, France
Kaci Souhila CRIL, Université d'Artois, France
Kechadi Tahar UCD, Irlande
Khalid BENABDESLEM LIESP, Université Lyon 1, France

Kheddouci H. Université de Lyon I
Khemmoudj Mohand Ou Idir Université Paris 13
Laafifi Yacine Université 8 mai 1945, Guelma
Lacomme Philippe LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Laskri Mohamed Tayeb Université Badji Mokhtar, Annaba
Leger Alain France Télécom
Lethi A. H. Université de Metz
Mahey Philippe LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Mazure Bertrand CRIL, CNRS Université d'ARtois
Merouani Hayett Université Badji Mokhtar, Annaba
Messine Frédéric ENSEEIHT, IRIT Toulouse (France)
Missaoui Rokia Universite de Quebec en Outaouais
Mohamed-Khireddine KHOLLADI Université Mentouri, Constantine
Nourine Rachid Université d'Oran
Nourine Lhouari LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Ouanes Mohand UMM Tizi-Ouzou
Petit Jean-Marc INSA de Lyon (France)
Pierre Fouilhoux LIP6, Paris
Rabhi Fethi UNSW, Sydney, Australie
Radjef MS. Université de Bejaia
Rey Christophe Université Blaise Pascal, Clermont-Fd
Sadi Bachir UMMTO, Tizi-Ouzou
Saidi Mohamed Université de Sétif
Sais Fatiha LRI, Université de Paris Sud
Sais Lakhdar CRIL
Salem Yassine Université de Sétif
Salim Haddadi Université 8 mai 1945, Guelma
Schneider Michel Université de Clermont-Ferrand II
Sellami Mokhtar Université Badji Mokhtar, Annaba
Spiteri Pierre INP, Toulouse
Tata Samir INT, Paris
Tchemisova Tatiana University of Aveiro, Portugal
Tighiouart Bornia Université Badji Mokhtar, Annaba
Toumani Farouk LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Trelat Emmanuel Université d'Orléans

Relecteurs additionnels

Mohamad Badra, LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Laurent D'orazio, LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Vincent Dubois, IRIN, Université de Nantes, France
Pierre Fouilhoux , LIP6, Paris, France
Adnene Guabtni, UNSW, Sydney, Australie
Ali khebizi, Université Badji Mokhtar, Annaba
Philippe Lacomme, LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Michel Liquière, LIRMM, Montpellier, France
Yannick Loiseau, LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand
Damien Lolive, ENSSAT, Lannion, France
Evi Syukur, UNSW, Sydney, Australie

Sur la Domination Localisatrice dans les Graphes

[1]Mustapha Chellali, [1]Malika Mimouni et [2]Peter J. Slater
[1]Laboratoire LAMDA-RO
Département de Mathématiques, Université de Blida
B.P. 270, Blida, Algerie.
E-mail: m_chellali@yahoo.com
[2] Département de Mathématiques et Département Science de l'informatique
Université de l' Alabama , Huntsville
Huntsville, AL 35899 USA.
E-mail: slaterp@email.uah.edu

Résumé: Un ensemble D de sommets dans un graphe $G = (V, E)$ est un ensemble dominant localisateur (EDL) si pour toute paire de sommets u, v de $V - D$ les ensembles $N(u) \cap D$ et $N(v) \cap D$ sont non vides et différents. Le nombre de domination localisatrice $\gamma_L(G)$ est le cardinal minimum d'un EDL de G, et le nombre de domination localisatrice supérieur, $\Gamma_L(G)$ est le cardinal maximum d'un EDL minimal de G. On présente différentes bornes sur $\Gamma_L(G)$ et $\gamma_L(G)$.

Mots clés: le nombre de domination localisatrice supérieur, le nombre de domination localisatrice.

1 Introduction

Soit $G = (V, E)$ un graphe simple d'ensemble de sommets V, et E l'ensemble des arêtes. Le *voisinage ouvert* $N(v)$ d'un sommet v est l'ensemble des sommets adjacents à v, le *voisinage fermé* de v est définit par $N[v] = N(v) \cup \{v\}$. Le *degré* de v est désigné par $d_G(v) = |N(v)|$. Le degré minimum d'un graphe G est noté $\delta(G)$.

Un ensemble $D \subseteq V$ est dit *dominant* si tout sommet de $V - D$ possède un voisin dans D. Le *nombre de domination* $\gamma(G)$ est le cardinal minimum d'un ensemble dominant dans G. Un ensemble $D \subseteq V$ est dit un *ensemble dominant localisateur* (EDL) s'il est dominant, et pour toute paire de sommets $x, y \in V - D$ satisfait $N(x) \cap D \neq N(y) \cap D$. Le *nombre de domination localisatrice* $\gamma_L(G)$ est le cardinal minimum d'un EDL de G, et le *nombre de domination*

localisatrice supérieur $\Gamma_L(G)$ est le cardinal maximum d'un EDL minimal de G. Un EDL de cardinal minimum est dit un $\gamma_L(G)$-ensemble, de même on définit un $\Gamma_L(G)$-ensemble. La domination localisatrice a été introduite par Slater [11, 12]. Concernant les études récentes sur la domination localisatrice on cite [3], [4] et [5]. Le *nombre de stabilité* $\beta_0(G)$ et le *nombre de domination stable* $i(G)$ sont le cardinal maximum et minimum d'un dominant stable dans G, respectivement.

Vu qu'aucun travail n'a été réalisé jusqu'à présent sur le nombre de domination localisatrice supérieur. Dans cet article on présente quelques bornes sur le nombre de domination localisatrice supérieur $\Gamma_L(G)$ ainsi que sur le nombre de domination localisatrice $\gamma_L(G)$ d'un graphe G. Avant d'exposer les principaux résultats, on commence par introduire les définitions et les notations suivantes.

Un sommet de degré un est appelé *feuille* et son voisin est appelé *support*. On désigne par $L(G)$ l'ensemble des feuilles de G, et par $S(G)$ l'ensemble des sommets supports de G, tels que $|L(G)| = l(G)$ et $|S(G)| = s(G)$. On désigne par T_x le sous-arbre induit par le sommet x et tous ses descendants dans l'arbre T. Le *diamètre* $\mathrm{diam}(G)$ d'un graphe G est la distance maximum de toutes les paires de sommets de G. La *couronne* d'un graphe G est le graphe construit à partir d'une copie de G, où pour chaque sommet $v \in V(G)$, un nouveau sommet v' et une arête pendante vv' sont ajoutés. On désigne par P_n et C_n la chaîne et le cycle de n sommets, respectivement.

2 Le nombre de domination localisatrice superieur

On commence par donner une borne supérieure sur $\Gamma_L(G)$ pour tout graphe G ayant au moins une arête.

Théorème 1 *Tout graphe non trivial connexe G d'ordre n satisfait $\Gamma_L(G) \leq n-1$. La borne est atteinte si et seulement G est un graphe complet ou bien une étoile.*

Preuve. Il est clair que la borne supérieure est dûe au fait que l'ensemble des sommets de G est un dominant localisateur mais il n'est pas minimal.

Soit S un $\Gamma_L(G)$-ensemble de taille $n-1$ et supposons que $V - S = \{u\}$. On suppose d'abord que $A = S - N(u) \neq \emptyset$. La minimalité de S implique que A est un ensemble indépendant car autrement il existe un sommet $z \in A$ ayant un voisin dans A, $S - \{z\}$ est un EDL de G, d'où la contradiction. Etant donné que G est connexe, tout sommet $v \in A$ a au moins un voisin dans S. Aussi $N(v) = N(u)$ autrement $S - \{v\}$ est un EDL de G, contradiction avec l'hypothèse. A présent si $|N(u)| \geq 2$, alors puisque tout sommet $u' \in N(u)$ est adjacent à tous les sommets de A, l'ensemble $S - \{u'\}$ est un EDL de G, ceci contredit la minimalité de S. Ainsi $|N(u)| = 1$, donc G est une étoile. Supposons maintenant que $A = \emptyset$. Si toutes les arêtes entre les sommets de S existent, alors G est un graphe complet. Par conséquent soient x, y deux sommets non adjacents de S. Si x a un voisin dans S alors $S - \{x\}$ est un EDL

de G, contradiction. Ainsi x est un sommet isolé dans S, de même pour y. Il s'ensuit que S est un ensemble indépendant, d'où G est une étoile.

La réciproque est évidente. ■

Dans ce qui suit on établit la valeur exacte du nombre de domination localisatrice supérieur pour les chaînes.

Théorème 2 *Pour toute chaîne* P_n,

$$\Gamma_L(P_n) = \begin{cases} 4k & si \quad n = 7k \\ 4k+1 & si \quad n = 7k+1 \ ou \ n = 7k+2 \\ 4k+2 & si \quad n = 7k+3 \ ou \ n = 7k+4 \\ 4k+3 & si \quad n = 7k+5 \\ 4k+4 & si \quad n = 7k+6 \end{cases}$$

Preuve. On procède par induction sur l'ordre n. Il est facile de vérifier que le résultat est vrai pour $n \leq 7$. Soit $n \geq 8$, on suppose que toute chaîne $P_{n'}$ d'ordre n', avec $1 \leq n' < n$ satisfait la propriété. Soit P_n une chaîne telle que $V(P_n) = \{u_1, u_2, ..., u_n\}$ et D un $\Gamma_L(P_n)$-ensemble quelconque. On admettra que $|D \cap \{u_1, u_2, ..., u_7\}| = 4$. On note d'abord que D ne contient pas trois sommets consécutifs, ainsi $|D \cap \{u_1, u_2, ..., u_7\}| \leq 4$. Supposons que $|D \cap \{u_1, u_2, ..., u_7\}| \leq 3$. Si $u_7 \in D$ alors u_2, u_4 doivent être dans D, par conséquent $D' = \{u_1, u_3\} \cup (D - \{u_2\})$ est un EDL minimal de P_n de taille supérieure à $|D|$, contradiction avec l'hypothèse. Donc $u_7 \notin D$. Il est clair que $D' = D \cap \{u_8, ..., u_n\}$ est un EDL de la chaîne $P_{n'} = P_n - \{u_1, u_2, ..., u_7\}$. Si D' est minimal alors $\{u_1, u_2, u_5, u_6\} \cup D'$ est un EDL minimal de taille supérieure à $|D|$. Ainsi on suppose que D' n'est pas minimal alors il existe un sommet $v \in D'$ tel que $D' - \{v\}$ est minimal pour $P_{n'}$. Par conséquent $\{u_1, u_2, u_5, u_6\} \cup (D' - \{v\})$ est un $\Gamma_L(P_n)$-ensemble puisque $|D \cap \{u_1, u_2, ..., u_7\}| \leq 3$. Par conséquent on a un EDL tel que $|D \cap \{u_1, u_2, ..., u_7\}| = 4$. Pour terminer la preuve on considère les deux cas suivants.

Cas 1 $u_7 \notin D$. D'après les observations précédentes, on peut supposer que $u_1, u_2, u_5, u_6 \in D$ et $u_3, u_4 \notin D$. Soit $P_{n'}$ la chaîne obtenue à partir de P_n en supprimant les sommets $u_1, u_2, ..., u_7$, donc $D - \{u_1, u_2, u_5, u_6\}$ est un EDL minimal de $P_{n'}$, d'où $\Gamma_L(P_{n'}) \geq \Gamma_L(P_n) - 4$. De plus tout $\Gamma_L(P_{n'})$-ensemble peut être étendu à un EDL minimal de P_n en lui ajoutant l'ensemble $\{u_1, u_2, u_5, u_6\}$, d'où $\Gamma_L(P_{n'}) = \Gamma_L(P_n) - 4$. En appliquant l'hypothèse d'induction sur $P_{n'}$ et en examinant cas par cas les valeurs de n' on obtient le résultat voulu.

Cas 2 $u_7 \in D$. On distingue les trois sous-cas suivants.

Cas 2.1 $u_6 \notin D$ et $u_8 \in D$. Etant donné que $|D \cap \{u_1, u_2, .., u_5\}| = 3$, disons $u_1, u_3, u_4 \in D$. Ainsi $D - \{u_1, u_2, ..., u_7\}$ est un EDL minimal de la chaîne induite par $V(P_n) - \{u_1, u_2, ..., u_7\} = V(P_{n'})$. D'où $\Gamma_L(P_{n'}) \geq \Gamma_L(P_n) - 4$, l'égalité vient du fait que tout $\Gamma_L(P_{n'})$-ensemble peut être étendu à un EDL minimal de P_n en lui ajoutant l'ensemble $\{u_1, u_2, u_5, u_6\}$. Par induction sur $P_{n'}$, et en considérant toutes les valeurs possibles de n' on obtient le résultat.

Cas 2.2 $u_6, u_8 \notin D$. Si $n = 8$ alors $\{u_1, u_2, u_5, u_6, u_8\}$ est un EDL minimal de P_8 de taille supérieure à $|D|$, d'où la contradiction. Ainsi $n \geq 9$. On suppose à

présent que $u_9 \in D$. D'après les faits cités ci-dessus, et sans perte de généralité, on supposera que $\{u_2, u_3, u_5\} \subset D$. Posons $D' = D - \{u_2, u_3, u_5, u_7\}$. On note que $\{u_2, u_3, u_5, u_7\}$ est un EDL minimal de la chaîne induite par $\{u_1, u_2, ..., u_8\}$. Si D' n'est pas un EDL minimal de $V(P_n) - \{u_1, u_2, ..., u_7\}$, alors u_8 et u_{10} ont un voisin unique u_9 dans D', par conséquent $\{u_1, u_6, u_8\} \cup (D - \{u_3, u_7\})$ est un EDL minimal de P_n de taille supérieure à $|D|$, d'où la contradiction. Donc D' est un EDL minimal de $P_{n'}$, où $n' = n - 7$. Il est facile de voir que $\Gamma_L(P_n) = \Gamma_L(P_{n'}) + 4$. Par induction sur $P_{n'}$, on obtient le résultat. Supposons maintenant que $u_9 \notin D$. Il est clair que $u_{10} \in D$. Suite aux faits, on suppose que $\{u_2, u_3, u_5\} \subset D$. On pose $D'' = D - \{u_2, u_3, u_5, u_7\}$, et P' la chaîne obtenue à partir de P_n en supprimant les sommets $u_1, u_2, ..., u_9$. Notons que D'' est un EDL de P'. Si D'' est minimal alors $D'' \cup \{u_1, u_2, u_5, u_6, u_8\}$ est un EDL minimal de P_n de taille supérieure à $|D|$, d'où la contradiction. Ainsi on suppose que D'' n'est pas minimal. Alors il existe un sommet $w \in D''$ tel que $D'' - \{w\}$ est un EDL minimal de P', donc l'ensemble $\{u_1, u_2, u_5, u_6, u_8\} \cup (D'' - \{w\})$ est un $\Gamma_L(P_n)$-ensemble ne contenant pas u_7, ce cas a été traité dans cas 1.

Cas 2.3 $u_6 \in D$. Alors $u_8 \notin D$, autrement $D - \{u_7\}$ est un EDL, ceci contredit la minimalité de D. De même $u_5 \notin D$, autrement $D - \{u_6\}$ est un EDL de G. Donc sans perte de généralité, on suppose $u_2, u_3 \in D$. Alors $\{u_5\} \cup (D - \{u_6\})$ est un $\Gamma_L(P_n)$-ensemble ne contenant ni u_6, ni u_8, ce cas a été considéré dans cas 2.2. ■

Théorème 3 *Si G est un arbre ou bien un graphe avec $g(G) \geq 5$, alors tout ensemble indépendant maximum S est un dominant localisateur minimal. De plus, si $\delta \geq 2$, alors $V - S$ est un dominant localisateur.*

Preuve. Soit S un $\beta_0(G)$-ensemble. On montre d'abord que S est un dominant localisateur de G. Si $\beta_0(G) = 1$, alors $G = K_1$ ou K_2, d'où le résultat est vrai. Supposons que $\beta_0(G) \geq 2$ et que S n'est pas un dominant localisateur de G. Dans ce cas, il existe au moins deux sommets $u, v \in V - S$ tels que $N(u) \cap S = N(v) \cap S$. Si u et v ont deux voisins communs dans S, alors le sous-graphe induit par u, v, ainsi que leurs voisins dans S contient un cycle C_4, ceci contredit le fait que $g(G) \geq 5$. Ainsi u et v ont un voisin commun unique dans S, disons w. Si $uv \in E$, alors $\{u, v, w\}$ induit un cycle C_3. Ainsi u et v ne sont pas adjacents, dans ce cas, $\{u, v\} \cup (S - \{w\})$ est un ensemble indépendant de taille supérieure à $|S|$, d'où la contradiction. On déduit que S est un dominant localisateur de G. Etant donné que S est un dominant minimal, ceci implique que S est un dominant localisateur minimal.

Cependant, si $\delta \geq 2$, alors puisque $g(G) \geq 5$, il n'existe pas dans S deux sommets ayant un voisinage commun dans $V - S$. D'où $V - S$ est un dominant localisateur. ■

En conséquence immédiate, on déduit les deux corollaires suivants.

Corollaire 4 *Si G est un arbre ou un graphe avec $g(G) \geq 5$, alors $\Gamma_L(G) \geq \beta_0(G) \geq \gamma_L(G)$.*

Corollaire 5 (Blidia et al. [5] 2008) *Si T est un arbre, alors $\beta_0(T) \geq \gamma_L(T)$.*

Corollaire 6 *Si G est un graphe d'ordre n, avec $\delta \geq 2$ et $g(G) \geq 5$, alors $\gamma_L(G) \leq n/2$.*

On note que la différence $\Gamma_L(G) - \beta(G)$ peut être arbitrairement large, même pour les arbres. Considérons l'arbre T_t obtenu à partir de $t \geq 1$ copies d'une chaîne P_6, en reliant le troisième sommet de chaque chaîne P_6 au troisième sommet de la chaîne suivante P_6. Il est clair que l'ensemble des supports et des feuilles est un $\Gamma_L(T_t)$-ensemble de taille $4t$, par contre $\beta_0(T_t) = 3t$.

Théorème 7 *Si T est un arbre non trivial d'ordre n, avec l feuilles, alors $\Gamma_L(T) \leq \dfrac{2n+\ell-2}{3}$, et cette borne est atteinte.*

Preuve. On procède par induction sur l'ordre n de T. Il est clair que le résultat est vrai si $\text{diam}(T) \in \{1,2,3\}$. On suppose que pour tout arbre T' d'ordre $n' < n$, avec l' feuilles satisfait $\Gamma_L(T') \leq \dfrac{(2n'+l'-2)}{3}$. Soient T un arbre d'ordre n, avec $\text{diam}(T) \geq 4$, et D un $\Gamma_L(T)$-ensemble quelconque de T.

S'il existe un sommet support u adjacent à au moins deux feuilles, alors on pose $T' = T - \{u'\}$, où u' est une feuille de u appartenant à D. Une telle feuille existe puisque D contient ou bien toutes les feuilles de u ou bien toutes les feuilles de u sauf une. Alors $D - \{u'\}$ est un EDL minimal de T', par conséquent $\Gamma_L(T') \geq |D| - 1$. Etant donné que $n' = n-1$, $l(T') = l-1$, par induction sur T', on obtient $\Gamma_L(T) \leq (2n+l-2)/3$. Par la suite, on supposera que $l - s(T) = 0$.

Supposons que T contient deux supports adjacents x, y. Soit T_x et T_y les sous-arbres obtenus à partir de T en supprimant l'arête xy. Alors $D_x = D \cap V(T_x)$ est un EDL minimal de T_x, de même $D_y = D \cap V(T_y)$ est un EDL minimal de T_y. Ainsi $\Gamma_L(T_x) + \Gamma_L(T_y) \geq |D_x| + |D_y| = \Gamma_L(T)$. Puisque $\text{diam}(T) \geq 4$, T_x ou T_y, disons T_y est de diamètre au moins deux. Par induction sur T_y on a $\Gamma_L(T_y) \leq \dfrac{2|V(T_y)| + |L(T_y)| - 2}{3}$. Si $\text{diam}(T_x) = 1$, alors $\Gamma_L(T_x) = 1 = \dfrac{2|V(T_x)| + 1 - 2}{3}$ et $l = |L(T_y)| + 1$. Si $\text{diam}(T_x) \geq 2$, alors par induction, $\Gamma_L(T_x) \leq \dfrac{2|V(T_x)| + |L(T_x)| - 2}{3}$, vu que $V(T_x) + V(T_y) = n$, et $l = L(T_x) + L(T_y)$, dans les deux cas, on obtient le résultat voulu. D'où on suppose qu'il n'existe pas deux supports adjacents dans T.

Enracinons l'arbre T en un sommet r d'excentricité maximum $\text{diam}(T) \geq 4$. Soit u un support à distance $\text{diam}(T) - 1$ de r sur la plus longue chaîne commençant par r. Soient v, w les parents de u, v respectivement, et u' l'unique feuille de u. Comme v ne peut pas être un support, le sous-arbre T_v de racine v est une étoile subdivisée.

On suppose d'abord que $v \in D$. Posons $T' = T - \{u', u\}$. Notons que D contient ou bien u ou u'. Ainsi $D \cap V(T')$ est un EDL de T', par conséquent

$\Gamma_L(T') \geq |D| - 1$ et $l(T') \leq l$. Par induction sur T', on obtient $\Gamma_L(T) \leq \frac{2n+\ell-2}{3}$. Supposons à présent que $v \notin D$, on distingue deux cas.

Cas 1. $d_T(v) = k \geq 3$. Si un des fils de v, disons z n'appartient pas à D, alors on pose $T' = T - \{z', z\}$, où z' est l'unique feuille de z. Ce cas est similaire à celui traité ci-dessus, donc $\Gamma_L(T) \leq \frac{2n+\ell-2}{3}$. Ainsi supposons que D contient tous les fils de v. Posons $T' = T - T_v$, il est clair que D ne contient aucune feuille de T_v, et $D \cap V(T')$ est un EDL minimal de T'. D'où $\Gamma_L(T') \geq |D| - k + 1$ et $l(T') \leq l - k + 1$, par induction sur T', on retrouve immédiatement le résultat.

Cas 2. $d_T(v) = k = 2$. Etant donné que $\text{diam}(T) \geq 4$, soit z le parent de w. Si $z = r$ alors $T = P_5$ et $\Gamma_L(T) \leq \frac{2n+\ell-2}{3}$. Ainsi $z \neq r$, pour compléter la preuve, on considère les trois situations suivantes de D. Rappelons que $v \notin D$, donc on a ou bien $u', w \in D$ et $u \notin D$ ou bien $u, w \in D$ et $u' \notin D$ ou bien $u, u' \in D$ et $w \notin D$.

Si on est dans la première situation, alors on pose $T' = T - \{u', u\}$. Dans la seconde situation, $D - \{u\}$ est un EDL minimal pour $T' = T - \{u', u\}$ ou pour $T' = T - \{u', u, v\}$, et le sous-arbre T' pour lequel $D - \{u\}$ est un EDL minimal sera considéré. Il est facile de vérifier que des deux situations citées ci-dessus, on obtient $\Gamma_L(T) \leq \frac{2n+\ell-2}{3}$.

Considérons à présent la dernière situation, c-a-d $u, u' \in D$ et $w \notin D$. Posons $T' = T - \{u', u, v\}$. Alors $D - \{u', u\}$ est un EDL minimal de T', d'où $\Gamma_L(T') \geq |D| - 2$. Par induction, et du fait que $n' = n - 3$, $l(T') = l$, il s'ensuit que $\Gamma_L(T) \leq \frac{2n+\ell-2}{3}$.

La borne est atteinte pour les étoiles non triviales. ■

La borne inférieure suivante sur le nombre de stabilité pour les graphs bipartis est donnée dans [2].

Proposition 8 (Blidia, Chellali, Favaron, Meddah [2] 2007) *Si G est un graphe biparti, alors*

$$\beta_0(G) \geq \frac{n + \ell(G) - s(G)}{2}.$$

Suite au Théorème 7 et la Proposition 8, on obtient une borne supérieure sur $\Gamma_L(T)$, pour les arbres en fonction de $\beta_0(T)$, le nombre de feuilles et de supports.

Corollaire 9 *Si T est un arbre non trivial, alors* $\Gamma_L(T) \leq \frac{4}{3}\beta_0(T) - \frac{1}{3}(\ell(T) - 2s(T) + 2)$.

On obtient aussi

Corollaire 10 *Si T est un arbre non trivial, alors* $\Gamma_L(T) - \beta_0(T) \leq \frac{1}{6}(n - \ell(T) + 3s(T) - 4)$.

3 Le nombre de domination localisatrice

On commence par rappeler les deux résultats donnés dans [4] sur le nombre de domination localisatrice pour la classe des arbres.

Théorème 11 (Blidia et al. [4] 2007) *Si T est un arbre d'ordre $n \geq 2$, alors* $\gamma_L(T) \leq \dfrac{(n + \ell(T) - s(T))}{2}$.

Théorème 12 (Blidia et al. [4] 2007) *Si T est un arbre d'ordre $n \geq 3$, alors* $\gamma_L(T) \geq \dfrac{(n + \ell(T) - s(T) + 1)}{3}$.

Proposition 13 *Si G est un graphe bloc avec $\delta \geq 2$, alors*

$$\gamma_L(G) + \beta_0(G) \leq n.$$

Preuve. Soit S un quelconque $\beta_0(G)$-ensemble. Alors tout sommet de S a au moins deux voisins dans $V - S$, étant donné que G est un graphe bloc ne contenant pas de C_4 et $K_4 - \{e\}$, chaque paire de sommets $x, y \in S$ satisfait $N(x) \cap (V - S) \neq N(y) \cap (V - S)$. Il s'ensuit que $V - S$ est un EDL de G, ceci implique que $\gamma_L(G) \leq |V - S| = n - \beta_0(G)$. ■

La borne de la proposition 13 est atteinte pour le graphe G_k formé à partir de k triangles partageant le même sommet.

Dans ce qui suit, on montrera que le nombre de domination localisatrice est borné inférieurement par le nombre de domination stable pour la classe des arbres.

Théorème 14 *Si T est un arbre, alors $\gamma_L(T) \geq i(T)$.*

Preuve. On procède par induction sur l'ordre de T. Si $\text{diam}(T) \in \{0, 1, 2, 3\}$, alors le résultat est vrai. Supposons que tout arbre T' d'ordre $n' < n$ satisfait $\gamma_L(T') \geq i(T')$. Soient T un arbre d'ordre n et de diamètre au moins quatre, et D un $\gamma_L(T)$-ensemble contenant tous les sommets supports. Si un quelconque sommet support, disons x de T est adjacent à deux feuilles ou plus, alors soit T' l'arbre obtenu à partir de T en supprimant une feuille x' de$x$, et qui appartient à D. Alors $D - \{x'\}$ est un EDL de T', d'où $\gamma_L(T') \leq \gamma_L(T) - 1$. Si S est un $i(T')$-ensemble quelconque, alors ou bien S ou $S \cup \{x'\}$ est un ensemble indépendant maximal de T, d'où $i(T) \leq i(T') + 1$. Par induction sur T', on retrouve le résultat voulu. Ainsi, supposons que tout sommet support de T est adjacent à une seule feuille.

A présent enracinons l'arbre T en un sommet r tel que r est une feuille d'excentricité maximum $\text{diam}(T)$. Soit u le sommet situé à une distance $\text{diam}(T) - 1$ de r, sur la plus longue chaîne commençant par r. Soit v le parent de u, et u' l'unique feuille de u.

Si v est un support, alors posons $T' = T - \{u, u'\}$. Par conséquent, $\gamma_L(T') \leq \gamma_L(T) - 1$ et $i(T) \leq i(T') + 1$. Par induction sur T', on a $\gamma_L(T) \geq i(T)$. Ainsi, v n'est pas un support, donc T_v est une étoile subdivisée. Comme D contient les sommets supports de T_v, alors $v \notin D$ (autrement on le remplace par son parent). On pose $T' = T - T_v$, il s'ensuit que $D \cap V(T')$ est un EDL de T', d'où $\gamma_L(T') \leq \gamma_L(T) - d_T(v) + 1$. De plus, tout $i(T')$-ensemble union l'ensemble des sommets supports de T_v est un ensemble indépendant maximal de T, ceci implique que $i(T) \leq i(T') - d_T(v) + 1$. Par induction sur T', on obtient $\gamma_L(T) \geq i(T)$. ■

D'après le Corollaire 4 et le Théorème 14 on obtient la chaîne d'inégalité suivante relativement aux paramètres de domination localisatrice ainsi que ceux de la stabilité pour tout arbre T :

$$i(T) \leq \gamma_L(T) \leq \beta_0(T) \leq \Gamma_L(T). \quad (1)$$

L'égalité est atteinte le long de cette chaîne si et seulement si $T = K_1$ ou T est une couronne d'un arbre T'. Cependant pour la classe des arbres T_t définis dans la section 2, on a $\Gamma_L(T_t) - i(T_t) = 4t - 2t = 2t$.

La classe des graphes G d'ordre n pair, et sans sommets isolés, avec $\gamma(G) = n/2$, a été caractérisée indépendemment par Payan et Xuong [10] et Fink, Jacobson, Kinch et Roberts [7].

Théorème 15 (Payan, Xuong [10] 1982 and Fink et al. [7] 1985) *Soit G un graphe d'ordre n pair, sans sommets isolés, alors $\gamma(G) = n/2$ si et seulement si chaque composante de G est ou bien un cycle C_4 ou bien une couronne d'un graphe connexe.*

Observation 16 *Soit T un arbre d'ordre $n \geq 3$, alors $\gamma(T) \leq \dfrac{n - \ell(T) + s(T)}{2}$. La borne est atteinte si et seulement si T est un arbre d'ordre $l(T) + s(T)$.*

Preuve. Il est clair que le résultat est vérifié si T est une étoile. Supposons que T n'est pas une étoile et soit T^* l'arbre obtenu à partir de T, en supprimant pour tout sommet support de T toutes ses feuilles sauf une. Etant donné qu'il existe un dominant minimum contenant tous les supports, on a $\gamma(T) = \gamma(T^*)$. De plus l'arbre T^* est d'ordre $n - l(T) + s(T)$, et d'après le Théorème célèbre d'Ore on a $\gamma(T^*) \leq \dfrac{n - \ell(T) + s(T)}{2}$. A présent, d'après le Théorème 15 on a $\gamma(T^*) = \dfrac{n - \ell + s}{2}$ si et seulement si T est une couronne d'un arbre T', d'où T est un arbre, où tout sommet est ou bien un support ou bien une feuille, ainsi T est d'ordre $l(T) + s(T)$.

La réciproque est évidente. ■

Rappelons qu'un ensemble $R \subseteq V(G)$ est dit *2-stable* dans G si pour deux sommets quelconques x et y de S on a $N[x] \cap N[y] = \emptyset$. Le cardinal maximum d'un ensemble *2-stable de G* noté $\rho(G)$ est appelé *le nombre de 2-stabilité.*

Proposition 17 *Pour tout graphe connexe non trivial G, on a*

$$\gamma_L(G) \leq n - \rho(G).$$

Preuve. Soit R un 2-stable maximum de G. Etant donné que $N[x] \cap N[y] = \emptyset$, pour toute paire de sommets distincts $x, y \in R$, $V - R$ est un dominant localisateur de G, d'où $\gamma_L(G) \leq |V - R| \leq n - \rho(G)$. ■

Farber [6] a prouvé que le nombre de domination et le nombre de 2-stabilité sont égaux pour tout graphe fortement triangulé, qui inclut la classe des arbres. Ainsi on énonce le corollaire suivant comme une conséquence de la proposition 17.

Corollaire 18 *Pour tout arbre non trivial T, on a $\gamma_L(T) + \gamma(T) \leq n$. La borne est atteinte si et seulement si T est un arbre d'ordre $l(T) + s(T)$.*

Preuve. On suppose que $\gamma_L(T) + \gamma(T) = n$. Si T est une étoile alors il est d'ordre $l(T) + s(T)$. Donc supposons que T n'est pas une étoile alors d'après le Théorème 11 et la remarque 16 on a $\gamma_L(T) = \dfrac{n + \ell(T) - s(T)}{2}$, et $\gamma(T) = \dfrac{n + \ell(T) - s(T)}{2}$. Il s'ensuit que T est un arbre d'ordre $l(T) + s(T)$.
La réciproque est évidente. ■

Le théorème suivant est une extension de la borne supérieure du Théorème 11 pour les graphes bipartis ne contenant pas de cycles C_4. On supposera que $l(P_2) = s(P_2) = 2$.

Théorème 19 *Si G est un graphe biparti connexe, sans sommets isolés et sans cycles C_4, alors $\gamma_L(G) \leq (n + l(G) - s(G))/2 \leq \Gamma_L(G)$.*

Preuve. Si G est un arbre alors d'après le Corollaire 4, la proposition 8 et le Théorème 11 le résultat est vrai. Supposons que G n'est pas un arbre. Soit D l'ensemble des feuilles de G choisi comme suit: pour tout sommet support u de G adjacent à deux feuilles ou plus dans G, on met dans D toutes les feuilles de u sauf une. Alors $|D| = l(G) - s(G)$. Considérons maintenant le sous-graphe G' obtenu à partir de G, en supprimant toutes les feuilles de G. Puisque G n'est pas un arbre, et sans C_4, G' est non trivial, donc il admet une unique bipartition en deux sous-ensembles indépendants non vides A et B. Il est clair que toute feuille de G' est un support dans G et tout sommet de G' qui n'est pas une feuille est de degré au moins deux dans G'. Posons $A' = A - S(G)$ et $B' = B - S(G)$, sans perte de généralité on peut supposer que $|A| \leq |B|$. Vu que G est un graphe biparti sans cycle C_4, l'ensembe B' (resp A') ne contient pas deux sommets ayant un voisinage commun dans $D \cup S(G) \cup A'$ (resp $D \cup S(G) \cup B'$). Ainsi chacun des ensembles $D \cup S(G) \cup A'$ et $D \cup S(G) \cup B'$ est un EDL minimal de G. Par conséquent, $\gamma_L(T) \leq |D \cup S(G) \cup A'|$ et $\Gamma_L(G) \geq |D \cup S(G) \cup B'|$. Du fait que $|A'| \leq (n - l(G) - s(G))/2 \leq |B'|$, on en déduit le résultat. ■

Notons que la borne supérieure sur $\gamma_L(G)$ n'est pas vérifiéede si G est un graphe biparti contenant un cycle C_4. Pour éclaircir ce point, considérons le cycle C_4 en attachant un nouveau sommet à un sommet du cycle. Il est clair que $\gamma_L(G) = 3 > (n + \ell(G) - s(G))/2$.

Théorème 20 *Si G est un graphe connexe d'ordre $n \geq 2$, ayant au plus un cycle, alors $\gamma_L(G) \leq (n + l(G) - s(G) + 1)/2$.*

Preuve. Si G est un arbre alors d'après le Théorème 11 le résultat est vrai. Ainsi, on suppose que G contient un cycle C. Il est clair que si $G = C$ alors $\gamma_L(G) \leq (n+1)/2$. Donc on suppose que $G \neq C$, par conséquent, G contient un sommet de degré au moins trois. Supposons que l'inégalité n'est pas vérifiée et que G est le plus petit unicycle connexe, tel que $\gamma_L(G) > (n + l(G) - s(G) + 1)/2$. On suppose aussi que parmi tous ces graphes, G est celui qui contient le moins d'arêtes.

On suppose d'abord que tous les supports sont sur le cycle C. Si C contient un seul sommet support b, alors soit A un ensemble indépendant maximum de $G[V(C)]$ qui contient b. Alors $|A| = \lfloor |C|/2 \rfloor$, et A union l'ensemble des feuilles L_b de b est un ensemble dominant localisateur de G de taille au plus $(n + |L_b|)/2 = (n + \ell(G) - s(G) + 1)/2$, contradiction avec notre hypothèse. Ainsi C contient au moins deux supports.

Supposons que C contient deux supports consécutifs x et y tels que la distance entre x et y est au moins trois. Soit H l'ensemble des sommets sur la chaîne reliant x à y. Ainsi les sommets de H induisent une chaîne d'ordre $|H| \geq 2$. Posons $G' = G - H$. Alors $n' = n - |H|$, $\ell(G') = \ell(G)$, $s(G') = s(G)$. G' et $P_{|H|}$ sont des arbres, d'où d'après le Théorème 11, $\gamma_L(G') \leq (n' + \ell(G') - s(G'))/2$ et $\gamma_L(P_{|H|}) \leq (|H| + 1)/2$. Ceci entraîne que

$\gamma_L(G) \leq (n' + \ell(G') - s(G'))/2 + (|H|+1)/2 = (n + \ell(G) - s(G) + 1)/2$, d'où la contradiction. Ainsi la distance entre deux supports consécutifs sur C est un ou deux. Par conséquent $n = s(G) + \ell(G) + k$, où k est le nombre de sommets de degré deux. Il est clair que $k \leq s(G)$. Soit $L'(G)$ l'ensemble des feuilles de G en prenant pour chaque support toutes ses feuilles sauf une. Ainsi $|L'(G)| = l(G) - s(G)$, il est facile de voir que $S(G) \cup L'(G)$ est un dominant localisateur de G, d'où $\gamma_L(G) \leq s(G) + (l(G) - s(G)) = l(G) < (n + l(G) - s(G) + 1)/2$, ceci est contradictoire avec notre hypothèse. Il s'ensuit que G contient au moins un support qui n'appartient pas à C.

Soit u un support de G situé à une distance maximale de C. Soit v le voisin de u sur l'unique chaîne de u à C. On suppose d'abord que $d_G(v) \geq 3$, posons $G' = G - (L_u \cup \{u\})$, où L_u est l'ensemble des feuilles de u. Si S' est un $\gamma_L(G')$-ensemble, alors $S' \cup L_u$ est un ensemble dominant localisateur de G, d'où $\gamma_L(G) \leq \gamma_L(G') + |L_u|$. Puisque G' est d'ordre n' inférieur à n et $n' = n - (|L_u| + 1)$, $\ell(G') = \ell(G) - |L_u|$ et $s(G') = s(G) - 1$ ceci entraîne que

$$\gamma_L(G) \leq (n' + \ell(G') - s(G') + 1)/2 + |L_u| = (n + \ell(G) - s(G) + 1)/2$$

contradiction. Ainsi on suppose que $d(v) = 2$ et soit w le second voisin de v sur l'unique chaîne de v à C. Si $d(w) = 2$ ou $d(w) \geq 3$ et w n'est pas un

support, alors on pose $G' = G - (L_u \cup \{u\})$. Donc G' satisfait le théorème et on a $n' = n - (|L_u| + 1)$, $\ell(G') = \ell(G) - |L_u| + 1$, $s(G') = s(G)$. Etant donné que tout $\gamma_L(G')$-ensemble peut être étendu à un dominant localisateur de G en lui ajoutant L_u, on obtient

$$\gamma_L(G) \leq (n' + \ell(G') - s(G') + 1)/2 + |L_u| = (n + \ell(G) - s(G) + 1)/2.$$

D'où la contradiction. Ainsi on suppose que $d_G(w) \geq 3$ et w est un support, posons $G' = G - (L_u \cup \{u, v\})$. Alors G' satisfait le théorème et on a $n' = n-(|L_u|+2)$, $\ell(G') = \ell(G)-|L_u|$, $s(G') = s(G)-1$. Puisqu'il existe un $\gamma_L(G')$-ensemble qui contient w, un tel ensemble pourra être augmenté à un dominant localisateur de G en lui ajoutant $\{u\} \cup L_u - \{u'\}$, où u' est une feuille quelconque de u. Il s'ensuit que $\gamma_L(G) \leq (n'+\ell(G')-s(G')+1)/2+|L_u| < (n+\ell(G)-s(G)+1)/2$, contradiction avec l'hypothèse. D'où $\gamma_L(G) \leq (n + l(G) - s(G) + 1)/2$. Ceci achève la preuve. ■

References

[1] M. Blidia, M. Chellali and O. Favaron, Independence and 2-domination in trees. *Australasian Journal of Combinatorics,* 33 (2005) 317–327.

[2] M. Blidia, M. Chellali, O. Favaron, N. Meddah, On k-independence in graphs with emphasis on trees. *Discrete Math.* 307 (2007) 2209–2216.

[3] M. Blidia, M. Chellali, R. Lounes and F. Maffray, Characterizations of trees with unique minimum locating-dominating sets. *Submitted.*

[4] M. Blidia, M. Chellali, F. Maffray, J. Moncel and A. Semri, Locating-domination and identifying codes in trees. *Australasian Journal of Combinatorics,* 39 (2007) 219–232.

[5] M. Blidia, O. Favaron and R. Lounes, Locating-domination, 2-domination and independence in trees. *Australasian Journal of Combinatorics,* 42 (2008) 309–316.

[6] M. Farber. Domination, independent domination and duality in strongly chordal graphs. *Disc. Appl. Math.* 7 (1984) 115–130.

[7] J. F. Fink, M.S. Jacobson, L.F. Kinch and J. Roberts, On graphs having domination number half their order, *Period. Math. Hungar.* **16** (1985) 287–293.

[8] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

[9] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater (eds), *Domination in Graphs: Advanced Topics*, Marcel Dekker, New York, 1998.

[10] C. Payan and N. H. Xuong, Domination-balanced graphs, *J. Graph Theory* **6** (1982) 23–32.

[11] P. J. Slater. Domination and location in acyclic graphs. *Networks* 17 (1987) 55–64.

[12] P. J. Slater. Dominating and reference sets in graphs. *J. Math. Phys. Sci.* 22 (1988) 445–455.

A new Algorithm for Finding the Non-Dominated Set for the MOILP Problem

Mohamed El-Amine Chergui Meriem Aït Mehdi
Moncef Abbas

Abstract

Our study is about a new technique to generate the non-dominated set for the MOILP problem. While most of researchers solve initially an ILP problem, the proposed method starts with an optimal solution of an LP problem and uses a branching process to find locally an integer solution. Then, an efficient cut which deletes only dominated vectors from the feasible set in the criteria space is built. The results show that our method is better than Sylva & Crema's one since it's 15 times faster, makes thrice less simplex iterations and generates almost 4 times fewer nodes on average.

Keywords: Multiobjective, Integer Linear Programming, Branch and Bound, Cutting plane, Non-dominated Solutions.

1 Introduction

Multiobjective integer linear programming problem is an extension of the classical single-objective integer programming motivated by a variety of real world applications in which it is necessary to consider two or more conflicting objective functions that are to be optimized over a feasible set of decisions. The discrete nature of these problems implies that they are non-convex in general. Such problems are commonly encountered in many areas of human activity including business, management, engineering and many other areas where decision-making requires consideration of competing objectives. Examples of the use of MOILPs can be found in capital budgeting [2], location analysis [5], and engineering design [8].

It is well known that for multiobjective linear programming problems (MOLP), the set of non-dominated solutions is exactly the set of solutions that can be obtained by solving a weighted-sum scalarization of the objective functions [7]. But the presence of discrete variables in MOLP problems makes this result invalid. Indeed, there often exist non-dominated solutions, which cannot be found by solving any weighted-sum of the objectives. This is true even in cases where the constraint matrix is totally unimodular [16]. These solutions are called

non-supported non-dominated solutions, whereas the remaining are called supported non-dominated solutions. Since non-supported non-dominated solutions often constitute the majority of the non-dominated set of discrete problems, they cannot be ignored in the decision process [17]. Moreover, they contribute essentially to the difficulty of MOILP problems. Several methods have been developed to solve the MOILP problem, ranging from exact methods that find all non-dominated solutions [13], [3] and [15], to interactive methods that only find the best solution for the decision-maker [4], [1] or meta-heuristics methods that generate approximate solutions [4].

The MOILP problem is difficult to solve since it is classified as a Multi Objective Combinatorial Optimization problem MOCO and it has been extensively studied in the literature [4]. For instance in [9], the authors have worked on an implicit enumeration based algorithm which consists on solving a sequence of single objective function problems, which are progressively more constrained through the addition of constraints that eliminate solutions dominated by known non-dominated vectors producing a sequence of efficient solutions sorted by the selected objective function value. In [12], a variation of the [9] algorithm is proposed, maximizing a positive combination of all the k objective functions that provides a non-dominated solution for the MOILP problem. The considered ILPs are augmented at each stage by $(k+1)$ new constraints and k bivalent variables.

In this paper, an exact method based on the simplex method to generate the set of all non-dominated vectors is presented. The main originality of the proposed approach is to exploit the results of the simplex tables to generate constraints that when added to a LP problem, eliminate domains containing only dominated vectors. To do this, a sequence of progressively more constrained LP problems combined with a classical branching process to search for feasible integer solutions is solved. This means that no ILP problem is solved during all the process of search for the non-dominated vectors but only feasible integer solutions are generated.

The sections described in this paper are organized as follows : after a formal introduction of the problem in section 2, the principle of the method is reported in section 3. The main theoretical results allowing to justify various stages of the algorithm suggested in the previous section are developed in section 4. Computational results are discussed in section 5 and a final section concludes.

2 Definitions and notations

We consider the following general multiobjective program (MOP):

$$\begin{cases} \max(z_i = f_i(x)) \quad i = 1, \ldots, k \\ x \in X \end{cases} \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is the feasible set and f_i, $i = 1, \ldots, k$, are real-valued functions.

Problem (1) is called a multiobjective linear program (MOLP) if:

$$f_i(x) = c^i x \quad \forall\, i = 1, \ldots, k$$

and

$$X = \{x \in \mathbb{R}^n \mid Ax = b,\, x \geq 0\}$$

where the $k \times n-$matrix $C = \left(c^i\right)_{i=1,\ldots,k}$, the $m \times n-$matrix A and the $m-$vector b are real.

Moreover, if the variables are integers, we obtain a multiobjective integer linear program (MOILP):

$$\begin{cases} \max Cx \\ x \in X \\ x \text{ integer} \end{cases} \tag{2}$$

where we assume that all components of the matrix C, the matrix A and the vector b are integers and the feasible set in the decision space X is a nonempty, compact polyhedron set.

The set of all feasible criterion vectors Y of MOLP problem is defined as follows:

$$Y = \left\{z \in \mathbb{R}^k \mid z_i = c^i x,\, x \in X\right\} = f(X) \tag{3}$$

The set Y is also nonempty and compact.

An integer solution $x \in X$ is called *efficient* if there does not exist another integer solution $\overline{x} \in X$ such that $c^i\overline{x} \geq c^i x$ for all $i = 1, \ldots, k$ with strict inequality for at least one of the objectives. The criterion vector $z = (z_1, \ldots, z_k)$ corresponding to x is called *non-dominated* vector. The set of efficient (also said Pareto optimal) solutions of MOILP problem will be denoted by E, the set of non-dominated vectors by ND throughout the paper.

Let $\Lambda = \left\{\lambda \in \mathbb{R}^k \mid \lambda > 0,\, \sum\limits_{i=1}^{k} \lambda_i = 1\right\}$ be the set of all positive weighting vectors. Then, for a fixed $\lambda \in \Lambda$, the weighted-sum program corresponding to MOLP is given by

$$\begin{cases} \max Z = \sum\limits_{i=1}^{k} \lambda_i c^i x \\ x \in X \end{cases} \tag{4}$$

It is well known, see for examlpe Steuer [11], that:
If x^* is an optimal solution of (4), then x^* is efficient for the MOLP problem. If x^* is an efficient solution for the MOLP problem, then there exists $\overline{\lambda} \in \Lambda$ such that x^* is optimal for (4).

But the last result does not generalize the nonlinear or discrete MOLPs since there are other efficient solutions, called nonsupported efficient solutions, which

cannot be found using any weighted-sum of objectives. The optimal solutions of (4) are called supported efficient solutions.

To enumerate the non-dominated set of the MOILP problem (2), we define a linear program (P_l), $l \geq 0$, as follows:

$$\begin{cases} \max Z = \sum_{i=1}^{k} c^i x \\ x \in X_l \end{cases} \tag{5}$$

where $X_0 = X$. k lines are added to the classical simplex tableau, each line corresponding to one criterion, in order to follow the criteria evolution. Let us note x_l^* the first integer solution obtained after solving (P_l) by the simplex method and, if necessary, the branching process, we then define the following sets:

$$H_l^i = \left\{ j \in N_l \,|\, \widehat{c}_j^i > 0 \right\} \tag{6}$$

where N_l, is the indices set of non-basic variables and $\widehat{c}_j^i$, is the component j of the reduced cost vector of the i^{th} criterion,

$$K_l = \left\{ i \in \{1, \ldots, k\} \mid H_l^i \neq \varnothing \right\} \tag{7}$$

indicates the set of criteria that could be improved and

$$X_{l+1} = \bigcup_{i \in K_l} \left\{ x \in X_l \,|\, c^i x \geq c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \left\lfloor \widehat{c}_j^i \right\rfloor x_j + \max\left\{1, \left\lfloor \widehat{c}_{j_0}^i \right\rfloor\right\} \right\} \tag{8}$$

where $\widehat{c}_{j_0}^i = \min_{j \in H_l^i} \left\{ \widehat{c}_j^i \right\}$ for $i \in K_l$, $\lfloor a \rfloor$ indicates the greatest integer smaller than a.

As we can notice, the construction of the set X_{l+1} is based on the following constraint that can be regarded as a cutting plane for the MOILP problem:

$$c^i x \geq c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \left\lfloor \widehat{c}_j^i \right\rfloor x_j + \max\left\{1, \left\lfloor \widehat{c}_{j_0}^i \right\rfloor\right\} \tag{9}$$

We define a constraint as *an efficient cutting plane* if it doesn't eliminate non-dominated integer solutions.

3 Principle of the Method

The proposed approach generates all the non-dominated integer solutions without computing all the feasible ones. It is based on a branching process and uses efficient cutting planes. At each stage, a LP problem is solved and the criteria evolve in a dynamic way in the corresponding augmented simplex table. Then, a branching process is carried out to detect an integer solution. Recall that

unlike other existing methods, ours doesn't need to search for an optimal integer solution, but only a nearby feasible integer one of the continuous optimal solution. When such a solution is obtained, the corresponding criterion vector is compared to those already found. The increasing directions of the criteria are used to build efficient cutting planes in order to avoid the exploration of domains containing only dominated integer solutions.

All operations described below are identified in nodes and branches in a structured tree. A node l of the tree is saturated if the corresponding program (P_l) is not feasible or if $K_l = \varnothing$.

If the optimal solution x of program (P_l) is not integer, let $x_j = \alpha_j$ be one fractional coordinate. The node l is then separated into two nodes which are imposed by the additional constraints respectively: $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$. This process is carried out until an integer feasible solution x_l^* is reached. Then, for each criterion i in the set K_l, we have to solve a new program (P_k), $k > l$, after adding the constraint $c^i x \geq c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \widehat{c}_j^i x_j + \max\left\{1, \widehat{c}_{j_0}^i\right\}$, where $\widehat{c}_{j_0}^i = \min_{j \in H_l^i} \left\{\widehat{c}_j^i\right\}$, which ensures an improvement for the i^{th} criterion to search for non-dominated solutions.

Algorithm

Step 1. (*Initialization*)
$X_0 := X$, $l := 0$ and $ND := \varnothing$; (non-dominated set of problem (P)).
Solve the linear program (P_0) at node 0 and let x be an optimal solution.
If x is not integer, go to STEP 2a, else go to STEP 2b.

Step 2 (*General Step*)
As long as there is an unsaturated node in the tree, do:
Choose the first created node l of the tree, not yet saturated and solve the corresponding linear program (P_l). If program (P_l) has not feasible solutions, then the corresponding node l is saturated. Else, let x be an optimal solution. If x is not integer, go to STEP 2a. Else, go to STEP 2b.

Step 2a Choose one coordinate x_j of x such that $x_j = \alpha_j$, with α_j fractional number, and separate the current node l of the tree into two nodes: add the constraint $x_j \leq \lfloor \alpha_j \rfloor$ in the first node, the constraint $x_j \geq \lfloor \alpha_j \rfloor + 1$ in the second node and go to STEP 2.

Step 2b Let x_l^* be the integer solution. If Cx_l^* is not dominated by z, for all solution $z \in ND$, then $ND := ND \cup \{Cx_l^*\}$. If it exists $z \in ND$ such that z is dominated by Cx_l^*, then $ND := ND \backslash \{z\} \cup \{Cx_l^*\}$. Determine the sets N_l and K_l.
If $K_l = \varnothing$, then the corresponding node l is saturated, go to STEP 2.
Else, for each index $i \in K_l$, add the constraint $c^i x \geq c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \left\lfloor \widehat{c}_j^i \right\rfloor x_j$

$+\max\left\{1,\ \lfloor \widehat{c}^i_{j_0} \rfloor\right\}$ to obtain $|K_l|$ new programs (P_k), $k > l$.
Go to STEP 2.

4 Main Result

In this section, justifications of steps described in the above method are established. The following result shows that, at each step l of the method, no non-dominated solution can be deleted when we consider the set $X_{l+1} \subset X_l$. Consider the optimal simplex tableau with the integer solution x^*_l.

Theorem 1. *Let x be an integer solution in the set X_l such that $Cx \neq Cx^*_l$. If the vector Cx is non-dominated then, x is located in the set X_{l+1}.*

Proof. Let x be an integer solution in the set X_l then, from the current optimal simplex tableau one can write the following:

$$c^i x = c^i x^*_l + \sum_{j \in N_l} \widehat{c}^i_j x_j, \quad \forall i \in \{1, \ldots, k\} \tag{10}$$

If $i \in \{1, \ldots, k\} \setminus K_l$ then, $\widehat{c}^i_j \leq 0\ \forall j \in N_l$. Hence, $\sum\limits_{j \in N_l} \widehat{c}^i_j x_j \leq 0$ and $c^i x \leq c^i x^*_l$.

If $i \in K_l$ then:

$$c^i x = c^i x^*_l + \sum_{j \in N_l \setminus H^i_l} \widehat{c}^i_j x_j + \sum_{j \in H^i_l} \widehat{c}^i_j x_j \tag{11}$$

Suppose that $x \notin X_{l+1}$ then,

$$x \notin \left\{ x \in X_l \,|\, c^i x \geq c^i x^*_l + \sum_{j \in N_l \setminus H^i_l} \left\lfloor \widehat{c}^i_j \right\rfloor x_j + \max\left\{1,\ \left\lfloor \widehat{c}^i_{j_0} \right\rfloor\right\} \right\} \quad \forall i \in K_l.$$

Because the solution x belongs to the set X_l, the following inequality is true:

$$c^i x < c^i x^*_l + \sum_{j \in N_l \setminus H^i_l} \left\lfloor \widehat{c}^i_j \right\rfloor x_j + \max\left\{1,\ \left\lfloor \widehat{c}^i_{j_0} \right\rfloor\right\}, \forall i \in K_l \tag{12}$$

In the other hand,

$$c^i x \geq c^i x^*_l + \sum_{j \in N_l \setminus H^i_l} \left\lfloor \widehat{c}^i_j \right\rfloor x_j + \sum_{j \in H^i_l} \left\lfloor \widehat{c}^i_j \right\rfloor x_j, \forall i \in K_l \tag{13}$$

Using the two last inequalities, we obtain:

$$\sum_{j \in H^i_l} \left\lfloor \widehat{c}^i_j \right\rfloor x_j < \max\left\{1,\ \left\lfloor \widehat{c}^i_{j_0} \right\rfloor\right\}, \forall i \in K_l$$

1st case :

Suppose that for a given $i \in K_l$, we have: $\max\left\{1,\ \left\lfloor \widehat{c}^i_{j_0} \right\rfloor\right\} = \left\lfloor \widehat{c}^i_{j_0} \right\rfloor$

Hence:

$$\sum_{j \in H_l^i} \frac{\lfloor \widehat{c}_j^i \rfloor}{\lfloor \widehat{c}_{j_0}^i \rfloor} x_j < 1$$

and according to the definition of $\widehat{c}_{j_0}^i$, $\frac{\lfloor \widehat{c}_j^i \rfloor}{\lfloor \widehat{c}_{j_0}^i \rfloor} \geq 1, \forall j \in H_l^i$, indicating that $x_j = 0$, $\forall j \in H_l^i$.

Consequently,

$$c^i x = c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \widehat{c}_j^i x_j$$

and hence:

$$c^i x \leq c^i x_l^*$$

2nd case :

Suppose that for a given $i \in K_l$: $\max\left\{1, \lfloor \widehat{c}_{j_0}^i \rfloor\right\} = 1$

Using the inequality (12), we can write:

$$c^i x < c^i x_l^* + \sum_{j \in N_l \setminus H_l^i} \lfloor \widehat{c}_j^i \rfloor \, x_j + 1$$

which implies that:

$$c^i x \leq c^i x_l^*$$

We conclude that:

$$Cx \leq Cx_l^*$$

Now, taking account of the assumption that $Cx \neq Cx_l^*$, we can say that the vector Cx is dominated by the vector Cx_l^*. □

Corollary 1. *The constraint (9) is an efficient cutting plane.*

Proof. It is clear that (9) is an efficient valid constraint since all integer efficient solutions in the current domain X_l check this constraint according to the above theorem. In the other hand, replacing x by x_l^* in (9) leads to an absurdity. Hence, the current integer solution x_l^* does not satisfy (9). In conclusion, we can say that the constraint (9) is an efficient cutting plane. □

Theorem 2. *The algorithm described below generates all the non-dominated solutions and terminates after a finite number of iterations.*

Proof. The feasible solutions set X being compact, it contains a finite number of integer solutions. At each step l of the algorithm, one determines an integer solution x_l^* when it exists. By taking account of the first theorem and the corresponding corollary, the solution $x_l^* \in X_l$ is eliminated as well as a subset of integer solutions with dominated criterion vectors when X_{l+1} is considered. Otherwise, the set K_l is empty, showing that no criterion can be improved and the corresponding node can be saturated. □

5 Computational results

Our method as well as Sylva & Crema's one were implemented in a Matlab program and tested on randomly generated multiobjective integer linear programming problems with two and four objective functions. The program was run on a 3.6 GHz Pentium VI processor. Objective functions and constraint coefficients are uncorrelated integers uniformly distributed between -10 and 99. For each constraint, the right-hand side value is set to $\delta\%$ of the sum of its coefficients where $\delta \in \{50, 33.33, 25, 20\}$. For each instance (n, m, k), a serie of 20 problems were solved and the complete non-dominated set ND was generated for all these problems.

The results show that our method is better than Sylva & Crema's one in CPU time and memory space used since it's 15 times faster, makes thrice less simplex iterations and generates almost 4 times fewer nodes on average.
It can also be seen that our method solves more ILP problems since, in average, it solves 213 times more ILP problems than Sylva & Crema's method. But, taking into account the first result, we can explain this phenomenon by the great size of the ILP problems solved by Sylva & Crema's method, $k + 1$ constraints and k variables are added with each non-dominated solution generated, whereas those of our method are of much smaller size. Is it not known that "divide to conquer?".

In the other hand, the computational results in [12] show clearly that only examples giving few non-dominated solutions are reported by the authors since the CPU time is too long for problems having high cardinality of the non-dominated set.

Instances			CPU time (s)		Simplex iterations		Nodes		ILPs		$\|ND\|$	
n	m	k	mean	max	mean	max	mean	max	mean	max	mean	max
10	5	2	2.42	4.79	11593.70	19028	1985.80	3316	632.20	2077	12.70	19
10	5	4	1.85	2.33	8890.90	10962	1604.20	2144	363	689	22.50	29
15	5	2	14.56	17.87	73514.70	94325	9479.60	12844	1616.10	3007	8.70	15
15	5	4	16.63	29.08	78295.90	123688	9929.20	15660	2183.20	3673	14.20	19
15	10	2	10.42	13.43	46891.70	61643	5489.20	7230	761.10	1237	9	18
15	10	4	10.06	12.51	41578	52343	4638.20	6150	1325.50	1723	13.70	21
20	5	2	73.11	112.40	340615.30	538587	35255.20	57696	4568	6657	10.80	16
20	5	4	112.48	152.41	384403.30	503130	37741.40	50564	8510.50	13134	9.80	15
20	10	2	40.96	50.65	164056.40	207198	14656.60	19030	1842.10	2665	8.50	13
25	5	2	326.22	714.84	1307155	2384816	107830	198782	11349.60	15667	6.50	11
20	10	4	67.18	98.42	196713.60	294863	16942.20	25066	4421.10	7254	79.80	112
25	10	4	219.87	268.81	570774.80	713534	40738.60	50778	8927.10	10972	74.20	117
30	10	4	664.21	859.06	1532269.10	2044898	94314.60	129766	18476.70	23373	93.40	171

Table 1: The results obtained by our method

Instances			CPU time (s)		Simplex iterations		Nodes		ILPs		$\|ND\|$	
n	m	k	mean	max	mean	max	mean	max	mean	max	mean	max
10	5	2	51.52	147.22	70153.20	141764	13615.20	30642	13.70	20	12.70	19
10	5	4	966.71	2086.60	187561.30	263698	31157	41722	23.50	30	22.50	29
15	5	2	121.88	295.56	192683.60	284717	29259.60	40102	9.70	16	8.70	15
15	5	4	1457.09	2792.39	685757.70	1193515	86266.80	183884	15.20	20	14.20	19
15	10	2	160.57	695.11	186601.60	512210	23268.40	56198	10	19	9	18
15	10	4	1050.86	2798.83	361024.90	694283	37592	64470	14.70	22	13.70	21
20	5	2	1027.58	2327.26	1174943.40	1924930	133722	206460	11.80	17	10.80	16
20	5	4	3881.56	9160.68	1714838.50	3898105	179433.40	373924	10.80	16	9.80	15
20	10	2	458.31	913.09	552355.10	863246	55655.40	85004	9.50	14	8.50	13
25	5	2	2231.01	4961.45	3176294.80	5730612	303307	565600	7.50	12	6.50	11

Table 2: The results obtained by Sylva & Crema's method

6 Conclusion

In this paper, a new exact method combining the well-known principle of branching in integer linear programming with a new efficient cut is described to generate all integer non-dominated solutions of a MOILP problem, without solving any ILP problem along the steps of the algorithm. The added cuts are built from the simplex tableau giving an integer solution, and strongly dependent on the criteria vectors. The method is dedicated to general MOLP problems with integer as well as zero-one decision variables. Compared to the Sylva & Crema's one, our method is better in CPU time and memory space since it solves linear programs of small size, and is not influenced by the number of criteria. In the other hand, the tree structure of the proposed algorithm can be parallelized in order to allow the resolution of big size problems.

Acknowledgement

This work was supported by the Laboratory LAID3, USTHB.

References

[1] M. J. Alves, J. Clímaco. A review of interactive methods for multiobjective integer and mixed-integer programming, European Journal of Operational Research 180 (2007) 99-115

[2] K. Bhaskar. A Multiple Objective Approach to Capital Budgeting, Accounting and Business Research 9 (1979) 25-46

[3] J. Clímaco, C. Ferreira, M. Captivo. Multicriteria integer programming: An overview of the different algorithmic approaches, multicriteria integer programming, J. Climaco ed, Multicriteria Analysis, Springer, Berlin, (1997) 248-258

[4] M. Ehrgott, J. Figueira, X. Gandibleux (editors). Multiobjective Discrete and Combinatorial Optimization, Annals of Operations Research 147, 2006

[5] C. Ferreira, J. Clímaco, J. Paixão. The location-covering problem: a bicriterion interactive approach, Investigación Operativa 4(2) (1994) 119-139

[6] D.T. Galligan, J.D. Ferguson. Application of Linear Programming in Bull Selection for a Dairy Herd, JAVMA 206 (1995) 173-176

[7] H. Isermann. Proper Efficiency and the Linear Vector Maximum Problem, Operations Research 22 (1974) 189-191

[8] P. Kere, J. Koski. Multicriterion optimization of composite laminates for maximum failure margins with an interactive descent algorithm, Structural and Multidisciplinary Optimization 23(6) (2002) 436-447

[9] D. Klein, E. Hannan. An Algorithm for Multiple Objective Integer Linear Programming Problem, European Journal of Operational Research 9 (1982) 378-385

[10] M.B. McConnel, D.T. Galligan. The Use of Integer Programming to Select Bulls Across Breeding Companies with Volume Price Discounts, J. Dairy Sci. 87 (2004) 3542-3549

[11] R.E. Steuer. Multiple Criteria Optimization: Theory, Computation and Applications, Wiley, 1985

[12] J. Sylva, A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs, European Journal of Operational Research 158(1) (2004) 46-55

[13] J. Teghem, P. Kunsch. A survey of techniques to determine the efficient solutions to multi-objective integer linear programming, Asia Pacific Journal of Operations Research 3 (1986) 95-108

[14] P.R. Tozer, J.R. Stokes. Using Multiple Objective Programming in a Dairy Cow Breeding Program, J. Dairy Sci. 84 (2001) 2782-2788

[15] E. L. Ulungu, J. Teghem. Multi-objective Combinatorial Optimization Problem: A Survey, Journal of Multi-Criteria Decision Analysis 3 (1994) 83-104

[16] E.L. Ulungu, J. Teghem. The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems, Foundations of Computing and Decision Sciences 20(2) (1994) 149-165

[17] M. Visee, J. Teghem, M. Pirlot, E.L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem, Journal of Global Optimization 12 (1998) 139-155

Sur le nombre d'alliance offensive globale dans les arbres

Mohamed Bouzefrane et Mustapha Chellali
Laboratoire LAMDA-RO, Département de Mathématiques
Université de Blida.
B.P. 270, Blida, Algérie
E-mail: m_bouzefrane@yahoo.fr; m_chellali@yahoo.com

Résumé. Pour un graphe $G = (V, E)$, un sous-ensemble $S \subseteq V$ est un dominant si chaque sommet dans $V - S$ a au moins un voisin dans S. Un dominant S est une alliance offensive globale si pour chaque sommet v dans $V - S$, au moins la moitié des sommets de son voisinage fermé est dans S. Le nombre de domination $\gamma(G)$ est le cardinal minimum d'un ensemble dominant de G et le nombre d'alliance offensive globale $\gamma_o(G)$ est le cardinal minimum d'une alliance offensive globale de G. Nous montrons d'abord que chaque arbre T d'ordre au moins trois avec ℓ feuilles et s sommets supports satisfait $\gamma_o(T) \geq (n - \ell + s + 1)/3$, et on caractérise les arbres extrémaux atteignants cette borne inférieure . On donne par la suite aussi une caractérisation des arbres pour lesquels les nombres de domination et d'alliance offensive globale sont égaux.

Mots-clés: nombre d'alliance offensive globale, nombre de domination, arbres.

Classification AMS : 05C69

1 Introduction

Dans la vie réelle, une alliance est un groupe d'entités qui s'unissent pour une cause commune. Les applications des alliances sont répandues aux coalitions de défense nationales, d'associations sociales et d'affaires. Motivé par ces diverses applications, Hedetniemi, Hedetniemi, et Kristiansen [5] ont présenté plusieurs types d'alliances dans les graphes, parmi lesquelles les alliances offensives globales considérées dans cet article. Les alliances offen-

sive ont été considérées par exemple dans [1] et [2]. Pour plus de détails sur la domination dans les graphes et ses variations, voir les livres de Haynes, Hedetniemi et Slater [3], [4]. Pour mieux définir les alliances, nous avons besoin de quelques définitions supplémentaires.

Dans un graphe simple $G = (V, E)$ d'ordre n, le *voisinage ouvert* d'un sommet $v \in V$ est $N_G(v) = N(v) = \{u \in V \mid uv \in E\}$ et le *voisinage fermé* de v est $N_G[v] = N[v] = N(v) \cup \{v\}$. Le *degré* d'un sommet v de G noté par $\deg_G(v)$ est le cardinal de son voisinage ouvert. Un sommet de degré un est appelé un *sommet pendant* ou une *feuille* et son voisin est appelé un *sommet support*. Si v est un sommet support, alors L_v désigne l'ensemble des feuilles attachées à v. On note aussi l'ensemble des feuilles d'un graphe G par $L(G)$, l'ensemble des sommets supports par $S(G)$, et soit $|L(G)| = \ell$, $|S(G)| = s$. Un arbre T est une *double étoile* s'il contient exactement deux sommets qui ne sont pas des feuilles. Une *étoile subdivisée* SS_q est obtenue à partir d'une étoile $K_{1,q}$ en subdivisant chaque arête exactement par un sommet.

Pour un graphe $G = (V, E)$, un sous-ensemble S de V est un *dominant* si tout sommet dans $V - S$ a au moins un voisin dans S. Un ensemble dominant S est dit *alliance offensive globale (aog)* si pour tout sommet $v \in V - S$, $|N[v] \cap S| \geq |N[v] - S|$. Le *nombre de domination* $\gamma(G)$ est le cardinal minimum d'un ensemble dominant de G, et le *nombre d'alliance offensive globale* $\gamma_o(G)$ est le cardinal minimum d'une alliance offensive globale de G. Il est claire que pour tout graphe G, $\gamma_o(G) \geq \gamma(G)$ et que tout graphe a une alliance offensive globale, puisque $S = V$ est un tel ensemble.

Dans cet article, on montre que tout arbre T d'ordre au moins trois avec ℓ feuilles et s sommets supports satisfait $\gamma_o(T) \geq (n - \ell + s + 1)/3$. D'autre part on caractérise les arbres extrémaux atteignant cette borne inférieure et on donne aussi une caractérisation des arbres pour lesquels le nombre de domination et le nombre d'alliance offensive globale sont égaux..

2 Borne inférieure

On commence par donner un couple d'observations.

Observation 1 *Si G est un graphe connexe d'ordre au moins trois, alors il existe un $\gamma_o(G)$-ensemble qui contient tous les sommets supports.*

Observation 2 *Soit T un arbre obtenu à partir d'un arbre non trivial T' en attachant l'ètoile $K_{1,t}$ de centre x par une arête xz à un sommet support z de T'. Alors $\gamma_o(T) = \gamma_o(T') + 1$ et $\gamma(T) = \gamma(T') + 1$.*

Preuve. Par l'Observation 1 il y'a un $\gamma_o(T)$-ensemble D qui contient tous les sommets supports, d'ou $x, z \in D$. Donc $D - \{x\}$ est un aog de T' et $\gamma_o(T') \leq \gamma_o(T) - 1$. Puisque tout $\gamma_o(T')$-ensemble peut être étendu en une aog de T en ajoutant le sommet x, alors $\gamma_o(T) \leq \gamma_o(T') + 1$. Ce qui nous donne $\gamma_o(T) = \gamma_o(T') + 1$. D'autre part si D' est un $\gamma(T')$-ensemble, alors $D' \cup \{x\}$ est un ensemble dominant de T, ce qui implique que $\gamma(T) \leq \gamma(T') + 1$. L'égalité est atteinte par le fait que x, z appartiennent à un $\gamma(T)$-ensemble et un tel ensemble sans le sommet x est un dominant de T'. ■

Soit $\mathcal{F}$ la famille des arbres d'ordre aux moins trois qui peuvent être obtenus de r étoiles disjointes en ajoutant premièrement $r - 1$ arêtes incidentes seulement avec les centres des étoiles, le graphe ainsi obtenu est connexe, ensuite en subdivisant chaque nouvelle arête exactement une seule fois.

Théorème 3 *Soit T un arbre d'ordre $n \geq 3$ avec ℓ feuilles et s sommets supports. Alors $\gamma_o(T) \geq (n-\ell+s+1)/3$ avec l'egalitè atteinte si et seulement si $T \in \mathcal{F}$.*

Preuve. Soit $T \in \mathcal{F}$. Alors T contient $|S(T)| - 1$ sommets de degré deux et le reste des sommets sont des feuilles et des supports. Il s'ensuit que $n = \ell+2s-1$ et donc $s = (n-\ell+s+1)/3$. Maintenant il est claire que chaque $\gamma_o(T)$-ensemble contient au moins $|S(T)|$ sommets et donc $\gamma_o(T) \geq |S(T)|$. L'égalité est obtenu par le fait que $S(T)$ constitue une alliance offensive globale de T, ce qui implique que $\gamma_o(T) = |S(T)| = (n - \ell + s + 1)/3$.

Pour prouver que si T est un arbre d'ordre $n \geq 3$, alors $\gamma_o(T) \geq (n - \ell + s + 1)/3$ avec égalité seulement si $T \in \mathcal{F}$, on utilise une induction sur l'ordre n. Si le diam$(T) = 2$, alors T est une étoile avec $\gamma_o(T) = 1 = (n-\ell+s+1)/3$ et donc $T \in \mathcal{F}$. Si le diam$(T) = 3$, alors $\gamma_o(T) = 2 > (n - \ell + s + 1)/3$. On suppose maintenant que tout arbre T' d'ordre $3 \leq n' < n$ avec ℓ' feuilles et s' sommets supports satisfait $\gamma_o(T') \geq (n' - \ell' + s' + 1)/3$ avec l'égalité atteinte si et seulement si $T \in \mathcal{F}$. Soit T un arbre d'ordre n et de diamètre au moins quatre, ayant ℓ feuilles et s sommets supports.

Nous enracinons T à un sommet r d'excentricité maximum diam$(T) \geqslant 4$. Soit u un sommet support à distance maximum de r, v le père de u, et w le père de v dans l'arbre enraciné . Notons que $\deg_T(w) \geq 2$. Soit D un $\gamma_o(T)$-ensemble ne contenant aucune feuille. Notons par T_x le sous arbre induit par le sommet x et ses descendants dans l'arbre enraciné T. Nous distinguons entre trois cas.

Cas 1. v est un sommet support . Soit $T' = T - L_u \cup \{u\}$. Alors $n' = n - 1 - |L_u| \geq 3$, $\ell' = \ell - |L_u|$ et $s' = s - 1$. Par l'Observation 2, $\gamma_o(T) = \gamma_o(T') + 1$ et par induction sur T', on obtient $\gamma_o(T) > (n - \ell + s + 1)/3$.

Cas 2. v n'est pas un sommet support mais $\deg_T(v) \geq 3$. Ainsi tout fils de v est un sommet support . Soit k le nombre de fils de v et B l'ensemble des feuilles dans T_v. On suppose d'abord que $\deg_T(w) \geq 3$ et soit $T' = T - T_v$. Alors $n' = n - |B| - k - 1 \geq 3$, $\ell' = \ell - |B|$ et $s' = s - k$. Puisque D contient tous les fils de v et ne contient pas v (sinon remplacer v par w), $D \cap V(T')$ est une aog de T'. Il s'ensuit que $\gamma_o(T') \leq \gamma_o(T) - k$ et par induction sur T' on a

$$\gamma_o(T) \geq (n' - \ell' + s' + 1)/3 + k \geq (n - \ell + s + 1 + k - 1)/3.$$

Donc $\gamma_o(T) > (n - \ell + s + 1)/3$ puisque $k \geq 2$.

Maintenant on suppose que $\deg_T(w) = 2$ et soit $T' = T - (T_v - \{v\})$. Alors $n' = n - |B| - k \geq 3$, $\ell' = \ell - |B| + 1$ et $s' = s - k + 1$. Il est claire que D contient tous les fils de v et ne contient pas v (sinon remplacer v par w) et donc D doit contenir w. Ainsi $D \cap V(T')$ est une aog de T' et donc $\gamma_o(T') \leq \gamma_o(T) - k$. Par induction sur T' on a

$$\gamma_o(T) \geq (n' - \ell' + s' + 1)/3 + k \geq (n - \ell + s + 1 + k)/3$$

et donc $\gamma_o(T) > (n - \ell + s + 1)/3$.

Cas 3. $\deg_T(v) = 2$. Alors $u, w \in D$ et $v \notin D$. Supposons que $\deg_T(w) = 2$ ou $\deg_T(w) \geq 3$ et w n'est pas un sommet support. Soit $T' = T - L_u \cup \{u\}$. Alors $D \cap V(T')$ est une aog de T' et donc $\gamma_o(T') \leq \gamma_o(T) - 1$. En utilisant l'induction sur T' et puisque $n' = n - 1 - |L_u| \geq 3$, $\ell' = \ell - |L_u| + 1$ et $s' = s$, on obtient

$$\gamma_o(T) \geq (n' - \ell' + s' + 1)/3 + 1 > (n - \ell + s + 1)/3.$$

On suppose enfin que $\deg_T(w) \geq 3$ et w est un sommet support. Soit $T' = T - L_u \cup \{u, v\}$. Alors $D \cap V(T')$ est une aog de T', $n' = n - 2 - |L_u| \geq 3, \ell' = \ell - |L_u|$ et $s' = s - 1$. D'ou par induction sur T', on a

$$\gamma_o(T) \geq \gamma_o(T') + 1 \geq (n' - \ell' + s' + 1)/3 + 1 \geq (n - \ell + s + 1)/3.$$

De plus si $\gamma_o(T) = (n - \ell + s + 1)/3$, alors on a l'ègalité le long de cette chaine d'inégalités. En particulier $\gamma_o(T') = (n' - \ell' + s' + 1)/3$. Ainsi par l'hypothèse d'induction sur T', $T' \in \mathcal{F}$. Il s'ensuit que $T \in \mathcal{F}$. ■

3 Arbres T avec $\gamma_o(T) = \gamma(T)$

Observation 4 *SoitT un arbre obtenu à partir d'un arbre non trivial T' en attachant une étoile subdivisée SS_k, $k \geq 2$, de centre x par une arête xy à un sommet y de T'. Alors*

a) $\gamma_o(T') \leq \gamma_o(T) - k$, *avec l'égalité atteinte si y appartient à un $\gamma_o(T')$-ensemble ou une majorité stricte de son voisinage fermé appartient à un $\gamma_o(T')$-ensemble.*

b) $\gamma(T) = \gamma(T') + k$.

Preuve. (a) Par l'Observation 1 Il y a un $\gamma_o(T)$-ensemble S qui contient tous les sommets supports de l'étoile subdivisée ajoutée. Aussi nous pouvons supposer que $x \notin S$ (sinon le remplacer par y). Ainsi $S \cap V(T')$ est une aog de T' et donc $\gamma_o(T') \leq \gamma_o(T) - k$. Maintenant si y appartient à un $\gamma_o(T')$-ensemble ou la majorité stricte de son voisinage appartient à un certain $\gamma_o(T')$-ensemble, alors de de tels ensembles peuvent être étendus à une aog de T en ajoutant l'ensemble des sommets supports de SS_k. Il s'ensuit que $\gamma_o(T) \leq \gamma_o(T') + k$ et l'égalité sera donc atteinte.

(b) est facile à montrer. ■

Afin de caractériser les arbres pour lesquels les nombres de domination et d'alliance offensive globale sont égaux; nous définissons la famille $\mathcal{F}$ de tous les arbres T qui peuvent être obtenus à partir d'une séquence T_1, T_2, ..., T_k ($k \geq 1$) d'arbres, où $T_1 = P_2$, $T = T_k$, et si $k \geq 2$, T_{i+1} est obtenu récursivement à partir deT_i par l'une des quatre opérations définies ci-dessous. Considérons l'un des sommets de T_1 un support et l'autre une feuille.

- **Opération** $\mathcal{O}_1$: Attacher un nouveau sommet en le joignant à un sommet support quelconque de T_i.
- **Opération** $\mathcal{O}_2$: Attacher une chaine $P_2 = xy$ en joignant x à un sommet support z quelconque de T_i .
- **Opération** $\mathcal{O}_3$: Attacher une étoile subdivisée SS_k, $k \geq 2$, de centre u en joignant u à un sommet v de T_i avec la condition que si v n'appartient pas à un $\gamma_o(T_i)$-ensemble D, alors la majorité stricte de $N_{T_i}[v]$ sont dans D.
- **Opération** $\mathcal{O}_4$: Attacher une chaine $P_3 = xyz$ en joignant x à un sommet quelconque de T_i appartenant à un $\gamma_o(T_i)$-ensemble.

Lemme 5 *SiT* $\in \mathcal{F}$, *alors* $\gamma_o(T) = \gamma(T)$.

Preuve. On utilise une induction sur le nombre d'opérations k exécutées pour construire T. La propriété est vraie pour $T_1 = P_2$. Supposons que la propriété est vraie pour tous les arbres de $\mathcal{F}$ construits par $k-1 \geq 0$ opérations. Soit $T = T_k$ avec $k \geq 2$, $T' = T_{k-1}$, et soit D un $\gamma_o(T)$-ensemble qui ne contient aucune feuille de T. Nous examinons les cas suivants:

Il est claire que si T est obtenu à partir de T' par l'Opération $\mathcal{O}_1$, alors $\gamma_o(T') = \gamma_o(T)$, $\gamma(T') = \gamma(T)$ et donc $\gamma_o(T) = \gamma(T)$.

Si T est obtenu à partir de T' par l'opération $\mathcal{O}_2$, alors par l'Observation 2, $\gamma_o(T) = \gamma_o(T') + 1$et $\gamma(T) = \gamma(T') + 1$. En Utilisant l'induction sur T' on trouve que $\gamma_o(T) = \gamma(T)$.

Si T est obtenu à partir deT' par l'Opération $\mathcal{O}_3$, alors par l'Observation 4, $\gamma_o(T) = \gamma_o(T') + k$ et $\gamma(T) = \gamma(T') + k$. Par induction sur T', on a $\gamma_o(T) = \gamma(T)$.

Finallement on suppose que T est obtenu à partir de T' par l'Opération $\mathcal{O}_4$. Soit $w \in V(T')$ le voisin de x. Alors $y \in D$ et $x \notin D$ (sinon le remplacer par w). Ainsi $D \cap V(T')$ est une aog de T' et on a $\gamma_o(T') \leq \gamma_o(T) - 1$. Maintenant puisque w appartient à un $\gamma_o(T')$-ensemble, un tel ensemble peut être étendu à une aog de T en ajoutant y. Donc $\gamma_o(T) \leq \gamma_o(T') + 1$ et l'égalité est obtenue. Aussi on peut voir facilement que $\gamma(T) = \gamma(T') + 1$. Par induction sur T', nous obtenons le résultat désiré. ■

Théorème 6 *Soit T un arbre. Alors $\gamma_o(T) = \gamma(T)$ si et seulement si $T = K_1$ ou bien $T \in \mathcal{F}$.*

Preuve. Il est claire que si $T = K_1$, alors $\gamma_o(T) = \gamma(T)$. Si $T \in \mathcal{F}$, alors par le lemme 5, $\gamma_o(T) = \gamma(T)$. Maintenant pour prouver la condition nécessaire on utilise une induction sur l'ordre n de T. Il est évident que $\gamma_o(K_1) = \gamma(K_1)$ et donc on suppose que $n \geq 2$. Si $n = 2$, alors $T = P_2$ et T appartient à $\mathcal{F}$. Si $n = 3$, alors $T = P_3$ qui appartient à $\mathcal{F}$ puisque il est obtenu à partir de P_2 en utilisant l'Opération $\mathcal{O}_1$. Supposons que tout arbre T' d'ordre n' tel que $2 \leq n' < n$, satisfaisant $\gamma_o(T') = \gamma(T')$ est dans $\mathcal{F}$.

Soit T un arbre d'ordre n tel que $\gamma_o(T) = \gamma(T)$. Si T est une étoile $K_{1,t}$, alors $\gamma_o(T) = \gamma(T)$ et $T \in \mathcal{F}$ parcequ'elle est obtenue à partir de P_2 en utilisant l'Opération $\mathcal{O}_1$. Si T est une étoile double, alors $\gamma_o(T) = \gamma(T)$ et $T \in \mathcal{F}$ car elle est obtenue à partir de P_2 en utilisant l'Opération $\mathcal{O}_2$ et $\mathcal{O}_1$. Ainsi on peut supposer que T est de diamètre au moins quatre.

Si un sommet support de T, disons x, est adjacent à deux feuilles ou plus, alors soit T' l'arbre obtenu à partir de T en supprimant une feuille

adjacente à x. Alors $\gamma_o(T') = \gamma_o(T)$, $\gamma(T') = \gamma(T)$ et donc $\gamma_o(T') = \gamma(T')$. Par induction sur T', on a $T' \in \mathcal{F}$. Donc $T \in \mathcal{F}$ car il est obtenu par T' en utilisant l'Opération $\mathcal{O}_1$. Dorénavant on peut supposer que tout sommet support de$T$ est adjacent à exactement une feuille.

Nous enracinons maintenant T à un sommet r d'excentricité maximum, $\text{diam}(T) \geqslant 4$. Soit v un sommet support à distance maximum de r, u le père de v, et w le père de u dans l'arbre enraciné. Soit v' l'unique feuille adjacente à v. Notons que $\deg_T(w) \geq 2$. Soit D un $\gamma_o(T)$-ensemble qui ne contient aucune feuille. Nous distinguons entre trois cas.

Cas 1. u est un sommet support . Soit $T' = T - \{v, v'\}$. Alors par l'Observation 2, $\gamma_o(T) = \gamma_o(T')+1$ et $\gamma(T) = \gamma(T')+1$. Ainsi $\gamma_o(T') = \gamma(T')$ et par induction sur T', on a $T' \in \mathcal{F}$. Il s'ensuit que $T \in \mathcal{F}$ et T est obtenu à partir de T' en utilisant l'Opération $\mathcal{O}_2$.

Cas 2. u n'est pas un sommet support, mais a au moins un fils en plus de v comme support. Ainsi T_v est une étoile subdivisée. Soit $T' = T - T_v$. Alors par l'Observation 4, $\gamma_o(T') \leq \gamma_o(T) - k$ et $\gamma(T) = \gamma(T') + k$, ou k est le nombre de fils de v. Supposons maintenant que $\gamma_o(T') < \gamma_o(T) - k$, alors

$$\gamma_o(T') < \gamma_o(T) - k = \gamma(T) - k = (\gamma(T') + k) - k = \gamma(T')$$

et donc $\gamma_o(T') < \gamma(T')$, une contradiction. D'ou $\gamma_o(T') = \gamma_o(T) - k$ et $D' = D \cap V(T')$ est un $\gamma_o(T')$-ensemble. Il s'ensuit que $\gamma_o(T') = \gamma(T')$. Notons que si $w \notin D'$, alors puisque D est un $\gamma_o(T)$-ensemble, $|N_{T'}[w] \cap D'| > |N_{T'}[w] - D'|$. Appliquons l'hypothèse d'induction sur T', $T' \in \mathcal{F}$, et donc $T \in \mathcal{F}$ parcequ'il est obtenu à partir de T' en utilisant l'Opération $\mathcal{O}_3$.

Cas 3. u est de degrè deux. Soit $T' = T - \{v', v, u\}$. On peut voir que $\gamma(T') = \gamma(T) - 1$. Aussi $v \in D$, $u \notin D$ (sinon le remplacer par w), et donc $w \in D$. Par conséquent $D \cap V(T')$ est une aog de T' et $\gamma_o(T') < \gamma_o(T) - 1$. Maintenant si $\gamma_o(T') < \gamma_o(T) - 1$, alors

$$\gamma_o(T') < \gamma_o(T) - 1 = \gamma(T) - 1 = (\gamma(T') + 1) - 1 = \gamma(T')$$

et ainsi $\gamma_o(T') < \gamma(T')$, une contradiction. Donc $\gamma_o(T') = \gamma_o(T) - 1$ et $D \cap V(T')$ est un $\gamma_o(T')$-ensemble qui contient w. Il s'ensuit que $\gamma_o(T') = \gamma(T')$ et par l'hypothèse d'induction sur T', $T' \in \mathcal{F}$. Par conséquent $T \in \mathcal{F}$ et il est obtenu à partir de T'en utilisant l'Opération $\mathcal{O}_4$. ■

References

[1] M. Chellali, Offensive alliances in bipartite graphs. *J. Combin. Math. Combin. Comput.* Accepté.

[2] O. Favaron, G. Fricke, W. Goddard, S. M. Hedetniemi, S. T. Hedetniemi, P. Kristiansen, R. C. Laskar et D. R. Skaggs, Offensive alliances in graphs. *Discuss. Math. Graph Theory* 24 (2)(2004) 263-275.

[3] T. W. Haynes, S. T. Hedetniemi et P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

[4] T. W. Haynes, S. T. Hedetniemi et P. J. Slater (eds), *Domination in Graphs: Advanced Topics*, Marcel Dekker, New York, 1998.

[5] S. M. Hedetniemi, S. T. Hedetniemi et P. Kristiansen, Alliances in graphs. *J. Combin. Math. Combin. Comput.* 48 (2004) 157–177.

A Simple Linear Time Algorithm for Bipartite Almost Distance-Hereditary Graphs Recognition

souad Slimani and Méziane Aïder

Mathematics Institut, USTHB, BP 32, El Alia,
Bab Ezzouar. 16 111, Algers, Algéria
{slimanisouad8}@gmail.com
{m-aider}@usthb.dz
http://www.USTHB.dz

Abstract. In a graph, the distance between two vertices represents the length of the shortest path connecting them. In a bipartite almost distance-hereditary graph, the length of the shortest $\{u, v\}$-path of G is at most equal to $d(u, v) + 2$. This class has been introduced by Aïder [2] who characterized them in terms of forbidden induced subgraphs.
In this paper, we use this characterization [2] to give two polynomials time algorithms to recognize these classes.

Key words: Distance-hereditary, Forbidden configurations , Bipartite graphs, Polynomial algorithms

1 Introduction

The notion of distance, has been used in graphs to construct the class of *'distance-hereditary graphs'*, which the distance between any two non-adjacent of any connected subgraph of such a connected graph is the same as the distance between these two vertices in the original graph. This class of graphs induces by Howorka [9], which established some properties and given the first characterization in terms of forbidden subgraphs. Other metric and algorithmic characterizations were montioned in [4, 7]. Adapting this concept to the bipartite graphs, Aïder [2] introduced the class *'almost bipartite distance-hereditary graphs'*, where the length of the shortest $\{u, v\}$-path of G is at most equal to $d(u, v) + 2$, who characterized it in terms of forbidden induced subgraphs. In this paper, we use this characterizations to study the recognition problem of this class.

1.1 Preliminaries

All graphs occurring in this paper are connected, finite,undirected, loopless and without multiple edges G=(V,E) where V is the vertex set and E is the edge set. Given a subset S of V, the induced subgraph $< S >$ of G is the maximal

subgraph of G with vertex set S. Denoted by $G - X$ (or $X \subseteq V$) the subgraph of G induced by $V - X$.

An induced path of G is a non redundant connection of a pair of vertices . The length of a shortest path between two vertices u and v is called distance and is denoted by $d_G(u, v)$; moreover the length of a longest induced path between the same vertices is denoted by $D_G(u, v)$.
A cycle C_n is a path $(x_1, x_2, ..., x_n, x_1)$. For convenience, the (x, y) part of a path P (respectively a cycle C) is denoted by $P(x, y)$ (respectively $C(u, v)$). A chord of a cycle C_n is an edge that joins non consecutive vertices of C_n. The chord distance of C_n, denoted by $cd(C_n)$, is defined to be the minimum number of consecutive vertices in C_n such that every chord of C_n is incident to some of such vertices. Each cycle on n vertices with chord distance equal either to 0 or to 1 will be referred to as a $1C_n$-configuration. The $1C_n$-biconfiguration is a bipartite $1C_n$-configuration. $P \cup Q$ induce a cycle denoted $C\{u, v\}$. The split composition of the graphs $G_1 = (V_1 \cup m_1, E_1)$ and $G_2 = (V_2 \cup m_2, E_2)$ with respect to m_1 and m_2 is the graph $G_1 * G_2$ having vertex set $V = V_1 \cup V_2$ and edge set $E = E'_1 \cup E'_2 \cup \{(u, v)/u \in N(m_1); v \in N(m_2)\}$, where $E'_i = \{(x, y) \in E_i / x, y \in V_i\}$. (See Fig.1, where a dashed line is used for a possible edge which can included in the graph and a dotted line for an induced path, possibly trivial(if the ends of the path coincide)).

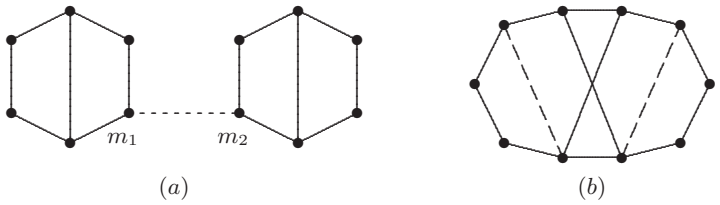


Fig. 1. The split composition of two C_6 with respect to m_1 and m_2: a dashed (dotted) line represents eventual edges (path) in the cycle

Definition 1. *Let C_1 and C_2 two configurations, v_1 and v_2 two vertices of C_1 and C_2 such that $|N(v_1)| = |N(v_2)|$. For each $r \in N \cup \{-1\}$:*

- *$C_1 \varphi_r C_2$: The composition of C_1 and C_2 obtained by connecting C_1 and C_2 by a path $C(v_1, v_2)$ of length "r".*
- *$C_1 \varphi_{-1} C_2$: is the Spilt composition of C_1 and C_2 (e.g. $C_1 * C_2$) with respect to the vertices v_1 and v_2.*

Observe that the graphs in Fig.1. (a) represent $C_6 \varphi_r C_6$ called $2C_6$-configuration of type (a) and the Fig.1. (b) represent $C_6 \varphi_{-1} C_6$ called $2C_6$-configuration of type (b).

Let us define a $2C_6$-biconfiguration of type (a) to be a graph obtained by combining two $1C_6$-biconfigurations by connecting by an induced path two vertices which are not endpoints of chords in the cycle (see Fig.1(a)) and a $2C_6$-biconfiguration of type (b) to be the graph isomorphic to the graph in Fig.1(b).

1.2 Recognition Problem for the Bipartite Almost Distance Hereditary Graphs

Definition 2. *Let k be a positive integer. A graph G is k-distance hereditary if and only if for any connected induced subgraph H of G we have:*

$$\forall u, v \in H,\ d_H(u,v) \leq d_G(u,v) + (k-1).$$

Aïder [1] has given a characterization of 2-distance-hereditary graphs called *'almost distance-hereditary graphs'*, by providing all forbidden configurations, and he has study the metric characterizations. Bessedik [5] and Rautenbach [10] have characterized independently the 3-distance-hereditary graphs. Other characterizations in terms of forbidden subgraphs involving chord distance properties were established for k-distance-hereditary graphs (for $k \geq 1$) by Cicerone and Di Stefano [6] and by Aïder and Meslem [3].

Theorem. [2]
Let G be a connected bipartite graph. Then G is bipartite almost distance hereditary if and only if G contains none of the following configurations as induced subgraphs:

1. Chordless cycles of length greater than or equal to 8
2. $2C_6$-biconfigurations
3. The graphs in Fig .2.

This theorem can be used to derive two polynomial algorithms to solve the recognition problem for the class of bipartite almost distance hereditary graphs.

Proposition 1. There exists a polynomial time algorithm to test whether a given graph G contains as induced subgraph, a cycle C_n with $n \geq 8$ and $cd(C_n) \leq 2$.

Proof. Let $G = (V, E)$ be a connected bipartite graph.
In G, it exists a cycle C_n with $n \geq 8$ and $cd(C_n) \leq 2$ if and only if there are in G two nodes x and y such that all the following conditions hold:

a) There exist two nodes u and v such that: $p_G(x,y) = (x,u,v,y)$;
b) There exists an induced path $P_G(x,y)$ such that $|P_G(x,y)| \geq 5$;
c) Every chord (if any) in the cycle C_n induced by $p_G(x,y) \cup P_G(x,y)$ is incident to u or to v.

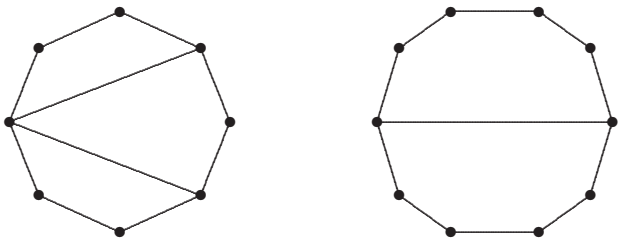


Fig. 2. Forbidden induced subgraph for the class of bipartite almost distance hereditary graphs

Let M be the set of $I_G(x, y)$ containing all the nodes (except x and y) that belong to a shortest $\{x, y\}$-path $[p_G(x, y)]$.
Moreover, if $d_{G-M}(x, y) \geq 5$, let $X = I_{G-M}(x, y) \cap N(x)$ and $Y = I_{G-M}(x, y) \cap N(y)$.
If $P_G(x, y)$ exists, then it must necessarily be one the following paths(see fig.3.):

P_1: a $\{x, y\}$-induced path containing neither nodes of X nor nodes of Y;
P_2: a $\{x, y\}$-induced path containing neither nodes of X but containing one node of Y;
P_3: a $\{x, y\}$-induced path containing one node of X but none nodes of Y;
P_4: a $\{x, y\}$-induced path containing one node of X and one node of Y.

This Procedure analysis every pair of nodes $\{x, y\}$ at distance 3 in G, and tests whether an induced path of type $P_i, 1 \leq i \leq 4$, between x and y exists in G. The procedure runs in time polynomial in the size of input graph G.
Hence, the total time to perform the procedure is $o(n^2)$.

Proposition 2. There exists a polynomial time algorithm to test whether a given connected bipartite graph G contains, as induced subgraph, a $2C_6$-biconfiguraion of type (a).

In what follows, we give a algorithm for checking whether G contains a $2C_6$-biconfiguraion of type (a).

This algorithm considers all the possible distinct pairs of cycles A and B with 6 nodes that are induced subgraphs of G. If $A \cup B$ induced a bipartite connected subgraph F, then either F is not a $2C_6$-biconfiguration of type (a) or F is a $2C_6$-biconfiguration of type (a).

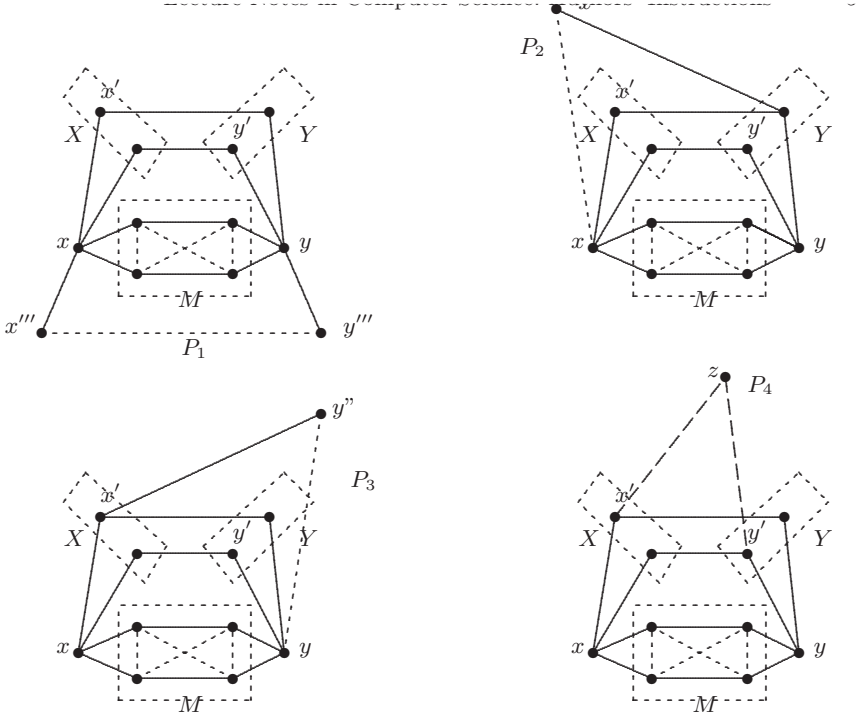


Fig. 3. P_1, P_2, P_3 et P_4:are the four kinds of induced paths from x to y, where direct line (resp. dotted, dashed) represents a path of length = 1(resp.≥ 3, ≥ 2)

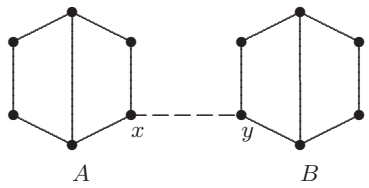


Fig. 4. $2C_6$_biconfiguration of type(a): where dashed lines represent eventual path

Algorithm 1 Testing the existence of a cycle C_n, $n \geq 8$, with $cd(C_n) \leq 2$

Require: A bipartite connected graph $G = (V, E)$
Ensure: True if and only if there exists C_n, $n \geq 8$, in G such that $cd(c_n) \leq 2$.

for each $\{x, y\} \in G$ such that $d_G(x, y) = 3$ **do**
  Compute M
  if x and y are connected in $G \setminus M$ **then**
    if $d_{G \setminus M}(x, y) > 4$ **then**
      return (true), there exists P_1,X and Y are empty
    else
      if both X and Y are note empty **then**
        Compute $X = I_{G \setminus M}(x, y) \cap N(x)$ and $Y = I_{G \setminus M}(x, y) \cap N(y)$
        if x and y are connected in $G \setminus (M \cup X)$ **then**
          Return (true), there exists P_2
        else
          both P_1 and P_2 do not exist
          if x and y are connected in $G \setminus (M \cup Y)$ **then**
            Return (true), there exists P_3
          else
            P_1, P_2, and P_3 do not exist
            for each pair $\{x', y'\}$ such that $x' \in X$, $y' \in Y$ and $(x', y') \notin E$ **do**
              if x' and y' are not connected in the subgraph $G \setminus [M \cup (X \setminus \{x'\}) \cup (Y \setminus \{y'\})]$ **then**
                Return(true), there exists P_4
              end if
            end for
          end if
        end if
      end if
    end if
  end if
end for

Return(false)

Algorithm 2 Looking for a $2C_6$ biconfiguration of type (a) in a graph G

Require: A bipartite connected graph $G = (V, E)$ with $|V| \geq 12$ and $|E| \geq 14$
Ensure: A $2C_6$-biconfiguration of type (a) exists as induced subgraph of G.

```
1: for all A ≡ C6, B ≡ C6' two distinct induced subgraphs of G do
2:    if cd(A) ≤ 1, cd(B) ≤ 1, and ⟨A ∪ B⟩ is not connected then
3:       for all x ∈ A, y ∈ B such that d_A(x) = d_B(y) = 2 do
4:          G_xy := G \ N[A \ x] ∪ N[B \ y]
5:          if x and y are connected in G_xy then
6:             Return, there exists a 2C6-biconfiguration of type (a)
7:          end if
8:       end for
9:    end if
10: end for
11: Return(false)
```

If $cd(A) \leq 1$, $cd(B) \leq 1$, and F is not connected, then A and B could belong to a $2C_6$-biconfiguration of type (a)(line 2). Indeed, the algorithm properly selects two nodes x and y such that $x \in A$ and $y \in B$ (line 3), and it tries to find a path P connecting them. P is looked for the subgraph G_{xy} obtained by removing from G all nodes in $N[A \setminus x] \cup N[B \setminus y]$.
If x and y remain connected in G_{xy}, this means that the desired path P exists. Notice that the cycle at line 1 is executed $o(n^{12})$ times.

Finally, we have the following:

Theorem The recognition problem for the bipartite almost distance hereditary graphs can be solved in polynomial time.

1.3 Open problems

In this paper, we have studied a class of graph constituting a parametric extension of the graphs. Within this framework, we were interested in the bipartite almost distance hereditary graph, introduced by Aïder[2], who given a characterization of these graphs in terms of forbidden subgraphs. According to this characterization, we could solve the problem of recognition of these graphs by developing two polynomial algorithms.

In spite of the results provided in this work, many problems are left open:

1. It will be interesting to give other characterizations of the bipartite almost distance hereditary graphs by using other proprieties (like the constructive operations,...)
2. and it will be interesting to give the metric study.

References

1. Aïder, M.: Almost Distance Hereditary Graphs. In: Discrete Mathrmatics, Vol. 242, 1–16(2002).
2. Aïder, M.: Bipartite Almost Distance-Hereditary Graphs. In: Discrete Mathrmatics, Vol. 308, 865–871(2008).
3. Aïder, M., Meslem, K.: On an Extension of Distance-Hereditary Graphs. In: Discrete Maths (2008) doi:10.1016/j.disc.2007.12.102.
4. Bandelt, H.J., Mulder, H.M.: Distance-Hereditary Graphs. J.Combin,seris B41, 182–208(1986)
5. Bessedik, M.: Sur quelques Proprietes Metriques des Graphes. Magister thesis: USTHB Algers, Algeria(1997).
6. Cicerone, S., Di Stefano, G.:$(k,+)$-Distanc-hereditary graphs. In: Jornal of Discret Algorithms , vol. 1 {3–4}, pp. 281– 302 (2003)
7. D'Arti,A., Moscarini, M., Mulder,M.: On the isomorphism problem for distance-hereditary graphs. In: Report 9241/ A econometric Institue: Erasmus University Rotterdam (1988).
8. Hammer, P. L., Maffray, F.:Complety Separable Graphs. In: Discret Applied Mathematics, vol. 27, pp. 85– 99 (1990).
9. Howorka, E.: A Characterization of Distance Hereditary Graphs. Quart. J. Math. Oxford (2) 28, 417–420(1977).
10. Rautenbach, D.: Graphs with Small Additive Stretch Number. In: Discussions Mathematics, Graph Theory, vol. 24, pp. 291–301 (2004).

Résolution d'un problème de programmation bi-niveaux linéaire par la méthode DC

A. Anzi* and M.S. Radjef **

Laboratoire de modélisation et d'optimisation des systèmes
(LAMOS), Université de Béjaia, Algérie

Résumé Nous proposons un algorithme pour la résolution d'un problème d'optimisation bi-niveaux linéaire. Dans un premier temps, ce problème est reformulé comme un programme d'optimisation non linéaire en exploitant les conditions d'optimalité de Karush-Kuhn-Tucker associées au problème du Suiveur. Après avoir utilisé une pénalité exacte pour le programme mathématique obtenu, nous proposons une décomposition sous une forme DC pour laquelle nous développons un algorithme DCA. Enfin, l'algorithme proposé est testé sur un ensemble de problèmes déjà étudiés dans la littérature.

Mots clés : programmation bi-niveaux linéaire, programmation DC, conditions KKT, algorithme DCA, pénalité exacte.

1 Introduction

La programmation bi-niveaux est utilisée pour la modélisation et la résolution des processus de décision ayant une structure hiérarchique. Cette classe de programmes constitue une branche de la programmation mathématique dans laquelle les contraintes sont déterminées, en partie, par un autre problème d'optimisation.

La programmation bi-niveaux est motivée par la théorie des jeux statiques et non coopératifs de Stackelberg appliquée aux problèmes économiques. Dans ces problèmes, le niveau supérieur est appelé Leader et le niveau inférieur Suiveur. Le contrôle des variables de décision est partitionné entre les preneurs de décision qui cherchent à optimiser leurs fonctions objectifs individuelles. Le Leader prend, le premier, sa décision dans l'objectif d'optimiser sa fonction de gains. Le Suiveur observe la décision du Leader et construit sa décision. Comme les ensembles des décisions sont interdépendants, la décision du Leader affecte l'ensemble des décisions et les gains du Suiveur et vice versa.

Les problèmes qui peuvent être modélisés par la programmation bi-niveaux sont nombreux et on rencontre des applications en économie [16], en transport urbain [13], en contrôle de pollution [2], ... etc (voir [8][14]), sans oublier le domaine de la guerre qui est considéré parmi les premières applications de cette

* Département de Recherche Opérationnelle, Université de Béjaia, Algérie.
** Laboratoire LAMOS, Université de Béjaia, Algérie.

classe de problèmes. En effet, bien que Candler et Norton (1977) furent les premiers à utiliser la terminologie *programmation à deux niveaux ou à plusieurs niveaux* dans un rapport de la Banque Mondiale, les toutes premières formulations liées au problème de programmation bi-niveaux sont apparues dans l'œuvre des auteurs J. Bracken et J. McGill (1973) consacrée à ce dernier domaine, et avec l'appellation "*programmes mathématiques avec des problèmes d'optimisation dans les contraintes*".

La programmation bi-niveaux linéaire est l'un des modèles de base de l'optimisation bi-niveaux, où les fonctions objectifs et les contraintes du Leader et du Suiveur sont linéaires. Un programme bi-niveaux linéaire (PBL) se formule comme suit :

$$(PBL)\quad \begin{cases} \max\limits_{x} \; F(x,y) = c_1^t x + d_1^t y, \\ \quad s.c. \;\; A_1 x + B_1 y \leq b_1, \\ \qquad x \geq 0; \\ \qquad \max\limits_{y} \; f(x,y) = c_2^t x + d_2^t y, \\ \qquad\quad s.c. \;\; A_2 x + B_2 y \leq b_2, \\ \qquad\qquad y \geq 0, \end{cases} \tag{1}$$

où $F : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \longrightarrow \mathbb{R}$; $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \longrightarrow \mathbb{R}$ sont les fonctions objectifs du Leader et du Suiveur respectivement ; c_1, $c_2 \in \mathbb{R}^{n_1}$; d_1, $d_2 \in \mathbb{R}^{n_2}$; $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$; A_1-$m_1 \times n_1$-matrice, B_1-$m_1 \times n_2$-matrice, A_2-$m_2 \times n_1$-matrice et B_2-$m_2 \times n_2$-matrice. Le problème du Suiveur, pour une décision x du Leader, est donné par

$$\max\{c_2^t x + d_2^t y : A_2 x + B_2 y \leq b_2, y \geq 0\}. \tag{2}$$

Il a été prouvé que la solution du problème (1) est un certain point extrême du domaine des contraintes qui est un polyèdre [10]. En dépit de sa simplicité apparente, le (PBL) est un problème complexe. Beaucoup d'approches ont été proposées dans la littérature pour sa résolution [10][15]. Les plus populaires sont basées sur sa reformulation en un programme mathématique qui est un programme linéaire avec une contrainte de complémentarité non linéaire. C'est cette contrainte de complémentarité qui constitue la difficulté majeure dans cette approche.

Dans ce papier, nous traitons le (PBL) par une technique d'optimisation non convexe et non différentiable basée sur la programmation DC (Difference of Convex functions) et DCA (DC Algorithm). La programmation DC et l'algorithme DCA ont été introduits par P.D.Tao en 1986. Ils ont été intensivement développés par P.D. Tao et L.T. Hoai An depuis 1993 pour devenir maintenant classiques et de plus en plus populaires. DCA est une méthode de descente sans recherche linéaire pour la résolution d'un programme DC général. Plus précisément, c'est une méthode primale-duale de sous-gradient basée sur l'optimalité locale et la dualité en optimisation DC. L'algorithme DCA a été appliqué avec succès pour un grand nombre de problèmes d'optimisation non convexe dans différents domaines de la science appliquée [4][6]. La globalité des solutions calculées par DCA n'est pas toujours garantie. Cependant, il a été remarqué,

qu'avec un bon choix du point initial, il converge souvent vers la solution globale.

En exploitant les conditions d'optimalité KKT associées au problème (2) du Suiveur, nous réécrivons le (PBL) sous forme d'un programme mathématique de la forme (voir [10], *proposition 5.2.2*)

$$\begin{cases} \max\limits_{x,y} \; F(x,y) = c_1^t x + d_1^t y \\ \qquad\quad A_1 x + B_1 y \leq b_1 \\ \qquad\quad A_2 x + B_2 y + w = b_2 \\ \qquad\quad B_2^t u - v = d_2 \\ \qquad\quad v^t y + u^t w = 0 \\ \qquad\quad x \geq 0,\; y \geq 0,\; u \geq 0,\; v \geq 0,\; w \geq 0, \end{cases} \tag{3}$$

où $w \in \mathbb{R}^{m_2}$ est une variable d'écart, $v \in \mathbb{R}^{n_2}$ et $u \in \mathbb{R}^{m_2}$ sont des variables duales. En utilisant une pénalité exacte, nous reformulons le programme (3) comme un programme de minimisation concave sous contraintes linéaires auquel nous proposons une décomposition DC. Enfin, nous développons un algorithme DCA pour la résolution de ce dernier programme.

2 Reformulation via une pénalité exacte

Dans cette section nous utilisons une pénalité exacte pour réécrire le problème (3) sous forme d'un programme de minimisation concave. Avant d'aborder ce point, nous donnons quelques définitions associées au problème (1).

Définition 1. *[10]*
Domaine des contraintes :

$$S = \{(x,y) : A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2, x \geq 0, y \geq 0\}.$$

Ensemble réalisable du Suiveur, pour $x \in \mathbb{R}^{n_1}$ fixé :

$$S(x) = \{y \in \mathbb{R}^{n_2} : B_2 y \leq b_2 - A_2 x, y \geq 0\}.$$

La projection de S sur l'ensemble des décisions du Leader :

$$P(X) = \{x \in \mathbb{R}^{n_1} : \exists y \in \mathbb{R}^{n_2}, A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2, x \geq 0, y \geq 0\}.$$

L'ensemble des réactions rationnelles du Suiveur, pour x fixé :

$$R(x) = \{y \in \mathbb{R}^{n_2} : y = \arg\max[f(x,\hat{y}) : \hat{y} \in S(x)]\}.$$

La région induite :

$$RI = \{(x,y) \in S,\, y \in R(x)\}.$$

La région induite *RI* représente l'ensemble réalisable du (PBL) sur lequel le Leader peut optimiser sa fonction objectif.

Pour s'assurer que le (PBL) est bien posé et possède une solution, les hypothèses suivantes seront supposées [10] :

H 1. *l'ensemble $R(x)$ est réduit à un singleton.*

H 2. *l'ensemble S est non vide et compact.*

L'hypothèse **H1** est nécessaire pour le cas, où il y a absence de coopération entre le Leader et le Suiveur.
En ajoutant une variable d'écart $e \geq 0$ au problème (3), nous obtenons

$$\begin{cases} \max\limits_{x,y} \; F(x,y) = c_1^t x + d_1^t y \\ \qquad\qquad A_1x + B_1y + e = b_1 \\ \qquad\qquad A_2x + B_2y + w = b_2 \\ \qquad\qquad B_2^t u - v = d_2 \\ \qquad\qquad v^t y + u^t w = 0 \\ \qquad\qquad x \geq 0,\; y \geq 0,\; u \geq 0,\; v \geq 0,\; w \geq 0, e \geq 0. \end{cases} \tag{4}$$

Introduisons maintenant quelques notations utiles pour la suite du document.
$z = \begin{pmatrix} x\ y\ e\ w\ v\ u \end{pmatrix}^t \in \mathbb{R}^n$, $c = \begin{pmatrix} -c_1\ -d_1\ 0\ 0\ 0\ 0 \end{pmatrix}^t \in \mathbb{R}^n$,
$A = \begin{pmatrix} A_1 & B_1 & I_{m_1} & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix} \in R^{m\times n}$, $b = \begin{pmatrix} b_1 \\ b_2 \\ d_2 \end{pmatrix} \in \mathbb{R}^m$,
$E_u = \begin{pmatrix} 0\ 0\ 0\ 0\ 0\ I_{m_2} \end{pmatrix}$, $E_v = \begin{pmatrix} 0\ 0\ 0\ 0\ I_{n_2}\ 0 \end{pmatrix}$,
$E_w = \begin{pmatrix} 0\ 0\ 0\ I_{m_2}\ 0\ 0 \end{pmatrix}$, $E_y = \begin{pmatrix} 0\ I_{n_2}\ 0\ 0\ 0\ 0 \end{pmatrix}$,
où I_k est une $k \times k$ matrice identité; 0 est la matrice nulle avec la dimension appropriée pour chaque cas, avec $n = n_1 + 2n_2 + m_1 + 2m_2$; $m = m_1 + m_2 + n_2$.
En utilisant ces notations, nous aurons :
$u^t w = (E_u z)^t (E_w z) = z^t (E_u^t E_w) z = z^t D^1 z$, et
$v^t y = (E_v z)^t (E_y z) = z^t (E_v^t E_y) z = z^t D^2 z$,
ce qui donne : $u^t w + v^t y = z^t D^1 z + z^t D^2 z = z^t D z \quad$ avec $\; D^1 + D^2 = D$.
Notons que les éléments $d_{ij} (i = \overline{1,n},\; j = \overline{1,n})$ de la matrice D sont tous non négatifs. En posant $Dz = q(z)$, notre problème peut s'écrire

$$P(z) \begin{cases} \min F(z) = c^t z \\ \quad Az = b, \\ \quad z^t q(z) = 0, \\ \quad z \geq 0. \end{cases}$$

avec $q(z) \geq 0,\; \forall z \geq 0$.

Considérons l'ensemble convexe $\mathcal{Z} = \{z \in \mathbb{R}^n : Az = b, z \geq 0\}$, et soit la fonction $\Psi : \mathbb{R}^n \to \mathbb{R}$ définie par $\quad \Psi(z) = \sum\limits_{i=1}^{n} \min\{q_i(z),\; z_i\}$.
Ψ est une fonction concave, finie et non négative sur $\mathcal{Z}$. Nous avons

$$\{z \in \mathcal{Z},\; z^t q(z) = 0\} = \{z \in \mathcal{Z},\; \Psi(z) \leq 0\}.$$

Le problème $P(z)$ peut être réécrit sous la forme

$$\alpha = \min\{F(z) \;:\; z \in \mathcal{Z},\; \Psi(z) \leq 0\}. \tag{5}$$

D'après ([3], *théorème 1*), si $\mathcal{Z}$ est non vide et borné, alors il existe une constante $k_0 \geq 0$, telle que pour tout $k \geq k_0$ le problème (5) est équivalent au problème pénalisé

$$\alpha(k) = \min\{F(z) + k\Psi(z) : z \in \mathcal{Z}\}. \tag{6}$$

3 DCA pour la résolution du problème (6)

Dans cette section, nous allons reformuler le problème (6) comme un programme DC. Puis, nous appliquons DCA pour sa résolution. Pour cela, nous donnons un bref aperçu sur la méthode DC et l'algorithme DCA.

3.1 Programmation DC

On utilise [20] pour les définitions usuelles en analyse convexe où les fonctions peuvent prendre les valeurs infinies $\pm\infty$. X désigne l'espace euclidien $\mathbb{R}^n$, muni du produit scalaire usuel noté $\langle ., . \rangle$. L'espace dual Y de X peut s'identifier à lui même. On note $\Gamma_0(X)$ l'ensemble de toutes les fonctions convexes, propres et semi-continues inférieurement sur X. Pour $g \in \Gamma_0(X)$, la fonction conjuguée, $g^* \in \Gamma_0(Y)$ de g est définie par : $g^*(y) = \sup\{\langle x, y\rangle - g(x) | x \in X\}$.

Soit $g \in \Gamma_0(X)$, $x_0 \in domg$ et $\epsilon > 0$. Alors $\partial_\epsilon g(x_0)$ représente le ϵ-sous différentiel de g au point x_0 et est donné par :

$$\partial_\epsilon g(x_0) = \big\{y_0 \in Y | g(x) \geq g(x_0) + \langle y_0, x - x_0\rangle - \epsilon,\ \forall x \in X\big\}.$$

Pour $\epsilon = 0$, on obtient $\partial g(x_0)$ qui représente le sous-différentiel usuel (exact) de g au point x_0. Les éléments de $\partial g(x_0)$ sont appelés sous-gradients.
Une fonction $\theta \in \Gamma_0(X)$ est dite convexe polyédrale sur l'ensemble $C \subset X$ [20], si $\theta(x)$ s'écrit sous la forme :

$$\theta(x) = \max\{\langle a_i, x\rangle - \beta_i : \quad i = 1, ..., m\} + \chi_C(x),$$

où $a_i \in \mathbb{R}^n, \beta_i \in \mathbb{R}, i = 1, ..., m$, C est un polyèdre convexe non vide et χ_C est la fonction indicatrice de l'ensemble C.

Considérons le programme DC standard

$$\alpha = \inf\{f(x) = g(x) - h(x) \ : \ x \in X\}, \tag{7}$$

où g et $h \in \Gamma_0(X)$ sont appelées les composantes DC de f et $g-h$ la décomposition DC de f. En utilisant la notion de conjugaison, on obtient le programme dual de (7) qui est donné par :

$$\beta = \inf\{h^*(y) - g^*(y) : y \in Y\}, \tag{8}$$

où g^* et h^* sont les fonctions conjuguées de g et h respectivement.
Pour le programme (7), on a les conditions nécessaires d'optimalité locale [7] :

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*) \tag{9}$$

$$\emptyset \neq \partial g(x^*) \cap \partial h(x^*). \tag{10}$$

Un point qui satisfait la condition (10) est appelé point critique de (7).
Un programme DC est dit polyédral si l'une des fonctions g ou h est polyédrale convexe. Pour cette classe de programme DC, la condition d'optimalité (9) est aussi suffisante [5].

3.2 Algorithme DCA

L'algorithme DCA est basé sur la construction de deux séquences $\{x^i\}$ et $\{y^i\}$ candidates à être solutions optimales locales pour le problème primal et dual respectivement et qui vérifient les propriétés suivantes [1] :

1. $\{g(x^i) - h(x^i)\}$ et $\{h^*(y^i) - g^*(y^i)\}$ décroissent à chaque itération.
2. Si $(g-h)(x^{i+1}) = (g-h)(x^i)$(resp. $(h^*-g^*)(y^{i+1}) = (h^*-g^*)(y^i)$) l'algorithme s'arrête à l'itération $i+1$ et le point x^i (resp. y^i) est un point critique de $g-h$ (resp. h^*-g^*).
3. Sinon toute valeur d'adhérence x^* de la séquence $\{x^i\}$ (resp. y^* de la suite $\{y^i\}$) est un point critique de $g-h$ (resp. h^*-g^*).

Ces séquences sont générées de la manière suivante : x^{i+1} (resp. y^i) est solution du problème convexe (11) (resp. (12)) défini par :

$$(P_i) \quad \alpha_i = \inf\{g(x) - [h(x^i) + \langle x - x^i,\ y^i\rangle] \mid x \in X\}, \tag{11}$$

$$(D_i) \quad \inf\{h^*(y) - [g^*(y^{i-1}) + \langle x^i,\ y - y^{i-1}\rangle] \mid y \in Y\}. \tag{12}$$

Ainsi, DCA peut avoir l'interprétation suivante : à chaque itération, on remplace dans le programme DC primal (7) (resp. dual(8)) h (resp. g^*) par sa minorante affine définie par $h(x^i)+\langle x-x^i, y^i\rangle$ (resp. $g^*(y^i)+\langle y-y^i, x^{i+1}\rangle$), ce qui donne le problème (P_i) (resp. (D_i)). DCA opère donc une double linéarisation en utilisant les sous-gradients de h et g^*, ce qui donne le schéma suivant :

$$y^i \in \partial h(x^i) \ \ et \ \ x^{i+1} \in \partial g^*(y^i). \tag{13}$$

Rappelons qu'il existe deux formes de DCA : DCA complet [4][6] et DCA simplifié. Le schéma (13) correspond à la forme simplifiée qui est préférable en pratique car elle est moins coûteuse. L'algorithme DCA qui découle de (13) est donné par :

Algorithme DCA simplifié

0 : x^0 donné.
itération i
1 : Calculer $y^i \in \partial h(x^i)$.
2 : Calculer $x^{i+1} \in \partial g^*(y^i)$.
3 : Si test d'arrêt vérifié **Stop** ; sinon $i \leftarrow i+1$ et **aller** en 1.

3.3 Décomposition DC pour le problème (6)

Considérons le problème (6) et notons $\chi_{\mathcal{Z}}$ la fonction indicatrice de l'ensemble $\mathcal{Z}$. Soient g et h deux fonctions définies par

$$g(z) = \chi_{\mathcal{Z}}(z) \;\text{ et }\; h(z) = -F(z) - k\Psi(z). \tag{14}$$

Donc g et h sont convexes et le problème (6) est un programme DC de la forme

$$\min\{g(z) - h(z) : z \in \mathbb{R}^n\}. \tag{15}$$

L'application de DCA pour le problème (15) revient à calculer les deux séquences $\{t^i\}$ et $\{z^i\}$ définies par

$$t^i \in \partial h(z^i) \;\text{ et }\; z^{i+1} \in \partial g^*(t^i).$$

En utilisant les règles de calcul en analyse convexe [20], nous calculons t^i et z^{i+1}.

Calcul de t^i : Nous avons $t^i \in \partial\big(-c^t z^i - k\sum_{j=1}^{n} \min\{q_j(z^i),\ z_j^i\}\big)$, donc $t^i = -c + k\theta^i$, avec $\theta^i \in \sum_{j=1}^{n} \partial\big(\max\{-D_j z^i,\ -z_j^i\}\big)$, où $q_j(z^i) = D_j z^i$. Soit

$$\theta^i = -\sum_{j=1}^{n} \begin{cases} D_j^t, & si \;\; z_j^i > D_j z^i, \\ e_j, & si \;\; z_j^i < D_j z^i, \\ \gamma e_j + (1-\gamma) D_j^t, & si \;\; z_j^i = D_j z^i, \end{cases} \tag{16}$$

où D_j est la j-*ème* ligne de D, e_j est le j-*ème* vecteur unitaire de $\mathbb{R}^n$ et $\gamma \in [0,1]$.

Calcul de z^{i+1} : Soit $z = (x, y, e, w, v, u)^t$. Le calcul de z^{i+1} revient à résoudre le programme linéaire

$$\arg\min\{-\langle z,\ t^i\rangle : z \in \mathcal{Z}\}, \tag{17}$$

Les composantes (v^{i+1}, u^{i+1}) étant les variables duales, elles se calculent en résolvant le programme dual du problème (2) donné par

$$\min\{u^t(b_2 - A_2 x^{i+1}) : B_2^t u - v = d_2,\ u \geq 0,\ v \geq 0\}. \tag{18}$$

Algorithme 1 : DCA pour le problème (6)

1 : Soit z^0 point initial, poser $i = 0$, $\epsilon > 0$, $k \in \mathbb{R}_+$, $\gamma \in [0,1]$ et $\lambda > 0$.
2 : Calculer $t^i = (t_x^i, t_y^i, t_e^i, t_w^i, t_v^i, t_u^i) \in \partial h(z^i)$ i.e.
$t^i = -c + k\theta^i$, où θ^i est calculée en utilisant (16)
3 : Calculer $z^{i+1} = (x^{i+1}, y^{i+1}, e^{i+1}, w^{i+1}, v^{i+1}, u^{i+1})$ en utilisant (17).
4 : Si $y^{i+1} \in \arg\max\big\{f(x^{i+1}, y) : B_2 y \leq b_2 - A_2 x^{i+1},\ y \geq 0\big\}$, **alors** aller à **5**, **sinon** aller à **7**
5 : Calculer (v^*, u^*) solution du problème (18).
Donc $z^{i+1} = (x^{i+1}, y^{i+1}, e^{i+1}, w^{i+1}, v^*, u^*)$.
6 : Si $\|z^{i+1} - z^i\|/(\|z^i\| + 1) \leq \epsilon$ **alors**
arrêter, z^{i+1} est solution optimale de (1), **sinon** aller à **7**
7 : Poser $z^i = z^{i+1}$, $i = i + 1$, $k = k + \lambda$ et aller à **2**.

Remarques :
• Le problème (6), avec la décomposition (14), est un programme DC polyédral, puisque $g(z) = \chi_{\mathcal{Z}}(z)$ est polyédrale [20]. Donc DCA appliqué à (6) possède une convergence

finie [4][6].
• Le test de l'étape 4 a été rajouté pour tester la faisabilité de la solution (x^{i+1}, y^{i+1}) pour le (PBL). La vérification de ce test implique que $y^{i+1} \in R(x^{i+1})$ (voir définition 1). Et puisque $(x^{i+1}, y^{i+1}) \in S$, nous aurons alors $(x^{i+1}, y^{i+1}) \in RI$ ce qui implique que (x^{i+1}, y^{i+1}) est une solution réalisable du (PBL).

3.4 Point initial pour l'algorithme 1

Pour obtenir un bon point initial pour DCA, nous avons utilisé encore DCA pour la résolution du problème (19) dont la valeur optimale est connue (égale à 0)

$$0 = \min\{\Psi(z) : \bar{A}z = \bar{b}, z \geq 0\} \tag{19}$$

où $\bar{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix}$ et $\bar{b} = \begin{pmatrix} 0 \\ b_2 \\ d_2 \end{pmatrix}$

4 Résultats numériques

L'algorithme présenté est codé en MATLAB. Nous l'avons testé sur un ensemble de problèmes (voir ci-dessous). Le point initial z^0 est calculé en résolvant le problème (19). Nous avons remarqué que la valeur du paramètre γ n'a pas d'influence sur les résultats de l'algorithme, donc nous avons pris comme valeur $\gamma = 0.5$. Le test d'arrêt de l'algorithme sera vérifié s'il est inférieur ou égal à $\epsilon = 10^{-3}$. Le temps est reporté en secondes.

Problèmes test

P1 [21] :
$$\begin{cases} \max F(x,y) = 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ x \geq 0 \\ \max f(x,y) = -x_1 - 2x_2 - y_1 - y_2 - 2y_3 \\ 0x_1 + 0x_2 - y_1 + y_2 + y_3 \leq 1 \\ 2x_1 + 0x_2 - y_1 + 2y_2 - 0.5y_3 \leq 1 \\ 0x_1 + 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1 \\ y \geq 0 \end{cases}$$

P2 [7] :
$$\begin{cases} \max F(x,y) = x + 3y \\ x \geq 0 \\ \max f(x,y) = x - 3y \\ -x - 2y \leq -10 \\ x - 2y \leq 6 \\ 2x - y \leq 21 \\ x + 2y \leq 38 \\ -x + 2y \leq 18 \\ y \geq 0 \end{cases}$$

P3 [17] :
$$\begin{cases} \max F(x,y) = 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ x_1 + 2x_2 - y_3 \leq 1.3 \\ x \geq 0 \\ \max f(x,y) = -2y_1 - y_2 - 2y_3 \\ 0x_1 + 0x_2 - y_1 + y_2 + y_3 \leq 1 \\ 4x_1 + 0x_2 - 2y_1 + 4y_2 - y_3 \leq 2 \\ 0x_1 + 4x_2 + 4y_1 - 2y_2 - y_3 \leq 2 \\ y \geq 0 \end{cases}$$

P4 [11] :
$$\begin{cases} \max F(x,y) = x + y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -4x - 3y \leq -19 \\ x + 2y \leq 11 \\ 3x + y \leq 13 \\ y \geq 0 \end{cases}$$

$$\text{P5 [19]} : \begin{cases} \max F(x,y) = -x - 5y \\ x \geq 0 \\ \max f(x,y) = 0x + y \\ -x - y \leq -8 \\ -3x + 2y \leq 6 \\ x + 4y \leq 48 \\ x - 5y \leq 9 \\ y \geq 0 \end{cases} \qquad \text{P6 [12]} : \begin{cases} \max F(x,y) = x_1 + y_1 - 4y_2 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 0y_1 + y_2 \\ x + y_1 + y_2 \leq 3 \\ -x - y_1 + y_2 \leq -1 \\ -x + y_1 + y_2 \leq 1 \\ x - y_1 + y_2 \leq 1 \\ y \geq 0 \end{cases}$$

$$\text{P7 [12]} : \begin{cases} \max F(x,y) = -0.4x_1 - y_1 - 5y_2 + 0y_3 + 0y_4 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 0y_1 + 0.5y_2 - y_3 - 2y_4 \\ -0.1x_1 - y_1 - y_2 + 0y_3 + 0y_4 \leq -1 \\ 0.2x_1 + 0y_1 + 1.25y_2 + 0y_3 - y_4 \leq -1 \\ -x + 6y_1 + y_2 - 2y_3 + 0y_4 \leq 1 \\ y \geq 0 \end{cases} \qquad \text{P8 [10]} : \begin{cases} \max F(x,y) = -x + 4y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -x - y \leq -3 \\ -2x + y \leq 0 \\ 2x + y \leq 12 \\ -3x + 2y \geq -4 \\ y \geq 0 \end{cases}$$

$$\text{P9 [21]} : \begin{cases} \max F(x,y) = 2x_1 - x_2 - 0.5y_1 \\ x_1 + x_2 \leq 2 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 0x_2 + 4y_1 - y_2 \\ -2x_1 + y_1 - y_2 \leq -2.5 \\ x_1 - 3x_2 + y_2 \leq 2 \\ y \geq 0 \end{cases} \qquad \text{P10 [21]} : \begin{cases} \max F(x,y) = -x + 4y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -2x + y \leq 0 \\ 2x + 5y \leq 108 \\ 2x - 3y \leq -4 \\ y \geq 0 \end{cases}$$

$$\text{P11 [9]} : \begin{cases} \max F(x,y) = 2x_1 - x_2 - x_3 + 2x_4 + x_5 - 3.5x_6 - y_1 - 1.5y_2 + 3y_3 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 2x_2 + 0x_3 + 0x_4 - x_5 + 0x_6 + 3y_1 - y_2 - 4y_3 \\ -x_1 + 0.2x_2 + 0x_3 + 0x_4 + x_5 + 2x_6 - 4y_1 + 2y_2 + y_3 \leq 12 \\ x_1 + 0x_2 + x_3 - 2x_4 + 0x_5 + 0x_6 + 0y_1 - 4y_2 + y_3 \leq 10 \\ 5x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 3.2x_6 + 2y_1 + 2y_2 + 0y_3 \leq 15 \\ 0x_1 - 3x_2 + 0x_3 - x_4 + x_5 + 0x_6 - 2y_1 + 0y_2 + 0y_3 \leq 12 \\ -2x_1 - x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0y_1 - y_2 + y_3 \leq -2 \\ 0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 - y_1 - 2y_2 - y_3 \leq -2 \\ 0x_1 - 2x_2 - 3x_3 + 0x_4 - x_5 + 0x_6 + 0y_1 + 0y_2 + 0y_3 \leq -3 \\ y \geq 0 \end{cases}$$

Les résultats sont donnés dans le tableau 1 où nous utilisons les notations suivantes :

k : le paramètre de pénalité.
λ : un pas pour augmenter le paramètre de pénalité.
temps : est le temps d'exécution de l'algorithme.
Iter : le nombre d'itérations de l'algorithme.
$(F^*; f^*)$: représentent les valeurs optimales obtenues du Leader et Suiveur respectivement.

Problème	$(k;\lambda)$	*solution*	$(F^*;f^*)$	*temps*	*Iter*
P1	(1 ;0.1)	-	-	-	-
	(1 ;0.5)	(0 0.9 0 0.6 0.4)	(29.2 ; -3.2)	4.06	19
	(1 ;1)	(0 0.9 0 0.6 0.4)	(29.2 ; -3.2)	2.48	11
	(5 ;1)	(0 0.9 0 0.6 0.4)	(29.2 ; -3.2)	2.07	7
	(5 ;5)	(0 0.9 0 0.6 0.4)	(29.2 ; -3.2)	1.26	3
	(10 ;5)	(0 0.75 0 0.5 0)	(23 ; -2)	0.7	2
P2	(0.01 ;0.01)	-	-	-	-
	(0.1 ;0.01)	(16 11)	(49 ; -17)	3.14	3
	(0.1 ;0.1)	(16 11)	(49 ; -17)	6.92	6
	(1 ;1)	(16 11)	(49 ; -17)	0.65	2
	(5 ;1)	(12 3)	(21 ; 3)	0.17	2
	(5 ;5)	(12 3)	(21 ; 3)	1.23	2
	(10 ;5)	(12 3)	(21 ; 3)	0.18	2
P3	(5 ;5)	-	-	-	-
	(10 ;5)	(0 0.78 0 0.43 0.26)	(21.36 ; -0.95)	1.18	2
	(15 ;5)	(0 0.78 0 0.43 0.26)	(21.36 ; -0.95)	1.44	2
	(20 ;5)	(0 0.78 0 0.43 0.26)	(21.36 ; -0.95)	1.20	2
P4	(0.1 ;0.1)	-	-	-	-
	(1 ;0.1)	(4 1)	(5 ; -1)	0.09	2
	(1 ;1)	(4 1)	(5 ; -1)	1.09	4
	(5 ;1)	(4 1)	(5 ; -1)	0.7	2
	(5 ;5)	(4 1)	(5 ; -1)	0.6	2
	(10 ;5)	(4 1)	(5 ; -1)	0.6	2
P5	(0.1 ;0.1)	-	-	-	-
	(1 ;0.1)	(2 6)	(-32 ; 6)	1.5	3
	(1 ;1)	(2 6)	(-32 ; 6)	1.7	3
	(5 ;1)	(2 6)	(-32 ; 6)	0.5	2
	(5 ;5)	(2 6)	(-32 ; 6)	0.6	2
	(10 ;5)	(2 6)	(-32 ; 6)	0.56	2
P6	(0.01 ;0.01)	(2 1 0)	(3 ; 0)	0.82	3
	(0.1 ;0.1)	(2 1 0)	(3 ; 0)	0.62	3
	(1 ;0.1)	(1 1 1)	(-2 ; 1)	0.56	1
	(1 ;1)	(1 1 1)	(-2 ; 1)	0.6	1
	(5 ;1)	(1 1 1)	(-2 ; 1)	0.58	1
	(10 ;5)	(1 1 1)	(-2 ; 1)	0.75	1
P7	(0.1 ;0.01)	-	-	-	-
	(0.1 ;0.1)	(0 0 1 0 2.25)	(-5 ; -4)	0.68	2
	(1 ;1)	(0 0 1 0 2.25)	(-5 ; -4)	0.6	2
	(5 ;1)	(0 0 1 0 2.25)	(-5 ; -4)	0.56	2
	(5 ;5)	(0 0 1 0 2.25)	(-5 ; -4)	1.06	2
	(10 ;5)	(0 0 1 0 2.25)	(-5 ; -4)	0.96	2
P8	(0.1 ;0.1)	-	-	-	-
	(1 ;0.1)	(4 4)	(12 ; -4)	3.96	8
	(1 ;1)	(4 4)	(12 ; -4)	1.51	3
	(5 ;1)	(2 1)	(2 ; -1)	0.56	2
	(5 ;5)	(2 1)	(2 ; -1)	0.21	2
	(10 ;5)	(2 1)	(2 ; -1)	0.65	2

P9	(0.1 ;0.1)	-	-	-	-
	(1 ;0.1)	(2 0 1.5 0)	(3.25 ; 6)	2.12	2
	(1 ;1)	(2 0 1.5 0)	(3.25 ; 6)	1.40	4
	(5 ;1)	(2 0 1.5 0)	(3.25 ; 6)	1.42	2
	(5 ;5)	(2 0 1.5 0)	(3.25 ; 6)	1.39	2
	(10 ;5)	(2 0 1.5 0)	(3.25 ; 6)	1.43	2
P10	(0.1 ;0.1)	-	-	-	
	(1 ;0.1)	(19 14)	(37 ; -14)	0.64	2
	(1 ;1)	(19 14)	(37 ; -14)	0.17	2
	(5 ;1)	(19 14)	(37 ; -14)	1.28	6
	(5 ;5)	(19 14)	(37 ; -14)	0.5	2
P11	(0.1 ;0.1)	-	-	-	-
	(0.5 ;0.1)	(0 4 0 15 9.2 0 0 0 2)	(41.2 ; -9.2)	1.17	2
	(0.5 ;0.5)	(0 4 0 15 9.2 0 0 0 2)	(41.2 ; -9.2)	2.5	7
	(1 ;0.5)	(0 2 0 11 19.6 0 2 0 0)	(37.6 ; -13.6)	0.7	2
	(1 ;1)	(0 2 0 11 19.6 0 2 0 0)	(37.6 ; -13.6)	0.7	2
	(5 ;1)	(1 0 0 6 21 0 2 0 0)	(33 ; -19)	2.64	6
	(5 ;5)	(1 0 0 6 21 0 2 0 0)	(33 ; -19)	1.2	2
	(10 ;5)	(0.5 0 0 0 3.93 3.92 0 1 0)	(-8.04 ; -4.93)	0.6	2

Tab. 1 : Résultats numériques pour l'algorithme 1

Le signe "−" signifie que l'algorithme n'a pas pu résoudre le problème.

Commentaires : D'après les résultats numériques du tableau 1, on peut voir que :

(*i*) L'algorithme possède une convergence finie et rapide ; le nombre moyen d'itérations est de 2.

(*ii*) Le temps d'exécution de l'algorithme est petit. Ce résultat est normal car seuls des systèmes linéaires sont résolus à chaque itération de l'algorithme.

(*iii*) La convergence de l'algorithme vers la solution globale ou locale est fortement liée au choix du paramètre de pénalité ainsi que le pas d'augmentation de ce paramètre.

(*iv*) Le choix du paramètre de pénalité, k, dépend fortement du problème testé.

5 Conclusion

Nous avons présenté un algorithme d'optimisation DC pour la résolution du problème de programmation bi-niveaux linéaire, où le problème du Suiveur est remplacé par les conditions d'optimalité KKT associées. Le problème est ensuite reformulé comme un programme DC à l'aide d'une pénalité exacte et une décomposition DC adéquate. L'algorithme proposé est simple et rapide, puisque seuls des programmes linéaires sont résolus à chaque itération. Les résultats numériques montrent, qu'avec un bon choix du paramètre de pénalité, qui dépend fortement du problème testé, l'algorithme converge souvent vers la solution globale.

Références

1. F.B. Akoa : Approches de points intérieurs et de la programmation DC en optimisation non convexe. Codes et simulations numériques industrielles. Thèse de Doctorat, Institut national des sciences appliquées de Rouen. (2005)
2. M.A. Amouzegar, K. Moshirvaziri : Determining optimal pollution control policies : An application of bilevel programming. European J. Oper. Res., 119 : 100-120, (1999)
3. L.T.H. An, P.D. Tao : A continuous approach for globally solving linearly constrained quadratic zero-one programming problems. Optimization, 50 : 93–120, (2001)
4. L.T.H. An, P.D. Tao : Convex analysis approach to dc programming : theory and applications. Acta Mathematica Vietnamica, 22 : 289-355, (1997)
5. L.T.H. An, P.D. Tao : Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. J. Glob. Optim., 11 : 253-285, (1997)
6. L.T.H. An, P.D. Tao : The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. Annals Oper. Res., 133 :23–46, (2005)
7. G. Anandalingam, D. J. White : A solution for the linear static Stackelberg problem using penalty functions. IEEE Trans. on Aut. Cont., 35 : 1170–1173, (1990)
8. G. Anandalingam, T. L. Friesz : Hierarchical optimization : An introduction. Annals of Oper. Res., 34 : 1–11, (1992)
9. J.F. Bard : An efficient point algorithm for a linear two stage optimization problem. Oper. Res., 31 : 670-684, (1983)
10. J.F. Bard : Practical bilevel optimization : algorithms and applications. Kluwer academic publishers, Dordrecht (1998)
11. O. Ben-Ayed, C. E. Blair : Computational difficulties of bilevel linear programming. Oper. Res., 38 : 556-560, (1990)
12. M. Campêlo, S. Dantas, S. Scheimberg : A note on a penalty function approach for solving bilevel linear programs. J. Glob. Optim., 16 : 245–255, (2000)
13. J. Clegg, M. Smith, Y. Xiang, R. Yarrow : Bilevel programming applied to optimising urban transportation. Transport. Res., Part B 35 : 41-70, (2001)
14. B. Colson, P. Marcotte, G. Savard : Bilevel programming : A survey. 4OR A Quarterly, J. Oper. Res., (2007)
15. S. Dempe : Foundations of Bilevel Programming. Kluwer academic publishers, Dordrecht (2000)
16. J. Fortuny-Amat, B. McCarl : A representation and economic interpretation of a two level programming problem. Oper. Res., 321 : 783-792, (1981)
17. S.R. Hejazi, A. Memariania, G. Jahanshahloob, M.M. Sepehria. Solving method for a class of bilevel linear programming based on Genetic Algorithms. Comp. and Oper. Res., 29 : 1913–1925, (2005)
18. L. Kuen-Ming, W. Ue-Pyng, S. Hsu-Shih, E. Stanley Leec : A hybrid neural network approach to bilevel programming problems. App. Math. Letters, (2006)
19. D. Li Zhu, Q. Xu, Z. Lin : A homotopy method for solving bilevel programming problem. Nonlinear Analysis, 57 : 917–928, (2004)
20. R.T. Rockafellar : Convex analysis. Princeton, USA (1970)
21. H. Tuy, A. Migdalas, N. T. Hoai-Phuong : A novel approach to Bilevel nonlinear programming. J. Glob. Optim., 38 : 527–554, (2007)

A characterization of locating-total domination edge critical graphs

Mostafa BLIDIA[1] and Widad DALI[2]

[1] Université de Blida, Laboratoire Lambda R-O, B.P. 270, Blida Algérie
[2] Université d'Alger, Dept. R-O, Alger, Algérie

Abstract. For a graph $G = (V, E)$ without isolated vertices, a subset D of vertices of V is a total dominating set, or just TDS of G, if every vertex in V is adjacent to a vertex in D. The total domination number $\gamma_t(G)$ is the minimum cardinality of a TDS of G. A subset D of V which is a total dominating set, is locating-total dominating set, or just LTDS of G, if for any two distinct vertices u and v of $V(G) - D$, $N(u) \cap D \neq N(v) \cap D$. The locating-total domination number $\gamma_L^t(G)$ is the minimum cardinality of a locating-total dominating set of G. A graph G is said to be a locating-total domination edge removal critical graph, or just γ_L^{t+}-ER-critical graph, *if* $\gamma_L^t(G-e) > \gamma_L^t(G)$ for all e non-pendant edge of E. The purpose of this paper is to characterize the class of γ_L^{t+}-ER-critical graphs.

Keywords: locating-domination, critical graph.

1 Introduction

Various types of criticality with respect to domination parameters (such as vertex and edge removal, vertex and edge addition) have been studied see for example [2] for surveys and references. In this paper we investigate graphs which are critical upon edge removal with respect to the locating total domination number.

Unless stated otherwise we follow the notation and terminology of [2]. Specifically, $N_G(v) = \{u \in V(G) \mid uv \in E(G)\}$ and $N_G[v] = N(v) \cup \{v\}$ denoted the *open* and *closed neighborhood,* respectively, of a vertex v of a graph $G = (V(G), E(G))$. A vertex of degree one is called a *pendant vertex* (or a *leaf*) and its neighbor is called a *support vertex.* We denote by $S(G)$ (resp. $L(G)$) the set of support vertices (resp. leaves) of G and by $L_v(G)$ the set of leaves adjacent to a support vertex v. A support vertex v is *strong (*resp. *weak)* if $|L_v| \geq 2$ (resp.$|L_v| = 1$). An edge incident with a leaf is called a pendant edge. We call the *core* of G the subset $V(G)\backslash(S(G) \cup L(G))$. The subgraph induced in G by a subset of vertices S is denoted $G[S]$. A subset S is an independent set if no edge exists between any two vertices of $G[S]$. We denote by $K_{1,p}$, $p \geq 1$ a star. Recall that a galaxy is a forest in which each component is a star, that is, every edge of a galaxy is pendant edge. If confusion is unlikely we omit the (G) from the above notation.

For a graph $G = (V, E)$ without isolated vertices, a subset D of vertices of V is a *total dominating set* (TDS) of G if every vertex in V is adjacent to a

vertex in D. The total domination number $\gamma_t(G)$ is the minimum cardinality of a TDS of G. A subset D of V which is a TDS is *locating-total dominating set,* or just LTDS of G, if for any two distinct vertices u and v of $V(G) - D$, $N(u) \cap D \neq N(v) \cap D$. The *locating-total domination number* $\gamma_L^t(G)$ is the minimum cardinality of a LTDS of G. Note that locating-total domination was introduced by Haynes, Henning and Howard [4].

By $\mu(G)$-set of G, where $\mu(G)$ is a domination parameter, we mean a vertex-set of G realizing $\mu(G)$, e.g., a $\gamma_t(G)$-set of G is a TDS X of G with $|X| = \gamma_t(G)$.

In this paper, we study the effects on increasing the locating-total domination number when an edge is deleted. Such problems have been considered before for some domination parameters. Sumner and Blitch [3] were the first introducing edge critical graphs for domination number.

When we remove a non-pendant edge e from a graph G, $G-e$ remains without isolated vertices, the locating-total domination number can increase, decrease or remains unchanged, e.g., if G is a P_5 then $\gamma_L^t(G) = 3$ and $\gamma_L^t(G - e) = 4$ for all e non-pendant edge of E. If G is a clique K_4 then $\gamma_L^t(G) = 3$ and $\gamma_L^t(G - e) = 2$ for all $e \in E$. If G is a P_6 then $\gamma_L(G) = \gamma_L(G - e) = 4$ for all e non-edge of E.

A graph G is said to be *a locating-total domination edge removal critical graph*, or just γ_L^{t+}-ER-critical graph*, if* $\gamma_L^t(G-e) > \gamma_L^t(G)$ for all e non-pendant edge of E.

The purpose of this paper is to give a descriptive characterization of the class of γ_L^{t+}*-ER*-critical graphs. In a similarly way, we have characterized in [1] the class of γ_L^+-ER-critical graphs.

2 Preliminary results

The following results will be of use throughout the paper.

Observation 1. *For every graph G, the set $S(G)$ of all support vertices is contained in every $\gamma_L^t(G)$-set and for each $v \in S(G)$, every $\gamma_L(G)$-set contains at least $|L_v|$ vertices in $\{v\} \cup L_v$*

Proposition 1. *If D is a $\gamma_L^t(G)$-set of a γ_L^{t+}-ER-critical graph $G = (V, E)$, then $V - D$ is an independent set.*

Proof. If an edge e exists in $G[V - D]$, then D is also an LTDS of $G - e$. Hence $\gamma_L^t(G - e) \leq \gamma_L^t(G)$, which contradicts that G is γ_L^{t+}-ER-critical graph.

Proposition 2. *If D is a $\gamma_L^t(G)$-set of a γ_L^{t+}-ER-critical graph $G = (V, E)$, then $G[D]$ is a galaxy.*

Proof. Suppose to the contrary that $G[D]$ is not a galaxy. So, $G[D]$ contains a non-pendant edge e. Since $G[D] - e$ is a subgraph without isolated vertices, D is LTDS of $G - e$ and $\gamma_L^t(G - e) \leq |D| = \gamma_L^t(G)$, so G is not a γ_L^{t+}-ER-critical graph, a contradiction.

Definition 3. Let $H = (V, E)$ be a connected graph which satisfies the following conditions:
(1) $V = X \cup Y$.
(2) $G[X]$ is a galaxy and Y is an independent set.
(3) For every y in Y and for every nonempty subset $X' \subseteq N(y)$ there exists an unique $y' \in Y$ such that $N(y') = X'$.
(4) For every support vertex x in X, $N(x) \cap Y \neq \emptyset$.

Let $\mathcal{H}$ be the set of all such graphs.

Examples:
- $P_4, P_5 \in \mathcal{H}$.
- For $p \geq 1, K_{1,p} \in \mathcal{H}$.

Fig. 1. Graph H of $\mathcal{H}$

Remark. Let H be a graph in $\mathcal{H}$.
X is a $\gamma_L^t(H)$-set.
The subgraph induced by the core of H is an independent set and for all vertex x in the core of H, $N(x) \cap Y = \emptyset$.

3 Characterization

Lemma 4. *If $H \in \mathcal{H}$, then H is a γ_L^{t+}-ER-critical graph.*

Proof. By definition of a locating-total domination edge removal critical graph, every star $K_{1,p}, p \geq 1$ is a γ_L^{t+}-ER-critical graph. Let $H = (V, E)$ be a graph in $\mathcal{H}$ different from $K_{1,p}, p \geq 1$. Delete any non-pendant edge $e = xy$. We have to consider only two cases.

Case 1. $x \in X$ and $y \in Y$.

By Definition 3, there exists $y' \in Y$ such that $N(y) - \{x\} = N(y')$, so X is not a LTDS of $H - e$. Now by Observation 1, Remark and since all neighbors of y are support vertices, $\gamma_L^t(H - e) \geq |X \cup \{y\}| = |X| + 1 = \gamma_L^t(H) + 1$. Hence, H is a γ_L^{t+}-ER-critical graph.

Case 2. $x \in X$ and $y \in X$.

By Remark, we have to consider two subcases.

Subcase 2.1. x and y be support vertices in H and one is a weak support, without loss of generality, let x be a weak support such that $(N(x)-\{y\})\cap X = \emptyset$. One of neighbors of x, different from y, must be in every LTDS of $H-e$, so by Observation 1, $\gamma_L^t(H-e) \geq |X|+1 = \gamma_L^t(H)+1$. Hence, H is a γ_L^{t+}-ER-critical graph.

Subcase 2.2. y is a support vertex in H and x is a vertex in the core of H, by Definition 3, $N(x)$ is a set of support vertices and $(N(y)-\{x\})\cap X = \emptyset$, so X is not a LTDS of $H-e$. Now by Observation 1, Remark, $\gamma_L^t(H-e) \geq |X \cup \{y'\}| = |X|+1 = \gamma_L^t(H)+1$, where y' is a vertex in Y adjacent to y. Hence, H is a γ_L^{t+}-ER-critical graph.

The following result characterize the class of γ_L^{t+}-ER-critical graphs.

Theorem 5. *A nontrivial connected graph $G=(V,E)$ is a γ_L^{t+}-ER-critical graph if and only if $G \in \mathcal{H}$*

Proof. The 'if ' part follows from Lemma 4, so let us prove the 'Only if' part. If $G = K_{1,p}, p \geq 1$, then $G \in \mathcal{H}$. Let $G=(V,E)$ be a connected γ_L^{t+}-ER-critical graph different from $K_{1,p}, p \geq 1$. Let X be a $\gamma_L^t(G)$-set of G. By Proposition 1 and Proposition 2, $Y = V - X$ is an independent set and $G[X]$ is a galaxy. Hence, part **(1)** and **(2)** of the Definition 3 are proved. Now, it remains to prove Condition **(3)** and Condition **(4).**

Proof of Condition **(3)**. For that, let $y \in Y$, $N(y) = \{x_1, ..., x_k\}, k \geq 1$ and $X' \subseteq N(y)$. We consider the following cases.

Case 1. $|X'| = k$. If $k = 1$, then $X' = N(y) = \{x_1\}$ and y is a pendant vertex. As X is a $\gamma_L^t(G)$-set of G, y is a unique vertex in Y such that $N(y) = X'$. If $k \neq 1$, then since X is a $\gamma_L(G)$-set of G, y is the unique vertex in Y such that $N(y) = X'$.

Case 2. $2 \leq |X'| \leq k-1$. If $|X'| = k-1$. Without loss of generality, let $X' = \{x_1, ..., x_{k-1}\}$, then it must exist an unique vertex $y^l \in Y$ with $N(y^l) = \{x_1, ..., x_{k-1}\}$, for otherwise D is a LTDS of $G-e$ with $e = yx$ and $x \in N(y) - N(y^l)$ which contradicts that G is γ_L^{t+}-ER-critical graph. We repeat this process for $y^l \in Y$ with $N(y^l) = \{x_1, ..., x_l\}$ where $2 \leq l \leq k-2$. Consequently, there exists $y^l \in Y$ with $N(y^l) = X'$.

Proof of Condition **(4).** Let x be a support vertex of G in $G[X]$. We suppose that $N(x) \cap Y = \emptyset$. So $N[x] \subset X$, as G is not a star, there exist at least a vertex $y \in N(x) \backslash L_x$, where L_x is the set of leaves adjacent to x, then for a pendent vertex $x' \in L_x$, the subset $X - \{x'\}$ is a LTDS of G smaller than X, a contradiction.

Notice that a disconnected graph G is γ_L^{t+}-ER-critical graph if and only if each component of G is γ_L^{t+}-ER-critical graph. So we have the following result.

Corollary 6. *A nonempty graph $G = (V, E)$ is γ_L^{t+}-ER-critical graph if and only if G is the union of graphs of $\mathcal{H}$.*

References

1. M. Blidia and W. Dali, A characterization of a locating-domination edge critical graphs. Submitted to *Australasian Journal of Combinatorics.*.
2. T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Fundamentals of Domination in Graphs. Marcel Dekker, New York, 1998.
3. D.P. Sumner and P. Blitch, Domination critical graphs. *J. Combin. Theory Ser.* B 34 (1983) 65-76.
4. T.W.Haynes, M. A. Henning and J. Howard, Locating and total dominating sets in trees. *Discrete Applied Mathematics* 154 (2006), 1293-1300.

This article was processed using the LaTeX macro package with LLNCS style

Identification des paramètres d'un modèle de diffusion par la méthode combinée Adomian/Alienor

Benzitouni Radhia , Manseur Salah

Laboratoire LAMDA-RO, Départment de Mathématiques, Université Saad Dahlab de Blida.

B.P. 270, Blida, Algeria.

(r_benzitouni@yahoo.fr) , (smanseur@yahoo.fr)

Résumé

Dans cet article, on considère le problème d'identification paramétrique d'un modèle représenté par une équation aux dérivées partielles (EDP) de type parabolique. Ce problème consiste à déterminer les paramètres inconnus du modèle en minimisant une fonctionnelle d'erreur. L'utilisation de la méthode combinée Adomian/Alienor permet de ramener le problème d'identification des paramètres en un problème de minimisation d'une fonction à une seule variable.

L'identification des paramètres d'un modèle biologique de diffusion a été étudié, et les résultats numériques obtenus sont satisfaisants.

MOTS CLÉS : Identification paramétrique, méthode Adomian, méthode Alienor, modèle de diffusion.

1 Introduction

De nombreux problèmes du monde réel, peuvent être représentés par une formulation mathématique (équations différentielles, équations aux dérivées partielles, équations intégrales,...etc). Les équations aux dérivées partielles interviennent aujourd'hui dans la modélisation de nombreux phénomènes provenant de la biologie, de l'économie, de la géologie et même de la finance. Ces modèles mathématiques dépendent en général de paramètres et le problème se pose alors de les déterminer, c'est à dire les identifier.

Dans cet article, on s'intéresse au problème d'identification des paramètres inconnus d'un modèle ([3],[8],[5]). Ce problème peut être résolu par les méthodes de maximum de vraisemblance, de sous-espace ou de fréquence ([6]).

Le problème d'identification paramétrique se ramène à la minimisation d'une fonctionnelle qui calcule la somme des carées des écarts entre les données expérimentales et les données calculées. Donc ce problème va conduire directement à un problème d'optimisation.

Dans ce travail, on a utilisé la méthode décompositionnelle d'Adomian et celle d'Alienor ([2],[3]) pour la résolution du problème.

La méthode décompositionnelle d'Adomian ([2],[3],[4],[1]) permet d'exprimer la solution du modèle sous forme de série convergente, et dépendante explicitement des paramètres inconnus. Cette méthode est basée sur la recherche d'une solution sous la forme d'une série et sur la décomposition en série de l'opérateur non linéaire en utilisant des polynômes appelés "polynômes d'Adomian".

La méthode Alienor ([1],[9],[7]) permet de transformer un problème de minimisation d'une fonction de n variables à un problème de minimisation d'une fonction d'une seule variable, en utilisant des transformations réductrices qui permettent de construire des courbes $\alpha-$denses afin de densifier l'espace $\mathbb{R}^n$.

La combinaison de ces deux méthodes permet de ramener le problème d'identification des paramètres en un problème de minimisation d'une fonction à une seule variable. Cette technique a été utilisée pour l'identification de modèles décrits par des équations différentielles (systéme d'équations différentielles) ([8]). Ici, on considère un problème modélisé par une équation aux dérivées partielles (EDP).

L'article est organisé comme suit: dans la section 2, on donnera une formulation du problème d'identification. Dans la section 3, on présentera les deux méthodes Adomian et Alienor, puis on les appliquera au problème d'identification. Un modèle biologique de diffusion sera étudié dans la section 4, et on terminera par une conclusion.

2 Formulation du problème

Considérons le problème d'identification des paramètres d'un modèle représenté par une équation aux dérivées partielles (EDP) de type parabolique de la forme ([5]):

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - KC, \qquad 0 \leq x \leq 1, \quad 0 \leq t \leq 1 \tag{1}$$

sous les conditions:

$$C(x,0) = \varphi(x), \qquad 0 \leq x \leq 1 \tag{2}$$

$$C(0,t) = g(t), \qquad 0 \prec t \leq 1 \tag{3}$$

$$C(1,t) = h(t), \qquad 0 \prec t \leq 1 \tag{4}$$

où φ, g et h sont des fonctions supposées connues, D et K sont des paramètres positifs inconnus.

L'équation (1) est une équation de diffusion, qui peut représenter par exemple la diffusion des antibiotiques ou des substances antitumorales dans le cerveau. Dans ce cas, $C(x,t)$ représente la concentration du médicament.

Le problème d'identification paramétrique consiste à déterminer les deux paramètres inconnus D et K (qu'on suppose constants) à partir de données expérimentales, en minimisant une fonctionnelle J définie par:

$$J = \sum_{j=1}^{m} \sum_{i=1}^{l} (C_i^j - C(x_i, t_j))^2 \tag{5}$$

Dans (5), les C_i^j sont les données expérimentales mesurées aux instants t_j, $j = 1, ..., m$ et aux positions x_i, $i = 1, ..., l$, les $C(x_i, t_j)$ représentent les valeurs calculées en résolvant l'EDP, aux instants t_j, $j = 1, .., m$ et aux positions x_i, $i = 1, ..., l$.

Donc pour résoudre ce problème, il est nécessaire de définir la fonctionnelle à minimiser qui dépend du type de problème considéré, et de choisir une technique adéquate pour effectuer la minimisation de cette fonctionnelle. Pour notre part, on a choisi la méthode combinée Adomian/Alienor.

3 Méthodes mathématiques

3.1 Méthode décompositionnelle d'Adomian

La méthode décompositionnelle d'Adomian est utilisée pour résoudre des équations fonctionnelles linéaires et non linéaires de différents types (différentielles, aux dérivées partielles, intégrales, algébriques,...etc).

Considérons l'équation fonctionnelle (sous la forme canonique) suivante:

$$u - N(u) = f \tag{6}$$

où N est un opérateur non linéaire, f est une fonction connue et u est l'inconnue de l'équation (6).

La méthode décompositionnelle d'Adomian consiste à chercher la solution u (si elle existe) sous forme d'une série:

$$u = \sum_{n=0}^{+\infty} u_n \tag{7}$$

et à décomposer l'opérateur non linéaire $N(u)$ en série ([3]) :

$$N(u) = \sum_{n=0}^{+\infty} A_n(u_0, u_1, ..., u_n) \tag{8}$$

où les A_n sont appelés"polynômes d'Adomian" qui dépendent de $u_0, ..., u_n$ et sont obtenus à partir des relations suivantes ([3]):

$$v = \sum_{i=0}^{+\infty} \lambda^i u_i \ , \ N(\sum_{i=0}^{+\infty} \lambda^i u_i) = \sum_{i=0}^{+\infty} \lambda^i A_i \tag{9}$$

D'où

$$A_n = \frac{1}{n!}\frac{d^n}{d\lambda^n}[N(\sum_{i=0}^{n}\lambda^i x_i)]_{\lambda=0}\ ,\ n = 0, 1, 2, ... \tag{10}$$

où λ est un paramétre introduit par convenance.

En remplaçant les expressions (7) et (8) dans (6) on peut écrire:

$$\sum_{n=0}^{+\infty} u_n - \sum_{n=0}^{+\infty} A_n = f \tag{11}$$

En identifiant les deux membres de l'équation (11), on obtient :

$$\begin{cases} u_0 = f, \\ u_1 = A_0(u_0), \\ u_2 = A_1(u_0, u_1), \\ \vdots \\ u_{n+1} = A_n(u_0, ..., u_n), \quad n = 0, 1, ... \end{cases} \tag{12}$$

La solution exacte de l'équation est déterminée à partir de la formule (12). En pratique, puisqu'on ne peut pas calculer tous les termes de la série, on se contente d'une approximation de la solution, en prenant une série tronquée d'ordre s:

$$\varphi_s = \sum_{n=0}^{s-1} u_n, \quad \text{avec } \lim_{s\longrightarrow+\infty} \varphi_s = u. \tag{13}$$

A partir de la formule (10), les polynômes d'Adomian sont déterminés ([12]):

$$\begin{cases} A_0(u_0) = N(u_0), \\ A_1(u_0, u_1) = u_1 \frac{d}{du_0} N(u_0), \\ A_2(u_0, u_1, u_2) = u_2 \frac{d}{du_0} N(u_0) + \frac{u_1^2}{2!}\frac{d^2}{du_0^2} N(u_0), \\ A_3(u_0, u_1, u_2, u_3) = u_3 \frac{d}{du_0} N(u_0) + u_1 u_2 \frac{d^2}{du_0^2} N(u_0) + \frac{u_1^3}{3!}\frac{d^3}{du_0^3} N(u_0) \\ \qquad\qquad\qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad\qquad\qquad . \end{cases} \tag{14}$$

De même, dans ([3],[13],[14]) les polynômes A_n sont calculés par les formules:

$$\begin{cases} A_0(u_0) = N(u_0), \\ A_n(u_0, u_1, ..., u_n) = \sum_{\alpha_1+\alpha_2+...+\alpha_n=n} N^{(\alpha_1)}(u_0) \frac{u_1^{(\alpha_1-\alpha_2)}}{(\alpha_1-\alpha_2)!} ... \frac{u_{n-1}^{(\alpha_{n-1}-\alpha_n)}}{(\alpha_{n-1}-\alpha_n)!} \frac{u_n^{\alpha_n}}{\alpha_n!}, \quad n = 1, 2, ... \end{cases} \tag{15}$$

Pour justifier la convergence de la série $\sum u_n$ et donc la série $\sum A_n$, ([3]) on peut utiliser les propriétés des séries substituées dans une autre série, en supposant que l'opérateur non linéaire N soit une contraction ($\|N\| \prec \delta \prec 1$). D'autres résultats de convergence sont donnés dans ([2]) et ([10]), sous l'hypothèse que l'opérateur N soit indéfiniment différentiable (au sens de Frechet).

3.2 Application de la méthode décompositionnelle d'Adomian à la résolution de l'EDP

Considérons l'EDP (1) avec les conditions (2),(3)et (4).

En posant $L_x = \frac{\partial^2}{\partial x^2}$ et $L_t^{-1} = \int_0^t (.)ds$, l'équation (1) s'écrit sous la forme canonique:

$$C(x,t) = C(x,0) + L_t^{-1}(DL_xC - KC) \tag{16}$$

On cherche la solution sous la forme:

$$C(x,t) = \sum_{n=0}^{+\infty} C_n(x,t) \tag{17}$$

On remplace l'expression (17) dans l'expression (16), on peut écrire:

$$\sum_{n=0}^{+\infty} C_n(x,t) = C(x,0) + L_t^{-1}(DL_x(\sum_{n=0}^{+\infty} C_n(x,t))) - K(\sum_{n=0}^{+\infty} C_n(x,t))) \tag{18}$$

On obtient les termes de la série solution, en identifiant les deux membres de l'équation (18) comme suit:

$$\begin{cases} C_0 = C(x,0) = \varphi(x), \\ C_{n+1} = L_t^{-1}(DL_xC_n - KC_n),\ n \geq 0 \end{cases} \tag{19}$$

3.3 La Méthode Alienor

La méthode Alienor est une méthode d'optimisation globale ([1],[3]). Cette méthode est proposée par Y.Cherruault et A.Guillez ([11]) permet de ramener la minimisation des fonctions de n variables à la minimisation des fonctions d'une seule variable, en utilisant des transformations réductrices.

Rappelons quelques définitions et résultats.

Définition 1: Un sous-ensemble $S \subseteq \mathbb{R}^n$ est dit $\alpha - dense$ ($\alpha \succ 0$), si pour tout point X de $\mathbb{R}^n$, il existe un point Y de S tel que $d(X,Y) \leq \alpha$, où d est la distance euclidienne de $\mathbb{R}^n$.

De plus, si S est défini par:

$$x_i = h_i(\theta),\ i = 1, ..., n \tag{20}$$

alors $h(\theta) = (h_1(\theta), h_2(\theta), ..., h_n(\theta))$ est appelée courbe $\alpha - dense$ ou transformation réductrice.

Dans cet article, nous avons utilisé les deux transformations réductrices suivantes:

3.3.1 $1^{ére}$transformation réductrice ([8])

Elle est définit par:

$$\begin{cases} x_1 = \theta, \\ x_i = h_i(\theta) = \frac{1}{2}(1 - \cos m^{i-1}\pi\theta),\ \theta \geq 0,\ i = 1, ..., n \ \text{ avec } m = 2 \text{ ou } m = 3. \end{cases} \tag{21}$$

Cette transformation réductrice est $\alpha - dense$ dans [0,1]^n.

3.3.2 $2^{éme}$transformation réductrice ([1])

$$x_i = h_i(\theta) = \cos \alpha_i \theta, \quad i = 1, ..., n \tag{22}$$

qui est $\alpha - dense$ dans [-1,1]^n. La suite $(\alpha_i)_i$ est une suite lentement croissante. Y.Cherruault (2004) a définit la suite $(\alpha_i)_i$ comme suit:

$$\begin{cases} \alpha_1 = 1, \\ \alpha_i = \frac{i+0.9}{i+1}, ...i = 2, ..., n. \end{cases} \tag{23}$$

Considérons le problème de minimisation:

$$\underset{(x_1, x_2, ..., x_n) \in \prod_{i=1}^{n} [a_i, b_i]}{Min} F(x_1, x_2, \ldots, x_n) \tag{24}$$

où F est une fonction continue sur $\mathbb{R}^n$ et vérifiant la condition de croissance à l'infini:

$$\lim_{x_1^2 + ... + x_n^2 \to \infty} F(x_1, \ldots, x_n) = +\infty \tag{25}$$

La méthode Alienor appliquée au problème (24) consiste à utiliser une transformation réductrice:

$$x_i = h_i(\theta) \quad \theta \geq 0,\ i = 1, \ldots, n, \tag{26}$$

où $h_i(\theta) \in C^\infty$ tel que $h(\theta) = (h_1(\theta), ..., h_n(\theta))$ soit une courbe α-*dense*. Le problème (24) est approché par un problème de minimisation d'une fonction à une seule variable:

$$\underset{\theta \in [\ 0,\ \theta_{\max}]}{Min} F^*(\theta) \tag{27}$$

où $F^*(\theta) = F(h_1(\theta), h_2(\theta), ..., h_n(\theta))$ et $\theta_{\max}$est la plus grande valeur que peut prendre θ.

Y.Cherruault en ([1]) a montré que tout minimum du problème (24) peut être approché par un minimum du problème (27).

Donc, la méthode Alienor transforme un problème de minimisation d'une fonction multivariables à un problème de minimisation d'une fonction monovariable. La fonction $F^*(\theta)$ qui en découle possède en général, plusieurs minima locaux. Mora, G., Cherruault, Y.,et Benabidallah A., dans ([7]), ont proposé une méthode d'optimisation basée sur l'O.P.O (Opérateur qui Préserve l'Optimisation),

qui permet d'éliminer progressivement tous les minima locaux pour ne conserver à la fin qu'un minimum global. l'idée de base consiste à appliquer un opérateur $T^{\varepsilon}_{F^*}(\theta)$ qui agit sur la fonction d'une seule variable $F^*(\theta)$.

3.4 Application de la méthode Adomian/Alienor au problème d'identification

Revenons à notre problème d'identification, qui consiste à minimiser la fonctionnelle J par rapport aux paramètres inconnus. Cette minimisation nécessite la résolution de l'EDP un grand nombre de fois, puisque la fonction J dépend implicitement des paramètres D et K à identifier. De plus, si on utilise par exemple, une méthode de type gradient, il est nécessaire de connaitre les dérivées de la fonctionnelle par rapport aux paramètres, ce qui n'est pas toujours facile.

L'utilisation de la méthode décompositionnelle d'Adomian à la résolution de l'EDP, permet d'avoir une solution approchée qui dépend explicitement des paramètres D et K, en prenant une série tronquée d'ordre s donnée par:

$$C(x,t) = \sum_{n=0}^{s-1} C_n(D,K,x,t) \tag{28}$$

où C_n , $n = 0, ..., s-1$, sont les termes de la série d'Adomian.

En remplaçant l'expression (28) dans l'expression (5), la fonctionnelle J dépendra explicitement des paramètres D et K.

La méthode Alienor permet alors de transformer le problème:

$$\min_{D,K} J(D,K) \tag{29}$$

en

$$\min_{\theta\in[0,\theta_{\max}]} J^*(\theta) \tag{30}$$

en utilisant la transformation réductrice

$$\begin{aligned} D &= h_1(\theta),\ \theta \geq 0 \\ K &= h_2(\theta),\ \theta \geq 0 \end{aligned} \tag{31}$$

où $h_1, h_2 \in C^\infty$ sont des fonctions choisies de manière à densifier l'espace $\mathbb{R}^2$. La fonction J^* est continue sur un domaine fermé et borné, elle possède au moins un minimum dans $[0, \theta_{\max}]$.

4 Résultats numériques

Reprenons le problème (1) défini par:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - KC, \qquad 0 \leq x \leq 1, \quad 0 \leq t \leq 1,$$

sous les conditions (2), (3) et (4):

$$C(x,0) = \varphi(x), \quad 0 \leq x \leq 1$$

$$C(0,t) = g(t), \quad 0 \prec t \leq 1$$

$$C(1,t) = h(t), \quad 0 \prec t \leq 1$$

Notre but est d'identifier les paramètres D et K, en minimisant la fonctionnelle d'erreur (5).

Comme condition initiale, on prend $\varphi(x) = e^{-(x-0.05)^2)}$, $0 \leq x \leq 1$.

Les valeurs de la solution de l'EDP $C(x_i, t_j)$ sont déterminées par la méthode décompositionnelle d'Adomian, en prenant une série tronquée d'ordre 4. La solution obtenue dépend explicitement des paramètres D et K, de la variable x et de la variable t :

$$C(x,t) = \sum_{n=0}^{3} C_n(D,K,x,t) \tag{32}$$

On suppose que les paramètres exacts sont $D = 0.06$ et $K = 0.05$, et on pose: $g(t) = 0.9$, $0 \prec t \leq 1$ et $h(t) = 0.4$, $0 \prec t \leq 1$.

Considérons aussi, la solution de l'EDP obtenue par la méthode des différences finies (schéma implicite) à partir des paramètres exacts comme données expérimentales C_i^j calculées en 10 instants de mesures et 10 positions, $i = 1, ..., 10$, $j = 1, ..., 10$.

La fonctionnelle J est déterminée à partir de la solution de l'EDP obtenue par la méthode d'Adomian et des données expérimentales simulées. on passe à sa minimisation par la méthode Alienor, en utilisant les deux transformations définit précedèment (paragraphe 3.3),pour deux variables D et K:

$1^{ére}$transformation réductrice:

$$\begin{cases} D = \theta \\ K = \frac{1}{2}(1 - \cos(3\pi\theta)) \end{cases}, \theta \geq 0 \tag{33}$$

$2^{éme}$ transformation réductrice:

$$\begin{cases} D = \cos(3\theta + 1) \\ K = \cos(3.1\theta + 1.1) \end{cases}, \theta \geq 0 \tag{34}$$

En utilisant la $1^{ére}$transformationn on approche la fonction $J(D,K)$ par la fonction $J_1^*(\theta)$ qui dépend d'une seule variable. Sa courbe représentative permet de situer le minimum global dans l'intervalle [0,1].

Plus précisément, le minimum de la fonction $J_1^*(\theta)$ est atteint pour $\theta = 0.05483$.

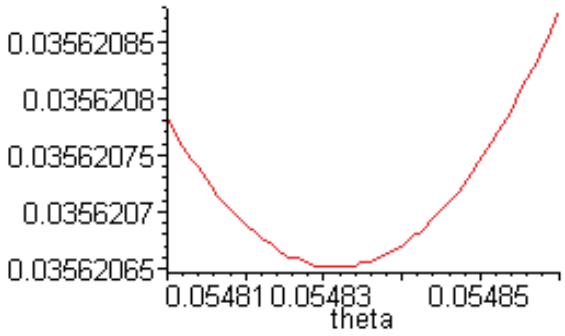


Figure 1: Courbe représentative de la fonction $J_1^*(\theta)$.

De même, en utilisant la $2^{éme}$ transformation réductrice, on approche la fonction $J(D, K)$ par la fonction $J_2^*(\theta)$. Le minimum global dans ce cas, est calculé par l'O.P.O, puisque la fonction $J_2^*(\theta)$ admet plusieurs minima locaux. La courbe de la fonction $J_2^*(\theta)$ est représentée par la figure 2:

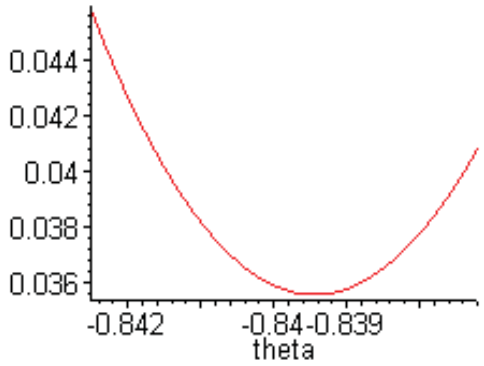


Figure 2: Courbe représentative de la fonction $J_2^*(\theta)$.

Le tableau suivant résume les valeurs des paramètres obtenus à partir des deux transformation réductrices, ainsi que la valeur de la fonctionnelle J.

Paramètres	$1^{ére}$transformation	$2^{éme}$transformation
D	0.054803	0.05377
K	0.06528	0.06983
La valeur de J	0.03562	0.03576

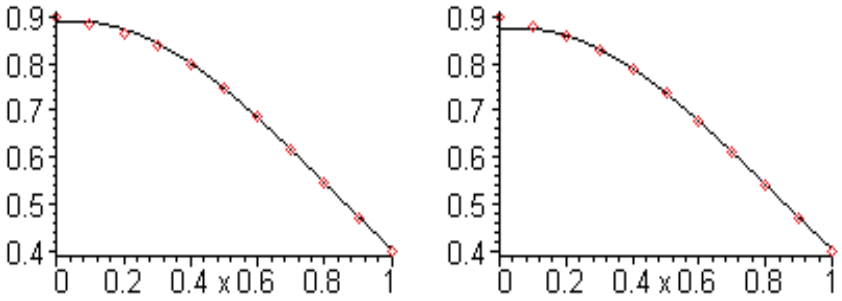
Table1: Valeurs des paramètres et de la fonction J obtenus aprés identification.

4.0.1 Discussion

On remarque d'aprés le tableau précèdent que les paramètres identifiés sont proches des paramètres exacts. Ces paramètres sont reportés dans la solution de l'EDP afin de les comparer avec les données

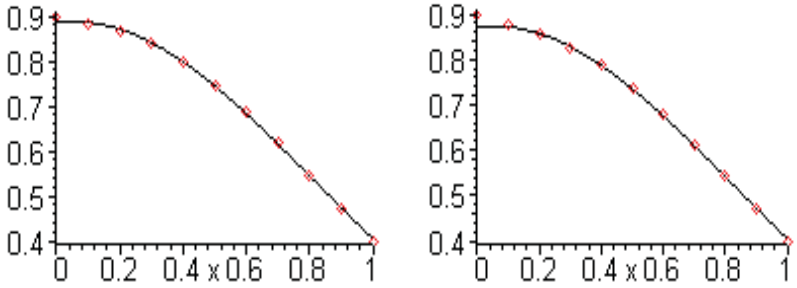
expérimentales.

La comparaison des courbes des données expérimentales (en points) avec les courbes de la solution en fonction des paramètres identifiés (lignes) sont illustrées par les figures suivantes:



t=0.7 t=0.8

Figure4: Superposition des courbes expérimentales (points) avec les courbes de la solution obtenue par la méthode Adomian/$1^{ére}$ transformation pour t=0.7 et t=0.8.



t=0.7 t=0.8

Figure2: Superposition des courbes expérimentales (points) avec les courbes de la solution obtenue par la méthode Adomian/$2^{éme}$ transformation pour t=0.7 et t=0.8.

On remarque qu'on a une bonne superposition des courbes, en utilisant les méthodes combinées Adomian/$1^{ére}$ transformation et Adomian/$2^{éme}$ transformation. On conclu donc que la combinaison de ces méthodes (Adomian et Alienor) aboutit à des résultats satisfaisants, avec des paramètres proches des paramètres exacts.

5 Conclusion

Le problème d'identification des paramètres d'un modèle représenté par une EDP parabolique a été étudié, en utilisant la combinaison des deux méthodes: la méthode décompositionnelle d'Adomian et la méthode Alienor. Ceci à permis de ramener le problème d'identification des paramètres à un problème de minimisation d'une fonction d'une seule variable.

Cette approche a été appliquée au problème d'identification des paramètres d'une EDP décrite par une équation de diffusion. La comparaison des solutions obtenues par la méthode combinée Adomian/Alienor avec les données expérimentales du modèle donne des résultats acceptables.

RÉFÉRENCES

[1]. Cherruault, Y.(1999). *Optimisation, méthodes locales et globales.* Presses Universitaires de France (P.U.F), paris.

[2]. Abbaoui, K. (1995). *Les fondements mathématiques de la méthode décompositionnelle d'Adomian et application à la résolution de problèmes issus de la biologie et de la médecine.* Thèse de l'Université Paris VI. Laboratoire MEDIMAT.

[3]. Cherruault, Y. (1998). *Modèles et méthodes mathématiques pour les sciences du vivant,* Presses Universitaires de France (P.U.F) Paris.

[4]. Khelifa, S. (2002). *Equations aux dérivées partielles et méthode décompositionnelle d'Adomian.* Thèse de Doctorat Es sciences. Université H.Boumedienne, Algeries.

[5]. Bellagoun, A., Cherruault, Y.,Meulemans, A.(1993).*Modelling and identification of drug diffusion in brain,* Bellman, Mathl, comput. V.17, $n^0$9, pp.101-105.

[6].Ljung, L.(1987). *New System identification: theory for the user,* Int J, Information and system sciences series, Prentice Hall, En glewood Cliffs, NJ.

[7]. Mora, G., Cherruault, Y., Benabidallah, A. (2003). *Global optimisation-preserving operators* . Kybernetes, Vol. 32, n^0 9/10, pp. 1473-1480.

[8].Manseur, S., Messaoudi, N.(2005). *Identification des paramètres d'un systéme Immunitaire du HIV par la méthode combinée Adomian/Alienor,* Cosi'05, 12-14 Bejaïa, Algérie.

[9]. Benneouala, T., Cherruault, Y. (2005). *Alienor method for global optimization with large number of variables.* Kybernetes, Vol. 34, n^0 7/8, pp. 1104-1111.

[10] Abbaoui, K., Cherruault, Y., (1995). *New ideas for proving convergence of decomposition methods.* Computers Math. Applic. Vol. 29, n^0 7, pp. 103-108.

[11]. Cherruault, Y., Guillez, A. *(1993). Une méthode pour la recherche du minimum global d'une fonctionnelle.* CRAS, Paris, T296, série1, pp. 175-178.

[12] Adomian, G., (1991). *A review of the decomposition method and some recent results for nonlinear equations.* Computers.Math.Applic. Vol. 21, n^0 5, pp.101-127.

[13] Abbaoui, K., Cherruault, Y., (1994). *Convergence of the Adomian method applied to non-linear equations.* Mathematical and Computer Modelling, Vol 20. n^0 9, pp. 60-73.

[14] Abbaoui, K., Cherruault, Y., (1994). *Convergence of Adomian method applied to differential equations.* Mathematical and Computer Modelling, Vol. 28, n^0 5, pp 103-9.

Les graphes triangulés sont des graphes B_1-orientables

Merzak Taflis*, Bachir Sadi*

*Département de mathématiques, Faculté des sciences,
Université de Tizi-Ouzou

email: mertaflis@yahoo.fr ; sadibach@yahoo.fr

Abstract. Après la définition de la classe des graphes B_1-orientables par J.P.Spinrad, en 1997, et la conjecture énoncée à ce propos, par Urrutia affirmant qu'un graphe B_1-orientable est le graphe d'intersection de sous-arbres d'un graphe planaire [15], B.Sadi a écrit un algorithme polynomial de reconnaissance de cette classe, dans [14].
Notre travail consiste en l'écriture de deux algorithmes, l'un basé sur Lex-BFS dans la reconnaissance des graphes triangulés [13],le deuxième utilisant l'algorithme de Johnson et al. [8], pour la génération des cliques maximales dans un graphe. Ces deux algorithmes montrent que les graphes triangulés sont des graphes B_1-orientables. .

Mots clés : Orientation de graphes, graphes triangulés, ordre d'élimination simplicial.

1 Introduction .

Dans notre article, après avoir rappelé de la définition des graphes B_1-orientables [15], on a énoncé le problème sous forme d'un théorème,en disant que les graphes triangulés sont des graphes B_1-orientables. Pour prouver ce théorème, on a developpé deux algorithmes qui font des graphes triangulés des graphes B_1-orientables.
Le premier algorithme est constitué de deux parties ; la première, c'est Lex-BFS, et pour cela, on a cité les traveaux donnés par certains auteurs [7], sur lequels ce dernier (Lex-BFS) a été construit, dont le principe, est de donner un ordre d'élimination simplicial si et seulement si le graphe est triangulé. La deuxième partie de cet algorithme est un fragment q'on a ajouté à Lex-BFS, dont le rôle est de donner au graphe une orientation liée à l'ordre d'élimination simplicial des sommets.
Le deuxième algorithme a le même principe que le premier, c'est toujours donner un ordre d'élimination simplicial, et pour cela, on a utilisé l'algorithme de Johnson et al. pour la génération des cliques maximales dans un graphe [8], comme une première partie. Dans sa deuxième partie, on a complété l'algorithme de Johnson et al. par un algorithme qui donne un certain ordre d'élimination simplicial, à partir des cliques maximales générées par l'algorithme précédent

(Johnson et al.), où sur chaque étape de l'ordre, il oriente un ensemble d'arêtes, en s'appuyant sur la notion de sommet simplicial.
Dans la suite, on va donner plus de details, pour comprendre mieux le fonctionnement de ces deux algorithmes.

2 Enoncé du problème .

Définition 0.1: Soit $G = (X, E)$ un graphe non orienté avec X l'ensmble des sommets et E l'ensemble des arêtes. Notons xy (resp. (x, y)), l'arête xy (resp. l'arc (x, y)).
G est B_1-orientable s'il admet une orientation U sur toutes ses arêtes telle que: $x, y, z \in X$; $(x, y) \in U$ et $(z, y) \in U \Rightarrow xz \in E$.
On note alors $G_\circ = (X, U)$ le graphe orienté, où U est la B_1-orientation cherchée.

Exemple 0.2: Si G est une clique, toute orientation des arêtes de G est une B_1-orientation.

Théorème 0.3: Tout graphe triangulé est B_1-orientable.

La démonstration de ce théorème se fait par l'un ou l'autre des deux algorithmes qu'on propose dans ce travail.
Le premier algorithme est basé sur la reconnaissance des graphes triangulés(Lex-BFS), le deuxième sur la génération des cliques maximales (Johnson et al.) dans un graphe. Mais avant de les écrire, nous rappelons simplement quelques éléments.

3 Rappel .

Définition 0.4: On note C_k, $k \geq 4$, le cycle de longueur ≥ 4, sans corde (arête reliant deux sommets non -consécutifs d'un cycle).
Un graphe $G = (X, E)$ est triangulé s'il ne contient pas de C_k.

Exemple 0.5: Arbre, clique.

Notation 0.6: Soit $G = (X, E)$ un graphe et $\acute{X} \subseteq X$ un sous-ensemble de sommets. Alors $G[\acute{X}]$ désigne le sous- graphe induit par les sommets de $\acute{X}$.

Le premier enseignement issu de la définition des graphes triangulés est que cette classe est une classe des graphes héréditaires. C'est ce qu'affirme le lemme suivant:

Lemme 0.7: Soit $G = (X, E)$ un graphe tringulé. Tout sous-graphe $H = G[\acute{X}]$ de G, où $\acute{X} \subseteq X$, est un graphe triangulé.

En effet, la suppression d'un quelconque ensemble S de sommets du graphe G ne peut en aucun cas créer de C_k, $k \geq 4$.

Définition 0.8: Un sommet x est simplicial si son voisinage $N(x)$ est une clique.

Définition 0.9: (équivalente) Un sommet x est simplicial si x appartient une seule clique maximale.

Définition 0.10: Un ordre $\sigma = [x_1, ..., x_i, ...x_n]$ est un ordre d'élimination simplicial si et seulement si pour tout $1 \leq i \leq n$, le sommet x_i est simplicial dans le graphe $G_i = G[x_i, ..., x_n]$.

Notation 0.11: Soit $\sigma = [x_1, ..., x_i, ..., x_n]$ un ordre. Dans la suite, E_i désignera toujours les sommets $\{x_i, ..., x_n\}$. Le sous-graphe induit par les sommets de E_i sera noté $G_i = G[x_i, ..., x_n]$.

Dirac, en 1961, a montré la propriété fondamentale suivante qui explique l'existence d'un ordre d'élimination simplicial dans les graphes triangulés.

Lemme 0.12: [Dir61] Tout graphe triangulé $G = (X, E)$ possède un sommet simplicial. Si G n'est pas un graphe complet, alors il possède au moins deux sommets simpliciaux non-adjacents.

Fulkerson et Gross, en 1965, ont caractérisé les graphes triangulés de la manière suivante:

Théorème 0.13: [FG65] Un graphe est triangulé si et seulement si il existe un ordre d'élimination simplicial.

Démonstration :$\Longrightarrow$) Cette implication découle de l'application récursive du lemme précédent: Dans un graphe triangulé, on trouve un sommet simplicial, on le supprime, le graphe obtenu reste un graphe triangulé.....
($\Longleftarrow$ Si un graphe n'est pas triangulé, alors il possède un cycle de plus de 3 sommets sans corde, et aucun sommet de ce cycle ne peut être éliminé simplicialement.

C'est toujours cette caractérisation (existence d'un ordre d'élimination simplicial) qui a mené à l'algorithme linéaire de reconnaissance des graphes triangulés (Lex-BFS) donné par Rose, Tarjan et Leuker en 1976. Il construit un ordre d'élimination simplicial à l'envers: il détermine d'abord le dernier sommet x_n de l'ordre et termine par le premier sommet x_1. Pour comprendre ce fonctionnement, nous devons utiliser un peu plus finement les conséquences de la proprieté de Dirac (lemme 0.12):

Lemme 0.14: Soit $G = (X, E)$ un graphe triangulé. Pour tout sommet $x \in X$,

il existe un ordre d'élimination simplicial σ terminant par x.

Preuve : Soit γ un ordre d'élimination simplicial de G et $\gamma(x) = i$. A partir de γ, nous allons construire un ordre d'élimination σ tel que $\sigma(n) = x$:
Si G est un graphe complet, il suffit d'échanger x et $\gamma(n)$ pour obtenir l'ordre d'élimination simplicial σ.
Si $i = n$, alors $\sigma = \gamma$. Supposons que $i \neq n$. Alors pour tout $1 \leq j < i$, prenons $\sigma(j) = \gamma(j)$. Par hypothèse x est simplicial dans le sous-graphe G_i. Or le lemme 0.12 montre qu'il existe dans G_i un sommet simplicial $y \neq x$. Donc on peut choisir $\sigma(i) = y$.
Si y n'est pas voisin de x dans G_i, le voisinage de x reste inchangé et reste donc une clique. Sinon y a été supprimé du voisinage de x. Mais tout sous-graphe d'une clique est une clique. Donc x reste simplicial dans le sous-graphe G_{i+1}.
Donc pour tout $i \leq k \leq n$, nous pouvons appliquer l'argument précédent. Ceci montre que σ peut être construit tel que $\sigma(n) = x$.

On peut donc imaginer un algorithme de reconnaissance des graphes triangulés qui commence par choisir le dernier sommet de l'ordre d'élimination simplicial. Il faut juste trouver une règle qui autorise à poursuivre cette stratégie à chaque étape sur le sous-graphe induit par les sommets non numérotés. C'est ce que fait Lex-BFS.

Les sommets sont numérotés dans l'ordre inverse de leur visite. Chaque sommet x possède une marque, $marque(x)$, qui correspond à la liste ordonnée de ses voisins numérotés. A chaque fois qu'un nouveau sommet x est numéroté, son numéro est ajouté à la fin de $marque(y)$ pour tout voisin non numroté y de x. Le nouveau sommet choisi est toujours un sommet ayant une marque maximum dans l'ordre lexicographique.

L'algorithme: Lex-BFS[RTL76]
Données: Un graphe $G = (X, E)$
Résultat: Un ordre σ des sommets de G
(1) Pour chaque sommet $x \in X$ faire:
$marque(x) \longleftarrow \emptyset$
(2) Pour $i = n$ à 1 faire:
(2-1) Choisir un sommet x non numéroté de marque maximum dans l'ordre lexicographique
$\sigma(i) \longleftarrow x$
(2-2) Pour chaque voisin non numéroté y de x faire:
$marque(y) \longleftarrow marque(y) \cup \{i\}$

Théorème 0.14: [RTL76] Lex-BFS calcule un ordre d'élimination simplicial si et seulement si le graphe G est triangulé. Sa complexité est $O(n + m)$, où $n = |X|$ et $m = |E|$.

4 Algorithme 1 .

Principe de l'algorithme:
L'algorithme comporte deux parties:
- La première c'est Lex-BFS dont le rôle est de donner un ordre d'élimination simplicial $\sigma = [x_1, ..., x_i, ..., x_n]$, où $x_i = \sigma(i)$ est un sommet simplicial pour le graphe $G_i = G[x_i, ..., x_n]$ pour tout $1 \leq i \leq n$.
- Dans la deuxième partie, l'idée principale est d'afficher tous les arcs de la forme (x_i, y), où x_i est simplicial dans le graphe G_i avec $y \in adj(x_i)$ pour tout $1 \leq i \leq n-1$.

L'algorithme:
Données: Un graphe $G = (X, E)$ triangulé non orienté
Résultat: Une B_1-orientation U de G
(0) Initialiser $U = \emptyset$
(1) Pour chaque sommet $x \in X$ faire:
$marque(x) \longleftarrow \emptyset$
(2) Pour $i = n$ à 1 faire:
(2-1) Choisir un sommet x non numéroté de marque maximum dans l'ordre lexicographique;
$\sigma(i) \longleftarrow x$
(2-2) Pour chaque voisin non numéroté y de x faire:
$marque(y) \longleftarrow marque(y) \cup \{i\}$
(3) Pour $i = 1$ à $n-1$ faire:
Si $U_i = \{(\sigma(j), \sigma(i)),$ où$\sigma(j) \in adj(\sigma(i)),$ et $j > i\} \neq \emptyset$ alors $U = U \cup U_i$

Preuve de l'algorithme:
On suppose que l'orientation trouvée n'est pas une B_1-orientation, donc $\exists (x, y) \in U$ et $(z, y) \in U$ et $xz \notin E$.
Les arêtes xy, zy se sont orientées à l'étape où y était un sommet simplicial, par conséquent, chaque couple dans le voisinage de y forme une arête, y compris xz.

5 Algorithme 2 .

Principe de l'algorithme:
Comme l'algorithme 1, cet algorithme est aussi constitué de deux parties:
- La première partie c'est l'algorithme de Johnson et al., dont le rôle est de générer toutes les cliques maximales dans un graphe . Il s'agit d'un algorithme qui utilise l'approche de parcours lexicographique, basé sur l'ordre total des sommets. L'idée principale est de pouvoir générer efficacement une clique maximale à partir de son prédécesseur dans l'ordre lexicographique, sachant que la première clique est donnée.
- La deuxième partie est un algorithme qui utilise l'approche de parcours lexicographique, basé sur l'ordre total des cliques maximales générées par l'algorithme de Johnson et al. Son principe est d'extraire les sommets simpliciaux à partir des cliques maximales (en utilisant la définition 0.9).

L'algorithme:

L'algorithme 2-a: (Johnson et al.)
Données: Soit $G = (X, E)$ un graphe triangulé non orienté
Résultat: Générer toutes les cliques maximales dans G
(1) $C_1 = \{1\}$
(2) Pour $i = 2$ à n faire:
Si $adj(i) \cap C_1 = C_1$ alors $C_1 = C_1 \cup \{i\}$
(3) Insérer C_1 dans Q
(4) $C = minQ$; afficher C
(5) Pour chaque sommet $i \in C$ faire:
(5-1) Pour tout sommet $j \in G$ non adjacent à i tq $i < j$ faire:
(5-1-1) Si: $(C \cap \{1, ..., j\} \cap adj(j)) \cup \{j\}$ est une clique maximale du sous-graphe induit par les sommets $\{1, 2, ..., j\}$ alors:
- Soit $\acute{C}$ la première clique maximale suivant l'orde lexicographique de G contenant $(C \cap \{1, ..., j\} \cap adj(j)) \cup \{j\}$
- Si $\acute{C} \notin Q$ alors Insérer $\acute{C}$ dans Q
(6) Faire: $Q = Q - \{C\}$; afficher Q
(7) (7-1) Si $Q = \emptyset$; STOP toutes les cliques sont générées
(7-2) Sinon; aller en (4)

L'algorithme 2-b:
Données: Soit $G = (X, E)$ un graphe triangulé non orienté, et $C_1, C_2, ..., C_k$ ses cliques maximales
Résultat: Une B_1-orientation de G
(1) Initialiser $i = 1$; $G = G_1 = (X_1, E_1) =$, et $j = 1$; $C_j = C_1$
(2) $P_i = C_j/(\cup_{h \neq j} C_h)$; afficher P_i
(3) (3-1) Si $P_i = \emptyset$ faire: $j \longleftarrow j + 1$; aller en (2)
(3-2) Sinon, faire:
(4) (4-1)prendre un sommet x de P_i
(4) (4-2) Orienter toutes les arêtes incidentes à x vers x
(5) Afficher $P_i - \{x\}$
(6) (6-1) Si $P_i - \{x\} \neq \emptyset$ faire:
$P_{i+1} = P_i - \{x\}$, $G_{i+1} = G_i - \{x\}$, $i \longleftarrow i + 1$; aller en (3-2)
(6-2) Sinon, faire:
(7)$G_{i+1} = G_i - x$
(8) (8-1) Si $E_{i+1} = \emptyset$; STOP; G est B_1-orientable
(8-2) Sinon, faire:$i \longleftarrow i + 1$; aller en (2)

Preuve de l'algorithme: Pour un sommet quelconque x de G, s'il y avait deux arcs entrant dans x, alors, leur orientation est obtenue à l'étape où on a orienté toutes les arêtes incidentes à x vers x, donc x est simplicial dont le voisinage qui contient les extrémités initiales de ces deux arcs forme une clique, par conséquent ces deux extrmités sont liées entre elles par une arête dans G, donc G ne peut être que B_1-orientable.

Complexité de l'algorithme: La complexité de cet algorithme est celle de l'algorithme de Johnson et al.; soit $O(n^3)$ par clique calculée [8], où $n = |X|$ et $m = |E|$.

remarque: On note que l'ensemble des cliques maximales dans le graphe G_{i+1}, obtenu partir de $G_i - \{x\}$, ne change pas par rapport à celui de G_i.
En effet, soit P_i l'ensemble des sommets simpliciaux dans G_i, d'après l'algorithme 2-b; P_i est dans une et une seule clique maximale de G_i (soit C_i), la suppression d'un sommet quelconque x de G_i, ne change rien des cliques maximales C_h, pour tout $h \neq j$ dans G_{i+1}, sinon x, n'est pas simplicial, par conséquent, C_i deviendra dans G_{i+1}, la nouvelle clique maximale $C_i - \{x\}$.

References

1. R. Bannari, Etude d'une heuristique pour générer une clique de cardinal maximum d'un graphe, Mémoire de DEA en recherche opérationnelle et productique, Université Blaise Pascal-Clermont II, 2002.
2. C.Berge, Les problèmes de colorations en theorie des graphes, Publ, Inst. Statist, Univ. Paris 9, 1960.
3. C.Berge, Gaphes et Hypergraphes, Dunod, 1970.
4. G.A. Dirac, On rigid circuit graphes, Abh. Math. Sem, Univ. Hambourg, 25, 1961.
5. D.R. Fulkerson and O.A. Gross, Incidence matrices and Interval graphs, Pacific Journal Math, 15: 835-855, 1965.
6. M. Goldberg and T. Spencer, A new parallel algorithm for the maximal independet set problem, Proc 28^{th} IEEEFOCS, page 161-165, 1987.
7. M.C.Golumbic, Algorithms Graph Theory and Perfect Graphs, Academic Press, New York University, 1980.
8. D.Johnson and Yannakakis, On generating all maximal independent sets - information processing, Letters 27, pages 119-123,1988.
9. M. Luby, A simple parallel algorithm for the maximal independet set problem, Proc, 17^{th} ACMSTOC, 1985.
10. K. Makino and T. Uno, New algorithms for enumerating all maximal cliques, pages 560-8531, 2004.
11. Christophe Paul, Parcours en Largeur Lexicographique: Un algorithme de partitionnement, application aux graphes et généralisation, Thèse de DOCTORAT en informatique, Université Montpellier II, 1998.
12. Donald J.Rose, Triangulated graphs and elimination process, J. Math. Anal, 32: 597-609, 1970.
13. Donald J.Rose, R.Endre Tarjan, and George S.Leucker, Algorithmic aspects of vertex elimination of graphs, SIAM Journal of computing, 5(2): 266-283, June 1976.
14. B.Sadi, B_1-orientable graphs, Revue sciences et technologie, Université de constantine.
15. J.P.Spinrad, Liste of open problems, manuscript, 1997.
16. Tsukiyama, M.Ide, H.Ariyoshi and I.Shirakawa, A new algorithme for generating all the independent sets, SIAM J, 6: 505-517, 1977.
17. J.R. Walter, Representation of rigid cicle graphs, Ph.D. thesis, Wayne State Univ.

This article was processed using the LaTeX macro package with LLNCS style

Calcul d'invariant dans les ensembles partiellement ordonnés

[1]Djamel TALEM et Bachir SADI

[1]Département des Mathématiques, facultés des Sciences. Université Mouloud Mammeri. Tizi Ouzou.

Abstract. Soit $P = (P, \leq_p)$ un ensemble partiellement ordonné et $D(P)$ est l'ordre strict défini sur l'ensemble des antichaînes maximales de P par: A, B deux *antichaînes* maximales de P, $A < B$ si et seulement si,$\forall a \in A, \exists b \in B$ tel que $a < b$.On note par $cdev(P)$, et on lit$\ll$ $chaîne - diviation \gg$ de P, $cdev(P)$ est un paramètre qui est égal au plus petit entier naturel i tel que $D^i(P)$ est un ordre total, oú $D^i(P) = D(D(...D(P)))$, i fois. Dans[1], T.Y.Kong et Ribemboim ont montré que $cdev(P) \leq 2d(P) - 1$, oú d(P) est le nombre d'éléments de la plus longue chaîne de P. Dans [2], Bachir Sadi a introduit un paramètre $I(P)$ appelé l'inclinaison de P et a montré que: $cdev(P) \leq 2d(P) - 2I(P) + 1$.
Dans cet article nous donnons une condition nécessaire et suffisante pour qu'un ordre P atteigne une borne supérieur $2d(P) - 1$, c'est à dire $cdev(P) = 2d(P) - 1$, et nous déterminons la valeur exacte du paramètre $cdev(P)$ pour un ordre d'intervalles et d'inclinaison 1.

Mots clés: Ordre, chaîne, antichaîne.

1 Introduction et definitions.

Un ensemble ordonné est un couple$(X, \leq_p)$ oú X est un ensemble non vide et $\leq_p$ est une relation d'ordre définie sur X. On note $P = (X, \leq_p)$ ou P l'ensemble ordonné appelé brièvement ordre P. Une chaîne de P est un sous-ensemble d'éléments de X deux à deux comparables. Elle a pour longueur le nombre de ses éléments. On note $d(P)$ la longueur de la plus longue chaîne de P. Une antichaîne est un sous-ensemble d'éléments de X deux à deux incomparables. Une chaîne (resp. antichaîne) de P est dite maximale si elle n'est pas strictement incluse dans une autre chaîne (resp. antichaîne). Soit P un ensemble partiellement ordonné, on note $D(P)$ l'ordre strict défini sur les antichaînes maximales de P par: A, B deux antichaînes maximales de $P, A < B$ si seulement si $\forall a \in A, \exists b \in B$ tel que, $a < b$. D'une manière générale $D^i(P)$ est l'ordre défini sur l'ensemble des antichaînes maximales de $D^{i-1}(P)$. On note par $cdev(P)$, et on lit$\ll$ $cha^ine - diviation \gg$ de P, $cdev(P)$ est un paramètre qui est égal au plus petit entier naturel i tel que $D^i(P)$ est un ordre total, où $D^i(P) = D(D(...D(P)))$, i fois. On pose $Min(P) = \{x \in P / Pred(x) = \emptyset\}$, où $Pred(x)$ est l'ensemble des prédécesseurs de x, excepté x. Soit l'application $rang$, notée rg, définie sur P à valeurs dans

N par: $x \in P, rg(x) = \begin{cases} 0 \text{ si } x \in Min(P). \\ max\{rg(y)/y \in Pred(x)\} + 1, sinon\ . \end{cases}$

N_i=$\{x \in P/rg(x) = i\}$ est le $(i+1)$-ieme niveau de P. N_i est dit complet si,N_i est une antichaîne maximale de P, c-à-d, $N_i \in D(P)$. A un niveau N_i on associe M_i qui est un sous-ensemble de $D(P)$ tel que, $M_i = \{A \in D(P)/A \cap N_i \neq \emptyset, et A \cap N_{i+1} \neq \emptyset\}$. Pour tout élément $x \in N_i correspondent deux ensembles, W^+(x) = \{y \in N_{i+1}/x <_p y\}$ et $W^-(x) = \{y \in N_{i-1}/y <_p x\}$. Pour une antichaîne maximale A de P, on appelle inclinaison de A, la quantité $I(A) = \max_{x,y \in A}\{|rg(y) - rg(x)|\}$. L'inclinaison de P est $I(P) = \max_{A \in D(P)}\{I(A)\}$.
Un couple (x, y) de P est une couverture, et on note $x <_\bullet y$ si, $x <_p y$ et $\forall z \in P$ tel que, $x \leq_P z \leq_P y$, alors, $z = x$ ou $z = y$. Un couple (x, y) de P est un saut si, x et y sont incomparables. Un ordre P est d'intervalles [3] s'il ne contient pas la somme disjointe $C_2 + C_2$ de deux chaînes de longueur 2, autrement dit, s'il n'existe pas deux couvertures (x, y), (t, z) telles que (x, t), (y, z) soient des sauts. On note par I_1 l'ensemble des ordres P d'inclinaison 1. Un ordre P est integrable dans I_1, s'il existe un ordre $Q \in I_1$ tel que, $D(Q) = P$.

Nous verrons par la suite que la classe des ordres d'intervalles possède de bonnes propriétés, ce qui laisse son etude dans ce contexte moins compliquée, et il est facile de voir qu'au cours du calcul de $D^i(P)$ pour un ordre quelconque, il existe un entier naturel $i_0 \in \{1, 2, ..., cdev(P)\}$ tel que, $\forall i \geq i_0, D^i(P)$ est un ordre d'intervalle, donc la connaissance de $cdev(P)$ dans cette classe rend son étude moins difficile dans un ordre quelconque.

2 Les ordres intégrables.

Proposition 2.1. Si $P \in I_1, D(P) \in I_1$, alors:
1) $D^2(P)$ ne contient pas $C_2 + C_2$.
2) Si $A_1^1 < A_2^1 < A_3^1$ dans $D^2(P)$, alors: $\forall A_3 \in A_3^1, \forall A_1 \in A_1^1$, on a, $A_1 < A_3$.
3)$I(D^2(P)) = 1$.
Preuve.
1) Soient $A_1^1, A_2^1, B_1^1, B_2^1$ dans $D^2(P)$ tels que, $(A_1^1, B_1^1), (B_2^1, A_2^1)$ sont des couvertures et (A_1^1, A_2^1) est un saut dans $D^2(P)$. Il s'agit de montrer que $B_1^1 < B_2^1$, c-à-d, (B_1^1, B_2^1) n'est pas un saut dans $D^2(P)$.
$A_1^1, A_2^1, B_1^1, B_1^1$ sont des antichaînes maximales de $D(P)$, on a deux cas:

Cas 1: $I(A_1^1) = 0$.
Ainsi A_1^1 est un niveau de $D(P)$, soit $A_1^1 = N_i^1$, (A_1^1, A_2^1) est un saut, donc $A_1^1 \cap A_2^1 \neq \emptyset$ et A_2^1 rencontre N_{i-1}^1 ou N_{i+1}^1, on suppose sans perte de généralité que A_2^1 rencontre N_{i-1}^1 c-á-d, $A_2^1 \cap N_{i-1}^1 \neq \emptyset$. On a, $B_2^1 < A_2^1$, ainsi $\forall B_2 \in B_2^1, rg(B_2) < i$ ...(1).
$A_1^1 < B_1^1$, donc $\exists B_1 \in B_1^1$ tel que, $rg(B_1) \geq i + 1$ ...(2).
De (1) et (2), on a, $rg(B_1) - rg(B_2) > 1$. B_1, B_2 sont des éléments de $D(P)$, et

$I(D(P)) = 1$, donc $B_2 < B_1$, et $B_2^1 < B_1^1$, et par conséquent $D^2(P)$ ne contient pas $C_2 + C_2$.
Un raisonnement identique si $I(A_2^1) = 0$.

Cas 2: $I(A_1^1) = I(A_2^1) = 1$.
Ainsi, il exist deux niveaux N_i^1, N_{i+1}^1 de $D(P)$ tels que,
$A_1^1 \cap N_i^1 \neq \emptyset$ et $A_1^1 \cap N_{i+1}^1 \neq \emptyset$
$A_2^1 \cap N_i^1 \neq \emptyset$ et $A_2^1 \cap N_{i+1}^1 \neq \emptyset$.

Soit $B_2 \in B_2^1$, on a $B_2^1 < A_2^1$, donc $rg(B_2) \leq i$ ...(1).

$A_1^1 < B_1^1$, donc $B_1^1 \cap N_{i+2}^1 \neq \emptyset$ ou B_1^1 est strictement au dessus de N_{i+2}^1, donc $\exists B_1 \in B_2^1$ tel que, $rg(B_1) \geq i + 2$ ...(2).
De (1) et (2), il vient que, $rg(B_1) - rg(B_2) \geq 2$. Et comme $I(D(P)) = 1$, on a, $B_2 < B_1$, et $B_2^1 < B_1^1$, d'où $D^2(P)$ ne contient pas $C_2 + C_2$.

2) Soient, $A_1 \in A_1^1, A_2 \in A_3^1$, on a,
$A_1^1 < A_2^1$, donc $\exists B_1 \in A_2^1$ tel que, $A_1 < B_1$ ...(1).
$A_2^1 < A_3^1$, donc $\exists B_2 \in A_2^1$ tel que, $B_2 < A_3$ ...(2).
A_1, A_3, B_1, B_2 sont des antichaînes maximales de P. Si B_1 ou B_2 sont des niveaux de P, alors, il est facile de voir que $A_1 < A_3$, on suppose qu'il existe N_i, N_{i+1} deux niveaux de P verifiant:
$B_1 \cap N_i \neq \emptyset, B_1 \cap N_{i+1} \neq \emptyset$ ...(3).
$B_2 \cap N_i \neq \emptyset, B_2 \cap N_{i+1} \neq \emptyset$ ...(4).
Soit $a_1 \in A_1$, d'après (1) et (3), $rg(a_1) \leq i$ ...(5).
Et d'après (2) et (4), $A_3 \cap N_{i+2} \neq \emptyset$ ou A_3 est strictement au dessus de N_{i+2}, donc $\exists a_3 \in A_3$ tel que, $rg(a_3) \geq i + 2$ ...(6).
De (5) et (6), on a, $rg(a_3) - rg(a_1) \geq 2$, et comme $I(P) = 1$, alors, $a_1 < a_3$, d'oú $A_1 < A_3$.

3) Si, $I(D(P)) = 2$, alors, il existe $A_0^1, A_1^1, A_2^1, A_3^1$ dans $D^2(P)$, tel que, $A_1^1 < A_2^1 < A_3^1$, et $(A_0^1, A_1^1), (A_0^1, A_2^1), (A_0^1, A_3^1)$ soient des sauts, ainsi A_0^1 rencontre A_1^1, A_2^1, A_3^1, c-à-d, il existe $A_1 \in A_1^1, A_3 \in A_3^1$ tel que, (A_1, A_3) est un saut. Or, d'apres 2) c'est impossible.

2.1-Proprietés.
Si P est intégrable dans I_1, alors:

1. $D(P)$ ne contient pas $C_2 + C_2$.
2. Si $A < B < C$ dans $D(P)$, alors, $\forall a \in A, \forall c \in C, a < c$.
3. $I(D(P)) = 1$.
4. (A, B) est un saut dans $D(P) \Longleftrightarrow A \cap B \neq \emptyset$.

3 Les ordres d'intervalles.

Proposition 3.1 Soit P un ordre d'intervalles, $N_0, N_1, ..., N_k$ ses niveaux, alors:

1. $\forall i = 1, 2, ..., k-1, \exists x \in N_i$, $W^+(x) = N_{i+1}$
2. Si N_i est complet, $i \geq 2$, alors, $\exists x \in N_i$, $W^-(x) = N_{i-1}$
3. $\forall x, y \in N_i$, $W^+(x) \subseteq W^+(y)$, ou $W^+(y) \subseteq W^+(x)$.
4. $\forall x, y \in N_i$, $W^-(x) \subseteq W^-(y)$, ou $W^-(y) \subseteq W^-(x)$.
5. $\exists x \in N_i, y \in N_{i+1}$, et $\forall A \in M_i$, on a, $x, y \in A$

Preuve:
1) Soit $x \in N_i$ tel que, $|W^+(x)|$ est maximum sur N_i. Si $W^+(x) \neq N_{i+1}$, alors, $\exists y \in N_{i+1}/W^+(x)$, ainsi (x, y) est un saut. Soit $a_0 \in N_i$ tel que, $y \in W^+(a_0)$. $|W^+(x)|$ est maximum, donc $W^+(x)/W^+(a_0)$ contient au moins un élément a_1. Il s'ensuit que, $(x, y), (a_0, a_1)$ sont des sauts, et $(x, a_1), (a_0, y)$ sont des couvertures, donc $(x, a_1), (a_0, y)$ est un $C_2 + C_2$. Absurde car P est un ordre d'intervalles.

2) Un raisonnement analogue à 1).

3) Si $x_1 \in W^+(x)/W^+(y)$, et $y_1 \in W^+(y)/W^+(x)$, il vient que, $(x, y_1), et(y, x_1)$ sont des sauts, et donc $(x, x_1), (y, y_1)$ est un C_2+C_2. Absurde car P est d'intervalles.

4) Un raisonnement analogue à 3).

5) Soient $x \in N_i$ tel que, $|W^+(x)|$ est minimum sur N_i, $y \in N_{i+1}$ tel que, $|W^-(y)|$ est minimum sur N_{i+1}, et $A \in M_i$. Soit $a \in A$, alors,
a n'appartient pas à $W^+(x)$, car sinon, on aurait, $a \in W^+(z)$ pour tout $z \in N_i$, et donc A n'est pas un élément de M_i.
a n'appartient pas à $W^-(y)$, car sinon, on aurait, $a \in W^-(z)$ pour tout $z \in N_{i+1}$, et donc A n'est pas un élément de M_i. D'oú $xety$ sont dans A.

Soit P un ordre d'intervalles, $N_0, N_1, ..., N_k$ ses niveaux. Si les niveaux de P sont tous complets, alors, dans $D(P)$ on a autant de niveaux que dans P, et pour tout, $i = 1, 2, ..., k$. $rg(N_i) = i$, et pour tout $A \in M_i$, $rg(A) = i$ dans $D(P)$, et au moins le dernier niveau de $D(P)$ n'est pas complet.

Soit P un ensemble ordonné de niveaux $N_0, N_1, ..., N_k$. Pour $i = 1, 2, ..., k$, on note $Min^+(N_i) = \{x \in N_i/|W^+(x)| \text{ est minimum}\}$.

On associe à P l'ensemble $U^0(P)$ défini par l'algorithme suivant:

0)$U(P) = \emptyset$.
1) Soient $x_0 \in Min^+(N_0), x_1 \in N_1/W^+(x_0)$ tels que, $|W^+(x_1)|$ est minimum sur $N_1/W^+(x_0)$.
2)
i- Si $W^+(x_1) = N_2$, soient $y_1 \in Min^+(N_1)$, et $x_2 \in N_2/W^+(y_1)$ tels que, $|W^+(x_2)|$ est minimum sur $N_2/W^+(y_1)$, aller á 2).

ii- Si $W^+(x_1) \neq N_2$, on pose $U(P) = U(P) \cup \{x_1\}$, et soit $x_2 \in N_2/W^+(x_1)$ tels que, $|W^+(x_2)|$ est minimum sur $N_2/W^+(x_1)$, aller á 2).

On répéte la procédure 2) jusqu'à ce qu'on parcourt tous les niveaux de P.

A la fin, on a:
a) Si N_k est complet, on pose $U^0(P) = U(P)$.
b)Si N_k n'est pas complet, et la procédure 2) s'arréte au niveau N_{k-2}, c-á-d, s'il existe $x \in U(P) \cap N_{k-2}$ tel que, $y \in N_{k-1}/W^+(x)$, $W^+(y) = N_k$, on pose $U^0(P) = U(P) \cup \{x_k\}$ avec $x_k \in N_k$.
c) Si N_k n'est pas complet, et la procédure 2) s'arréte au niveau N_{k-1} et il existe $y_{k-2} \in N_{k-2}, y_{k-1} \in Min^+(N_{k-1})$,et $y_k \in N_k$ tel que, $(y_{k-2}, y_{k-1}), (y_{k-1}, y_k)$ sont des sauts, on pose $U^0(P) = U(P) \cup \{y_k.\}$.

Remarque: A tout élément $x_i \in N_i \cap U^0(P), i \neq k$, correspondent deux éléments $x_{i-1} \in N_{i-1}, x_{i+1} \in N_{i+1}$ tels que, $(x_{i-1}, x_i), (x_i, x_{i+1})$ sont des sauts, autrement dit: $\exists A_{i-1} \in M_{i-1}, A_i \in M_i$ tels que, $x_i \in A_{i-1} \cap A_i \cap N_i$.

Proposition 3.2:
Si P un ordre d'intervalle et $N_0, N_1, ..., N_k$ sont ses niveaux, alors:
Il exist un ordre d'intervalle Q ayant au moins les (k) premiers niveaux complets, et vérifiant: $d(Q) = d(P), U^0(Q) = U^0(P), etcdev(Q) = cdev(P)$.

Preuve:
Soient $N_i, i \neq k$ un niveau non complet de P, et $x_i \in Min^+(N_i)$, il suffit de relier les éléments de $Min^+(N_{i-1})$ aux éléments de $N_i/\{x_i\}$.

Proposition 3.3:
Soit P un ordre d'intervalle dans I_1, $N_0, N_1, ..., N_k$ ses niveaux qui sont tous complets, alors: $|U_0(P)| = |U^0(D(P))| - 1$.

Preuve:
Soit $x_i \in U(P)$. D'après la remarque précédente,$\exists A_{i-1} \in M_{i-1}, A_i \in M_i$ tels que, $x_i \in A_{i-1} \cap A_i \cap N_i$, et comme les niveaux de P sont tous complets,alors, dans $D(P)$ on a, A_{i-1}, A_i, N_{i+1} sont respectivement dans les niveaux $N^1_{i-1}, N^1_i, N^1_{i+1}$ de $D(P)$, et $(A_{i-1}, A_i), (A_i, N_{i+1})$ sont des sauts. Ainsi la procédure 2) appliquée à $D(P)$ passera forcément par le point A_i, donc, $A_i \in U(D(P))$, d'où, $|U(P)| = |U(D(P))|$. D'un autre coté, $N^1_k = \{N_k\}$ est le dernier niveau de $D(P)$ qui n'est pas complet, donc d'après la définition de U^0 on a, $U^0(D(P)) = U(D(P)) \cup \{N_k\}$, et finalement:$|U_0(P)|=|U(P)| = |U(D(P))| = |U^0(D(P))| - 1$.

Proposition 3.4:
Soit P un ordre d'intervalles dans I_1 et $N_0, N_1, ..., N_k$ sont ses niveaux. Si N_k est l'unique niveau de P incomplet, alors: $|U_0(P)| = |U^0(D(P))| + 1$.

Preuve:
On a deux cas:
Cas 1:$U^0(P) = U(P)$.
Alors, $\exists x_{k-1} \in N_{k-1} \cap U(P)$, et $\forall A_{k-2} \in M_{k-2}$ on a, $A_{k-2} \cap Min^+(N_{k-1}) \doteq \emptyset$. Ainsi, $\forall A_{k-2} \in M_{k-2}, \exists A_{k-1} \in M_{k-1}$ tel que, $A_{k-2} < A_{k-1}$, ce qui veut dire que, N^1_{k-1} qui est le dernier niveau de $D(P)$ est complet, donc $N^1_{k-1} \cap U^0(D(P)) = \emptyset$, et comme l'antichaîne maximale contenant le saut (x_{k-1}, x_k) appartient à N^1_{k-1}, alors, à x_k ne correspond aucun élément de $U^0(D(P))$. D'oú $|U_0(P)| = |U^0(D(P))| + 1$.

Cas 2: $U^0(P) = U(P) \cup \{x_k\}$, avec $x_k \in N_k$. On a, deux cas:
1)$\exists x_{k-1} \in N_{k-1} \cap U(P)$, et $A_{k-2} \in M_{k-2}$ tel que, $A_{k-2} \cap Min^+(N_{k-1}) \neq \emptyset$, il vient que,$N^1_{k-1}$ qui est le dernier niveau de $D(P)$ n'est pas complet, et à x_k ne correspond aucun élément de $U^0(D(P))$, car le niveau qui le contient disparait dans le passage à $D(P)$, d'où $|U_0(P)| = |U^0(D(P))| + 1$.
2) $\exists x_{k-2} \in U^0(P) \cap N_{k-2}$ et $\forall x_{k-1} \in N_{k-1}/W^+(x_{k-2})$, on a $W^+(x_{k-1}) = N_k$, ici aussi à x_k ne correspond aucun élément de $U^0(D(P))$, car le niveau qui le contient disparait dans le passage à $D(P)$, d'où $|U_0(P)| = |U^0(D(P))| + 1$.

4 Ordre contenant un sous ordre faible défini par deux niveaux consécutifs.

Un ordre est dit faible si les seules antichaîne maximales de cet ordre sont ses niveaux.

Proposition 4.1:
Soit P un ordre et $N_0, N_1, ..., N_k$ sont ses niveaux. On suppose qu'il existe i tel que, (N_i, N_{i+1}) est un ordre faible $(M_i = \emptyset)$. Si P_1 est le sous-ordre induit par l'union des niveaux $N_0, N_1, ..., N_i$ et P_2 est le sous-ordre induit par l'union des niveaux $N_{i+1}, N_{i+2}, ..., N_k$,alors:
1) Pour $j = 1, 2, ..., cdev(P)$, le dernier niveau de $D^j(P_1)$ et le premier niveau de $D^j(P_2)$ forment un sous-ordre faible.
2) $Cdev(P) = \max(cdev(P_1), cdev(P_2))$.

Preuve:
1) Soient N^{j-1}_i le dernier niveau de $D^{j-1}(P_1)$, N^{j-1}_{t+1} le premier nieau de $D^{j-1}(P_2)$, et supposons que, le couple $(N^{j-1}_t, N^{j-1}_{t+1})$ induit un ordre faible. Et Soient N^j_s, N^j_{s+1} deux niveaux consécutifs de $D^j(P)$ contenant respectivement N^{j-1}_t, N^{j-1}_{t+1}, et montrons que,(N^j_s, N^j_{s+1}) est un sous-ordre faible dans $D^j(P)$.
On a, N^j_s est constitué par N^{j-1}_t et éventuellement par les antichaînes maximales de $D^{j-1}(P)$ qui rencontrent simultanément N^{j-1}_{t-1}, etN^{j-1}_t, et N^j_{s+1} est constitué par N^{j-1}_{t+1} et par les antichaînes maximales de $D^{j-1}(P)$ qui rencontrent simultanément N^{j-1}_{t+1}, etN^{j-1}_{t+2}, donc tout élément de N^j_s est inferieur à tous les éléments de N^j_{s+1}, ainsi les seules antichaînes maximales de (N^j_s, N^j_{s+1}) sont N^j_s,

etN_{s+1}^j, ce qui veut dire que, (N_s^j, N_{s+1}^j) est un sous-ordre faible de $D^j(P)$.

2) On pose $cdev(P_1) = i_1$, et $cdev(P_2) = i_2$, il vient que,
Si $i_1 < i_2$, alors, $D^{i_2}(P)$ est une chaîne .
Si $i_2 < i_1$, alors, $D^{i_1}(P)$ est une chaîne .
D'oú, $cdev(P) = \max(i_1, i_2)$.

5 Ordre d'intervalles vérifiant, $cdev(P) = 2d(P) - 1$

.

Proposition 5.1. P un ordre d'intervalles. Une condition nécessaire et suffisante pour que, $Cdev(P) = 2d(P) - 1$, est que les trois conditions suivantes soient vérifiées:
1) $I(P) = 1$.
2) $M_i \neq \emptyset$, pour tout i.
3) $U^0(P) = \emptyset$.

Preuve:
La condition est necessaire.
Dans [2], Bachir SADI a prouv l'ingalit $cdev(P) \leq 2d(P) - 2I(P) + 1$ pour tout ordre non total.Si $I(P) > 2$, alors $cdev(P) < 2d(P) - 1$.
S'il existe i tel que, $M_i = \emptyset$, alors P contient un sous-ordre faible defini par le couple (N_{i-1}, N_i), et d'après la prop.4.1, $cdev(P) = \max(cdev(P_1), cdev(P_2)) < 2d(P) - 1$.
Si, $U^0(P) = \{x\}$, montrons par récurrence sur $d(P)$ qu' on a aussi $cdev(P) < 2d(P) - 1$.
Pour $d(P) = 2$ P est un ordre biparti non connexe, et d'après [2], $cdev(P) = 2 < 2d(P) - 1 = 3$.
On suppose que cela reste vrai pour tout ordre P vérifiant $d(P) \leq n$ et $|U^0(P)| = 1$. Soit P un ordre tel que, $d(P) = n + 1$ et $|U^0(P)| = 1$. On a deux cas:
Cas 1: Le dernier niveau de P n'est pas complet.
Grace à la proposition 3.2, on peut toujours supposer que les autres niveaux de P sont complets, ainsi, $d(D(P)) = d(P) - 1 = n$, et le dernier niveau de $D(P)$ est complet, d'après la prop. 3.4, $|U_0(D(P))| = |U^0(P)| - 1 = 0$.
Dans $D^2(P)$, on a, $d(D^2(P) = d(D(P)) = n$, et d'après la proposition 3.3, on a, $|U_0(D^2(P))| = |U^0(D(P))| + 1 = 1$, et l'hypothèse de recurence appliquée à $D^2(P)$ nous donne, $cdev(D^2(P)) < 2d(D^2(P)) - 1 = 2n - 1$.
D'un autre cote on a, $cdev(P) = 2 + cdev(D^2(P))$, d'oú, $cdev(P) < 2 + 2n - 1 = 2(n + 1) - 1 = 2d(P) - 1$.

Cas 2: le dernier niveau de P est complet.
D'après la prop. 3.3, on a,$|U_0(D(P))| = |U^0(P)| + 1 = 2$. $d(D(P)) = d(P) = n + 1$, et le dernier niveau de $D(P)$ n'est pas complet. Dans $D^2(P)$, on a, $d(D^2(P)) = d(D(P)) - 1 = n$, et d'après la proposition 3.4, $|U_0(D^2(P))| =$

$|U^0(D(P))| - 1 = 1$, et par l'hypothèse de récurrence appliquée à $D^2(P)$,on a, $cdev(D^2(P)) < 2d(D^2(P)) - 1 = 2n - 1$.
D'un autre coté on a, $cdev(P) = 2 + cdev(D^2(P))$, d'oú, $cdev(P) < 2 + 2n - 1 = 2(n+1) - 1 = 2d(P) - 1$.

La condition est suffisante.
On la montre par récurrence sur $d(P)$.
Pour $d(P) = 2$, P est un ordre biparti connexe et non faible, donc $cdev(P) = 3 = 2d(P) - 1$. ([2]).
On suppose que pour $d(P) \leq n$, $cdev(P) = 2d(P) - 1$.
Soit P un ordre tel que, $d(P) = n+1$, les niveaux de P sont tous complets, donc $d(D(P)) = d(P) = n+1$, et $|U^0(D(P)| = |U^0(P)| + 1 = 1$(proposition3.3), et le dernier niveau de $D(P)$ n'est pas complet.
Dans $D^2(P)$, on a, $d(D^2(P)) = d(D(P)) - 1 = n$, et $|U^0(D^2(P)| = |U^0(D(P))| - 1 = 0$, et par l'hypothèse de récurrence appliquée à $D^2(P)$,on a, $cdev(D^2(P)) = 2d(D^2(P)) - 1 = 2n - 1$.
D'un autre coté on a, $cdev(P) = 2 + cdev(D^2(P))$, d'oú, $cdev(P) = 2 + 2n - 1 = 2(n+1) - 1 = 2d(P) - 1$.

6 *cdev*(*P*) pour un ordre d'intervalle.

Lemme: Soitent P un ordre d'intervalle, $D(P), D^2(P), ..., D^{2t}(P)$ une suite d'ordres obtenue à partir de P. On suppose que, pour $i = 1, 2, ..., 2t$, on a:
1) Les niveaux de $D^i(P)$ sont tous complets, si i pair.
2) Les niveaux de $D^i(P)$ sont tous complets except le dernier niveau, si i impair.
Alors: pour i pair, on a, $|U^0(D^i(P)| = |U^0(P)|$, et $d(D^i(P)) = d(P) - \frac{i}{2}$.

Preuve:
On suppose que cela est vrai pour $2j < 2t$, c-à-d, $|U^0(D^{2j}(P)| = |U^0(P)|$, et $d(D^{2j}(P)) = d(P) - j$.
On a, les niveaux de $D^{2j}(P)$ sont complets, donc
$|U^0(D^{2j+1}(P)| = |U^0(D^{2j}(P))| + 1$ (prop.3.3), et $d(D^{2j+1}(P)) = d(D^{2j}(P))$, et le dernier niveau de $D^{2j+1}(P)$ n'est pas complet.
Dans $D^{2j+2}(P)$, on a,
$d(D^{2j+2}(P)) = d(D^{2j+1}(P)) - 1 = d(D^{2j}(P)) - 1$, et
$|U^0(D^{2j+2}(P)| = |U^0(D^{2j+1}(P))| - 1 = |U^0(D^{2j}(P)|$ (prop.3.4).
Et d'après hypothèse on a,
$d(D^{2j+2}(P)) = d(P) - j - 1 = d(P) - (j+1) = d(P) - \frac{2(j+1)}{2}$,
et $|U^0(D^{2j+2}(P)| = |U^0(P)|$.

Théorème:
Soit P un ordre d'intervalle dans I_1 ne contenant pas un ordre faible comme sous-ensemble induit par l'union des niveaux, alors: $cdev(P) = 2d(P) - |U^0(P)| - 1$.

Preuve :
On utilise la demonstration par récurrence sur $|U^0(P)|$.
Pour $|U^0(P)| = 0$, $cdev(P) = 2d(P) - 1$(prop.5.1).
On suppose que, pour $|U^0(P)| \leq n$, $cdev(P) = 2d(P) - |U^0(P)| - 1$.
Soit P tel que, $|U^0(P)| = n + 1$, on a deux cas :
Cas 1: le dernier niveau de P n'est pas complet.
On peut toujours supposer que les autre niveaux sont complets (prop.3.2), il vient que, $|U^0(D(P))| = |U^0(P)|-1 = n$, et $d(D(P)) = d(P)-1$, et d'après l'hypothèse de récurrence, $cdev(D(P)) = 2d(D(P))-|U^0(D(P)|-1 = 2(d(P)-1)-|U^0(P)|$.
Et comme $cdev(P) = cdev(D(P))+1$, alors,$cdev(P) = 2(d(P)-1)-|U^0(P)|+1 = 2d(P) - |U^0(P)| - 1$.

Cas2:
le dernier niveau de P est complet.
Soit $t = 2j + 2$ le premier entier naturel tel que, $D^{2j+2}(P)$ a son dernier niveau non complet.
D'après le lemme précédent $D^{2j}(P)$ vérifie:
$|U^0(D^{2j}(P)| = |U^0(P)|$
$d(D^{2j}(P)) = d(P) - j$.

Le dernier niveau de $D^{2j}(P)$ étant complet, donc,

$|U^0(D^{2j+1}(P))| = |U^0(D^{2j}(P))| + 1 = |U^0(P)| + 1 = n + 2$
$d(D^{2j+1}(P)) = d(D^{2j}(P)) = d(P) - j$.

Le dernier niveau de $D^{2j+1}(P)$ n'est pas complet, donc,

$|U^0(D^{2j+2}(P)| = |U^0(D^{2j+1}(P))| - 1 = |U^0(P)| = n + 1$
$d(D^{2j+2}(P)) = d(D^{2j}(P)) - 1 = d(P) - j - 1$.
Le dernier niveau de $D^{2j+2}(P)$ n'est pas complet, alors,

$|U^0(D^{2j+3}(P)| = |U^0(D^{2j+2}(P))| - 1 = |U^0(P)| - 1 = n$
$d(D^{2j+3}(P)) = d(D^{2j+2}(P)) - 1 = d(P) - j - 2$.
Et par hypothèse de récurrence, on a,
$cdev(D^{2j+3}(P)) = 2d(D^{2j+3}(P))-|U^0(D^{2j+3}(P)|-1 = 2(d(P)-j-2)-n-1 = 2d(P) - 2j - n - 5$.
Et comme, $cdev(P) = 2j + 3 + cdev(D^{2j+3}(P)$, alors,
$cdev(P) = 2j + 3 + 2d(P) - 2 - n - 5 = 2d(P) - n - 2 = 2d(P) - (n + 1) - 1$,
d'où $cdev(P) = 2d(P) - |U^0(P)| - 1$.

Bibliographie:

[1] T.Y.Kong and P.Ribemboim, Channing of partially ordred sets, C.R.Acad.Sci. Paris, t. 319, Série I,p.533-537,1994.

[2] Bachir Sadi, Suite d'ensembles partiellement ordonnés, ARIMA, vol.4, 2006.

[3] P.C. Fishburn. Intransitive indifference intervals. J. Math. Psych, 7:144-149 (1970).

Pareto multiobjectifs pour la régulation par Bi-colonie d'un déplacement multimodal en mode perturbé

Aloui Abdelouhab, Tahraoui Mohamed Amine

Département d'Informatique,
Université de Béjaia, Algérie
{aaloui_abdel,tahraoui_info}@yahoo.fr

Résumé L'objectif de ce travail est la réalisation d'un système de régulation d'un réseau de transport multimodal en mode perturbé. Ceci en utilisant la métaheuristique d'Optimisation par Colonie de Fourmis noté(OCF) avec la particularité d'utilisation de deux colonies(C1 et C2) au lieu d'une habituellement. Le but est de construire deux populations de solutions. La première population, notée P1, construite par C1 et la deuxième P2, construite par C2. L'objectif principal de cette méthode est de générer une variété de solutions Pareto Optimales diversifiées dans l'espace de recherche, telle que la construction de P2 se fait en évitant les solutions de P1, de sorte qu'une solution de secours ait une probabilité élevée de se trouver dans P2, si jamais une perturbation vient de surgir et que les solutions de P1 deviennent obsolètes. L'implémentation de cette approche et les résultats des tests illustrent sa validité.

Key words: Métaheuristique, colonie de fourmis, optimisation combinatoire, transport multimodal, plus court chemin, Front Pareto

1 Introduction

Les déplacement de tous genre font partie de la dynamique de la vie de tous les jours, par conséquent, résoudre les problèmes liés aux moyens de transport est une nécessité(planification, exploitation, régulation). Dans ce travail, il sera abordé le problème de la régulation dans un réseau de transport multimodal, c'est à dire, proposer un Système d'Aide à la Régulation (SAR) afin d'assurer une bonne qualité de service, même en cas de perturbation.
Le transport multimodal bien qu'il présente des avantages, fait naître de nouveaux défis, de nouvelles contraintes, auxquels il faut répondre, à savoir, comment planifier, comment optimiser, comment réguler son réseau de transport et quoi faire en cas de perturbation, etc.
A l'origine évaluer un réseau de transport revient, de prime à bord à déterminer le meilleur itinéraire à emprunter pour réaliser un déplacement d'un point source à un point destination, selon un certain nombre de considérations. Ce qui revient aussi à résoudre une variante particulière du problème de Plus Court Chemin dans un réseau Multimodal(PCCM).

Le problème de PCCM a été traité dans un travail [2] dans lequel nous avons montré la performance de la métaheuristique OCF, or il ne suffit pas d'offrir un service dans le cas de fonctionnement normal du réseau, mais aussi, pouvoir accompagner le client dans le cas où le réseau est endommagé. Cette manière de faire est appelée régulation, par laquelle nous cherchons des chemins alternatifs en cas de dommages dans le réseau et c'est cela qui exige d'avoir non pas une solution mais une population de solutions.
Ce travail est organisé en sections, les sections 2 et 3 présentent la problématique et le contexte de la régulation, ainsi que la prise en charge des perturbation. La section 4 présente Pareto OCF, la suite du papier est consacré à l'approche proposée ainsi qu'aux résultats obtenus.

2 problématique et contexte de la régulation

La prise en compte des perturbations dans un réseau de transport multimodal constitue un enjeu important d'un point de vue opérationnel, car les passagers ne disposent d'aucune information sur le trafic et les congestions au niveau du réseau. Ces informations sont difficiles à mettre en œuvre pour de nombreuses raisons : organisationnelles, économiques, juridiques et techniques..., etc. [1]
Plusieurs types de perturbations peuvent affecter la régularité du trafic. En général, une perturbation concerne un ensemble de stations dans une zone ou bien un ensemble de modes de transport [1].
Afin de réagir en cas de perturbation, la gestion en temps réel est très importante. Dans ce cas, un processus de régulation est réalisé en temps réel, afin de contourner où amortir ces perturbations, qui peuvent aussi apparaître simultanément.
Le processus de régulation contient plusieurs tâches difficiles allant de la détection des perturbations à la prise de décision [3]. Le régulateur doit constamment faire face à tous les incidents en temps réel. Il doit prendre en un temps limité des décisions immédiates pour traiter ces incidents.

Dans [5,3], les auteurs divisent le processus de régulation en deux modules : Systèmes d'Aide à l'Exploitation (SAE), Système d'Aide à la Régulation (SAR). Bien que les SAE et SAR sont importants,la littérature ne rapporte que très peu de travaux, parmi les approches de résolution utilisées dans ce domaine, on peut citer : les algorithmes génétiques [1,6], les systèmes multi agent [7] et la théorie de la logique floue [4].
C'est le deuxième module qui est retenu dans ce travail. Une fois le SAE identifié la source de perturbation, la cause de perturbation, etc). Le SAR commence par une analyse rapide de l'incident identifié. Il s'agit d'un premier constat pouvant donner une estimation de l'impact de la perturbation afin de mesurer son étendue. Puis le SAR propose au régulateur des actions ainsi que leurs effets sur l'état actuel du réseau. A ce moment le régulateur choisit une action et le SAR effectue l'implémentation.
Le régulateur se trouve parfois devant des cas de perturbations très difficiles, ceci explique l'importance d'améliorer les services de SAR.

L'objectif de ce travail est donc de développer un système d'aide à la régulation, par lequel Nous voulons proposer au régulateur pendant le voyage plusieurs solutions pour l'aider à choisir la plus adéquate en cas d'une perturbation. Ainsi, la tâche du régulateur sera simplifiée, ce qui améliore la qualité du service rendu aux voyageurs.
Le processus de régulation est illustré par la figure 1.

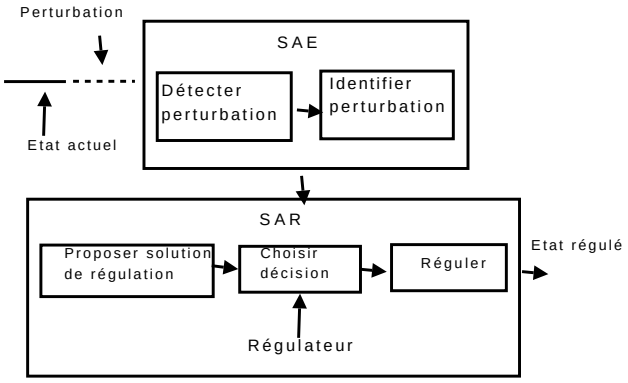


Fig. 1. Le déroulement du processus de régulation en temps réel

3 Prise en charge des perturbations

Pour gérer les cas de perturbations, nous avons besoin d'une population finale dont les solutions sont dispersées (diversifiées) sur toute la zone et utilisant le plus de modes possibles. Autrement dit, si nous disposons de solutions robustes représentatives des différentes régions et des différents modes de transport, nous augmentons aussi la probabilité de trouver une solution de secours en temps réel lors des perturbations. Sur ce principe, Zidi et Hammadi [1] ont développé une méthode d'optimisation génétique qui permet de proposer plusieurs solutions au régulateur pour prendre rapidement la meilleure décision en cas de perturbation. La diversification des solutions est assurée par deux variables (A_{ij}, M_i) permettant de contrôler les opérateurs de croissement génétique.
A_{ij} est le nombre d'arcs communs entre deux solutions (i) et (j), minimiser cette variable permet d'augmenter l'exploration des solutions sur la zone de transport. La deuxième variable M_i est le nombre de modes utilisés dans une solution, la maximisation des M_i permet d'améliorer la population en garantissant la multimodalité des solutions ainsi que la diversification. L'inconvénient majeur de la stratégie de Zidi et Hammadi [1] est qu'elle élimine arbitrairement de nombreuses solutions optimales, qui pourraient pourtant s'avérer intéressantes et utiles quand une perturbation surjet. Comme il est plus difficile d'établir en même temps la diversification et l'intensification des solutions dans la population finale. Notre proposition sépare le contrôle de l'intensification et la diversification entre deux colonies de fourmis. La colonie C1 construit ses solutions en

favorisant l'intensification et la colonie C2 en favorisant la diversification. De cette manière, nous augmentons la chance de trouver un chemin de secours en cas de perturbation. Le principal apport de notre contribution par rapport à l'approche de Zidi et Hammadi [1] réside dans la qualité de solutions proposées au régulateur.

4 Approche Bi-colonie d'aide à la régulation

4.1 Système d'optimisation proposé : stratégie de résolution

L'objectif principal n'est pas de définir une seule solution d'une perturbation, mais de rechercher un ensemble de meilleures solutions possibles. L'adaptation de ce principe par la méthode d'optimisation par colonie de fourmis, repose sur une coopération de deux colonies (C1, C2) qui s'exécutent séquentiellement l'une après l'autre (C1 puis C2), tel que C1 propose une population (P1) de k meilleures solutions. Ce qui permet à l'utilisateur de choisir celle qui lui convient le mieux en cas de perturbation lors de son passage. Malheureusement, il n'existe aucun mécanisme de recherche qui permet de diversifier les solutions dans la première colonie, c'est à dire, il y a une possibilité où une perturbation peut affecter toutes les solutions de C1, c'est à dire aucune solution de P1 ne convient pour continuer le trajet prévu. Pour surmonter ce problème, nous avons ajouté une deuxième colonie C2. C2 utilise un mécanisme de diversification qui évite les solutions de C1, afin de proposer une autre population (P2), appelée solutions de secours. Autrement dit, en cas de perturbation qui touche toutes les solutions de P1, cette population ne représentera habituellement plus de solutions valides à la nouvelle instance de problème, et la meilleure solution pour la nouvelle instance peut être trouvée facilement et rapidement à partir des solutions de secours proposées par C2.

5 Pareto OCF pour la résolution multi-objectifs du PCCM

Dans cette section, nous donnons l'algorithme Pareto OCF pour l'optimisation multicritère à exécuter par une seule colonie de fourmis.De simples modifications nous permettrons don adaptations au principe de Bi-colonies.
L'algorithme Pareto OCF nous permet de donner les différentes solutions qui couvrent le front Pareto optimal tout en utilisant une colonie hétérogène [8,9,10], où chaque fourmi donne une importance relative (poids) aux critères d'optimisation différemment, pour qu'il soit possible de diversifier la recherche dans l'espace des solutions réalisables. Dans les deux approches [8,9], les valeurs des poids changent d'une manière déterministe. Dans [10], les valeurs des poids changent d'une manière aléatoire. Afin d'accélérer la convergence de notre approche vers les solutions de compromis, les valeurs des poids changent dynamiquement d'une

fourmi à une autre en fonction de la distance normalisée entre l'ensemble des solutions Pareto optimales et le point idéal pour chaque objectif.

- Les objectifs à minimiser sont : le temps du trajet (f_1), les temps d'attentes (f_2), et le coût de transport (f_3).

- Le but est de trouver un ensemble Q de solutions Pareto optimales au lieu de trouver une seule solution.

- Pour guider les fourmis dans l'espace de recherche, on propose trois informations heuristiques η_1, η_2, η_3 pour les trois fonctions objectifs f_1, f_2, f_3 et deux autres variables w_1, w_2 pour déterminer l'influence relative de chaque heuristique, où w_1 est le poids accordé à η_1 et η_2, w_2 est le poids accordé à η_3.

- Les informations heuristiques de déplacement du nœud courrant i vers le nœud j par un mode de transport x en quittant i à une date t sont :

$$\eta_{1[ij(x,t)]} = \frac{1}{v_{ij(x,t)}},\ \eta_{2[ij(x,t)]} = \frac{1}{\theta_{ij(x,t)}},\ \eta_{3[ij(x,t)]} = \frac{1}{\phi_{ij(x,t)}}$$

- On donne la règle de déplacement qui consiste à choisir à la fois le successeur du nœud courant i à visiter,le mode de transport x à utiliser et la date de départ t correspondante par :

$$P_{ij(x,t)} = \begin{cases} \frac{\tau^{\alpha}_{ij(x,t)}(w_1 \times (\eta_{1[ij(x,t)]} \times \eta_{2[ij(x,t)]}) + w_2 \times \eta_{3[ij(x,t)]})^{\beta}}{\sum_{k \notin tabu} (\tau_{ik\,\forall(w,l)})^{\alpha}(w_1 \times (\eta_{1[ik\,\forall(w,l)]} \times \eta_{2[ik\,\forall(w,l)]}) + w_2 \times \eta_{3[ik\,\forall(w,l)]})^{\beta}} j \notin tabou \\ 0 \qquad\qquad Sinon \end{cases} \quad (1)$$

- L'ensemble Pareto Q est mis à jour après que chaque fourmi y construit une solution.

- Les solutions de l'ensemble Q sont utilisées pour mettre à jour la mémoire de phéromone.

Cette méthode a été formalisée en intégrant les éléments décrits précédemment dans une procédure générique qui se déroule en cinq étapes (voir l'algorithme 1).

Etape1 : Recherche du point idéal L'optimisation de chaque objectif pris séparément ait permis de trouver les différentes solutions optimales. Le vecteur de solutions correspond alors au point idéal et peut être exprimé par : $S^* = \{s_1^*, s_2^*\}$ Où :

- s_1^* : La solution optimale qui minimise le temps total de transport (temps du trajet et les temps d'attentes)
- s_2^* : La solution optimale qui minimise le coût total de transport.

Etape2 : Initialisation Cette phase consiste tout d'abord à initialiser les

Algorithm 1 PMO-OCF

Début
Etape1 : Recherche du point idéal s^*
Etape2 :
  Initialiser les paramètres OCF.
  Initialiser la trace de phéromone à τ_{max}.
  Initialiser l'ensemble Pareto Q à un ensemble vide.
Etape3 :
repeat
  for chaque fourmis k $\in [1, m]$ **do**
    $s_k \leftarrow Construction - Solution()$
    Evaluer la solution s_k obtenue selon les différents objectifs et mettre à jour l'ensemble Pareto Q avec les solutions non dominées.
    if (il y a un mis à jour dans l'ensemble Q) **then**
      Calculer le vecteur de poids $W_{k+1} = (w_1, w_2)$
    end if
  end for
  Etape4 :
  Evaluer les solutions de l'ensemble Pareto Q selon les différents objectifs et Mettre à jour la mémoire de phéromone en respectant les limites $[\tau_{min}, \tau_{max}]$
until un critère d'arrêt est vérifié.
Etape5 : Affichage des solutions Pareto Optimales.
Fin.

paramètres spécifiques à l'OCF, tel que le nombre de fourmis, le nombre maximal de cycles, le paramètre d'évaporation de la trace de phéromone, l'intervalle $[\tau_{min}, \tau_{max}]$, ainsi que les paramètres de la règle de transition (α, β et le vecteur de poids W). Les pistes sont initialisées à leur valeur maximale τ_{max}, afin de garantir une exploration plus large de l'espace de recherche durant les premiers cycles. L'ensemble Pareto des solutions non dominées Q est initialisé à l'ensemble vide.

Etape3 :Déroulement d'un cycle La construction de solution progresse tant que la fourmi n'arrive pas à la ville destination. A chaque étape dans la construction, le choix de transition est effectué selon la règle de déplacement présentée par la formule 1, cette règle est définie proportionnellement à un facteur phénoménal τ et un facteur heuristique qui regroupe les trois fonctions objectifs à optimiser f_1, f_2, f_3. Ces trois fonctions étant pondérés par $W = \{w_1, w_2\}$ qui déterminent leur importance relative. Notons que les valeurs des poids changent dynamiquement d'une solution à une autre en fonction de la distance normalisée entre l'ensemble de solutions Pareto optimales et le point idéal pour chaque objectif. Formellement, les valeurs de poids sont calculées en utilisant la formule suivante :

$$\begin{cases} w_1 = \sqrt{\frac{1}{|Q|}\sum_{j=1}^{|Q|}(T_{SD}(s_j) - T_{SD}(s_1^*))^2} \\ w_2 = \sqrt{\frac{1}{|Q|}\sum_{j=1}^{|Q|}(f_3(s_j) - f_3(s_2^*))^2} \end{cases} \quad (2)$$

qui garantit, d'une part, que la valeur de chaque poids appartient à l'intervalle $[0, 1]$ et d'autre part, pour que la somme de ces poids soit égale à 1 tel que : $w_i = \frac{w_i}{w_1+w_2}$ avec i=1,2.

Cette méthode de calcul permet d'assurer la notion d'hétérogénéité, afin d'élargir légèrement l'espace de recherche et ainsi être en mesure d'atteindre des compromis intéressants se trouvant à proximité. De plus, elle permet d'obtenir des poids qui convergent vers le point idéal.

Suite à la construction d'une solution, l'évaluation de cette dernière est faite pour chacun des objectifs du problème. Les relations de dominance sont alors vérifiées pour chacune des solutions et les solutions Pareto sont conservées dans un ensemble Q. Formellement, cette opération est représentée par :

$$\text{Si } (S) \text{ n'est pas dominée par } \forall T \in Q \text{ Alors } Q = S \cup Q - \{\forall T \in Q, T \leq S\}$$

Etape4 : La mise à jour de phéromone Cette opération s'effectue en deux étapes (formule 3). Dans une première étape, toutes les traces de phéromone sont diminuées, pour simuler l'évaporation en multipliant chaque composant phéromonal par un ratio de persistance ρ tel que $0 < \rho < 1$, dans une deuxième étape, les fourmis trouvant des solutions Pareto optimales déposent une quantité Δ de phéromone.

$$(I) \begin{cases} \forall (i,j) \in A \tau_{ij\forall(x,t)} = \rho \times \tau_{ij\forall(x,t)} \\ \tau_{ij(x,t)} = \tau_{min} \; si \; \tau_{ij(x,t)} \prec \tau_{min} \end{cases} \tag{3}$$

$$(II) \begin{cases} \forall T \in Q : \forall (i,j,x,t) \in T \tau_{ij(x,t)} = \tau_{ij\forall(x,t)} + \Delta \\ \tau_{ij(x,t)} = \tau_{max} \; si \; \tau_{ij(x,t)} \succ \tau_{max} \end{cases}$$

Etape5 :Vérification de conditions d'arrêt Le processus itératif prend fin lorsque les conditions d'arrêt sont atteintes. Dans notre approche, l'algorithme s'arrête après un nombre maximum de cycles $tmax$. La suite de ce travail présente comment utiliser deux colonies pour la prise en charge des perturbations.

5.1 Bi-PMO-OCF

pour pouvoir exploiter l'algorithme PMO-OCF par les deux colonies simultanément, on doit le modifié de sorte à assurer les deux points suivants :

- Les meilleures solutions de chaque colonie (C1 ou C2) sont conservées dans une population finie
- Un mécanisme de diversification de recherche entre les deux colonies C1 et C2 est introduit.

  Alors le pseudo-code de l'approche Bi-PMO-OCF des deux colonies Ci (i=1,2) est identique à PMO-OCF, sauf que nous avons ajouté une procédure

(voir Algorithme 2) composée de 2 phases pour mettre à jour la population de solutions P. Dans la première phase, pour chaque solution fournie s, les relations de dominance sont vérifiées pour construire l'ensemble Pareto Q. la deuxième phase de cette procédure consiste à utiliser l'ensemble Q pour ne conserver dans P que les solutions Pareto et les meilleures solutions dominées (ensemble R) identifiées par la méthode de classement proposée (voir Algorithme 3). La définition de la population P varie selon la taille de l'ensemble Q :

$$P = \begin{cases} Q & \mid Q \mid \geq k \\ Q \cup R & Sinon \end{cases} \qquad (4)$$

Dans le deuxième cas, il faut compléter l'ensemble Pareto Q par un ensemble R qui contient $k- \mid Q \mid$ solutions dominées, ici nous proposons de construire l'ensemble R par la sélection des solutions les plus proches de la frontière Pareto. Cette sélection est effectuée grâce à une fonction $d(s)$ qui peut être définie dans l'espace de solutions réalisables en exploitant le concept de dominance Pareto comme suit :

$$d(s) = \mid \forall t \in Q \;\; t > s \mid \qquad (5)$$

Rappelons que $t > s$ signifie que la solution t domine la solution s.
Afin de mieux expliquer le principe de base de cette fonction, considérons l'exemple de minimisation de deux objectifs F1 et F2 de la figure 2. Dans cet exemple, les huit solutions s_i $(i = 1, \ldots, 8)$ sont représentées dans le plan défini par les deux fonctions F1 et F2, le front Pareto de ce plan est défini par quatre solutions. L'évaluation de la fonction $d(s)$ des solutions non dominées au sens de Pareto est indiquée dans le tableau de la même figure.

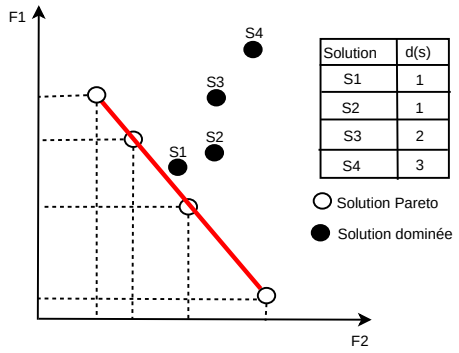


Fig. 2. Classement des solutions dominées en utilisant la fonction d(s)

En observant la figure 2, nous pouvons constater que :
– La méthode de calcul proposée permet d'affecter aux meilleures solutions dominées la valeur du $d(s)$ la plus petite. Ainsi, les solutions ayant la

plus petite valeur de $d(s)$ auront plus de chances d'être choisies dans la construction de l'ensemble R.

– $d(s_i) = d(s_j) \nRightarrow s_i <> s_j$, c'est le cas par exemple pour les deux solutions s_1 et s_2 ($d(s_1) = d(s_2)$, mais s_1 domine s_2).

Algorithm 2 update-population(Q, s, R)

Début
$N \leftarrow \emptyset$
if s n'est pas dominée par $\forall t \in Q$ **then**
  $N \leftarrow Q - \{\forall t \in Qs > t\}$
  $Q \leftarrow Q - N \cup \{s\}$
else
  $R \leftarrow R \cup \{s\}$
end if
if $|Q| < k$ **then**
  $R \leftarrow R \cup N$
  if $|R| + |Q| < k$ **then**
    classement(R)
    supprimer les $|R| + |Q| - k$ derniers solutions de R
  end if
else
  $R \leftarrow \emptyset$
end if
$Q \leftarrow Q \cup R$
Fin

Algorithm 3 update-population(R)

Début
Les solutions de l'ensemble R sont triées dans un ordre croissant en fonction de la valeur de $d(s)$.
Les solutions ayant la même valeur de $d(s)$ sont triées en ordre croissant en fonction de l'évaluation de la somme $\sum_{i \leq |F|} f_i(s)$
Fin

La procédure de classement décrite dans l'algorithme 3 est appliquée pour trier les solutions dominées afin de construire l'ensemble R.

En observant le pseudo-code de l'algorithme 3, nous pouvons remarquer que la deuxième étape de classement permet de trier les solutions ayant la même valeur de $d(s)$ en évitant le cas suivant :

$$d(s_i) = d(s_j) \; et \; s_i > s_j \Longrightarrow s_i \; est \; classee \; avant \; s_j \qquad (6)$$

Ce cas est impossible de se produire puisque l'équation 7 est toujours valide dans la procédure de classement proposée.

$$\sum_{t\leq|F|} f_t(s_i) < \sum_{t\leq|F|} f_t(s_j) \Longrightarrow s_j \not\succ s_i \qquad (7)$$

6 Résultats expérimentaux

Pour valider notre approche, plusieurs tests sont réalisés sur des graphes de 60 et 100 noeuds, le nombre de modes disponibles est fixé à trois. Cinq problèmes (P1,. . .,P5) sont construits selon les points de départ et destination sélectionnés dans ces graphes, pour lesquels nous avons examiné plusieurs tests.

Le tableau 1 présente les paramètres spécifiques à l'OCF des deux colonies (C1 et C2) utilisés pour tester l'approche proposée dans ce travail. Ces paramètres ont été fixés après une série de tests préliminaires appliqués sur des réseaux aléatoires de différentes tailles. Les paramètres de la 2^{eme} colonie doivent être choisi afin d'éviter la convergence vers la population de solutions proposée par la 1^{ere} colonie. Les deux paramètres les plus sensibles pour contrôler la convergence d'un algorithme sont probablement le nombre maximum de cycles $tmax$ et le paramètre d'évaporation ρ

pr	t_{max}	M	α	β	ρ	$[\tau_{min},\tau_{max}]$	Δ	K
C1	150	80	2	1	0.9	[0.07,6]	0.005	15
C2	60	60	2	1	0.6	[0.07,6]	0.05	15

Tab. 1. Paramètres OCF adoptés pour tester l'algorithme Bi-PMO-OCF

La détermination de la taille de la population k est très spécifique aux caractéristiques du problème traité, en particulier aux difficultés rencontrées pour produire des solutions faisables pour faire face aux perturbations. Dans cette expérimentation, une valeur limite admissible de la taille de la population est de 15 solutions pour chaque colonie. En dessous de cette valeur, un manque de diversification est constaté.

Notre système propose au client la meilleure solution trouvée par l'algorithme Bi-colonies. A ce niveau, on distingue deux types de perturbations : une perturbation se produit avant le début de voyage et une autre pendant le voyage. Le premier type de perturbation demande juste de rechercher une autre solution admissible dans la population de solutions, pour le deuxième cas, on doit déterminer tout d'abord la position courante du client. Le régulateur se charge de déduire les meilleurs scénarii possibles qui permettent au client de continuer son déplacement sans problème vers le noeud prévu.

Après avoir choisie une solution finale (la meilleure solution), nous générons des perturbations aléatoires pour calculer les chances de trouver une solution

alternative dans les deux populations proposées par l'exécution de l'algorithme Bi-colonie. Notant P la probabilité de trouver une solution alternative en cas de perturbations.

Après plusieurs scénarii de perturbations générées aléatoirement, nous avons obtenu des statistiques montrant l'évolution de la probabilité moyenne (avec et sans) considération des solutions trouvées par la deuxième colonie. Les résultats obtenus sont présentés dans le tableau 2.

		P1	P2	P3	P4	P5
C1	P	96%	80%	65.25%	90%	90%
C1,C2	P	100%	96%	95%	100%	98%

Tab. 2. Résultats de tests obtenus (perturbation pendant le voyage) de l'approche Bi-PMO-OCF

Ce tableau montre que les probabilités de trouver une solution alternative en cas d'une perturbation sont plus élevées si l'on considère les solutions trouvées par la 2^{eme} colonie, ce qui explique l'intérêt de cette colonie en terme de diversification. Notons que dans le cas ou une perturbation se produit avant le début de voyage, il existe toujours une chance (100%) de trouver une autre solution de secours.

7 Conclusion

La régulation du trafic en temps réel des réseaux de transport multimodal est une tâche de plus en plus complexe qui nécessite l'élaboration d'un SAR. Dans ce travail on a proposé une nouvelle approches de résolution du problème de PCCM tout en évitant l'inconvénient majeur de la stratégie adopté par Zidi et Hammadi [1] qui réside dans la difficulté d'établir en même temps la diversification et l'intensification des solutions dans la population finale. Dans notre approche, il s'agit d'une nouvelle idée de recherche d'itinéraire, qui s'appuie sur une coopération de deux colonies pour générer une population de chemins optimums et diversifiés sur toute la zone de transport. La diversification est assurée par le principe de la construction de la deuxième population de solutions en évitant les solutions de la première population. Les résultats obtenus par cette version montrent l'efficacité et l'intérêt pratique de notre approche.

Références

1. k . Zidi and S . Hammadi, *Algorithme génétique avec contrôle des opérateurs pour l'optimisation multicritère d'un déplacement dans un réseau de transport multimodal*, Workshop avec école intégrée Méthodologies et Heuristiques pour l'Optimisation des Systèmes Industriels 24-26 Avril,Hammamet ,Tunisie, 2005.

2. A. Tahraoui,Z. Hebbas ,A. Aloui *Using ACO for solving a multimodal transport problem*, Meta'08,29-31 octobre,Hammamet ,Tunisie, 2008.
3. K. Zidi,M. Salah ,*SARR : système d'aide à la régulation et la reconfiguration des réseaux de transport multimodal*, http ://hdl.handle.net/1908/1067, Université des sciences et technologies de Lille,2007.
4. M. Ould sidi, S. Hammadi, S. Hayat and P. Borne,*Approche floue d'un système d'évaluation des stratégies de régulation d'un réseau de transport* , In S. C. et al.,editor, Applications of Evolutionary Computing - EvoWorkshops,2004.
5. K. Bouamrane,T. Bonte and M. Sevaux,and C. Tahon,*SART : un système d'aide à la décision pour la régulation d'un réseau de transport bimodal* , Proceedings of the workshop Mèthodologies et Heuristiques pour l'Optimisation des Systèmes Industriels, MHOSI 2005, Hammamet, Tunisie, avril,2005.
6. B. F.Chaar and S. Hammadi, *Régulation spatio-temporelle d'un réseau de transport multimodal,*, e-STA - Sciences et Technologie de l'Automatique, Vol 2, 2005.
7. B. F.Chaar and S. Hammadi, *Modélisation multi-agent et aide à la décision : application à la régulation des correspondances dans les réseaux de transport urbain* , thèse de doctorat, Université de Lille, décembre 2002.
8. S. Iredi,D. Merkle and M. Middendorf,*Bi-Criterion Optimization with Multi Colony Ant Algorithms*, journal = "Lecture Notes in Computer Science", volume = "1993", pages = "359–372", year = "2001", url = "citeseer.ist.psu.edu/iredi00bicriterion.html
9. D. Pinto, B. Baran,*Solving multiobjective multicast routing problem with a new ant colony optimization approach*, booktitle="Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking , address="Cali, Columbia",year="2005", pages="11–19",
10. K. Doerner, K. F,W. Gutjahr,R. Hartl,C. Strauss, and C. Stummer,*Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection*, European Journal of Operational Research,year="2006",volume="171",pages="830–841",

Méthode de support à deux phases pour la résolution des problèmes de programmation linéaire à variables bornées : Comparaison numérique

M. Bentobache et M. O. Bibi

Laboratoire de Modélisation et d'Optimisation des Systèmes
Département de Recherche Opérationnelle
Université de Béjaia, 06000 (Algérie)
mbentobache@yahoo.com, mohandbibi@yahoo.fr

Résumé Dans cet article, nous allons présenter la méthode primale de support pour la résolution des problèmes de programmation linéaire à variables bornées [4]. Ensuite, on traitera deux techniques pour l'initialisation de cette dernière, à savoir : la technique d'ajout de plusieurs variables artificielles et celle d'ajout d'une seule variable artificielle. Afin de comparer les performances de la méthode de support avec celles de la méthode du simplexe, nous avons réalisé une implémentation numérique de plusieurs variantes de la méthode de support et celle de la méthode des deux phases du simplexe avec plusieurs variables artificielles, et ce, sous le langage de programmation Matlab. Enfin, des résultats numériques sur le temps d'exécution et le nombre d'itérations sont présentés pour les différentes variantes, appliquées aux problèmes-tests de la librairie netlib [11].

Keywords : Solution réalisable de support, variable artificielle, méthode du simplexe, méthode de support à deux phases, Netlib.

1 Introduction

Dans [1], une méthode, dite méthode de support à deux phases avec une seule variable artificielle pour résoudre les problèmes de programmation linéaire à variables non-négatives et bornées a été développée. Le principe général de cette dernière est le suivant : lors de la première phase, on commence par rechercher un support initial avec la méthode de Gauss-Jordan ; puis on procède à la recherche d'une solution réalisable initiale, et ce, en résolvant un problème auxiliaire avec une seule variable artificielle et ayant une solution réalisable de support initiale évidente. Dans la seconde phase, on résoud le problème original avec la méthode primale de support développée par R. Gabasov et F.M. Kirillova [4], qui est une généralisation de la méthode du simplexe [3]. Dans [2], nous nous sommes occupés de la résolution des problèmes à variables non-négatives. Nous avons alors proposé deux approches pour la recherche d'un support initial : la première

consiste à appliquer la méthode d'élimination de Gauss avec pivot partiel au système d'équations correspondant aux contraintes principales et l'autre consiste à transformer les contraintes d'égalité en contraintes d'inégalité, ce qui nous donne ainsi un support évident mais augmente la dimension du problème. Une comparaison numérique avec la méthode du simplexe sur des problèmes-tests à variables non-négatives de la librairie netlib a été ainsi réalisée. Dans ce papier, nous avons porté notre intérêt à l'étude des performances de la technique utilisant une seule variable artificielle pour résoudre les problèmes de programmation linéaire à variables bornées.

Le papier est organisé comme suit : dans la deuxième section, nous allons présenter quelques rappels sur la méthode primale de support pour la résolution des problèmes de programmation linéaire à variables bornées. Dans la troisième section, on présentera les différentes techniques pour l'initialisation de la méthode de support. La quatrième section sera consacrée à la présentation de quelques résultats numériques qui nous permettront de comparer les performances de la méthode de support à variables bornées avec celles de la méthode du simplexe à variables bornées. On terminera cet article par une conclusion.

2 Méthode primale de support à variables bornées

2.1 Position du problème et définitions

Le problème de programmation linéaire à variables bornées se présente sous la forme standard suivante :

$$\max\ z = c^T x, \tag{1}$$

$$Ax = b, \tag{2}$$

$$l \leq x \leq u, \tag{3}$$

où c, x sont des n-vecteurs ; b un m-vecteur ; A est une matrice d'ordre $(m \times n)$, avec $rang A = m < n$; l et u sont des n-vecteurs. Dans ce qui suit, on supposera que $||l|| < \infty$, $||u|| < \infty$. Définissons les ensembles d'indices suivants :

$$I = \{1, 2, \ldots, m\},\ \ J = \{1, 2, \ldots, n\},\ \ J = J_B \cup J_N,\ \ J_B \cap J_N = \emptyset,\ \ |J_B| = m.$$

On peut alors écrire et fractionner les vecteurs et la matrice A de la manière suivante :

$$l = l(J) = (l_j, j \in J),\ u = u(J) = (u_j, j \in J);$$

$$x = x(J) = (x_j, j \in J),\ x = (x_B, x_N),\ x_B = x(J_B) = (x_j, j \in J_B),$$

$$x_N = x(J_N) = (x_j, j \in J_N);\ c = c(J) = (c_j, j \in J),\ c = (c_B, c_N),$$

$$c_B = c(J_B) = (c_j, j \in J_B),\ c_N = c(J_N) = (c_j, j \in J_N);$$

$$A = A(I, J) = (a_{ij}, i \in I, j \in J) = (a_1, a_2, \ldots, a_n) = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{pmatrix},$$

$$a_j = (a_{1j}, a_{2j}, \ldots, a_{mj})^T, j = \overline{1,n};\ A_i = (a_{i1}, a_{i2} \ldots, a_{in}), i = \overline{1,m};$$
$$A = (A_B, A_N),\ \ A_B = A(I, J_B),\ \ A_N = A(I, J_N).$$

Un vecteur x vérifiant les contraintes (2)-(3) est appelé *solution réalisable* du problème (1)-(3). Une solution réalisable x^0 est dite optimale si $z(x^0) = c^T x^0 = \max\ c^T x$, où x est pris parmi toutes les solutions réalisables du problème (1)-(3). D'autre part, une solution réalisable x^ϵ est appelée ϵ-*optimale* ou *suboptimale* si

$$z(x^0) - z(x^\epsilon) = c^T x^0 - c^T x^\epsilon \leq \epsilon,$$

où x^0 est une solution optimale du problème (1)-(3) et ϵ un nombre positif ou nul choisi à l'avance. Soit un sous-ensemble d'indices $J_B \subset J$ tel que $|J_B| = |I| = m$. L'ensemble J_B est alors appelé *support* si $\det A_B = \det A(I, J_B) \neq 0$. Le couple $\{x, J_B\}$ formé de la solution réalisable x et du support J_B est appelé *solution réalisable de support* (SRS). La solution réalisable de support est dite non-dégénérée si $l_j < x_j < u_j,\ j \in J_B$. Définissons le vecteur des multiplicateurs de Lagrange π, ainsi que le vecteur des coûts réduits Δ :

$$\pi^T = c_B^T A_B^{-1},\ \ \Delta^T = \Delta^T(J) = \pi^T A - c^T = (\Delta_B^T, \Delta_N^T),$$

où $\Delta_B^T = c_B^T A_B^{-1} A_B - c_B^T = 0,\ \Delta_N^T = c_B^T A_B^{-1} A_N - c_N^T$.

Théorème 1 *(Critère d'optimalité [4]). Soit $\{x, J_B\}$ une SRS du problème (1)-(3). Alors les relations :*

$$\begin{cases} \Delta_j \geq 0 \text{ pour } & x_j = l_j, \\ \Delta_j \leq 0 \text{ pour } & x_j = u_j, \\ \Delta_j = 0 \text{ pour } & l_j < x_j < u_j,\ \ j \in J_N, \end{cases} \tag{4}$$

sont suffisantes pour l'optimalité de la solution réalisable x. Ces mêmes relations sont aussi nécessaires dans le cas où la solution réalisable de support est non-dégénérée.

On appelle *estimation de suboptimalité* la quantité $\beta(x, J_B)$ définie par :

$$\beta(x, J_B) = \sum_{\Delta_j > 0, j \in J_N} \Delta_j (x_j - l_j) + \sum_{\Delta_j < 0, j \in J_N} \Delta_j (x_j - u_j). \tag{5}$$

On a alors le théorème suivant [4] :

Théorème 2 *(Condition suffisante de suboptimalité). Soit $\{x, J_B\}$ une SRS du problème (1)-(3) et ϵ un nombre positif ou nul arbitraire. Si $\beta(x, J_B) \leq \epsilon$, alors la solution réalisable x est ϵ-optimale.*

2.2 Schéma de l'algorithme primal de support

Soit $\{x, J_B\}$ une solution réalisable initiale de support et ϵ un nombre arbitraire positif ou nul. Le schéma de l'algorithme présente ainsi les étapes suivantes :

(1) Calculer $\pi^T = c_B^T A_B^{-1}$, $\Delta_j = \pi^T a_j - c_j,\ j \in J_N$;

(2) calculer $\beta(x, J_B)$ avec la formule (5) ;

(3) si $\beta(x, J_B) = 0$, alors l'algorithme s'arrête avec $\{x, J_B\}$ une SRS optimale.

(4) Si $\beta(x, J_B) \leq \epsilon$, alors l'algorithme s'arrête avec $\{x, J_B\}$ une SRS ϵ-optimale. Sinon, aller à l'étape (5) ;

(5) déterminer l'ensemble des indices non-optimaux :

$$J_{NNO} = \{j \in J_N : [\Delta_j < 0 \text{ et } x_j < u_j] \text{ ou } [\Delta_j > 0 \text{ et } x_j > l_j]\};$$

(6) choisir l'indice j_0 tel que $|\Delta_{j_0}| = \max\limits_{j \in J_{NNO}} |\Delta_j|$;

(7) calculer la direction d'amélioration d en utilisant les relations

$$d_{j_0} = -\operatorname{sign} \Delta_{j_0}, d_j = 0,\ j \neq j_0,\ j \in J_N;\ d(J_B) = -A_B^{-1} A_N d(J_N) = -A_B^{-1} a_{j_0} d_{j_0};$$

(8) calculer $\theta_{j_1} = \min\limits_{j \in J_B} \theta_j$, où θ_j est déterminé par la formule

$$\theta_j = \begin{cases} (u_j - x_j)/d_j, \text{ si } d_j > 0\,; \\ (l_j - x_j)/d_j, \text{ si } d_j < 0\,; \\ \infty, \qquad\qquad \text{si } d_j = 0\,; \end{cases}$$

(9) calculer θ_{j_0} en utilisant la formule

$$\theta_{j_0} = \begin{cases} x_{j_0} - l_{j_0}, \text{ si } \Delta_{j_0} > 0\,; \\ u_{j_0} - x_{j_0}, \text{ si } \Delta_{j_0} < 0\,; \end{cases}$$

(10) calculer $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}$;

(11) calculer $\bar{x} = x + \theta^0 d$, $\bar{z} = z + \theta^0 |\Delta_{j_0}|$. Aller en (12) ;

(12) calculer $\beta(\bar{x}, J_B) = \beta(x, J_B) - \theta^0 |\Delta_{j_0}|$;

(13) si $\beta(\bar{x}, J_B) = 0$, alors l'algorithme s'arrête avec $\{\bar{x}, J_B\}$ une SRS optimale.

(14) Si $\beta(\bar{x}, J_B) \leq \epsilon$, alors l'algorithme s'arrête avec $\{\bar{x}, J_B\}$ une SRS ϵ-optimale. Sinon, aller à l'étape (15) ;

(15) si $\theta^0 = \theta_{j_0}$, alors on pose $\bar{J}_B = J_B$;

(16) si $\theta^0 = \theta_{j_1}$, alors on pose $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$;

(17) on pose $J_B = \bar{J}_B$, $x = \bar{x}$. Aller à l'étape (1).

3 Initialisation de la méthode de support à variables bornées

Considérons le problème de programmation linéaire sous forme générale suivante :

$$\max\ z = \tilde{c}^T \tilde{x}, \tag{6}$$

$$A^1 \tilde{x} \leq b^1, \tag{7}$$

$$A^2 \tilde{x} = b^2, \tag{8}$$

$$\tilde{l} \leq \tilde{x} \leq \tilde{u}, \tag{9}$$

où $\tilde{c}, \tilde{x}, \tilde{l}, \tilde{u}$ sont des vecteurs de $\Re^p$; A^1 est une matrice de dimension $(m_1 \times p)$, A^2 est une matrice de dimension $(m_2 \times p)$, $b^1 \in \Re^{m_1}$, $b^2 \in \Re^{m_2}$. Puisque dans ce travail on ne s'interesse qu'à la comparaison du temps d'exécution des différentes variantes de la méthode de support avec la méthode du simplexe, alors on supposera que $||\tilde{l}|| < \infty$, $||\tilde{u}|| < \infty$.

3.1 Technique d'ajout de plusieurs variables artificielles

Afin d'initialiser la méthode de support pour la résolution d'un problème de programmation linéaire à variables bornées écrit sous la forme standard (1)-(3), R. Gabasov et F.M. Kirillova [4,5] ajoutent autant de variables artificelles que de contraintes. Dans ce travail, nous nous intéressons à la résolution d'un problème sous la forme générale (6)-(9). De ce fait, nous déduisons d'une manière adéquate des bornes finies pour les variables d'écart ajoutées aux contraintes d'inégalité et nous n'ajoutons une variable artificielle que si c'est nécessaire. Ainsi le nombre de variables artificielles ajoutées sera inférieur ou égale au nombre de contraintes.

3.2 Technique d'ajout d'une seule variable artificielle

Dans cette technique, on commence d'abord par rechercher un support initial ; puis on procède à la recherche d'une solution réalisable initiale pour le problème original, et ce, en résolvant un problème auxiliaire avec une seule variable artificielle. Afin de rechercher un support initial, nous proposons trois approches :

Approche 1 : après transformation du problème (6)-(9) en un problème équivalent sous forme standard, on obtient le problème suivant :

$$\max\ z = c^T x, \tag{10}$$

$$Ax = b, \tag{11}$$

$$l \leq x \leq u, \tag{12}$$

où $c = \begin{pmatrix} \tilde{c} \\ 0 \end{pmatrix} \in \Re^n$, avec $n = p + m_1$; $x = \begin{pmatrix} \tilde{x} \\ x^e \end{pmatrix}$, où $x^e = \begin{pmatrix} x^e_1 \\ \vdots \\ x^e_{m_1} \end{pmatrix} = \begin{pmatrix} x_{p+1} \\ \vdots \\ x_n \end{pmatrix}$; $\tilde{A} = \begin{pmatrix} A^1 \\ A^2 \end{pmatrix}$, $\xi = \begin{pmatrix} I_{m_1} \\ 0_{(m_2 \times m_1)} \end{pmatrix}$, $A = (\tilde{A}, \xi)$; $b = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix} \in \Re^m$, avec $m = m_1 + m_2$; $l = \begin{pmatrix} \tilde{l} \\ 0 \end{pmatrix} \in \Re^n$, $u = \begin{pmatrix} \tilde{u} \\ u^e \end{pmatrix} \in \Re^n$, où u^e représente le vecteur des bornes supérieures des variables d'écart telles que $u^e = (u^e_i, i = \overline{1, m_1})$, $u^e_i = b_i - \tilde{A}_i h^i$, avec h^i un p-vecteur que nous déduisons de la manière suivante :

$$h^i_j = \begin{cases} \tilde{l}_j, & \text{si } \tilde{a}_{ij} > 0\,; \\ \tilde{u}_j, & \text{si } \tilde{a}_{ij} < 0\,; \\ 0, & \text{si } \tilde{a}_{ij} = 0. \end{cases} \tag{13}$$

En effet, on a $\tilde{l}_j \leq \tilde{x}_j \leq \tilde{u}_j, j = \overline{1,p}$. Donc si $\tilde{a}_{ij} > 0$, alors $-\tilde{a}_{ij}\tilde{x}_j \leq -\tilde{a}_{ij}\tilde{l}_j$; si $\tilde{a}_{ij} < 0$, alors $-\tilde{a}_{ij}\tilde{x}_j \leq -\tilde{a}_{ij}\tilde{u}_j$. D'où $x_i^e = b_i - \sum_{j=1}^{p} \tilde{a}_{ij}\tilde{x}_j \leq b_i - \sum_{j=1}^{p} \tilde{a}_{ij}h_j^i$, $i = \overline{1,m_1}$.

Appliquons l'algorithme de Gauss avec pivot partiel au système d'équations (11). Soit $J_B = \{j_1, j_2, \ldots, j_r\}$, où $r \leq m$, le support obtenu. Si $r < m$, alors on élimine les $m-r$ contraintes redondantes pour avoir une matrice des contraintes de rang complet en lignes. Sinon, la matrice A est déjà de rang complet en lignes, i.e., $rangA = m$.

Approche 2 : On transforme les m_2 contraintes d'égalité (8) en m_2+1 contraintes d'inégalité. On obtient alors le problème de programmation linéaire équivalent suivant [14] :

$$\begin{aligned} \max\ z &= \tilde{c}^T\tilde{x}, \\ A^1\tilde{x} &\leq b^1, \\ -A^2\tilde{x} &\leq -b^2, \\ e^TA^2\tilde{x} &\leq e^Tb^2, \\ \tilde{l} \leq \tilde{x} &\leq \tilde{u}, \end{aligned}$$

où $e = (1, 1, \ldots, 1)^T$ est un m_2-vecteur. L'ajout de $m_1 + m_2 + 1$ variables d'écart à ce problème nous donne le problème sous forme standard suivant :

$$\max\ z = c^Tx, \tag{14}$$

$$Ax = b, \tag{15}$$

$$l \leq x \leq u, \tag{16}$$

où $c = \begin{pmatrix} \tilde{c} \\ 0 \end{pmatrix} \in \Re^n$, $x = (x_1, \ldots, x_p, x_{p+1}, \ldots x_n)^T \in \Re^n$, avec $n = p+m_1+m_2+1$; $A = (\tilde{A}, I_m)$, où $\tilde{A} = \begin{pmatrix} A^1 \\ -A^2 \\ e^TA^2 \end{pmatrix}$, $b = \begin{pmatrix} b^1 \\ -b^2 \\ e^Tb^2 \end{pmatrix} \in \Re^m$, avec $m = m_1 + m_2 + 1$; $l = \begin{pmatrix} \tilde{l} \\ 0 \end{pmatrix} \in \Re^n$, $u = \begin{pmatrix} \tilde{u} \\ u^e \end{pmatrix} \in \Re^n$, où u^e représente le vecteur des bornes supérieures des variables d'écart telles que $u^e = (u_i^e, i = \overline{1,m}), u_i^e = b_i - \tilde{A}_ih^i$, avec h^i un p-vecteur calculé avec la formule (13).

Remarquons que A est une matrice de dimension $(m \times n)$ de rang complet en lignes. Par conséquent, le problème (14)-(16) admet un support évident $J_B = \{p+1, \ldots, n\}$.

Approche 3 : elle est similaire à l'approche 2, néanmoins cette fois on transforme les m_2 contraintes d'égalité (8) en $2m_2$ contraintes d'inégalité ($A^2\tilde{x} = b^2 \Leftrightarrow A^2\tilde{x} \leq b^2$ et $-A^2\tilde{x} \leq -b^2$)

Nous pouvons constater que dans les trois approches sus-mentionnées, on obtient un problème de programmation linéaire équivalent sous la forme suivante :

$$\max\ z = c^Tx, \tag{17}$$

$$Ax = b, \tag{18}$$

$$l \leq x \leq u, \tag{19}$$

où A est une matrice de dimension $(m \times n)$ de rang complet en lignes, b un m-vecteur quelconque et $J_B = \{j_1, j_2, \ldots, j_m\}$ un support. Maintenant que l'on dispose d'un support initial, on entame la procédure de recherche d'une solution réalisable pour le problème original. Pour ce faire, considérons le problème auxiliaire suivant :

$$\max\ \psi = -x_{n+1}, \tag{20}$$

$$Ax + \rho x_{n+1} = b, \tag{21}$$

$$l \leq x \leq u, \tag{22}$$

$$0 \leq x_{n+1} \leq 1, \tag{23}$$

où x_{n+1} est une variable artificielle et $\rho = b - Ax^+$, x^+ étant un n-vecteur choisi entre l et u. Remarquons que $y = \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} x^+ \\ 1 \end{pmatrix}$ est une solution réalisable évidente pour le problème auxiliaire. Appliquons alors l'algorithme de la méthode de support pour résoudre le problème auxiliaire, et ce, en commençant par la SRS initiale $\{y, J_B\}$, où $y = \begin{pmatrix} x^+ \\ 1 \end{pmatrix}$ et $J_B = \{j_1, j_2, \ldots, j_m\}$ un support trouvé avec l'une des trois approches sus-mentionnées. Soit $\{y^0, J_B^0\}$, avec $y^0 = \begin{pmatrix} x^0 \\ x_{n+1}^0 \end{pmatrix}$, la solution optimale obtenue.
Si $x_{n+1}^0 > 0$, alors le problème original (17)-(19) est irréalisable.
Sinon, deux cas peuvent se présenter :
1. Si $n+1 \notin J_B^0$, alors $\{x^0, J_B^0\}$ est une SRS pour le problème original (17)-(19).
2. Si $n+1 \in J_B^0$, alors on fait sortir l'indice artificiel $n+1$ de J_B^0 et on le fait remplacer par un indice approprié, et ce, selon la règle algébrique utilisée dans le simplexe. On obtient alors un nouveau support $\bar{J}_B^0$. Par conséquent, $\{x^0, \bar{J}_B^0\}$ est une SRS pour le problème original.

4 Résultats numériques

Afin d'effectuer une comparaison numérique entre les différentes méthodes que nous avons proposées et la méthode du simplexe, nous les avons programmées sous le langage de programmation Matlab 7.4.0 (R2007a).

Pour la conversion des problèmes-tests de la librairie Netlib [8,11] du format "mps" au format utilisable par Matlab ("mat"), nous avons utilisé le lecteur readmps de C. Keil [13]. Comme les problèmes existants dans Netlib présentent des bornes sur les variables qui peuvent être infinies, alors on a construit un échantillon de 32 problèmes ayant des bornes finies et une même valeur optimale de la fonction objectif que ceux de Netlib. Le principe de cette construction est comme suit : soit xl et xu les bornes obtenues après la conversion d'un problème donné du format "mps" au format "mat" et x^* la solution optimale que nous avons obtenue avec le solver lipsol de Matlab. Soit $xmin = \min\{x_j^*, j = \overline{1,p}\}$ et $xmax = \max\{x_j^*, j = \overline{1,p}\}$. Ainsi, la matrice des contraintes principales et le vecteur des coûts du problème construit restent les mêmes que ceux du problème

de Netlib et les nouvelles bornes inférieures, $\tilde{l}$, et supérieures, $\tilde{u}$, sont obtenues comme suit :

$$\tilde{l}_j = \begin{cases} xmin, \text{ si } xl_j = -\infty\,; \\ xl_j, \quad \text{si } xl_j > -\infty, \end{cases} \text{ et } \tilde{u}_j = \begin{cases} xmax, \text{ si } xu_j = +\infty\,; \\ xu_j, \quad \text{si } xu_j < +\infty. \end{cases}$$

Les colonnes 2, 3, 4 et 5 de la table 1 représentent le listing des caractéristiques de ces 32 problèmes, où **NC**, **NV**, **PCE** et **D** représentent respectivement le nombre de contraintes, le nombre de variables, le pourcentage des contraintes d'égalité (le rapport entre le nombre de contraintes d'égalité et le nombre total de contraintes, multiplié par 100) et la densité de la matrice des contraintes (le rapport entre le nombre d'éléments non nuls et le nombre total d'éléments de la matrice des contraintes, multiplié par 100).

Afin d'effectuer la comparaison, nous avons nommé les différentes méthodes comme suit :

Méth.1. SimplexPva : la méthode du simplexe à variables bornées avec plusieurs variables artificielles ;

Méth.2. SupportPva : la méthode de support à variables bornées avec plusieurs variables artificielles ;

Méth.3. Support1vaGauss : la méthode de support à variables bornées avec une seule variable artificielle là où le support initial est construit avec la première approche, i.e., la méthode de Gauss avec pivot partiel ;

Méth.4. Support1avFc1 : la méthode de support à variables bornées avec une seule variable artificielle là où le support initial est construit avec la seconde approche, i.e., le problème sous forme générale est transformé en un problème équivalent sous forme canonique en transformant les m_2 contraintes d'égalité en $m_2 + 1$ contraintes d'inégalité ;

Méth.5. Support1avFc2 : la méthode de support à variables bornées avec une seule variable artificielle là où le support initial est construit avec la troisième approche, i.e., le problème sous forme générale est transformé en un problème équivalent sous forme canonique en transformant les m_2 contraintes d'égalité en $2m_2$ contraintes d'inégalité.

Afin de calculer le vecteur des multiplicateurs de Lagrange ainsi que les composantes basiques de la direction de recherche, nous résolvons les systèmes d'équations : $A_B^T \pi = c_B$ et $A_B d_B = -a_{j_0} d_{j_0}$, et ce, en utilisant la décomposition LU de A_B. La mise à jour des facteurs L et U est faite, à chaque itération, en utilisant la formule de Sherman-Morrison-Woodbury [14].

Nous avons résolu les 32 problèmes listés dans la première colonne de la table 1 avec les méthodes 1, 2, 3, 4 et 5 sur un PC portable IBM Mobile Intel(R) Pentium (R) 4-M CPU 2.0 GHz, 512 MO de RAM, fonctionnant sous le système d'exploitation Windows XP SP2, en prenant les différentes tolérances comme suit : la tolérance d'optimalité : $OptimTol = 10^{-10}$; pour des raisons d'instabilité, la tolérance de pivotage, PivTol, est prise égale à 10^{-10} pour les problèmes stocfor1 et fit1p ; 10^{-9} pour le problème modszk1, 10^{-5} pour le problème bandm et 10^{-6} pour les autres problèmes ; la tolérance de suboptimalité : $SubOptimTol = 10^{-6}$. Le point initial x^+ est choisi comme suit : $x^+(J_B) = (l(J_B) + u(J_B))/2$ et $x^+(J_N) = l(J_N)$. Les résultats numériques

sont reportés dans les tables 1 et 2, où nit_1, nit et cpu, représentent respectivement le nombre d'itérations de la phase 1, le nombre d'itérations total et le temps moyen d'exécution en secondes de chaque méthode (on a effectué dix exécutions pour les problèmes "afiro" jusqu'à "ship04s" et trois exécutions pour les problèmes restants) ; cpu_1, cpu_2 et p, montrés dans les colonnes 4, 5 et 7 de la table 2, représentent respectivement le temps moyen nécessaire pour trouver le support initial, le temps moyen nécessaire pour avoir la solution optimale, et ce, sans inclure le temps cpu_1 ($cpu_2 = cpu - cpu_1$) et enfin le pourcentage du temps consommé par la recherche du support initial dans la méthode 3 ($p = cpu_1/cpu \times 100$).

Une lecture des tables 1 et 2 nous permet de constater que la méthode de support utilisant plusieurs variables artificielles (méthode 2) est la plus rapide des variantes de support. De plus, elle a un comportement voisin de celui de la méthode du simplexe pour la majorité des problèmes traités. La méthode utilisant l'algorithme de Gauss pour la recherche du support initial est la plus rapide des variantes de support utilisant une seule variable artificielle pour les problèmes de petite et moyenne dimension et la variante Support1avFc1 est la plus rapide pour les problèmes de grande dimension. Supposons maintenant que l'utilisateur dispose au préalable du support initial calculé avec l'algorithme de Gauss, alors en commençant de ce support initial, la variante Support1avGauss prendra les temps montrés dans la colonne cpu_2 de la table 2 (colonne 5) pour atteindre la solution optimale. L'examen de ces temps d'exécution montre qu'ils sont, pour la majorité des problèmes, nettement inférieurs à ceux de la méthode du simplexe si l'on dispose au préalable d'un bon support initial, ce qui est le cas dans certains problèmes de contrôle optimal [5], par exemple, où l'utilisateur peut utiliser son expérience pour choisir un bon support initial, la variante utilisant une seule variable artificielle est alors préférable à la méthode du simplexe utilisant plusieurs variables artificielles. Cependant, en absence d'information sur le support initial, la méthode du simplexe est la plus performante. En effet, en comparant les différentes variantes proposées de la méthode de support avec la méthode du simplexe avec plusieurs variables artificielles, on constate que la différence n'est pas grande pour les problèmes de petite dimension mais elle devient considérable pour les problèmes de taille moyenne et grande pour lesquels la méthode du simplexe prend un temps nettement inférieur. Néanmoins, pour certains problèmes (ceux qui ont des temps d'exécution écrits en gras), les différentes variantes de support sont meilleures que la méthode du simplexe. La lenteur de la méthode de support avec une seule variable artificielle s'explique par le fait que la majorité des problèmes à variables bornées traités ici ont un pourcentage considérable de contraintes d'égalité. De ce fait, la méthode Support1vaGauss perd beaucoup de temps dans la recherche du support initial pour les problèmes de moyenne et grande dimension (ce temps est reporté dans la colonne 4 de la table 2) et les variantes Support1avFc1 (méthode 4) et Support1avFc2 (méthode 5) sont lentes à cause de la transformation des contraintes d'égalité en contraintes d'inégalité, ce qui augmente la dimension du problème.

Tab. 1. Temps d'exécution et nombre d'itérations des méthodes 1 et 2

						Meth.1			Meth.2	
	NC	NV	PCE	$D(\%)$	nit_1	nit	cpu	nit_1	nit	cpu
afiro	27	32	29.63	9.61	10	17	0.10	10	17	**0.07**
kb2	43	41	37.21	16.2	74	121	0.29	43	135	0.53
sc50a	50	48	40	5.42	30	49	0.25	30	49	0.31
sc50b	50	48	40	4.92	40	53	0.26	40	53	0.34
adlittle	56	97	26.79	7.05	42	130	0.86	43	132	**0.83**
blend	74	83	58.11	7.99	65	126	0.79	65	128	**0.78**
share2b	96	79	13.54	9.15	154	193	1.28	115	162	**1.13**
sc105	105	103	42.86	2.59	76	123	0.96	68	112	**0.93**
stocfor1	117	111	53.85	3.44	88	106	0.73	90	108	0.94
israel	174	142	0	9.18	9	326	3.20	9	334	**2.62**
sc205	205	203	44.39	1.32	217	341	3.60	129	267	**2.31**
grow7	140	301	100	6.2	141	294	3.33	141	292	**3.09**
agg	488	163	7.377	3.03	77	119	3.07	75	126	4.90
bandm	305	472	100	1.73	1459	1723	34.52	2705	2924	69.71
sctap1	300	480	40	1.18	316	451	11.71	337	418	**11.07**
grow15	300	645	100	2.9	301	824	18.51	301	828	20.49
degen2	444	534	49.77	1.68	1423	2190	68.97	1418	2103	72.11
fit1d	24	1026	4.167	54.4	21	1379	51.77	21	1422	58.30
grow22	440	946	100	1.98	441	1464	50.39	441	1478	55.85
scsd6	147	1350	100	2.17	207	767	39.68	204	699	40.57
ship04s	402	1458	88.06	0.74	479	533	37.92	479	534	41.53
fit1p	627	1677	100	0.94	1531	2877	186.93	1639	3062	252.23
modszk1	687	1620	100	0.29	1103	2557	177.47	1001	2630	190.74
ship04l	402	2118	88.06	0.74	734	811	63.96	734	812	95.64
sctap2	1090	1880	43.12	0.33	924	1767	213.42	944	1849	254.64
scsd8	397	2750	100	0.79	547	1461	177.56	499	1278	198.04
ship08s	778	2387	89.72	0.38	822	956	149.64	823	959	**149.25**
ship12s	1151	2763	90.79	0.26	1344	1515	322.32	1347	1520	**313.98**
sctap3	1480	2480	41.89	0.24	1166	2171	482.76	1182	2448	503.16
stocfor2	2157	2031	52.99	0.19	2811	3400	704.43	4287	4900	947.45
ship08l	778	4283	89.72	0.38	1158	1414	493.60	1166	1424	514.18
ship12l	1151	5427	90.79	0.26	2643	2892	1726.58	2646	2895	1727.97

Tab. 2. Temps d'exécution et nombre d'itérations des méthodes 3, 4 et 5

			Meth. 3					Meth. 4			Meth. 5	
	nit1	nit	cpu1	cpu2	cpu	p(%)	nit1	nit	cpu	nit1	nit	cpu
afiro	21	24	0.01	0.16	0.18	7.99	19	30	0.20	11	21	0.15
kb2	21	81	0.04	0.47	0.51	8.05	3	136	0.96	3	120	0.96
sc50a	14	40	0.04	0.24	0.29	15.05	3	53	0.40	3	50	0.47
sc50b	3	35	0.04	0.25	0.29	14.55	3	55	0.43	3	57	0.53
adlittle	17	118	0.09	1.05	1.14	8.07	68	177	1.78	46	204	2.05
blend	76	111	0.47	0.82	1.29	36.45	3	145	1.49	3	264	2.51
share2b	116	137	0.07	1.23	1.29	5.18	117	156	1.78	129	178	2.01
sc105	27	76	0.32	0.84	1.17	27.79	3	115	1.55	3	117	1.86
stocfor1	59	85	0.31	0.42	0.73	41.89	78	110	**0.65**	58	138	1.66
israel	18	449	0.05	4.97	5.03	1.06	18	449	5.25	18	449	5.49
sc205	5	159	1.01	2.12	**3.13**	32.32	3	256	4.07	3	282	5.25
grow7	16	223	4.33	2.28	6.61	65.54	3	304	5.29	3	319	7.31
agg	75	118	0.31	2.56	**2.87**	10.8	118	167	4.06	117	171	4.37
bandm	131	394	23.96	5.70	**29.65**	80.79	585	893	**24.46**	495	944	**18.7**
sctap1	195	277	2.84	6.37	**9.21**	30.84	348	504	14.56	311	418	13.87
grow15	16	609	31.23	11.95	43.18	72.32	3	811	28.24	3	879	41.22
degen2	511	1092	14.91	28.64	**43.55**	34.24	1020	1938	77.00	1020	3530	150.36
fit1d	3	1372	0.10	52.75	52.85	0.18	3	1309	53.95	3	1297	54.35
grow22	11	949	83.40	27.53	110.92	75.18	3	1344	76.46	3	1418	104.05
scsd6	176	747	15.06	37.25	52.31	28.79	250	803	52.23	238	1567	113.39
ship04s	177	313	54.31	17.68	71.99	75.44	326	480	43.38	300	479	57.59
fit1p	197	1093	1566.82	49.48	1616.30	96.94	1663	2404	295.93	1417	2198	301.47
modszk1	930	2365	276.69	150.75	427.44	64.73	1100	2616	337.76	354	7095	893.52
ship04l	296	498	60.74	39.63	100.37	60.51	377	629	89.02	376	614	102.68
sctap2	210	796	158.72	88.18	246.91	64.28	699	1322	246.75	686	1353	232.71
scsd8	383	1484	160.70	211.56	372.26	43.17	486	2452	473.71	593	5127	1053.94
ship08s	297	499	353.25	54.80	408.05	86.57	780	1065	194.42	528	798	196.91
ship12s	696	1013	936.21	126.15	1062.36	88.13	957	1306	**320.28**	833	1246	404.45
sctap3	468	1303	326.17	214.96	541.12	60.28	1083	1959	576.34	954	1953	505.93
stocfor2	815	1801	2373.30	200.49	2573.78	92.21	973	2056	**649.80**	977	2968	785.91
ship08l	443	812	671.67	246.37	918.04	73.16	1019	1485	594.14	841	1211	585.88
ship12l	1686	2291	1859.71	1034.17	2893.88	64.26	1142	1778	**1158.73**	1051	1821	**1407.42**

5 Conclusion

Dans ce travail, nous avons implémenté et comparé plusieurs variantes de la méthode de support à variables bornées avec la méthode du simplexe. Les résultats numériques montrent que la méthode du simplexe demeure la plus rapide pour la résolution des problèmes de programmation linéaire à matrices creuses dont le nombre de contraintes d'égalité est assez important. Toutefois, lors de l'étude expérimentale que nous avons effectuée, nous avons constaté que la variation du point initial x^+ ainsi que du support initial J_B influe sur le temps d'exécution de la variante de support utilisant une seule variable artificielle. Par conséquent, une question se pose ici : existe t-il un choix judicieux du point initial et du support initial qui améliore au mieux le temps d'exécution de la méthode de support à variables bornées avec une seule variable artificielle ?

Références

1. Bentobache, M. : Nouvelle méthode pour la résolution des problèmes de programmation linéaire sous forme canonique et à variables bornées. Mémoire de magister, Université de Béjaia, (2005).
2. Bentobache, M. et Bibi, M.O. : Méthode de support à deux phases pour la résolution des problèmes de programmation linéaire à variables simples : Comparaison numérique, Actes du colloque COSI'08, Tizi Ouzou, Juin (2008) 314–325.
3. Dantzig, G. B. : Linear Programming and Extensions. Princeton University Press, Princeton, N.J., (1963).
4. Gabasov, R. and Kirillova, F. M. : Méthodes de programmation linéaire, volumes 1, 2 et 3. Edition de l'Université, Minsk (en russe), (1977, 1978 et 1980).
5. Gabasov, R., Kirillova, F.M. and Prischepova, S.V. : Optimal Feedback Control, Springer-Verlag, London, (1995).
6. Morgan, S.S. : A Comparison of Simplex Method Algorithms. Master thesis. University of Florida, (1997).
7. Vanderbei, R.J. : Linear Programming : Foundations and Extensions. Kluwer academic publichers, Princeton University, (2001).
8. Lustig, I. : An Analysis of an Available Set of Linear Programming Test Problems. Technical Report SOL 87-11. Systems Optimization Lab. Stanford University, Stanford, CA. (1987) 1–53.
9. Gass, S.I. : Linear Programming : Methods and Applications. McGraw-Hill Book Company, Inc., New York, (1964).
10. Bixby, R.E. : Implementing the Simplex Method : The Initial Basis. ORSA Journal on Computing, Vol. 4, No. 3. (1992) 1–18.
11. Netlib test problems. http ://www.netlib.org/lp.
12. Millham, C.B. : Fast Feasibility Methods For Linear Programming. Opsearch. Vol. 13, (1976) 198–204.
13. Keil, C. : Readmps Software. http ://www.ti3.tu-harburg.de/˜keil/
14. Ferris, M.C., Mangasarian, O.L. and Wright, S.J. : Linear Programming with MATLAB, MPS-SIAM Series on Optimization, (2007).

Modèle hybride spatio-temporel d'analyse d'images satellitales à base de réseaux Bayésiens dynamiques

Houcine Essid, Imed Riadh Farah et Henda Ben Ghzala
Laboratoire de Recherche en Informatique Arabisée et Documentique Intégrée (R.I.A.D.I), Ecole Nationale des sciences de l'informatique. Campus Universitaire de Manouba, 2010 Manouba, Tunis, Tunisie Tel : 216 71 600 444, Télécopieur : 216 71 600 449
e-mail : ehoucine@yahoo.fr
e-mail: riadh.farah@ensi.rnu.tn
e-mail: henda.benghzala@cck.rnu.tn

Vincent Barra
ISIMA - LIMOS - UMR CNRS 6158 - Campus Scientifique des Cézeaux Office B139 63177 AUBIERE CEDEX, France
Tel. : 33 4 73 40 74 92 Fax : 33 4 73 40 50 01
e-mail : vincent.barra@isima.fr

Résumé

L'interprétation automatique des images de télédétection est un problème encore difficile qui ne peut être simplement résolu par la seule performance des machines. Si de nombreux outils de traitements d'images ont simplifié l'extraction d'informations, le photo-interprète reste encore, par son expérience des images et du type d'application, seul juge pour l'enchaînement des traitements et l'évaluation de la qualité des résultats. Le travail d'interprétation se complexifie lors de l'analyse des changements à partir d'une série d'images espacées dans le temps, puisqu'il est nécessaire de tenir compte de la nature du changement et de la résolution des contraintes inhérentes aux images et nécessaire à la comparaison de celles-ci. Les réseaux Bayésiens sont actuellement une des techniques les plus intéressantes de l'intelligence artificielle. Ils proposent un formalisme puissant et intuitif pour représenter des informations incertaines et imprécises. Aujourd'hui, la contribution de ces réseaux ne réside plus seulement dans l'intégration de connaissances contextuelles dans la mesure où elle commence à apporter des solutions aux problèmes d'évolution dans le temps et l'espace. L'objectif de notre travail est de proposer une nouvelle approche d'interprétation pour l'analyse de la dynamique d'objets de scènes d'images satellitaires à base de chaînes de Markov cachées représentées par des réseaux Bayésiens dynamiques qui conduisent à des algorithmes rapides d'inférence et d'apprentissage. Nous avons validé notre approche sur des images landsat relatives au sud de la Tunisie.

Mots clés : Rréseaux Bayésiens, Chaînes de Markov Cachées, analyse spatio-temporelle, interprétation d'images satellitaires.

1. Introduction

De nos jours, beaucoup d'intérêt est donné a l'analyse d'images satellitaires qui est appliqué dans l'aménagement des surfaces urbaines ou rurales et de protection de l'environnement. Il constitue une technique d'observation moins coûteuse que des enquêtes de terrain et peut être utilisée aussi bien par des régions technologiquement développées, que par des régions défavorisées.

Les images présentent une grande quantité de données à traiter afin d'en extraire de l'information. Il serait donc intéressant d'avoir un (ou plusieurs) modèle(s) pour analyser et interpréter les images satellitaires même si elles sont entachées d'incertitude. Cette incertitude est d'origine multiple: possibilité d'erreur dans les données, ambiguïté de la représentation de l'information, incertitude sur les relations entre les diverses informations. Dans la littérature du traitement d'images, différentes méthodes ont été proposées et développées en se basant sur les réseaux Bayésiens. Ces études ont montré que les systèmes élaborés sous le modèle Bayésien pouvaient être efficaces mais présentent toujours certains inconvénients. Dans cet article, nous présentons une étude comparative de ces méthodes et ce afin de proposer une nouvelle approche hybride dynamique pour l'analyse d'images satellitaires.

Le présent article est organisé en trois sections. La première est consacrée aux modèles graphiques probabilistes basés sur les réseaux Bayésiens. Dans la deuxième nous présentons les différentes études de traitement d'images basés sur les réseaux Bayésiens et ses dérivés. On termine par une troisième section dans lequel nous proposons un modèle hybride dynamique pour l'analyse d'images satellitaires.

2. Les réseaux Bayésiens

Les réseaux Bayésiens [29] ainsi que leurs méthodes d'inférence ont été introduits dans [2, 3]. Brièvement, le formalisme des réseaux Bayésiens consiste à associer un graphe acyclique orienté à une distribution jointe de probabilités P(X) d'un ensemble de variables aléatoires $X= \{X_1, \ldots, X_n\}$. Les nœuds de ce graphique représentent les variables aléatoires, et les flèches codent les indépendances conditionnelles qui sont supposées dans la distribution jointe de probabilités. L'ensemble de toutes les relations d'indépendances conditionnelles se nomme les propriétés de Markov qui impliquent que, sachant ses parents, une variable est indépendante de toutes les autres variables du réseau à l'exception de ses descendants. Un réseau Bayésien est complètement défini par une structure de graphe S et un jeu de paramètres θ de probabilités conditionnelles des variables étant donné leurs parents. En effet, la distribution jointe de probabilités peut être exprimée sous une forme factorisée qui est:

$$P(X) = \prod_{i=1}^{n} P(Xi/\pi i),$$

où π_i dénote les parents de X_i dans S. Par ailleurs, il existe plusieurs déclinaisons des réseaux bayésiens [1] :

- Les réseaux Bayésiens multi-agents [4] sont utiles lorsque les informations ne sont disponibles que localement, et que pour diverses raisons (de confidentialité par exemple), les différents agents ne veulent pas partager leurs informations. Dans ce cas, il leur sera tout de même possible d'utiliser leurs informations

respectives sans pour autant les divulguer. Le résultat d'une inférence dans ce réseau particulier satisfera alors les différents partis.

- Les réseaux Bayésiens de niveau deux [14] sont utiles pour avoir une visualisation plus concise et donc plus lisible des relations de dépendance entre les attributs.
- Les réseaux Bayésiens orientés objets [5] sont utiles lorsqu'une sous structure est répétée à plusieurs endroits du réseau global, cela permet d'avoir une représentation plus économique et plus lisible, en particulier si le réseau global est constitué d'une répétition d'une sous structure particulière.
- Les diagrammes d'influence [6] sont une extension des réseaux Bayésiens qui introduit des nœuds de nouvelle nature liés à la problématique de l'aide à la décision.
- Les réseaux Bayésiens dynamiques
  [8] sont utiles pour modéliser des phénomènes dynamiques ou temporels, en utilisant un temps discret.
- Les réseaux Bayésiens continus [11] sont utiles pour modéliser des phénomènes temporels lorsque le temps est continu.
- Les filtres Bayésiens sont des réseaux Bayésiens dynamiques particuliers, ils ne possèdent qu'une variable d'état et une variable d'observation. Leurs variantes appelées les filtres de Kalman [9] ont été très utilisées pour des problématiques où le nombre d'états n'était pas discret.
- Les processus de décision markoviens [10] sont une extension des modèles dynamiques et des diagrammes d'influence, ils permettent de traiter des problèmes liés à la décision tout en prenant en compte un aspect temporel.
- Les réseaux Bayésiens causaux [13] également appelés les modèles markoviens, sont utiles si l'on veut s'assurer que toutes les dépendances codées ont une signification réelle. Ils ne doivent donc pas être utilisés comme de simples outils de calcul comme il serait possible de le faire avec un réseau Bayésien classique, il faut alors s'appuyer sur leur pouvoir expressif.
- Les réseaux Bayésiens multi-entités[12] sont une extension des réseaux Bayésiens et de la logique Bayésienne du premier ordre qui permet d'utiliser plusieurs 'petits' réseaux Bayésiens pour modéliser un système complexe. Ils contiennent des variables contextuelles, qui peuvent être de différentes natures, notamment être liés à des notions de décision.

3. Synthèse des travaux d'analyse d'images satellitaires par les réseaux Bayésiens

3-1. Introduction

L'analyse d'images satellitaires a fait l'objet de nombreuses recherches. Depuis 1984 plusieurs systèmes ont été développés en utilisant différentes architectures de réseaux

Bayésiens pour différentes applications sans tenir compte de l'aspect temporel et dynamique des images satellitaires. L'idée principale de notre méthode est d'utiliser une approche hiérarchique et dynamique et ce afin de prendre en considération la composante temps et la dynamique de la scène.

3-2. Synthèse des travaux

Sans vouloir être exhaustive, nous donnons ci-après les principales références qui ont servi à notre analyse et ce depuis 1996. Chacun d'eux est présenté suivant trois points de vue:

- Architecture de réseaux Bayésiens et dérivés
- Application dans le domaine de l'analyse d'images satellitaires: Interprétation, segmentation, classification, fusion, déconvolution.
- Prise en compte de l'aspect spatio-temporelle

a- Le système KUMAR : développé pour l'interprétation d'images. C'est à dire le processus de reconnaissance et de détection d'objets en vue de donner un sens à l'image. Ce système utilise un réseau Bayésien de niveau 2 [14].

b- Le système AMIT : conçu pour ressortir les régions intéressantes d'une image et fusionner des données issues de plusieurs capteurs. Ce système utilise les réseaux Bayésiens hiérarchiques [15].

c- Le système ANDRÉ: Il a été dédié à la déconvolution d'images satellitaires et aériennes. Ce système permet d'estimer les paramètres décrivant les propriétés de l'image que l'on cherche à reconstruire. Il se base sur un modèle Bayésien hiérarchique permettant de tenir compte des caractéristiques des images étudiées [16].

d- Le système JESSE: L'un des premiers systèmes qui à utilise les réseaux Bayésiens dans l'extraction de données d'une base d'images. La gestion de l'incertitude est base sur les croyances [30].

e- Le système ASCENDER II: L'utilisation des réseaux Bayésiens hiérarchique combinée avec la théorie de l'utilité permet de manipuler les informations à partir de plusieurs images 3D ayant différentes caractéristiques. [17].

f- Le système PIECZYNSKI: Ce système est dédié à la segmentation dans le traitement statistique des images par les méthodes Bayésiennes et les modèles de Markov [18].

g- Le système LAI: Utilisé pour estimer une variable écologique (l'index des zones abandonnées). Ce système est une excellente application des réseaux Bayésiens [19].

h- Le système IHBN: Il permet l'extraction de données d'une base d'images. Ce système utilise l'architecture Bayésienne combinée avec les arbres de décision pour l'apprentissage incrémental [20].

i- Le système BOUBCHIR : Il est conçu pour récupérer une image de bonne qualité, proche de l'image originale recueillie en sortie de tout capteur. Ce système utilise

l'estimation statistique Bayésienne dans le domaine des transformées multi-échelles parcimonieuses orientées et non orientées comme solution au problème de débruitage [21].

j- Le système TANAKA: Ce système utilise les réseaux Bayésiens et la propagation des croyances dans le traitement probabiliste des images. [22].

k- Le Système TIMO: Il permet l'extraction de données d'une base d'images. Ce système utilise le réseau Bayésien hiérarchique non supervisé et ce en se basant sur les caractéristiques des régions et du contexte. [23].

l- Le système KIN-MAN : Il présente une technique interactive d'extraction d'images en utilisant un apprentissage semi supervisé. Ce système utilise les réseaux Bayésiens continues. [24].

m- Le système CIRO: C'est un système qui combine les classificateurs en utilisant le réseau Bayésien pour atteindre une classification finale. Il est appliqué aux bases de données d'images [25].

On Remarque que la plupart des systèmes existant ont traités la composante extraction d'objets et peu d'entre eux qui ont résolu le problème de l'interprétation d'images. Parmi eux on peut citer le système Kumar [14] qui a utilisé l'approche Bayésienne simple sans tenir compte de l'aspect temporal et dynamique des images satellitaires. De même que pour la majorité des systèmes analysés. Pour remédier à cette insuffisance, nous contribuant par la proposition d'une approche qui utilise le réseaux Bayésien dynamiques comme représentation des modèles de Markov cachés hiérarchiques [8] pour l'interprétation des images satellitaires et ce en tenant compte de la composante temporelle.

4. Proposition d'un modèle hybride: Modèles de Markov Cachés Hiérarchiques représentés par Réseau Bayésien Dynamique

4-1. Réseaux Bayésiens Dynamiques

Certains système actuels d'analyse et d'interprétation d'images utilise une modélisation probabiliste par des modèles de Markov cachés (MMC) car ils savant bien s'adapter à la variabilité des observations qui sont supposées être gouvernées par un processus dynamique caché. Les hypothèses d'indépendance associées sont telles que le processus caché est markovien du premier ordre et chaque observation dépend seulement de la variable caché actuelle. Il y a cependant une question fondamentale concernant ces hypothèses de dépendance: sont-elles consistantes avec les données et avec tout type d'application ?

Pour remédier, on propose une approche flexible basée sur le formalisme des réseaux Bayésiens dynamiques (RBD) qui codent la distribution jointe de probabilités d'un ensemble de variables évoluant dans le temps. Les RBD sont une extension des réseaux Bayésiens classiques qui permet de représenter l'évolution temporelle des variables. Si on considère un ensemble $X[t] = \{X_1[t], \ldots, X_N[t]\}$ de variables évoluant

dans le temps, un RBD représente la distribution de probabilité jointe de ces variables pour un intervalle borné [0, T]. Cette probabilité est donnée par :

$$P(X[1], \ldots, X[T]) = \prod_{t=1}^{T} \prod_{t=1}^{n} P(X_i[t]/\pi_{it})$$

où π_{it} dénote les parents de $X_i[t]$.

Dans notre nouvelle approche, nous modélisons les dépendances entre les objets en créant des “interactions” entre les différents MMC correspondants aux différents niveaux de la hiérarchie. Pour cela, nous utilisons le formalisme des réseaux Bayésiens (RB) qui constitue un cadre idéal pour deux raisons majeurs. D’une part, grâce à leur structure graphique, les RB offrent un outil naturel pour représenter les dépendances entre les différentes variables d’un système donné. D’autre part, en exploitant les indépendances conditionnelles entre les variables, ils introduisent une certaine "modularité" dans les systèmes complexes. Ainsi, non seulement les RB fournissent un outil séduisant pour modéliser des systèmes complexes, mais aussi conduisent à des algorithmes rapides d’inférence et d’apprentissage.

4-2. Proposition d'un modèle hybride

Dans notre nouvelle approche, nous modélisons les dépendances entre les objets en créant des “interactions” entre les différents MMC correspondants aux différents niveaux de la hiérarchie. Pour cela, nous utilisons le formalisme des réseaux Bayésiens (RB) qui constitue un cadre idéal pour deux raisons majeures. D’une part, grâce à leur structure graphique, les RB offrent un outil naturel pour représenter les dépendances entre les différentes variables d’un système donné. D’autre part, en exploitant les indépendances conditionnelles entre les variables, ils introduisent une certaine "modularité" dans les systèmes complexes. Ainsi, non seulement les RB fournissent un outil séduisant pour modéliser des systèmes complexes, mais aussi conduisent à des algorithmes rapides d’inférence et d’apprentissage. La figure 1 représente la forme générale d'un MMC représenté par un RBD (Chaque noeud en cette structure représente une variable aléatoire $X_h[t]$ ou $X_0[t]$ dont la valeur indique l’état ou l’observation). La figure 2 donne un exemple avec T=4.

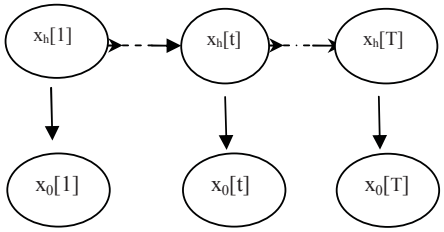


Figure 1

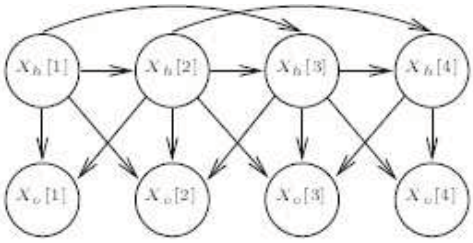


Figure 2

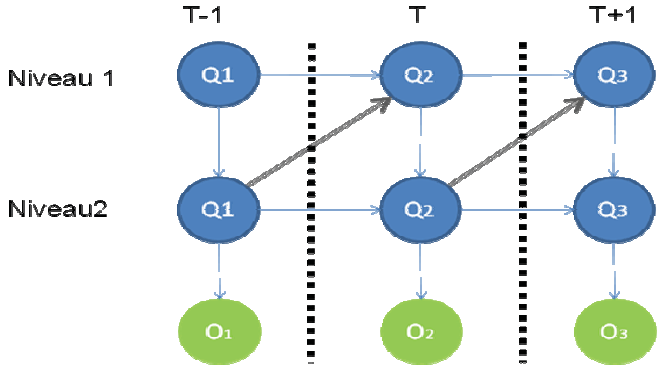


Figure 3: Un MMC hiérarchique de deux niveaux est représenté comme un RB dynamique

Dans cet article, étant donné une image satellitaire prise en deux temps différents t_1 et t_2, on se propose de les analyser et les interpréter par des MMC hiérarchiques représentée par des RBD. Mais, pour alimenter le réseau, il faut tout d'abord passer par la phase de segmentation qui extrait les régions d'intérêt dans l'image, et ce en utilisant la technique du FCM [28] pour obtenir des observations (O_1, …, O_n). Il n'y a pas de méthode universelle de segmentation qui peut être appliquée à tous les types d'images.
Dans ce travail, on a choisit d'appliquer la méthode fuzzy c-means (FCM). C'est une méthode adequate pour le cas des images satellitaires entachées d'imprécision. FCM est un algorithme non supervise qui permet d'affecter chaque point à une classe particulière avec un certain degré d'appartenance. Le nombre de classes étant fixé sans l'intervention de l'homme. Puis, le système détermine les dépendances et les paramètres qui représentent au mieux les données. Le principe de la représentation se fonde sur la prétention de stationnarité qui implique que la structure et les paramètres du RBD se répètent. La distribution de probabilité jointe est codée en utilisant un premier réseau et un réseau déroulé de transition. Cette distribution est obtenue en déroulant le réseau de transition pour un nombre suffisant de tranches de temps [27]. Le réseau initial indique les états initiaux de la distribution X[1]. Le réseau de

transition code la probabilité temps invariable de transition $P(X[t+1]|X[t])$. Le JPD pour un intervalle fini de temps est obtenu en déroulant le réseau de transition pour un nombre suffisant de tranches de temps.

L'exemple de la figure 3 représente un RBD de 2 niveaux, O1, O2 et O3 sont les observations à l'instant t-1, t et t+1. Les nœuds du réseau Bayésien sont regroupés par « tranches » temporelles, les variables Q2 d'une tranche t ne dépendent que d'autres variables de la même tranche issues d'un niveau supérieur ou de variables de la tranche précédente t−1 du même niveau ou du niveau inférieur.

5. Validation

Étant donné une image satellitaire composée de plusieurs objets. L'idée principale de notre approche est la suivante: pour chaque objet de l'image, au lieu de considérer un HMM indépendant dans chaque niveau, nous construisons un RBD plus complexe et uniforme en ajoutant des liens (orientés) entre les variables pour capturer les dépendances entre les différents niveaux. Une question naturelle est: quels sont les liens à ajouter? Probablement, la meilleure solution serait d'apprendre la structure graphique (les dépendances) à partir des données. Cependant, cette stratégie, appelée apprentissage structurelle, qui est extrêmement intéressante et que nous sommes en train d'étudier, n'est pas l'objectif de ce papier. Notre but ici est (d'abord) de fixer une structure graphique "raisonnable" puis évaluer si notre nouvelle approche hiérarchique est prometteuse. Contrairement à un HMM classique, notre modèle hiérarchique fournit une modélisation de la dynamique de l'image. A la différence d'une hiérarchie de réseaux Bayésiens classiques, notre RBD permet une interaction entre les niveaux. De plus, notre modèle utilise les informations contenues dans tous les nivaux. Dans le même domaine, un système hiérarchique fondé sur les réseaux bayésiens Classiques a été proposé dans [15] et [17]. Mais ces approches, contrairement à la notre, ne permettent ni une inférence exacte ni rapide et ne tiennent pas compte de l'aspect temporel. Par contre, notre approche est basée sur les RBD et la composante dynamique est prise en considération. Le problème d'inférence dans les RBD peut être résolu par un réseau Bayésien statique énorme et puis d'employer les algorithmes statique énorme et puis d'employer les algorithmes généraux d'inférence. Mais, cette approche augmente la complexité informatique en termes de mémoire. Par conséquent, l'inférence de RBD est exécutée en utilisant les opérateurs récursifs [26] qui mettent à jour l'état de la croyance du réseau pendant que les nouvelles observations deviennent disponibles. Afin de valider notre approche, on va considérer l'exemple suivant se rapportant à la zone de Matmata du sud de la Tunisie. On peut facilement tracer la matrice stochastique de transition. Prenons l'image de la région de Matmata en Tunisie de l'année 2000 et celle de l'année 1995 (figures 4 et 5). La segmentation par FCM [28] transforme l'image originale en une image classifiée, qui est comparée avec l'image " vérité terrain".

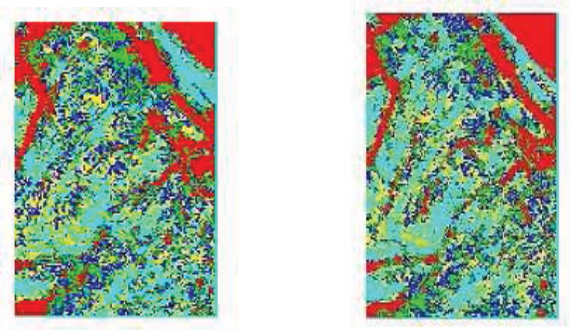


Figure 4 Les images de Matmata en 1995 et 2000

Légende

Classe 1	
Classe 2	
Classe 3	
Classe 4	
Classe 5	

D'où on peut déduire la matrice stochastique de transition suivante et ce par superposition et calcul de différence. Par exemple, 65,271% des pixels de la classe1 reste dans la classe 1 entre1995 et 2000, 17,829% ont migré vers la classe 2, 7,281 vers la classe 3, 3,158% à la classe 4 et 4,079 à la classe 5.

	Classe 1	**Classe 2**	**Classe 3**	**Classe 4**	**Classe 5**
Classe 1	65.271	17.829	7.281	3.158	4.079
Classe 2	19.434	32.970	18.850	8.524	3.725
Classe 3	11.293	29.856	33.195	22.945	10.947
Classe 4	3.128	14.742	28.149	34.570	23.434
Classe 5	0.874	4.604	12.525	30.803	57.815

L'avantage de la représentation d'un modèle de Markov caché hiérarchique par un RBD est la possibilité d'utiliser la procédure générique d'inférence et d'apprentissage de ces réseaux(chose que nous sommes en train d'étudier). De Plus, le temps de l'inférence dans un RBD est largement inférieur à celui dans un modèle de Markov caché.

6. Conclusions et perspectives

Dans cet article, nous avons présenté l'état de l'art des systèmes d'analyse et de traitement d'images satellitaires par les réseaux Bayésiens. Nous avons proposé une nouvelle approche basée sur les réseaux Bayésiens dynamiques et les modèles de Markov cachés hiérarchiques pour l'interprétation des images satellitaires. Comme perspectives, nous allons réaliser cette approche pour montrer son efficacité et la comparer avec les réalisations existantes. De plus, on sait que lorsqu'une sous structure (avec ses paramètres) apparaît de manière répétée dans un réseau Bayésien, Il est possible de représenter ce dernier à l'aide d'un réseau Bayésien dit orienté objet. Ces modèles sont particulièrement bien adaptés pour représenter les réseaux Bayésiens dynamiques. On peut envisager aussi d'introduire des modèles appelés les réseaux Bayésiens partiellement dynamiques, qui, comme leur nom l'indique, peuvent contenir à la fois des nœuds statiques et des nœuds dynamiques. Pour montrer la performance du modele proposé, on va faire d'autres test en prenant d'autres images et comparer les resultats obtenus aux resultas des techniques proposées en littérature.

7. Bibliographie

1. Olivier, F. De l'identification de structure de réseaux Bayésiens à la reconnaissance de formes à partir d'informations complète ou incomplètes. Thèse, Institut National des Sciences Appliquées de Rouen Novembre 2006.
2. Pearl, J. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann 1988.
3. Jensen, F. An introduction to Bayesian Networks. Taylor and Francis, London, United Kingdom. 13
4. Maes, S., Meganck, S. et Manderick, B. Identification of causal effects in multi-agent causal models. Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2005), (pp. 178–182)., Innsbruck.
5. Koller, D. et Pfeffer, A. Oject-oriented Bayesian network. Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), (pp. 302–313).
6. Howard, R. et Matheson, J. Influence diagrams. Howard, R. and Matheson, J., editors,Readings on the Principles and Applications of Decision Analysis, volume Il, 721–762.
7. Jensen, F. Bayesian Networks and Decision Graphs. Springer Verlag series : Statistics for Engineering and Information Science, ISBN : 0-387-95259-4.
8. Murphy, K. Dynamic Bayesian Networks : Representation, Inference and Learning. PhD thesis, University of california, Berkeley
9. Anderson, B. et Moore, J. Optimal Filtering. Prentice-Hall.
10. Kaelbling, L., Littman, M., et Cassandra, A. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101, 99–134.
11. Nodelman, U. et Horvitz, E. Ctbns for inferring users' presence and activities with extentions for modeling and evaluation. Technical Report MSR-TR-2003-97, Microsoft Research
12. Laskey, K. First-order Bayesian logic. George Mason University Department of Systems Engineering and Operations Research. http://ite.gmu.edu/˜klaskey/papers/La skey FOBL.pdf
13. Pearl, J. Causality : Models, Reasoning, and Inference. Cambridge, England : Cambridge University Press, ISBN : 0-521-77362-8.
14. Kumar, V.P. et Desai, U.B.Image Interpretation Using Bayesian Networks. IEEE Transactions On Patern Analysis And Machine Intelligence, Vol. 18, No. 1, Janvier 1996
15. Singhalt, A. , LuoS, J. et Brownt, C. A Multilevel Bayesian Network Approach to Image Sensor Fusion. Department of Computer Science Imaging Science Division University of Rochester Rochester, NY 14627
16. Jalobeanu, A. Modèles, estimation Bayésienne et algorithmes pour la déconvolution d'images satellitaires et aériennes. Thèse à l'Université de Nice-Sophia Antipolis.
17. Marengoni, M. , Han son, A. , Zilberstein, S. et Riseman, E. Decision Making and Uncertainty Management in a 3D Reconstruction System. IEEE Transactions On Pattern Analysis And

Machine Intelligence, Vol. 25, No. 7, Juillet 2003.
18. Pieczynski, W. Modèles de Markov en traitement d'images. Traitement du Signal 2003 – Volume 20 n° 3 – Spécial 2003.
19. Kalácska, M. , Sánchez-Azofeifa, G.A. , Caelli, T. , Rivard, B. et Boerlage, B. Estimating Lai From Satellite Imagery. IEEE Transactions On Geoscience And Remote Sensing, Vol. 43, No. 8, Août 2005.
20. Baice, L. et Senmiao, Y. Incremental Hybrid Bayesian Network in Content-Based Image Retrieval. IEEE, CCECE/CCGEI, Saskatoon, Mai 2005.
21. Boubchir L. Approches bayésiennes pour le débruitage des images dans le domaine des transformées multi-échelles parcimonieuses orientées et non orientées. Thèse Université De Caen/Basse-Normandie.
22. Tanaka, K. Bayesian Network and Probabilistic Image Processing —Statistical Aspect of Belief Propagation Method. Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan.
23. Li, Y. et Bretschneider, T.R. Semantic-Sensitive Satellite Image Retrieval. IEEE Transactions On Geoscience And Remote Sensing, Vol. 45, No. 4, Avril 2007.
24. Yang, M. , Guan, J. Qiu, G. et Lam, K. Semi-supervised Learning based on Bayesian Networks and Optimization for Interactive Image Retrieval. School of Computer Science and Information Technology, The University of Nottingham, Nottigham, NG8 1BB, UK.
25. De Stefano, C., D'Elia, C. , Marcelli, A. et Scotto di Freca, A. Using Bayesian Network for combining classifiers. 14th International Conference on Image Analysis and Processing (ICIAP 2007.
26. N. Friedman, Murphy, and S. Russel. Learning structure of dynamic probabilistic networks. In UAI'98, Madison, Wisconsin, 1998.
27. Tlemsani R. et Benyattou A., Application des réseaux bayésiens dynamiques à la reconnaissance en-ligne des caractères isolés. 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March 25-29, 2007 – TUNISIA, Setit 2007.
28. I. R. Farah, W. Boulila, K. Saheb Ettabaa, B. Solaiman, M. Ben Ahmed, "Interpretation of multi-sensor remote sensing images: Multi-approach fusion of uncertain information", IEEE Transactions on Geoscience and Remote Sensing, 2008
29. P. Naim, P. H. Wuillemin, Ph. Leray, O. Pourret et A. Becker, « Réseaux Bayésiens ». 3ème édition, Eyrolles. 2008.
30. J. Xin, et S.J. Jesse, «Relevance feedback for content-based image retrieval using Bayesian network". ACM International Conference Proceeding Series; Vol. 100. Proceedings of the

An Integer Monotone Optimization Method : Application to the Maximum Probability Problem

Fatima Bellahcene*, Ouiza Bouarab*, Faroudja Aumorassi*

*Département de Mathématiques, Faculté des sciences, Université Mouloud Mammeri, Tizi-Ouzou, Algérie.

E-mail: f_bellahcene@yahoo.fr, ouizabouarab@yahoo.fr, harasbit@yahoo.fr

Abstract : In this paper, we designed a novel method for solving nonlinear monotone integer optimization problems. The method is of branch-and-bound framework. The algorithm proceeds successively by refining the partition and removing integer boxes that do not contain an optimal solution, and finally terminates at an optimal solution in a finite number of iterations. We apply this method to solve the maximum probability problem which attempts to model uncertainty in the objective function by assuming that the input is specified in terms of a probability distribution, rather than by deterministic data given in advance. An illustrative example is given to clarify the theory discussed.

Keywords: Monotone optimization; Stochastic programming; Domain cut; Branch-and-bound.

1 Introduction

The rapid growth of global optimization is observed in recent years for problems arising in science and engineering. Such problems include finance, operations research, network and transportation problems, nuclear and mechanical design, chemical engineering design and control. Different algorithms are designed for solving various classes of global optimization problems when the decision variables are positive. Within each of these classes, different algorithms make assumption about the problem structure.

Mathematical programs with monotonic properties have been studied in the literature. Li and al. [5] proposed convexification and concavification schemes that transform a monotone function into either a convex or a concave function.

Tuy [14] developed a general framework for solving bounded monotonic programming problems to global optimality. The algorithm is based upon successively outer approximating the feasible region of a monotonic programming problem using a nested sequence of polyblocks, or unions of hyper-rectangles, and exploiting the fact that the minima of a non-decreasing function over a polyblock is at an extreme point of the polyblock. While the convexification and concavification schemes [5] require continuity and twice differentiable properties of the function, the Polyblock algorithm does not require any other properties

except monotonicity of functions. The efficiency of the Polyblock algorithm has been demonstrated in various applications such as linear and polynomial fractional programming (see for example [8]). SUN and al., [12] incorporate three basic strategies: partition, convexification and local search into the branch-and-bound framework in order to design an efficient algorithm for monotone optimization problems.

In many situations, however, fractional values of the variables are not physically meaningful. Therefore, modeling with nonlinear integer programs and the development of solution algorithms for such problems are of great interest to management scientists. Recently, Li and al. [7] developed a convergent Lagrangian dual and domain cut algorithm. The key issue in implementing his idea is to partition the non-rectangular domain, such that the lagrangian relaxation on the revised domain can be decomposed into others boxes. This algorithm is extended to solve multi-dimensional nonlinear knapsack problems with some modifications in the domain cut process and in the computation of the dual value on the integer sub-boxes [4]. A more efficient algorithm which combines the domain cut idea with a linear approximation method is presented in [6] to deal with the special structure of concave nonlinear knapsack problems. Mixed integer nonlinear programs are discussed in [13].

The aim of this paper is to present a novel solution method for monotone optimization problems that performs faster than the methods developed in [5] and [6]. We apply this method to solve the special stochastic integer problem namely the "maximum probability problem". For surveys, specifically on stochastic integer programming, we refer to the survey papers Klein Haneveld [3], Van der Vlerk [15] and Stougie and Van der Vlerk [11].

First, in Section 2, we review some basic concepts and results of monotonic optimization. In Section 3, a solution approach based on the domain partition scheme is outlined. In section 4, we state the maximum probability problem and reformulate it as a monotonic optimization problem. Finally, the paper closes with a numerical example illustrating how the method works in practice.

2 Characteristics of monotonic programs

We begin with a review of some basic concepts and results of monotonic optimization as discussed in [5,6]. The general form of a monotone global optimization problem is :

$$max\ f(x) \tag{1}$$

$$\text{subject to} \qquad g_i(x) \leq b_i, \quad i = 1, \ldots, m \tag{2}$$

$$x \in X = \{x \in Z^n \mid l_j \leq x_j \leq d_j,\ j = 1, ..., n\} \tag{3}$$

where f and all g_i are increasing functions of x_j on $[l_j, d_j]$ for $j = 1, ..., n$, $i = 1, ..., m$, functions f and g_i are not necessarily convex or separable.

The difficulty of designing a solution method for problem (1)-(3) lies in the non-convexity and non-separability of f and $g_i's$. Due to the non-convexity and non-separability, the classical branch-and-bound method and Lagrangian relaxation (decomposition) method are not directly applicable to this problem.

Definition 1 *: Let $[a,b] = \{x \in R^n \setminus a \leq x \leq b\}$ be a box in R^n. A function $f : [a,b] \to R$ is said to be increasing if $a \leq x \leq x' \leq b$ implies $f(x) \leq f(x')$; decreasing if $a \leq x \leq x' \leq b$ implies $f(x) \geq f(x')$; monotonic if it is either increasing or decreasing.*

Definition 2 *: A set $E \subset [a,b]$ is said to be downward (or normal) if $a \leq x' \leq x \in E$ implies $x' \in E$; upward (or reverse normal) if $b \geq x' \geq x \in E$ implies $x' \in E$.*

Definition 3 *:(1) The downward hull of a set $A \subset [a,b] \subset R^n_+$ is the set $A = \{y \in [a,b] \mid x \leq y \text{ for some } x \in A\}$. (2) The upward hull of a set $A \subset [a,b] \subset R^n_+$ is the set $A = \{y \in [a,b] \mid y \leq x \text{ for some } x \in A\}$.*

Definition 4 *: If A is a finite subset of [a, b] then the downward hull of A is called a polyblock with vertex set V.*

Proposition 5 *:(1) The maximum of an increasing function f (x) over a polyblock is achieved at a proper vertex of this polyblock.*
(2) The minimum of an increasing function f over an upward polyblock is achieved at a proper vertex of this upward polyblock.

Proof. : Let x^* be a maximizer of $f(x)$ over a polyblock P. Since a polyblock is the downward hull of its proper vertices, there exists a proper vertex x of P such that $x^* \in [a,x]$. Then $f(x) \geq f(x^*)$ because $x \geq x^*$, so x must be also an optimal solution. The proof of (2) is similar. ■

3 Methodological approach

Let $S = \{x \in X \subset Z^n \mid g_i(x) \leq b_i,\ i =: 1, ..., m\}$ and define

$$G\{x) = \max_{i=1,...,m} \{g_i(x) - b_i\} \tag{4}$$

The boundary of the constraints can be expressed as $\Gamma = \{x \in X \mid G(x) = 0\}$. Let $\langle \alpha, \beta \rangle$ be an integer box in X with $\alpha \in S$ and $\beta \notin S$. Suppose also that $G(\alpha) < 0$. Let x_b be an intersection point of the line $x = \lambda\alpha + (1-\lambda)\beta$, $0 \leq \lambda \leq 1$ and the boundary Γ. Since $G(\alpha) < 0$ and $G(\beta) > 0$, there must exist an x_b in X that satisfies $G(x_b) = 0$, i.e., $g_i(x_b) \leq 0$ for $i = 1, ..., m$ and there exists at least one i such that $g_i(x_b) = b_i$.

Denote by $\lfloor x \rfloor$ the integer vector with its i-th component being the maximum integer less than or equal to x_j, $j =: 1, ..., n$ and denote by $\lceil x \rceil$ the integer vector with its j-th component being the minimum integer greater than or equal to

x_j, $j =: 1, ..., n$. Let $x^F = \lfloor x_b \rfloor$ and $x^I = \lceil x_b \rceil$. Suppose that x_b is not integral (otherwise $x^F = x^I$). It is easy to see that x^F is a feasible point ($x^F \in S$) and x^I is infeasible ($x^I \notin S$).

Consider the integer boxes $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$. By the monotonicity of f and g_i, there are no feasible points better than x^F in $\langle \alpha, x^F \rangle$ and there are no feasible points x^I in $\langle x^I, \beta \rangle$. Therefore, when searching for an optimal solution to (1)-(3), we can remove integer boxes $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$ from $\langle \alpha, \beta \rangle$ for further consideration after comparing x^F with the incumbent solution.

3.1 Domain cut

We will refer the process of cutting non-promising integer boxes and partitioning a revised domain into sub-boxes as domain cut (see [7]). The domain cut is based on the monotone properties of f and g_i.

Theorem 6 *: Let $A = \langle \alpha, \beta \rangle$, $B = \langle \alpha, \gamma \rangle$ and $C = \langle \gamma, \beta \rangle$ where $\alpha \leq \gamma \leq \beta$. Then both $A \setminus B$ and $A \setminus C$ can be partitioned into at most n new integer boxes.*

$$A \setminus B = \bigcup_{j=1}^{n} \left(\prod_{k=1}^{j-1} \langle \alpha_k, \gamma_k \rangle \langle \gamma_j + 1, \beta_j \rangle \prod_{k=j+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{5}$$

$$A \setminus C = \bigcup_{j=1}^{j-1} \left(\prod_{k=1}^{j-1} \langle \gamma_k, \beta_k \rangle \langle \alpha_j, \gamma_j - 1 \rangle \prod_{k=j+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{6}$$

Theorem (6) shows that the set of the integer points left in $\langle \alpha, \beta \rangle$ after removing $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$ can be partitioned into a union of at most $2n - 1$ smaller integer boxes.

Based on the above discussion, we can derive an exact method for searching for an optimal solution of (1)-(3). The algorithm consists of two main steps: finding a feasible point x^F and an infeasible point x^I and generating integer boxes using the formulas (5) and (6). The best feasible solution obtained during the generation of integer boxes is kept as an incumbent solution. For the newly generated integer boxes, only the ones across the boundary Γ are needed to be kept for further consideration. Moreover, by the monotonicity of the problem, an integer box with the function value of its upper bound point less than or equal to the function value of the incumbent can be discarded. The algorithm proceeds successively by refining the partition and removing integer boxes that do not contain an optimal solution, and finally terminates at an optimal solution in a finite number of iterations.

3.2 The main algorithm

Step 0 : (Initialization). Let $l = (l_1, ..., l_n)$, $d = (d_1, ..., d_n)$. If l is infeasible, then problem (1)-(3) has no feasible solution; If d is feasible, then d is the optimal solution to (1)-(3), stop; Otherwise, set $x_{opt} = l$, $f_{opt} = f(x_{opt})$, $X^1 = \{\langle l, d\rangle\}$, and set $k = 1$.

Step 1: (Box Selection and Finding Boundary Point). Select an integer box $\langle \alpha, \beta\rangle \in X^k$. Set $X^k = X^k \backslash \langle \alpha, \beta\rangle$.
Finding the root λ^* of the following equation:

$$G\left[\lambda\alpha + (1-\lambda)\beta\right] = 0, \qquad \lambda \in [0, 1]$$

where G is defined in (4). Set $x_b = \lambda\alpha + (1-\lambda)\beta$. Set $x^F = \lfloor x_b \rfloor$.
If $x^F = x_b$ then set $x^I = x_b + e_j$, where e_j is the j-th unit vector in R^n with $x_b + e_j \leq \beta$. Otherwise, set $x^I = \lceil x_b \rceil$. If $f(x^F) > f_{opt}$,
set $x_{opt} = x^F$ and $f(x_{opt}) = f(x^F)$.

Step 2: (Partition and Remove).
(i) Apply the formula (6) to partition the set $\Omega_1 = \langle \alpha, \beta\rangle \setminus \langle x^I, \beta\rangle$ into a union of integer boxes. Let $x^F \in \langle \widehat{\alpha}, \widehat{\beta}\rangle \in \Omega_1$. Set $\Omega_1 = \Omega_1 \setminus \langle \widehat{\alpha}, \widehat{\beta}\rangle$
(ii) Apply the formula (5) to partition set $\Omega_2 = \langle \widehat{\alpha}, \widehat{\beta}\rangle \backslash \langle \widehat{\alpha}, x^F\rangle$.
(iii) Set $Y^k = \Omega_1 \cup \Omega_2$.
(iv) Perform the following for each integer box $\langle \alpha, \beta\rangle$ generated in the above partition process:
(a) If β is feasible, remove $\langle \alpha, \beta\rangle$ from Y^k. Furthermore if $f(\beta) > f(x_{opt})$ set $x_{opt} = \beta$ and $f_{opt} = f(\beta)$.
(b) If α is infeasible, remove $\langle \alpha, \beta\rangle$ from Y^k.
(c) If $f(\beta) < f_{opt}$, remove $\langle \alpha, \beta\rangle$ from Y^k.
(d) If α is feasible, β is infeasible and $f(\alpha) > f_{opt}$.
Set $x_{opt} = \alpha$ and $f_{opt} = f(\alpha)$.
Denote Z^k the set of integer boxes after the above removing process.

Step 3 : (Updating Integer Boxes). Removing all integer boxes $\langle \alpha, \beta\rangle$ in X^k with $f(\beta) < f_{opt}$. Set $X^k + 1 = X^k \cup Z^k$. If $X^k + 1 = \varnothing$, stop.
Otherwise, set $k = k + 1$ and go to Step 1.

Theorem 7 *This algorithm stops at an optimal solution to problem (1)-(3) within a finite number of iterations.*

Proof. : The finite convergence of the algorithm can be easily seen from the finiteness of X and the fact that at each iteration at least the integer points x^F and x^I are removed from X^*. Since the partition formulas (5) and (6) and the cutting process in Step 2 do not remove any integer point better than the incumbent x_{opt} the algorithm terminates with an optimal solution to (1)-(3). ■

4 The maximum probability problem

One of the prominent applications of the monotone optimization arises from the stochastic optimization problems witch attempts to model uncertainty in the data by assuming that (part of) the input is specified in terms of a probability distribution, rather than by deterministic data given in advance. Stochastic optimization models have become an important paradigm in a wide range of application areas, including transportation models, logistics, financial instruments, and network design. The widespread applicability of stochastic programming (SP) models has attracted considerable attention from the OR community, resulting in several books and papers. For an overview we refer to the introductory book of Kall and Wallace [2], the comprehensive books by Prekopa [9], Birge and Louveaux [1].

Consider the optimization problem

$$max\ Z = \ C^t x \tag{7}$$

$$\text{subject to} \qquad A_i x \leq b_i, \quad i = 1, \ldots, m \tag{8}$$

$$x \in X = \{x \in Z^n \mid l_j \leq x_j \leq d_j,\ j = 1, \ldots, n\} \tag{9}$$

where A is an $(m \times n)$ matrix, b is an m-vector and C is a $n-$random vector. Problem (7)-(9) is not well specified since the maximization of random objective is not mathematically defined. Therefore a revision of the modeling process is necessary, leading to so-called deterministic equivalents for (7)-(9). Several criteria can be used to derive equivalent problems (see, for example Slowinski and al.[10]). Each one records certain statistical characteristics of the stochastic objective. Three criteria are easily applicable. Their application requires knowing only the expected value and the variance of the stochastic objective. The expected value criterion generates central trend optimal solutions. The minimum variance criterion provides us with optimal solutions in which the stochastic objective is more concentrated around its expected value, and for this reason, can be considered a criterion of "not very risky" optimality.

The use of maximum probability criterion requires the collaboration of the decision maker who has to fix an aspiration level or a probability for the stochastic objective. However, it generates good solutions in terms of probability. For this reason, we think that it is very suitable for solving the stochastic problem (7)-(9) in spite of being less used then the previous ones. The problem is to determine the greater value u for which we can assert that, given the fixed probability p, the stochastic objective function exceeds the level u. This model can be expressed as :

$$max\ u \tag{10}$$

$$\text{subject to} \qquad P_r(C^t x \geq u) = p \tag{11}$$

$$A_i x \leq b_i,\ i = 1, \ldots, n \tag{12}$$

$$x \in X = \{x \in Z^n \mid l_j \leq x_j \leq d_j,\ j = 1, \ldots, n\} \tag{13}$$

4.1 Problem reformulation

Under the assumption that vector C has a normal distribution with mean value $\overline{C}$ and variance covariance matrix V, the constraint (11) takes the form:

$$\begin{aligned} P_r(C^t x \quad \geq \quad u) = p &\Longleftrightarrow P_r(C^t x \leq u) = 1 - p \\ &\Longleftrightarrow \quad P_r\left(\frac{C^t x - \overline{C}^t x}{\sqrt{x^t V x}} \leq \frac{u - \overline{C}^t x}{\sqrt{x^t V x}}\right) = 1 - p \\ &\Longleftrightarrow \quad \Phi\left(\frac{u - \overline{C}^t x}{\sqrt{x^t V x}}\right) = 1 - p \\ &\Longleftrightarrow \quad u = \overline{C}^t x + \Phi^{-1}(1-p)\sqrt{x^t V x} \end{aligned}$$

where $\Phi(.)$ is the distribution function of the standard normal distribution.

Then problem (10)-(13) reduces to the nonlinear integer programming problem

$$max\ u(x) = \overline{C}^t x + \Phi^{-1}(1-p)\sqrt{x^t V x} \tag{14}$$

$$\text{subject to} \qquad A_i x \leq b_i, i = 1, \ldots, m \tag{15}$$

$$x \in S = \{x \in Z^n \mid l_j \leq x_j \leq d_j,\ j = 1, \ldots, n\} \tag{16}$$

where u is an increasing function of x_j on $[l_j, d_j]$ for $j = 1, ..., n$.

Due to the non-separability of u, the classical branch-and-bound method and Lagrangian relaxation (decomposition) method are not applicable to problem (14)-(16). In the following, we show how the algorithm developed in this paper apply to this problem.

5 An illustrative example

Let $\overline{C}^t = (2, 3)$, $V = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$, $p = 0,998$, $\Phi^{-1}(1-p) = 2,87$.

Assume that only the linear constraint $x_1 + x_2 \leq 7.5$ must to be satisfied under the condition $1 \leq x_j \leq 6$, then problem (14)-(16) is formulated as

$max\ u(x) = 2x_1 + 3x_2 + 2,87\sqrt{2x_1^2 + x_2^2}$
subject to
$x_1 + x_2 \leq 7.5$
$x \in X = \{x \in Z^n \mid 1 \leq x_1 \leq 6, 1 \leq x_2 \leq 6\ \}$

The iterations of the following algorithm are described as follows:

Iteration 1:

Step 0 : let $l = (1,1)^t$; $d = (6,6)$; $x_{opt} = (1,1)^t$; $f_{opt} = 9.97$;
$X^1 = \{\langle l, d\rangle\}$, $k = 1$.

Step 1 : Select $\langle\alpha, \beta\rangle$=$\langle l, d\rangle$. Use the bisection procedure to find the root λ^* of the following equation:
$G[\lambda\alpha + (1-\lambda)\beta] = g(\lambda(1,1) + (1-\lambda)(6,6)) - 7,5 = 0,\ \ \lambda \in [0,1]$
$g(6 - 5\lambda, 6 - 5\lambda) = 0$, $\lambda \in [0,1]$.
Set $x_b = \lambda^*\alpha + (1-\lambda^*)\beta$, $x_b = (3.7735, 3.7735)^t$; $x^F = \lfloor x_b \rfloor = (3,3)^t$ and $x^I = \lceil x_b \rceil = (4,4)^t$. Since $u(x^F) = 29.91 > u_{opt} = 9.97$, we set $x_{opt} = (3,3)^t$; $u_{opt} = 29.91$.

Step 2 : Partition the set $\Omega_1 = \langle\alpha, \beta\rangle \setminus \langle x^I, \beta\rangle$ into two integer boxes.
$\Omega_1 = \{\langle(1,1)^t, (3,6)^t\rangle ; \langle(4,1)^t, (6,3)^t\rangle\} = \{B_1, B_2\}$
Since $x^F \in B_1$, remove B_1 from Ω_1. Set $\Omega_1 = B_2$ and Partition the set Ω_2.
$\Omega_2 = B_1 \setminus \langle(1,1)^t, x^F\rangle = B_1 \setminus \langle(1,1)^t, (3,3)^t\rangle = \langle(1,4)^t, (3,6)^t\rangle = B_3$.
Set $Y^1 = \Omega_1 \cup \Omega_2 = \{B_2, B_3\} = \{\langle(4,1)^t, (6,3)^t\rangle; \langle(1,4)^t, (3,6)^t\rangle\}$.
Remove all integers boxes $\langle\alpha, \beta\rangle$ in Y^1 with $u(\beta) \leq u_{opt}$.
Denote Z^1 the set of integer boxes after the above removing process.
We obtain $Z^1 = Y^1$.

Step 3 : Set $X^2 = Z^1 = \{B_2, B_3\}$

Iteration 2:

Step 1 : Select $\langle\alpha, \beta\rangle = \langle(4,1)^t, (6,3)^t\rangle\langle(4,1)^t, (6,3)^t\rangle$ in X^2.We get $u(6,3) = 46.83$ and $u(3,6) = 45.09$. Since $u(6,3) > u(3,6)$, we remove B_3 from X^2. We then set $X^2 = \{\langle(4,1)^t, (6,3)^t\rangle\}$.
The bisection procedure finds out $x_b = (5.25, 2.25)^t$, $x^F = (5,2)^t$ and $x^I = (6,3)^t$. Since $u(x^F) = 37.09 > u_{opt} = 29.91$,
set $x_{opt} = (5,2)^t$; $u_{opt} = 37.09$.

Step 2 : Partition the set $\Omega_1 = \langle\alpha, \beta\rangle \setminus \langle x^I, \beta\rangle$ into two integer boxes.
$\Omega_1 = \{B_1, B_2\} = \{\langle(4,1)^t, (5,3)^t\rangle ; \langle(6,1)^t, (6,2)^t\rangle\}$
Since $x^F = (5,2)^t \in B_1$, remove B_1 from Ω_1. Set $\Omega_1 = B_2$.
Since $(6,1)^t$ is feasible and $u(6,1) = 39.52 > u_{opt} = 37.09$, set $x_{opt} = (6,1)^t$ and $u_{opt} = 39.52$. Remove B_2 from Ω_1. We obtain $\Omega_1 = \varnothing$; $Z^2 = Y^2 = \varnothing$.

Step 3 : For $\langle(4,1)^t, (6,3)^t\rangle \in X^2$, since $x^F \in \langle(4,1)^t, (6,3)^t\rangle$, remove it from X^2. Thus $X^2 = \varnothing$ and $X^3 = X^2 \cup Z^2 = \varnothing$. Stop, the incumbent $x_{opt} = (6,1)^t$ is an optimal solution to the problem with $u_{opt} = 39.52$.

6 Conclusion

The main contribution of this study is an improved method for solving the equivalent deterministic nonlinear program of the maximum probability problem. This method exploit only the monotonic properties of the objective function and no convexification or linearisation is needed. It appears—on several examples— that the algorithm performs faster than the methods developed in [5] and [6]. However further experimental validation of this observation is needed.

References

[1] ABirge J.R., and Louveaux F.V., (1997), Introduction to Stochastic Programming, Springer Verlag, New York.

[2] Kall P., and Wallace S.W., (1994), Stochastic Programming, Wiley, Chichester, Also available as PDF file at http://www.unizh.ch/ior/Pages/Deutsch/ Mitglieder/Kall/bib/ka-wal-94.pdf.

[3] Klein Haneveld W.K., and van der Vlerk M.H., (1999), Stochastic integer programming: General models and algorithms, Ann. Oper. Res., 85,pages 39-57.

[4] Li, D., and White, D. J., (2000), p-th power Lagrangian method for integer programming, Annals of Operations Research, 98:151-170.

[5] Li, D., Sun, X. L., Biswal, M. P., and Gao, F., (2001), Convexification, concavification and monotonization in global optimization, Annals of Operations Research, vol. 105, pp. 213–226.

[6] Li, D., Sun, X.L. and McKinnon, K. (2005), An exact solution method for reliability optimization in complex systems, Annals of Operations Research 133, 129–148.

[7] Li, D., Sun, X. L., and Wang, J., (2005), Convergent Lagrangian methods for separable nonlinear integer programming: Objective level cut and domain cut methods, In J. Karlof, editor, Integer Programming: Theory and Practice, pages 19-36. Taylor & Francis Group, London.

[8] Phuong, N. T. H. and Tuy, H., (2004), A unified monotonic approach to generalized linear fractional programming, Journal of Global Optimization, vol. 26, pp. 229–259.

[9] Prekopa, A. , Ganszer, S., Deak I., K. Patyi, K., (1974), The stabil stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy, Proceedings of the Oxford International Conference-ed. Dempster.

[10] Slowinski R., and Teghem J., (1990), Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty, Kluwer Academic Publishers.

[11] Stougie L., and van der Vlerk M.H., (1997), Stochastic integer programming, In M. Dell Amico, F.Maffioli, and S. Martello, editors, Annotated Bibliographies in Combinatorial Optimization, chapter 9, pages 127-141, Wiley.

[12] Sun, X.L., McKinnon, K.I.M. and Li, D. (2001), A convexification method for a class of global optimization problems with applications to reliability optimization, Journal of Global Optimization 21, 185–199.

[13] Tawarmalani, M., and Sahinidis, N. V., (2004), Global optimization of mixed integer nonlinear programs: A theoretical and computational study, Mathematical Programming, 99:563-591.

[14] Tuy, H., (2000), Monotonic optimization: Problems and solution approaches, SIAM Journal of Optimization, vol. 11, no. 2, pp. 464–494.

Vander Vlerk M.H., (1996-2002), Stochastic programming bibliography. World Wide Web, http://mally.eco.rug.nl/spbib.html.

Bounds on the k-independence and k-chromatic numbers of graphs.

Mostafa Blidia*, Ahmed Bouchou† and Lutz Volkmann‡

April 24, 2009

Abstract

For an integer $k \geq 1$ and a graph $G = (V, E)$, a subset S of the vertex set V is k -independent in G if the maximum degree of the subgraph induced by the vertices of S is less or equal $k - 1$. The k-independence number $\beta_k(G)$ of G is the maximum cardinality of a k-independent set of G.

A set S of V is k-Co-independent in G if S is k-independent in the complement of G. The k-Co-independence number $\omega_k(G)$ of G is the maximum size of a k-Co-independent set in G. The sequences (β_k) and (ω_k) are weakly increasing.

We define the k-chromatic number or k-independence partition number $\chi_k(G)$ of G as the smallest integer m such that G admits a partition of his vertices into m k-independent sets and the k-Co-independence partition number $\theta_k(G)$ of G as the smallest integer m such that G admits a partition of his vertices into m k-Co-independent sets. The sequences (χ_k) and (θ_k) are weakly decreasing.

In this paper we mainly present bounds on these four parameters. Some of them are extensions of well-known classical results.

Keywords: k-independence, k-Co-independence, k-chromatic number, k-Co-independence partition number.

*Lamda-RO, Dept. Mathematics, University of Blida, B.P. 270, Blida, Algeria. E-mail: m_blidia@yahoo.fr.

†Dept. Mathematics, University of Blida, B.P. 270, Blida, Algeria. E-mail: bouchou.ahmed@yahoo.fr.

‡Lehrstuhl II f ür Mathematik, RWTH Aachen University, Templergraben 55, D-52056 Aachen, Germany. Email: volkm@math2.rwth-aachen.de

1 Terminology and introduction

We consider finite, undirected and simple graphs $G = (V, G) = (V(G), E(G))$ of order $|V(G)| = n(G)$ and size $|E(G)| = m(G)$. The open neighborhood of a vertex $v \in V$ is $N_G(v) = \{u \in V \mid uv \in E\}$, i.e, the set of all vertices adjacent with v. If $S \subseteq V(G)$, then $N_G(S) = \cup_{v \in S} N_G(v)$ is the open neighborhood of S. The closed neighbourhoods of v and S are $N_G[v] = N_G(v) \cup \{v\}$ and $N_G[S] = N_G(S) \cup S$. The degree of a vertex v of G is $d_G(v) = |N_G(v)|$. By $\Delta(G)$ and $\delta(G)$ we denote the maximum degree and the minimum degree of G. If $S \subseteq V(G)$, then $G[S]$ denotes the subgraph induced by the vertex set S. If $S \subset V$ and $x \in V \setminus S$, then we denote by $d_S(x)$ the number of edges from x to S.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two disjoint graphs. Their union $G = G_1 \cup G_2$ has the vertex set $V = V_1 \cup V_2$ and the edge set $E = E_1 \cup E_2$. Their join $G_1 + G_2$ consists of $G_1 \cup G_2$ together with the edge set $\{uv \,|\, u \in V_1,\, v \in V_2\}$. The composition $G = G_1[G_2]$ has $V = V_1 \times V_2$ as its vertex set and $u = (u_1, u_2)$ is adjacent with $v = (v_1, v_2)$ whenever (u_1 is adjacent with v_1) or ($u_1 = v_1$ and u_2 is adjacent with v_2). The cycle of order n is denoted by C_n.

For any parameter $\mu(G)$ associated to a graph property $\mathcal{P}$, we refer to a set of vertices with property $\mathcal{P}$ and cardinality $\mu(G)$ as a $\mu(G)$-set. An independent set S is a set of vertices whose induced subgraph has no edge. In [6, 7] Fink and Jacobson defined a generalization of the concepts of independence. For an integer $k \geq 1$ and a graph $G = (V, E)$, a subset D of V is $k-$dominating if every vertex in $V \setminus D$ has at least k neighbors in D. The k-dominanting number $\gamma_k(G)$ of G is the minimum cardinality of a k-dominating set of G. A subset S of V is k-independent in G if $\Delta(G[S]) < k$. The k-independence number $\beta_k(G)$ of G is the maximum cardinality of a k-independent set of G.

Since every k-independent set is $(k+1)$-independent, the sequence (β_k) is weakly increasing and thus

$$\beta(G) = \beta_1(G) \leq \beta_2(G) \leq \ldots \leq \beta_\Delta(G) < \beta_{\Delta+1}(G) = n.$$

More details and results on k-independence can be found in [2, 4, 5, 6, 7, 9, 13].

A set $S \subseteq V(G)$ is k-Co-independent in G if S is k-independent in the complement $\overline{G}$ of G; that is $\Delta(\overline{G[S]}) < k$. The k-Co-independence number $\omega_k(G)$ of G is the maximum size of a k-Co-independent set in G. Also the sequence (ω_k) is weakly increasing and so

$$\omega(G) = \omega_1(G) \leq \omega_2(G) \leq \ldots \leq \omega_{n-\delta-1}(G) < \omega_{n-\delta}(G) = n.$$

We define the k-chromatic number or k-independence partition number $\chi_k(G)$ of G as the smallest integer m such that G admits a partition of its vertices into m k-independent sets and the k-Co-independence partition number $\theta_k(G)$ of G as the smallest integer m such that G admits a partition of its vertices into m k-Co-independent sets. The sequences (χ_k) and (θ_k) are weakly decreasing and therefore

$$\chi(G) = \chi_1(G) \geq \chi_2(G) \geq \ldots \geq \chi_\Delta(G) > \chi_{\Delta+1}(G) = 1$$

as well as

$$\theta(G) = \theta_1(G) \geq \theta_2(G) \geq \ldots \geq \theta_{n-\delta-1}(G) > \theta_{n-\delta}(G) = 1.$$

For $k = 1$, the k-chromatic number of G is the chromatic number $\chi(G)$ of G, and the k-Co-independence partition number of G is the clique partition number $\theta(G)$ of G.

In the following we always assume that $k \geq 1$ is an integer. Since a k-Co-independent set S of G is a k-independent set of $\overline{G}$, we deduce that $\Delta(\overline{G[S]}) < k$ and $\Delta(\overline{G[S]}) + \delta(G[S]) = |S| - 1$. Thus $\delta(G[S]) > |S| - k - 1$. Equivalently, a set S is a k-Co-independent set if $\delta(G[S]) > |S| - k - 1$.

Observation 1 *Every graph G satisfies $\omega_k(G) = \beta_k(\overline{G})$ and $\theta_k(G) = \chi_k(\overline{G})$.*

When no confusion can arrise, we often write V, E, n, $d(v)$, $N(v)$, $\Delta, \overline{\Delta}, \delta, \overline{\delta}$ ... for $V(G)$, $E(G)$, $n(G)$, $d_G(v)$, $N_G(v), \Delta(G), \Delta(\overline{G}), \delta(G), \delta(\overline{G})$, ...

In this paper we present lower and upper bounds on $\beta_k(G)$, $\omega_k(G)$, $\chi_k(G)$ and $\theta_k(G)$. The special case $k = 1$ mostly leads to well-known classical results.

2 Relations between $\beta_k, \omega_k, \chi_k, \theta_k$

It is well known that $\omega(G) \leq \chi(G)$ and $\beta(G) \leq \theta(G)$ for every graph G. In the following we extend these inequalities.

Theorem 2 *If G is a graph such that $k \leq \min(\Delta, \overline{\Delta})$, then*

$$\omega_k(G) \leq (2k-1)\chi_k(G).$$

Proof. Let $S_1, S_2, \ldots, S_{\chi_k(G)}$ be a partition of the vertex set $V(G)$ into $\chi_k(G)$ k-independent sets. If B is a $\omega_k(G)$-set of G, then $\Delta(\overline{G[B]}) \leq k-1$. If

we define $A_i = B \cap S_i$ for all $i = 1, 2, \ldots, \chi_k(G)$, then A_i is a k-independent set in G as well as in $\overline{G}$ or $A_i = \emptyset$ for $i = 1, 2, \ldots, \chi_k(G)$. Thus $\Delta(G[A_i]) \leq k-1$ and $\Delta(\overline{G[A_i]}) \leq k-1$. Since $2m(G[A_i]) = \sum_{v \in A_i} d_{G[A_i]}(v) \leq |A_i|\,(k-1)$ and $2m(\overline{G[A_i]}) = \sum_{v \in A_i} d_{\overline{G[A_i]}}(v) \leq |A_i|\,(k-1)$, we obtain

$$\frac{|A_i|\,(|A_i|-1)}{2} = m(G[A_i]) + m(\overline{G[A_i]}) \leq |A_i|\,(k-1).$$

This implies $|A_i| \leq 2k-1$, and we deduce that

$$\omega_k(G) = |B| = \sum_{i=1}^{\chi_k(G)} |B \cap S_i| = \sum_{i=1}^{\chi_k(G)} |A_i| \leq (2k-1)\chi_k(G).$$

This completes the proof of Theorem 2. ■

The complement of the composition $G = G_1[G_2]$, where $G_1 = C_4$ and $G_2 = C_5$, is extremal for Theorem 2 when $k = 3$, because $\omega_3(\overline{G}) = 10$ and $\chi_3(\overline{G}) = 2$. Also the complement of the composition $C_n[C_5]$ with n even and $k = 3$ is extremal.

Observation 1 and Theorem 2 imply the next corollary.

Corollary 1 *If G is a graph such that $k \leq \min(\Delta, \overline{\Delta})$, then*

$$\beta_k(G) \leq (2k-1)\theta_k(G).$$

Let $k \geq 1$ be an odd integer, and let $G = G_1[G_2]$ be the composition, where G_1 is a graph such that $\beta(G_1) = \theta(G_1)$ and G_2 is a $(k-1)$-regular graph of order $2k-1$. Then we can see that G satisfies $\beta_k(G) = (2k-1)\theta_k(G)$.

In the book of C. Berge [1] we can find the inequalities $\chi(G)\beta(G) \geq n$, $\chi(G) + \beta(G) \leq n+1$, $\theta(G)\omega(G) \geq n$ and $\theta(G) + \omega(G) \leq n+1$ for every graph. In the following we generalize these results.

Observation 3 *If G is a graph with $k \leq \Delta$, then $\chi_k(G)\beta_k(G) \geq n(G)$.*

Proof. Let $S_1, S_2, \ldots, S_{\chi_k(G)}$ be a partition of the vertex set $V(G)$ into $\chi_k(G)$ k-independent sets. Since every S_i is a k -independent set, we conclude that $n(G) = |S_1| + |S_2| + \ldots + \left|S_{\chi_k}\right| \leq \chi_k(G)\beta_k(G)$. ■

Let $d \geq 2$ be an even integer, and let H_d be a $(d-2)$-regular graph of order d (that means that H_d is a complement of the union of $\frac{d}{2}$ copies of

K_2). Since $\beta_k(H_{pk}) = k$ and $\chi_k(H_{pk}) = p$ when k is even, the graph H_{pk} is extremal for Observation 3.

Observations 1 and 3 imply the next corollary.

Corollary 2 *Let G be a graph. If $k \leq \overline{\Delta}$, then $\theta_k(G)\omega_k(G) \geq n(G)$.*

Corollary 3 *Let G be a graph. If $k \leq \Delta$, then $\chi_k(G) + \beta_k(G) \geq 2\sqrt{n(G)}$.*

Proof. Observation 3 leads to $\chi_k(G)\beta_k(G) \geq n$. Thus we obtain $\chi_k(G) + \beta_k(G) \geq \chi_k(G) + \dfrac{n}{\chi_k(G)}$. A simple calculation shows that the minimum of the function $f(x) = x + \dfrac{n}{x}$ is $2\sqrt{n}$ when $0 < x \leq n$. Hence we receive at

$$\chi_k(G) + \beta_k(G) \geq \chi_k(G) + \frac{n}{\chi_k(G)} \geq 2\sqrt{n},$$

and the desired bound is proved. ■

Let $k \geq 2$ be an even integer, and let H_{k^2} be a $(k^2 - 2)$-regular-graph of order k^2. Since $\beta_k(H_{k^2}) = k$ and $\chi_k(H_{k^2}) = k$, the graph H_{k^2} is extremal for Corollary 3.

Corollary 4 *Let G be a graph. If $k \leq \overline{\Delta}$ then $\theta_k(G) + \omega_k(G) \geq 2\sqrt{n(G)}$.*

Theorem 4 *Let $G = (V, E)$ be a graph of order n. If $k \leq \Delta$, then*

$$k\chi_k(G) + \beta_k(G) \leq n + 2k - 1,$$

and the bound is best possible for odd k.

Proof. Let S be a $\beta_k(G)$-set of G. Then $V \setminus S$ can be partioned into $\left\lceil \frac{|V(G)\setminus S|}{k} \right\rceil$ k-independent sets. Since $\left\lceil \frac{|V\setminus S|}{k} \right\rceil = \left\lfloor \frac{|V\setminus S|-1}{k} \right\rfloor + 1$, we deduce that

$$\begin{aligned} \chi_k(G) &\leq \left\lceil \frac{|V \setminus S|}{k} \right\rceil + 1 \leq \left\lfloor \frac{|V \setminus S| - 1}{k} \right\rfloor + 2 \\ &\leq \frac{|V \setminus S| - 1}{k} + 2 = \frac{n - \beta_k(G) - 1}{k} + 2. \end{aligned}$$

This inequality chain yields to $k\chi_k(G) + \beta_k(G) \leq n + 2k - 1$, and the desired bound is proved.

Let $k \geq 1$ be an odd intger, and let M_{2k-1} be a $(k-1)$-regular graph of order $2k-1$. If we define $H = K_{kp+1} + M_{2k-1}$, then we observe that $\beta_k(H) = 2k-1$ and $\chi_k(H) = p+2$. It follows that $k\chi_k(H) + \beta_k(H) = n(H) + 2k - 1$, and thus the bound is best possible when k is odd. ■

Observation 1 and Theorem 4 imply the next result.

Corollary 5 *Let G be a graph. If $k \leq \overline{\Delta}$, then $k\theta_k(G)+\omega_k(G) \leq n+2k-1$, and the bound is sharp for odd k.*

Corollary 6 *Let G be a graph. If $k \leq \Delta$, then $\chi_k(G)\beta_k(G) \leq (n+2k-1)^2/4k$.*

Proof. Theorem 4 leads to $\beta_k(G) \leq n+2k-1-k\chi_k(G)$. Thus we obtain $\chi_k(G)\beta_k(G) \leq \chi_k(G)(n+2k-1-k\chi_k(G))$ A simple calculation shows that the maximum of the function $f(x) = x(n+2k-1-kx)$ est $\dfrac{(n+2k-1)^2}{4k}$ in $[1,n]$. thus $\chi_k(G)\beta_k(G) \leq \chi_k(G)(n+2k-1-k\chi_k(G)) \leq \dfrac{(n+2k-1)^2}{4k}$.

■

The star $K_{1,2k}$ is extremal for the precedent Corollary because $\beta_k(G) = 2k$, $\chi_k(G) = 2$ and $\chi_k(G)\beta_k(G) = 4k = (2k+1+2k-1)^2/4k$.

Corollary 7 *Let G be a graph, If $k \leq \Delta(\overline{G})$, then*

$$\theta_k(G)\omega_k(G) \leq (n+2k-1)^2/4k.$$

3 Bounds for $\beta_k, \omega_k, \chi_k, \theta_k$

The next result by Favaron [5] is the main tool for the proofs of our next two theorems.

Theorem 5 *(Favaron [5] 1985) If G is a graph, then every k-independent set D of G such that $k|D| - |E(G[D])|$ is maximum is a k-dominating set of G.*

Theorem 6 *Let G be a graph. If $k \leq \Delta$, then $\chi_k(G) \leq \dfrac{\Delta+k}{k}$.*

Proof. Let $S_1, S_2, \ldots, S_p$ be a partition of the vertex $V(G)$ such that S_1 is a k-independent set and a k-dominating set of G. In addition, let S_i be a k-independent set and a k-dominating set in $G[V(G) \setminus \overset{i-1}{\underset{j=1}{\cup}} S_j]$. In view of Theorem 5, such a partition exists. Then $d_{S_i}(x) \geq k$ for every vertex $x \in S_p$ and each $i \in \{1, 2, \ldots, p-1\}$. This implies that $d_G(x) \geq k(p-1)$ for each $x \in S_p$, and consequently $\Delta \geq k(p-1) \geq k(\chi_k(G) - 1)$. This leads to the desired upper bound for $\chi_k(G)$. ■

Let $G = v + H_k^1 + H_k^2 + \ldots + H_k^p$, where H_k^i is a copy of a $(k-2)$-regular graph H_k of even order k for every $i = 1, 2, \ldots, p$. Then G is extremal for Theorem 6, because $\chi_k(G) = p + 1$ and $\dfrac{\Delta + k}{k} = \dfrac{pk + k}{k}$.

Corollary 8 *Let G be a graph. If $k \leq \overline{\Delta}$, then $\theta_k(G) \leq \dfrac{\overline{\Delta} + k}{k}$.*

Theorem 6 and Observation 3 immediately implies the following well-known bound by Hopkins and Staton [10].

Corollary 9 *If G is a graph, and $1 \leq k \leq \Delta$ then:*

$$\beta_k(G) \geq \frac{n}{(1 + \left\lfloor \frac{\Delta(G)}{k} \right\rfloor)}$$

Corollary 10 *If G is a graph, and $1 \leq k \leq \Delta$ then:*

$$\omega_k(G) \geq \frac{n}{(1 + \left\lfloor \frac{\Delta(\overline{G})}{k} \right\rfloor)}$$

Theorem 7 *If G is a graph such that $\Delta \geq k$, then*

$$\chi_k(G) \leq \sqrt{\frac{2m(G)}{k^2} + \left(\frac{k-2}{2k}\right)^2} + \frac{3k-2}{2k}.$$

Proof. Let $S_1, S_2, \ldots, S_p$ be a partition of the vertex set $V(G)$ such that S_1 is a k-independent set and a k-dominating set of G. In addition, let S_i be a k-independent set and a k-dominating set of $G[V(G) - \cup_{j=1}^{i-1} S_j]$ for $i = 2, 3, \ldots, p$. By Theorem 5, such a partition exists. Since S_i is a k-dominating set of $G[V(G) - \cup_{j=1}^{i-1} S_j]$ for $i = 1, 2, \ldots, p-1$, it follows that $d_{S_i}(x) \geq k$ for each $x \in V(G) - \cup_{j=1}^{i} S_j$ and each $i = 1, 2, \ldots, p-1$.

Furthermore, we observe that $|S_i| \geq k$ for each $i = 1, 2, \ldots, p-1$, and therefore we obtain

$$\begin{aligned} m(G) &\geq k|S_2| + 2k|S_3| + \ldots + (p-2)k|S_{p-1}| + (p-1)k|S_p| \\ &\geq k^2(1 + 2 + \ldots + (p-2)) + k(p-1) \\ &= \frac{k^2(p-1)(p-2) + 2k(p-1)}{2} \end{aligned}$$

and thus

$$2m(G) \geq k(p-1)(k(p-2)+2).$$

The last inequality and a simple calculation lead to

$$\chi_k(G) - 1 \leq p - 1 \leq \sqrt{\frac{2m(G)}{k^2} + \left(\frac{k-2}{2k}\right)^2} + \frac{k-2}{2k},$$

and hence the theorem is proved. ■

Corollary 11 *If G is a graph such that $\overline{\Delta} \geq k$, then*

$$\theta_k(G) \leq \sqrt{\frac{2m(\overline{G})}{k^2} + \left(\frac{k-2}{2k}\right)^2} + \frac{3k-2}{2k}.$$

Theorem 7 immediately implies the following well-known bound by P. Hansen [8].

Corollary 12 *(Hansen [8] 1979) If G is a graph, then*

$$\chi(G) \leq \sqrt{2m(G) + \frac{1}{4}} + \frac{1}{2}.$$

Example 8 *Let t, k be integers such that $k \geq 1$ and $t \geq 2$, and let G be a complete t-partite graphs with the partite sets $V_1, V_2, \ldots, V_t$ such that $|V_1| = |V_2| = \ldots |V_{t-1}| = k$ and $|V_t| = 1$. Then $n(G) = k(t-1)+1$, $\chi_k(G) = t$ and $2m(G) = k(t-1)(k(t-2)+2)$. This leads to*

$$\sqrt{\frac{2m(G)}{k^2} + \left(\frac{k-2}{2k}\right)^2} = t - \frac{3k-2}{2k},$$

and thus

$$\chi_k(G) = t = \sqrt{\frac{2m(G)}{k^2} + \left(\frac{k-2}{2k}\right)^2} + \frac{3k-2}{2k}.$$

This example shows that Theorem 7 is the best possible.

Since $\Delta(G) = k(t-1)$, this example also shows that Theorem 6 is the best possible.

Lemma 9 *Let $n, p \geq 1$ and $r, t \geq 0$ be integers such that $n = tp + r$ and $r < p$. If $x_1, x_2, \ldots, x_p \geq 1$ are integers with $\sum_{i-1}^{p} x_i = n$, then*

$$\sum_{i=1}^{p} x_i^2 \geq \frac{n^2 - r^2}{p} + r \tag{1}$$

Proof. Assume, without loss of generality, that $x_1 \geq x_2 \geq \ldots \geq x_p$. First we will show that the sum in (1) is minimum when $x_1 \leq x_p + 1$. Suppose that $x_1 \geq x_p + 2$ and define $x_1' = x_1 - 1$, $x_p' = x_p + 1$ and $x_i' = x_i$ for $2 \leq i \leq p-1$. Obviously, $\sum_{i=1}^{p} x_i' = n$ but

$$\begin{aligned}\sum_{i=1}^{p} x_i^2 - \sum_{i=1}^{p} x_i'^2 &= x_1^2 - (x_1 - 1)^2 + x_p^2 - (x_p + 1)^2 \\ &= 2(x_1 - x_p - 1) \geq 2.\end{aligned}$$

Consequently, the sum in (1) is minimum when $x_1 \leq x_p + 1$. But then $x_i = t$ for $r + 1 \leq i \leq p$ and $x_i = t + 1$ for $1 \leq i \leq r$. Using $t = \dfrac{(n-r)}{p}$, we obtain

$$\begin{aligned}\sum_{i=1}^{p} x_i^2 &= \sum_{i=1}^{r} (t+1)^2 + \sum_{i=r+1}^{p} t^2 \\ &= r(t+1)^2 + (p-r)t^2 \\ &= t(pt + 2r) + r \\ &= \frac{n-r}{p}(n - r + 2r) + r \\ &= \frac{n^2 - r^2}{p} + r,\end{aligned}$$

and the proof of Lemma 9 is complete. ■

Theorem 10 *Let G be a graph of order n, and assume that $n = t\chi_k(G) + r$ with integers $t \geq 0$ and $0 \leq r < \chi_k(G)$. If $k \leq \Delta$, then*

$$\chi_k(G) \geq \max\left\{\frac{n^2 - r^2}{n^2 - 2m(G) + (k-1)n - r}, \frac{n^2}{n^2 - 2m(G) + (k-1)n}\right\} \tag{2}$$

Proof. Let $S_1, S_2, \ldots, S_p$ be a partition of $V(G)$ into $p = \chi_k(G)$ k-independent sets. Applying Lemma 9, we obtain

$$\begin{aligned} n(n-1) &= 2m(G) + 2m(\overline{G}) \geq 2m(G) + \sum_{i=1}^{p} 2m(\overline{G[S_i]}) \\ &\geq 2m(G) + \sum_{i=1}^{p} |S_i|(|S_i| - k) = 2m(G) + \sum_{i=1}^{p} |S_i|^2 - k\sum_{i=1}^{p} |S_i| \\ &\geq 2m(G) - kn + \frac{n^2 - r^2}{p} + r. \end{aligned} \tag{3}$$

Using $p = \chi_k(G)$, this easily leads to

$$\chi_k(G) \geq \frac{n^2 - r^2}{n^2 - 2m(G) + (k-1)n - r}. \tag{4}$$

In addition, we deduce from (3) that

$$n(n-1) \geq 2m(G) - kn + \frac{n^2 - r^2}{p} + r \geq 2m(G) - kn + \frac{n^2}{p},$$

and this yields

$$\chi_k(G) \geq \frac{n^2}{n^2 - 2m(G) + (k-1)n}. \tag{5}$$

Combining (4) and (5), we obtain the desired bound (2). ■

Let $H_{p(k+1)}$ be a $(p(k+1)-2)$-regular graph of order $p(k+1)$ with k odd. Then G is extremal for Theorem 10, because

$$\chi_k(G) = p = \frac{n^2}{n^2 - 2m(G) + (k-1)n}.$$

Because of $\theta_k(\overline{G}) = \chi_k(G)$ and $2m(G) + 2m(\overline{G}) = n^2 - n$, inequality (2) implies the next corollary.

Corollary 13 *Let G be a graph of order n, and assume that $n = t\theta_k(G) + r$ with integers $t \geq 0$ and $0 \leq r < \theta_k(G)$. If $k \leq \overline{\Delta}$, then*

$$\theta_k(G) \geq \max\left\{ \frac{n^2 - r^2}{2m(G) + kn - r}, \frac{n^2}{2m(G) + kn} \right\}.$$

The following well-known bound by Meyers and Liu [11] is a special case of Theorem 10.

Corollary 14 *(Meyers, Liu [11] 1972) If G is a graph of order n, then*

$$\chi(G) \geq \frac{n^2}{n^2 - 2m(G)}.$$

4 Nordhaus-Gaddum type results

In their now classical paper [12], Nordhaus and Gaddum established the inequality $\chi(G) + \chi(\overline{G}) \leq n + 1$. Improvements and generalizations of this inequality can be found in Section 9.1 of the monograph [9] by Haynes, Hedetniemi and Slater.

Theorem 11 *(Chartrand, Schuster [3] 1974) For any graph G of order n, we have:*

1. $\beta(G) + \beta(\overline{G}) \leq n + 1$ *and* $\omega(G) + \omega(\overline{G}) \leq n + 1$.
2. $\beta(G)\beta(\overline{G}) \leq \left\lceil \frac{n^2+2n}{4} \right\rceil$ *and* $\omega(G)\omega(\overline{G}) \leq \left\lceil \frac{n^2+2n}{4} \right\rceil$.

Next we present generalizations of these inequalities.

Theorem 12 *If G is a graph of order n such that $k \leq \min(\Delta, \overline{\Delta})$, then $\beta_k(G) + \beta_k(\overline{G}) \leq n + 2k - 1$ and $\beta_k(G)\beta_k(\overline{G}) \leq (n + 2k - 1)^2/4$.*

Proof. Let S be a $\beta_k(G)$-set of G, and let B be a $\beta_k(\overline{G})$-set of $\overline{G}$. If $A = B \cap S$, then $n \geq |S| + |B| - |A|$. Since $|A| \leq 2k - 1$ (see the proof of Theorem 2), it follows that $\beta_k(G) + \beta_k(\overline{G}) \leq n + 2k - 1$, and the first inequlity is proved. This implies that

$$\begin{aligned}(n + 2k - 1)^2 &\geq \big(\beta_k(G) + \beta_k(\overline{G})\big)^2 = \big(\beta_k(G) - \beta_k(\overline{G})\big)^2 + 4\beta_k(G)\beta_k(\overline{G}) \\ &\geq 4\beta_k(G)\beta_k(\overline{G}),\end{aligned}$$

and thus $\beta_k(G)\beta_k(\overline{G}) \leq (n + 2k - 1)^2/4$. ■

Let $k \geq 1$ be an odd integer, and let $G = G_1[G_2]$ be the composition, where G_1 is the complete graph K_p and G_2 is a $(k-1)$-regular graph of order $2k - 1$. Then we can see that G satisfies $\beta_k(G) + \beta_k(\overline{G}) = n + 2k - 1$.

In addition, let $G = (C_5 + C_5) \cup C_5$. Since $\beta_3(G) = 10$ and $\beta_3(\overline{G}) = 10$, we see that G is extremal for the first and the second inequality of Theorem 12 when $k = 3$.

Applying Observation 1 and Theorem 12, we obtain the next corollary.

Corollary 15 *If G is a graph of order n such that $k \leq \min(\Delta, \overline{\Delta})$, then $\omega_k(G) + \omega_k(\overline{G}) \leq n + 2k - 1$ and $\omega_k(G)\omega_k(\overline{G}) \leq (n + 2k - 1)^2/4$.*

Theorem 13 *If G is a graph of order n such that $k \leq \min(\Delta, \overline{\Delta})$, then* $\chi_k(G)\chi_k(\overline{G}) \geq \dfrac{n}{2k-1}$ *and* $\chi_k(G) + \chi_k(\overline{G}) \geq 2\sqrt{\dfrac{n}{2k-1}}$

Proof. Let $S_1, S_2, \ldots, S_{\chi_k(G)}$ be a partition of the vertex set $V(G)$ into $\chi_k(G)$ k-independents sets. Then each S_i is a k-Co-independent set of $\overline{G}$ and Theorem 2 implies that $|S_i| \leq \omega_k(\overline{G}) \leq (2k-1)\,\chi_k(\overline{G})$. Therefore we obtain

$$n = \sum_{i=1}^{\chi_k(G)} |S_i| \leq (2k-1)\,\chi_k(\overline{G})\chi_k(G),$$

and the first inequlity is proved. Now it follows that

$$\begin{aligned} 4\frac{n}{2k-1} &\leq 4\chi_k(G)\chi_k(\overline{G}) \leq \left(\chi_k(G) - \chi_k(\overline{G})\right)^2 + 4\chi_k(G)\chi_k(\overline{G}) \\ &= \left(\chi_k(G) + \chi_k(\overline{G})\right)^2, \end{aligned}$$

and this leads to the second inequality. ■

Let $k \geq 1$ be an odd integer, and let $G = G_1[G_2]$ be the composition, where $G_1 = C_4$ and G_2 is a $(k-1)$-regular graph of order $2k-1$. Then this composition is extremal for the first and the second inequality of Theorem 13, because $\chi_k(G) = 2$ and $\chi_k(\overline{G}) = 2$.

Corollary 16 *If G is a graph of order n such that $k \leq \min(\Delta, \overline{\Delta})$, then* $\theta_k(G)\theta_k(\overline{G}) \leq \dfrac{n}{2k-1}$ *and* $\theta_k(G) + \theta_k(\overline{G}) \geq 2\sqrt{\dfrac{n}{2k-1}}$

Since $\Delta(\overline{G}) = n - \delta(G) - 1$, Theorem 6 yields the following Nordhaus-Gaddum bound.

Corollary 17 *Let G be a graph of order n. If $k \leq \Delta$ and $k \leq \overline{\Delta}$, then*

$$\chi_k(G) + \chi_k(\overline{G}) \leq \frac{\Delta(G) - \delta(G) + n + 2k - 1}{k}$$

If $\Delta(G) + \Delta(\overline{G}) \leq n$, then $\Delta(G) + n - \delta(G) - 1 \leq n$ and thus $0 \leq \Delta(G) - \delta(G) \leq 1$.

So, if $\Delta(G) - \delta(G) = 0$, then $\chi_k(G) + \chi_k(\overline{G}) \leq \dfrac{n + 2k - 1}{k}$,

And if $\Delta(G) - \delta(G) = 1$, then $\chi_k(G) + \chi_k(\overline{G}) \leq \dfrac{n + 2k}{k}$.

Conjecture 14 *If G is a graph of order n, then*

$$\chi_k(G) + \chi_k(\overline{G}) \leq \left\lceil \frac{n + 2k - 1}{k} \right\rceil.$$

References

[1] C. Berge, Graphes et Hypergraphes, Edition Dunod (1973).

[2] Y. Caro and Z. Tuza, Improved lower bounds on $k-$independence, J. Graph Theory 15 (1991) 99-107.

[3] G. Chartrand and S. Schuster, On the independance numbers of complementary graphs, Trans. New York Acad. Sci., Series II, 36:247-251, 1974.

[4] O. Favaron, $k-$domination and $k-$independence in graphs, Ars Combin. 25 (1988) C 159-167.

[5] O. Favaron, On a conjecture of Fink and Jacobson concerning $k-$domination and $k-$independene, J. Combin. Theory Series B 39 N° 1 (1985) 101-102.

[6] J. F. Fink and M. S. Jacobson, $n-$domination in graphs, in : *Graph Theory with Applications to Algorithms and Computer.* John Wiley and sons, New York (1985) 283-300.

[7] J. F. Fink and M. S. Jacobson, $n-$domination, n-dependence and forbidden subgraphs, in : *Graph Theory with Applications to Algorithms and Computer.* John Wiley and sons, New York (1985) 301-311.

[8] P. Hansen, Upper bounds for the stabilty number of a graph, Rev. Roum. Math. Pures Appl. 24 (1979), 1195-1199.

[9] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.

[10] G. Hopkins and W. Staton, Vertex partitions and k-small subsets of graphs. Ars Combin 22 (1986), 19-24.

[11] B.R. Meyers and R. Liu, A lower bound on the chromatic number of a graph. Networks 1 (1971/72) 273-277.

[12] E.A. Nordhaus and J. W. Gaddum, On complementary graphs, Amer. Math. Monthly 63 (1956), 175-177.

[13] C. Stracke and L. Volkmann, A new domination conception, J. Graph Theory 17 (1993) 315-323.

Les graphes independence point critiques

Kamel Tablennehas
Département de Mathématiques, Faculté des Sciences, Université de Blida.
E-mail: tablennehas1@yahoo.fr

Résumé

Soit $G = (V, E)$ un graphe simple. Un sous-ensemble S de V est un dominant de G si tout sommet de $V - S$ est adjacent à au moins un sommet de S. Le cardinal minimum d'un ensemble dominant de G, noté $\gamma(G)$, est appelé nombre de domination. Un ensemble dominant stable d'un graphe G est un ensemble dominant dont le sous-graphe induit est un stable. Le cardinal minimum (resp. maximum) d'un ensemble dominant stable de G, noté $i(G)$ (resp. $\beta_0(G)$), est appelé nombre de domination stable (resp. nombre de stabilité). Etant donné un paramètre μ d'un graphe G, nous dirons que G est μ-point critique si $\mu(G_{ab}) < \mu(G)$ pour toute arête $ab \in E(G)$ et G est totalement μ-point critique si $\mu(G_{ab}) < \mu(G)$ pour tout couple de sommets $(a, b) \in V \times V$, où G_{ab} est le graphe obtenu par la contraction de l'arête ab ou l'identification de a et b.

Dans ce papier, on donne une condition nécessaire et suffisante pour qu'un graphe soit β_0-point critique ainsi qu'une caractérisation de quelques graphes β_0-point critiques, à savoir les arbres et les graphes sans $K_{1,3}$. Ensuite, on conjecture que la classe des arbres i-point critiques est équivalente à la classe des graphes i-excellent.

Keywords: domination stable, contraction, identification, graphe critique.

1 Introduction

On considère un graphe simple connexe $G = (V, E)$ ayant V comme ensemble de sommets et E comme ensemble d'arêtes.

Le nombre de sommets $|V|$ dans un graphe G est appelé ordre de G et noté souvent par n. Le voisinage ouvert d'un sommet est $N(v) = \{u \in V / uv \in E\}$, son voisinage fermé est $N[v] = N(v) \cup \{v\}$. Le degré d'un sommet

v, noté par $d_G(v)$ est $|N(v)|$. Un sommet de degré nul est dit isolé. Un sommet de degré un est appelé sommet pendant, et son voisin est dit sommet support. Un graphe G est dit **connexe** si pour toute paire de sommets du graphe il existe une chaîne les reliant. Un sommet v d'un graphe connexe est dit sommet d'articulation si le graphe $G - v$ n'est pas connexe. Le graphe **complet** d'ordre n, noté K_n, est le graphe simple dans laquel tous les sommets sont de degré $n - 1$. Ainsi deux sommets quelconques de K_n sont adjacents. Une **clique** est un sous-graphe complet d'un graphe G. Une clique de p sommets est notée $K_{p.}$
Etant donné un graphe B. Un graphe G est dit **sans** B, si le graphe G ne contient pas B comme sous-graphe induit. Un graphe $G = (V, E)$ est dit $\mathbf{k - parti}$. s'il existe une partition de V en k sous-ensembles $V_1V_2, ..., V_k$ tels que chacun des $G[V_i]$ ne contient aucune arête. Si $k = 2$ le graphe G est dit **biparti.** On appelle graphe **biparti complet,** un graphe biparti tel que pour tout sommet $u \in V_1$ et $v \in V_2, uv \in E$. Si $|V_1| = p$ et $|V_2| = q$ alors le graphe **biparti complet** est noté $K_{p,q}$. Un cas spécial d'un graphe biparti complet dans lequel $|V_1| = 1$ et $|V_2| = s$ est appelé une **étoile** et noté $K_{1,s}$. Le sommet de V_1 est appelé **centre** de l'étoile.Une couronne d'un graphe H notée par HoK_1 est obtenu par une copie de H où chaque sommet de H est adjacent à un sommet pendant.

Soit $G = (V, E)$ un graphe simple. Un sous-ensmble $S \subseteq V(G)$ est un ensemble dominant si tout sommet de $V - S$ est adjacent à au moins un sommet de S. Le cardinal minimum d'un ensemble dominant de G est appelé nombre de domination et est noté par $\gamma(G)$.

Un sous ensemble S de V est dit dominant stable de G si S est un dominant et le sous graphe induit par S ne contient pas d'arête. Le cardinal minimum (resp.maximum) d'un stable maximal de G noté $i(G)$ (resp.$\beta_0(G)$) est appelé le nombre de domination stable (resp.le nombre de stabilité) de G. Pour tout paramètre $\mu(G)$, un ensemble S de cardinal $\mu(G)$ vérifiant la propriété désirée est appelé $\mu(G)$-ensemble ou simplement μ-ensemble. On dit qu'un sommet est μ-**bon** s'il est contenu dans au moins un $\mu(G)$-ensemble et μ-**mauvais** sinon. Un graphe G est dit $\mu-$ **excellent** si tout sommet de G est μ-bon.

Si a et b sont deux sommets de G (a et b peuvent être adjacents ou non), alors on note par G_{ab} le graphe obtenu par la contraction de l'arête ab ou l'identification de a et b. Le nouveau sommet obtenu dans G_{ab} est noté par $\overline{ab}$. Nous dirons que G est μ-point critique si $\mu(G_{ab}) < \mu(G)$ pour toute arête $ab \in E$, et G est totalement μ-point critique si $\mu(G_{ab}) < \mu(G)$ pour tout couple de sommets $(a, b) \in V \times V$. Rappelons que la notion des graphes γ-point critiques a été introduite par Burton et Sumner [5].

Dans cet article, on va étudier les graphes à dominant stable point critiques.

2 Les graphes β_0-point critiques

2.1 Quelques résultats préliminaires

Observation 1 *Soient un graphe* $G = (V, E)$ *et* u, v *deux sommets de* $V(G)$. *Alors* $\beta_0(G) - 1 \leq \beta_0(G_{uv}) \leq \beta_0(G)$.

Preuve. Soit S un β_0-ensemble de G. Si $u, v \notin S$, alors S reste un β_0-ensemble du graphe G_{uv} par conséquent $\beta_0(G_{uv}) \geq \beta_0(G) \geq \beta_0(G) - 1$. Si $u, v \in S$, alors $\{\overline{uv}\} \cup S - \{u, v\}$ est un ensemble indépendant du graphe G_{uv}, par conséquent $\beta_0(G_{uv}) \geq \beta_0(G) - 1$. On suppose maintenant que $u \in S$ et $v \notin S$, alors $S - \{u\}$ est un β_0-ensemble du graphe G_{uv}, d'ou $\beta_0(G_{uv}) \geq \beta_0(G) - 1$.
Supposons que D est un β_0-ensemble du graphe G_{uv}. Si $\overline{uv} \notin D$, alors D est un dominant stable du graphe G, et donc $\beta_0(G) \geq \beta_0(G_{uv})$. On suppose que le sommet $\overline{uv} \in D$ et $uv \notin E(G)$, alors $\{u, v\} \cup D - \{\overline{uv}\}$ est un dominant stable du graphe G, par conséquent $\beta_0(G) \geq \beta_0(G_{uv}) + 1$ donc $\beta_0(G_{uv}) \leq \beta_0(G)$. On suppose maintenant que $\overline{uv} \in D$ et $uv \in E(G)$, alors $\{u\} \cup D - \{\overline{uv}\}$ est un dominant stable du graphe G, et par suite $\beta_0(G) \geq .\beta_0(G_{uv})$. Dans tous les cas on a $\beta_0(G) \geq \beta_0(G_{uv})$. ■

Observation 2 *Si* $G = (V, E)$ *est un graphe* β_0*-point critique alors* $\beta_0(G_{uv}) = \beta_0(G) - 1$ *pour toute arête* $uv \in E(G)$.

Preuve. C'est une conséquence de l'observation 1. ■

Une condition nécessaire et suffaisante pour qu'un graphe connexe soit β_0-point critique est donnée par le théorème suivant:

Théorème 3 *Un graphe* $G = (V, E)$ *connexe d'ordre* $n \geq 3$ *est* β_0*-point critique si et seulement si :*
i)− Pour tout stable maximum S *du graphe* G, $V - S$ *est un stable.*
ii)− Pour tout sommet $x \in V - S$ *on a* $d(x) \geq 2$.

Preuve. Soient G un graphe β_0-point critique et S un β_0-ensemble du graphe G.
i)− On suppose que le sous-graphe induit par l'ensemble $V - S$ n'est pas un stable, alors $V - S$ contient au moins une arête uv. En contractant l'arête uv, S reste un $\beta_0(G_{uv})$-ensemble et donc $\beta_0(G_{uv}) \geq \beta_0(G)$, contradiction

avec le fait que G est β_0-point critique.
$ii)-$ On suppose maintenant qu'il existe un sommet $x \in V - S$ tel que $d(x) = 1$. Alors x admet un unique voisin $y \in S$ et comme G est connexe d'ordre $n \geq 3$, le sommet y admet au moins un voisin $z \in V - S$ autre que le sommet x, par conséquent l'ensemble $S' = (S - \{y\}) \cup \{z\}$ est un stable tel que $V - S'$ n'est pas un stable, contradiction.
Pour la réciproque, soit S un stable maximum de G et soient u, v deux sommets adjacents tel que $u \in S$ et $v \in V - S$. Alors $S - \{u\}$ est un stable de G_{uv}, d'ou $\beta_0(G_{uv}) \geq |S| - 1$. Supposons que $\beta_0(G_{uv}) > |S| - 1$, alors d'aprés l'observation 1 $\beta_0(G_{uv}) = \beta_0(G)$. Soit D un $\beta_0(G_{uv})$-ensemble, si $uv \notin D$ alors $((V - D) - \{\overline{uv}\}) \cup \{u, v\}$ contient l'arête uv, contradiction avec les hypothèses du théorème, d'ou $\beta_0(G_{uv}) = \beta_0(G) - 1$. ■

Puisque tout graphe biparti admet une partiton unique en deux stables, alors d'aprés le théorème 3 , on dèduit le corollaire suivant:

Corollaire 4 *Si G n'est pas un graphe biparti alors G n'est pas $\beta_\circ$-point critique.*

Nous donnons ci-dessous une caractérisation des graphes totalement β_0-point critiques.

Théorème 5 *Un graphe $G = (V, E)$ est totalement β_0-point critique si et seulement si G est une étoile $K_{1,t}$ avec $t \geq 2$.*

Preuve. Si G est totalement β_0-point critique alors G est β_0-point critique. Si S est un $\beta_0(G)$-ensemble alors $V - S$ est un stable. On suppose que $|V - S| \geq 2$ et soient x, y deux sommets de $V - S$. Alors S reste un $\beta_0(G_{xy})$-ensemble, par conséquent $\beta_0(G_{xy}) \geq \beta_0(G)$, contradiction avec le fait que G est β_0-point critique. Donc $|V - S| = 1$ et G est une etoile $K_{1,t}$ avec $t \geq 2$. ■

2.2 Les arbres β_0-point critiques

Rappelons la définition d'un 2-dominant comme suit:

Définition 6 *Soit $G = (V, E)$ un graphe, un sous ensemble S de V est dit un ensemble 2-dominant de G si S est un dominant et pour tout sommet $v \in V - S, v$ est adjacent à deux sommets de S. Le nombre de domination double noté par $\gamma_2(G)$ est le cardinal minimum d'un ensemble 2-dominant de G.*

Dans [7] Blidia, Chellali et Favaron ont montré que dans les arbres le nombre 2-domination est borné inferieurement par le nombre de stabilité.

Théorème 7 (Blidia, Chellali et Favaron [7]) *Si T est un arbre, alors* $\gamma_2(T) \geq \beta_0(T)$.

Les mêmes auteurs de [7] ont caractérisé les arbres extrèmaux atteignant la borne inferieure.

Soit $\mathcal{F}$: la famille des arbres obtenu d'une séquence d'arbres $T_1, T_2, ..., T_k$ avec $k \geq 1$ tel que $T_1 = K_{1,t}, t \geq 2$ de centre w, $T = T_k$ et si $k \geq 2$ T_{i+1} est obtenu à partir de T_i par une des opérations suivantes. Poser $A(T_1) = L_w$.

Opération θ_1 : Attacher une étoile $K_{1,p}$ avec $p \geq 1$ de centre x par une arête à un sommet pendant y et poser $A(T_{i+1}) = A(T_i) \cup L_x$.

Opération θ_2 : Attacher une étoile $K_{1,p}$ avec $p \geq 1$ de centre x par une arête à un sommet y qui n'est pas pendant et poser $A(T_{i+1}) = A(T_i) \cup L_x$.

Opération θ_3 : Attacher une étoile $K_{1,p}$ avec $p \geq 1$ de centre x par une arête à un sommet y de $V(T_i) - A(T_i)$ et poser $A(T_{i+1}) = A(T_i) \cup L_x$.

Théorème 8 (Blidia, Chellali et Favaron [7]) *Soit T un arbre, alors les assertions suivantes sont équivalentes:*
a)- $\gamma_2(T) = \beta_0(T)$. *b)- $T = K_1$ ou $T \in \mathcal{F}$.*
c)- T admet un $\gamma_2(T)$-ensemble unique et $\beta_0(T)$-ensemble unique.

Par le théorème suivant on donne une condition nécessaire pour qu'un arbre soit β_0-point critique.

Théorème 9 *Si T est un arbre β_0-point critique, alors* $\beta_0(T) = \gamma_2(T)$.

Preuve. D'aprés le théorème 3, tout sommet $x \in V - S$ on a $d(x) \geq 2$ d'ou $\gamma_2(G) \leq \beta_0(G)$ pour tout graphe β_0-point critique et d'aprés le théorème 7 on a $\gamma_2(G) \geq \beta_0(G)$ pour les arbres. Par conséquent si G est un arbre alors $\beta_0(G) = \gamma_2(G)$. ■

Dans ce paragraphe nous proposons une caractérisation des arbres β_0-point critiques.

Soit $\mathcal{F}_1$ la famille des arbres T qui se construisent récursivement à partir d'un arbre $T_1 = K_{1,p}$ avec $p \geq 2$ de centre x, et pour $i \geq 1$ l'arbre T_{i+1} est obtenu à partir de T_i par l'opération θ. Poser $A(T_1) = L_{x.}$ (l'ensemble des sommets pendants adjacents à x).

Opération θ : Attacher une étoile $K_{1,t}$ avec $t \geq 1$ de centre u par une arête à un sommet $v \in A(T_i)$ et poser $A(T_{i+1}) = A(T_i) \cup L_u$.

Théorème 10 *Un arbre T d'ordre $n \geq 3$ est β_0-point critique si et seulement si $T \in \mathcal{F}_1$.*

Preuve. ($\Leftarrow$) Par induction sur le nombre $(k-1)$ d'opérations nécéssaire pour construire l'arbre T. Si $k = 1$ alors $T = T_1 = K_{1,p}$ pour $p \geq 2$, par conséquent T est β_0-point critique. On suppose que pour $k \geq 2$ la propriété est vraie pour tout arbre $T' \in \mathcal{F}_1$ construit avec moins de $(k-1)$ opérations. Soit T un arbre de la famille $\mathcal{F}_1$ construit à partir de l'arbre T' deja construit par $(k-1)$ opérations. Donc par induction T' est β_0-point critique. Il est clair que $\beta_0(T) = \beta_0(T') + |L_u|$, d'aprés le théorème 7, $A(T)$ est à la fois un unique $\gamma_2(T)$-ensemble et un $\beta_0(T)$-ensemble. Et par construction $V(T) - A(T)$ est un stable, donc d'aprés le théorème 3, $A(T)$ est un 2-dominant. Alors d'aprés 3 T est β_0-point critique.
($\Rightarrow$) Par induction sur l'ordre $n(T)$. pour $n = 2$ alors $T = P_2$ (T n'est pas β_0-point critique). Pour $n \geq 3$ et si $diam(T) = 2$ alors $T = K_{1,t}$ avec $t \geq 2$, par conséquent $T \in \mathcal{F}_1$. On suppose que tout arbre d'ordre $n' \geq 3$ β_0-point critique est dans $\mathcal{F}_1$. Soit T un arbre d'ordre $n > n'$, β_0-point critique. Puisque aucun arbre de $diam(T) = 3$ n'est β_0-point critique, on suppose que $diam(T) \geq 4$ et soit S un stable maximum de T. D'aprés le théorème 3 S est un 2−dominant et donc $\gamma_2(T) = \beta_0(T)$. Par conséquent tous les sommets pendants sont dans S. Soit v un sommet support pour lequel $V(T) - (L_v \cup \{v\})$ est un arbre (un tel sommet existe toujours) et u son unique voisin dans $V(T) - (L_v \cup \{v\})$. Dans ce cas $L_v \subset S$, $v \notin S$ et $u \in S$ (car sinon uv est une arête dans $V - S$, contradiction avec le fait que $V - S$ est un stable). Soit $T' = T - (L_v \cup \{v\})$ il est clair que $\beta_\circ(T) = \beta_\circ(T') + |L_u|$ et $u \in S \cap T'$. Montrons que T' est $\beta_\circ$-point critique: On suppose que T' n'est pas $\beta_\circ$-point critique, alors il existe une arête wr telle que $\beta_\circ(T'_{wr}) = \beta_\circ(T')$. Dans ce cas ona: $\beta_\circ(T) = \beta_\circ(T') + |L_v|$ et $\beta_\circ(T_{wr}) \geq \beta_\circ(T'_{wr}) + |L_v|$ par conséquent $\beta_\circ(T_{wr}) \geq \beta_\circ(T') + |L_v|$, et par suite $\beta_\circ(T_{wr}) \geq \beta_\circ(T)$, contradiction car T est $\beta_\circ$-point critique. Donc T' est $\beta_\circ$-point critique. Par inductionsur T', $T' \in \mathcal{F}_1$ et donc $T \in \mathcal{F}_1$ car il est obtenu à partir de T' par l'opération θ. ■

2.3 Les graphes sans $K_{1,3}$ β_0-point critiques

Nous donnons par le théorème suivant les graphes sans $K_{1,3}$ β_0-point critiques.

Théorème 11 *Les graphes sans $K_{1,3}$ β_0-point critiques sont les cycles C_{2k} et les chaînes P_{2k+1}.*

Preuve. Soit S un β_0-ensemble de G. Alors d'aprés le théorème 3 $V-S$ est un stable et pour tout $x \in V-S$ on a $d(x) \geq 2$. Comme G est sans $K_{1,3}$ alors $d(x)=2$ pour tout $x \in V-S$ et $d(x) \leq 2$ pour tout $x \in S$, par conséquent $d(x) \leq 2$ pour tout $x \in V$. Donc le graphe G est soit un cycle soit une chaine.

Cas1. $G = C_n$

• Si n est impair, le graphe G n'est pas β_0-point critique car pour tout stable S , $V-S$ n'est pas un stable.

• Si n est pair, alors $\beta_0(C_{2q}) = q$. En contractant une arête uv on obtient le cycle C_{2q-1}, d'ou $\beta_0(G_{uv}) = \beta_0(C_{2q-1}) = q-1$, par conséquent $\beta_0(G_{uv}) < \beta_0(G)$.

Cas2. $G = P_n$

• Si n est pair, alors $\beta_0(G) = \beta_0(P_{2q}) = q$. En contractant une arête uv on obtient le graphe $G_{uv} = P_{2q-1}$, d'ou $\beta_0(G_{uv}) = \beta_0(P_{2q-1}) = q$, par conséquent $\beta_0(G_{uv}) = \beta_0(G)$, contradiction.

• Si n est impair, alors $\beta_0(P_{2q+1}) = q+1$. On contractant une arête uv on obtient le graphe $G_{uv} = P_{2q}$, d'ou $\beta_0(G_{uv}) = \beta_0(P_{2q}) = q$, par conséquent $\beta_0(G_{uv}) < \beta_0(G)$. ■

3 Les graphes i -point critiques

3.1 Définitions et résultats préliminaires

Considérons une partition des sommets d'un graphe connexe $G=(V,E)$ en trois classes disjointes, selon l'effet de la supression d'un sommet v sur le nombre de domination stable $i(G)$.

Soit $V = V_i^o \cup V_i^- \cup V_i^+$, tels que $V_i^o = \{v \in V(G) : i(G-v) = i(G)\}$.
$V_i^- = \{v \in V(G) : i(G-v) < i(G)\}$
$V_i^+ = \{v \in V(G) : i(G-v) > i(G)\}$.

Définition 12 *Un sommet $v \in V$ est dit i -critique si $v \in V_i^-$.*

Définition 13 *Un graphe $G=(V,E)$ est dit i -critique si $V_i^- = V$.*

Par les observations suivantes nous donnons quelques propriétées des ensembles V_i^-, V_i^+ et V_i^o.

Observation 14 *Soient $G=(V,E)$ un graphe et v un sommet de V. Si $v \in V_i^-$ alors il existe un i- ensemble S contenant le sommet v.*

Preuve. Soient S un $i(G-v)$- ensemble. Si v est adjacent à un sommet $x \in S$ alors $i(G-v) \geq i(G)$, contradiction. D'ou v n'admet pas de voisin dans S. Donc $S \cup \{v\}$ est un stable maximal ce qui implique que $i(G) \leq |S|+1$ et puisque $i(G) > |S|$ on a $i(G) = |S|+1$. Par conséquent $S \cup \{v\}$ est un $i(G)$-ensemble contenant v. ■

Observation 15 *Soient $G = (V, E)$ un graphe et v un sommet de V. Alors $v \in V_i^-$ si et seulement s' il existe un i-ensemble S tel que v est sans sommets privés par rapport à S.*

Preuve. ($\Rightarrow$) Supposons que v admet des sommets privés par rapport à tous les $i(G)$-ensembles contenant v. Puisque $v \in V_i^-$, alors pour tout $i(G-v)$-ensemble D, on a $D \cup \{v\}$ est un $i(G)$-ensemble ne contenant aucun privés de v.

($\Leftarrow$) Soit un sommet v sans sommets priveés par rapport à un $i(G)$-ensemble S contenant v. Alors $S' = S - \{v\}$ domine le graphe $G-v$, par conséquent $i(G-v) < i(G)$. D'ou $v \in V_i^-$. ■

Observation 16 *Soit $G = (V, E)$ un graphe .*
1) *Si $v \in V_i^+$ alors v appartient à tout $i(G)$-ensemble S et de plus il admet au moins deux sommets privés non adjacents.*
2) *Si $v \in V_i^+$ alors v n'est adjacent à aucun sommet critique.*

Preuve. (1) Soient S un i-ensemble de G et $v \in V_i^+$. Alors par définition on a $i(G-v) > i(G)$. On suppose qu'il existe un $i(G)$-ensemble S' ne contenant pas le sommet v. Il est clair que $i(G-v) \leq i(G)$, contradiction avec la définition. Comme $i(G-v) > i(G)$, alors v admet des sommets privés dans $V-S$. On suppose que v admet un seul sommet privé dans $V-S$ disons a, alors $D = (S-\{v\}) \cup \{a\}$ est un $i(G)$-ensemble ne contenant pas le sommet v, contradiction.

(2) Supposons qu'il existe un sommet $u \in V_i^-$, d'aprés l'observation 14 il existe un $i(G)$- ensemble D contenant u mais comme $v \in D$ d'aprés (1), Par conséquent D n'est pas un stable, contradiction. ■

Observation 17 *Soient $G = (V, E)$ un graphe et $v \in V$. Si le sommet v n'est pas i-bon, alors $v \in V^o$.*

Preuve. conséquence des observations 14 et 16. ■

3.2 Les graphes i-point critiques tel que $i(G) = 2$

Il est à signaler que la contraction d'une arête d'un graphe connexe $G = (V, E)$ peut augmenter ou diminuer $i(G)$. Par exemple, la contraction de l'arête reliant les deux sommets supports d'une double étoile $S_{r,s}$ diminue $i(G)$, et la contraction de la l'arête uv du graphe G de la figure 1 fait augmenter $i(G)$

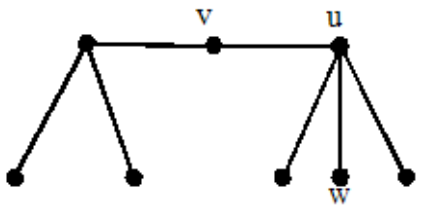


Figure1

Observation 18 *Les graphes P_{3n+1}, $C_{3n+1}, C_n oK_1$ sont i-point critiques.*

Soient $G = (V, E)$ un graphe connexe et S un $i(G)$-ensemble tel que $i(G) = 2$ et soit G_1 le graphe multipartis complet tel que chaque partie contient au moins trois sommets.

La structure des graphes γ-point critiques tels que $\gamma(G) = 2$ est caractérisée par Burton et Sumner [5] comme suit:

Lemme 19 *Soit G un graphe tel que $\gamma(G) = 2$,alors les sommets critiques de G sont exactement ceux qui sont adjacents aux sommets pendants dans $\overline{G}$.*

Théorème 20 (Burton et Sumner [5]) *Soit G un graphe d'ordre $n \geq 4$ avec $\gamma(G) = 2$. Alors G est γ-point critique si et seulement si $\overline{G}$ n'est pas complét, et chaque composante de $\overline{G}$ est soit une couronne soit une clique K_p, $p \geq 2$.*

Le théorème suivant caractérise les graphes totalement γ-point critiques tels que $\gamma(G) = 2$.

Théorème 21 (Burton et Sumner [5]) *Soit G un graphe d'ordre $n \geq 2$ avec $\gamma(G) = 2$. Alors G est totalement γ-point critique si et seulement si chaque composante de $\overline{G}$ est une couronne*

Par les deux théorèmes suivants on caractérise les graphes (totalement) i-point critiques ayant $i(G) = 2$.

Théorème 22 *Un graphe $G = (V, E)$ connexe différent de G_1 est i-point critique si et seulement si G est γ-point critique.*

Preuve. ($\Rightarrow$) Soit un graphe $G = (V, E)$ connexe tel que $i(G) = 2$. Alors aucun sommet de G ne puisse dominé tous les sommets donc $2 \leq \gamma(G) \leq i(G) = 2$, par conséquent $\gamma(G) = 2$. On suppose maintenant que le graphe G est i -point critique, implique $i(G_{ab}) = 1$ pour toute arête $ab \in E$, par conséquent $\gamma(G_{ab}) = 1$ pour toute arête $ab \in E$. D'ou le graphe G est γ-point critique.
($\Leftarrow$) Pour la réciproque. Supposons que le graphe G est γ -point critique, alors $\gamma(G_{ab}) = 1$ pour toute arête $ab \in E$. Par conséquent $i(G_{ab}) = 1$ pour toute arête $ab \in E$ et par suite G est i -point critique. ■

Théorème 23 *Un graphe $G = (V, E)$ connexe est totalement i-point critique si et seulement si G est totalement γ-point critique.*

Preuve. ($\Rightarrow$)Soit $G = (V, E)$ un graphe connexe tel que $i(G) = 2$, alors il est clair que $\gamma(G) = 2$ puisque G ne contient pas de sommet de degré $n - 1$. On suppose maintenant que le graphe G est i totalement point critique, alors $i(G_{ab}) = 1$ pour tout couple de sommets $a, b \in V$, par conséquent $\gamma(G_{ab}) = 1$ pour tout couple de sommets $a, b \in V$. Donc le graphe G est totalement γ point critique.
($\Leftarrow$) Pour la réciproque.Supposons que $\gamma(G_{ab}) = 1$ pour tout couple de sommets $a, b \in V$, alors $i(G_{ab}) = 1$ pour tout couple de sommets $a, b \in V$ et par suite G est totalement i point critique. ■

3.3 Les arbres i -point critiques

Dans ce paragraphe nous commençons par caractériser les arbres i-point critique ayant $i(G) = 2$..

Proposition 24 *Soit T un arbre. T est i-point critique si et seulement si $T = P_4$.*

Preuve. Soit $S = \{x, y\}$ un $i(T)$- ensemble,on note par P_x et P_y les voisins privés dans $V - S$ de x et y respectivement. Puisque T est un arbre P_x et P_y sont des stables. Supposons que $|P_x| \geq 2$. Alors la contraction de l'arête $xx', x' \in P_x$ ne fait pas diminuer $i(G)$, par conséquent $|P_x| \leq 1$ et

similairement $|P_y| \leq 1$. Soit $A \subset V - S$ l'ensemble des sommets communs entre x et y. Il est clair que $|A| \leq 1$ car sinon T contient un cycle, contradiction. On distingue deux cas possibles

Cas1. $|A| = 0$. Alors $|P_x| = |P_y| = 1$ et comme T est un graphe connexe alors $T = P_4$.

Cas2. $|A| = 1$. Alors $T = P_4$ ou $T = P_5$. Le P_5 est exclu car il n'est pas i-point critique.

La réciproque est simple à avoir. ■

Dans [8] Haynes T.W et Henning M.A ont caractérisé les arbres i-excellent comme suit:

Soit $\mathcal{F}$ est la famille des arbres obtenus par la séquence $T_1, T_2, ...T_j$ avec $j \geq 1$ tel que T_1 est un double étoiles $S_{r,r}$ pour $r \geq 1$ et $T = T_i$ et si $i \geq 2$ on a T_{i+1}peut étre obtenu récursivement à partir de T_i par une des deux opérations $\theta_1$où θ_2 tel que on note par $W(T)$ l'ensemble des sommets supports de T , un sommet v est de status A si il est support et de status B s' il est pendant.

Opération θ_1 : L'arbre T_{i+1}est obtenu à partir de T_i en ajoutant une étoile $k_{1,t}$ pour $t \geq 1$ de centre w relie par une arête wy tel que y est un sommet de T_i avec $sta(y) = A$ et en ajoutant aussi $(t-1)$ sommets pendants au sommet y.

Opération θ_2 : L'arbre T_{i+1}est obtenu à partir de $T_i \cup S_{t,t+1}$ en ajoutant une arête wy tel que w est un sommet de la double étoiles adjacent à $t \geq 0$ sommets pendants et y est un sommet de T_i avec $sta(y) = B$.

Théorème 25 (Haynes et Henning [8]) *Un arbre T est i- excellent si et seulement si $T \in \{K_1, K_2\}$ou $T \in \mathcal{F}$.*

Le résultat suivant à été fait en colloboration avec T.W. Haynes.

Théorème 26 *Si un arbre T d'ordre $n \geq 3$ est i- excellent alors T est i -point critique.*

Preuve. On suppose que T est i-excellent et on montre que T est i-point critique .D'aprés le théorème 25 la famille des arbres $T \in F$ est obtenu par une séquence d'arbres $T_1, T_2, ..., T_m$ tel que $T_1 = S_{r,r}$ avec $r \geq 1$ et pour $m \geq 2$, T_{i+1} est obtenu à partir de T_i en ajoutant une étoile $k_{1,t}$ (opération θ_1) ou bien une double étoile $S_{r,r+1}$ (opération θ_2).

On procéde par induction le long d'une séquence de m arbres pour construire l'arbre T.

Si $m = 1$ alors T est une double étoile $S_{r,r}$ avec $r \geq 1$ et par conséquent

T est i-point critique. On suppose, alors que c'est vérifié pour tout arbre $T \in F$ construit par une séquence de m arbre tel que $m \geq 2$.
Soit un arbre $T \in F$ construit par une séquence d'arbre $T_1, T_2, ...T_m$ et soient u et v deux sommets de T tel que $sta(u) = A$ et $sta(v) = B$. Pour des raisons de simplicité on note T_{m-1} par T'. Il est clair que pour tout sommet w de T' tel que $sta(w) = A$ on a $i(T') \leq i(T'_{-w})$, alors on considére deux cas possibles :
Cas.1. T est obtenu à partir de T' par l'opération θ_1.
On suppose que T est obtenu à partir de T' en ajoutant une étoie $k_{1,t}, t \geq 1$ de centre w relier par une arête wy tel que $y \in V(T')$, $star(y) = A$ et $(t-1)$ nouveaux sommets pendants adjaçents à y. Soient L_w l'ensemble des sommets pendants adjacents à w et L_y l'ensemble des sommets pendants adjacents à y. Par induction on a T' est i-point critique.

On montre d'abord que $i(T) = i(T') + t$, il est clair que l'ajout du L_w à tout $i(T')$ ensemble devient un dominant stable de T, alors $i(T) \leq i(T') + t$. Soient S un $i(T)$-ensemble et $S' = S \cap V(T')$, on suppose que $y \in S$, alors $L_w \subset S$ et $S - L_w$ est un dominant stable de T' donc $i(T') \leq |S| - t$. Si $y \notin S$, alors $(L_y \cup \{w\}) \subset S$ et S'est un dominant stable de T'_{-y} alors $i(T') \leq i(T'_{-y}) \leq |S'| = |S| - t$ et par conséquent $i(T') \leq i(T) - t$. D'ou $i(T') + t \leq i(T)$, alors $i(T) = i(T') + t$:

Pour toute arête $uv \in E(T')$ on a $i(T'_{uv}) < i(T')$, soit S' un $i(T'_{uv})$-ensemble pour une arête $uv \in E(T')$, alors $(S' \cup L_w)$ est un dominant stable de T_{uv} implique $i(T'_{uv}) \leq |S'| + |L_w| < i(T') + |L_w|$ donc $i(T'_{uv}) < i(T') + t$, par conséquent $i(T'_{uv}) < i(T')$ pour toute arête $uv \in E(T')$.

Maintenant on montre que toute arête $uv \in (E(T) - E(T'))$ est i-point critique tel que u est un sommet pendant de T avec $stat(u) = B$, donc il existe un ensemble S contenant le sommet u, par conséquent $S - \{u\}$ est un dominant stable de T_{uv} de taille $i(T) - 1$. D'ou $i(T_{uv}) < i(T)$.

Considérons l'arête wy tel que $stat(w) = stat(y) = A$, puisque T est i-excellent, on peut toujours selèctionné un $i(T)$- ensemble S contenant y tel que $w \notin S$ et $L_w \subset S$, dans ce cas $S - L_w$ est un dominant stable de T_{wy} de taille $|S| - |L_w| = i(T) - t$, par conséquent $i(T_{wy}) < i(T)$. D'ou T est i -point critique.

Cas.2. T est un arbre obtenu à partir de T' par l'opération θ_2.
On suppose que T est obtenu à partir de $T' \cup S_{t,t+1}$ tel que wy est une arête avec w est le sommet de $S_{t,t+1}$adjaçent à $t \geq 0$ sommets pendants et $y \in V(T')$ tel que $stat(y) = B$.
Soit z le sommet de $S_{t,t+1}$ adjaçent à $t + 1$ sommets pendants. Si $t \geq 1$ on note par L_w l'ensemble des t sommets pendants adjaçents à w et si $t = 0$ alors $L_w = \emptyset$, L_z c'est l'ensemble des sommets pendants adjacents à z.

Par hypothese T' est i-point critique, on commence d'abord à montrer que
$i(T) = i(T')+t+1$, il est clair qu'en ajoutant $L_w \cup \{z\}$ à tout $i(T')$-ensemble on obtient un dominant stable de T, et par conséquent $i(T) \leq i(T')+t+1$. Soient S un $i(T)$-ensemble et $S' = S \cap V(T')$. On suppose que $w \notin S$, alors $(L_w \cup \{z\}) \subset S$ et $S - L_w - \{z\}$ est un dominant stable de $T,'$ par conséquent $i(T') \leq |S| - (t+1)$. On suppose maintenant que $w \in S$, alors $(L_z \cup \{w\}) \subset S$.

On suppose que $y \notin p_n(w, S)$, alors $(S - L_z - \{w\}) \cup (L_w \cup \{z\})$ est un dominant stable de T de taille $|S| = i(T)$. On suppose maintenant que $y \in p_n(w, S)$, alors $(S' \cup \{y\})$ est un dominant stable de T', par conséquent $i(T') \leq |S'| + 1 = |S| - (t+1)$. alors $i(T') + t + 1 \leq i(T)$. D'ou $i(T) = i(T') + t + 1$.

On montre maintenant que T est i-point critique, pour cela il suffit de montrer que toute arête $uv \in E(T)$ est i-point critique. On suppose que $uv \in E(T')$ et S' un $i(T'_{uv})$-ensemble, puisque $|S'| < i(T')$ par induction, alors $S' \cup L_w \cup \{z\}$ est un dominant stable de T_{uv} et par conséquent $i(T_{uv}) \leq |S'|+t+1 < i(T')+t+1$. Alors toute arête $uv \in E(T')$ est i-point critique. On considére maintenant une arête wx tel que x est un sommet pendant adjaçent à w tel que $L_w \neq \emptyset$. Soit S un $i(T)$-ensemble contenant le sommet z donc $w \notin S$ et $L_w \subset S$, par conséquent $S - \{x]$ est un dominant stable de T_{wx}. Donc $i(T_{wx}) \leq |S| - 1 = i(T) - 1$, d'ou $i(T_{wx}) \leq |S| - 1 = i(T) - 1$ ainsi $i(T_{wx}) \leq i(T)$. Alors l'arête wx est i-point critique. De la même manière pour une arête zx tel que x est un sommet pendant adjacent à z. Finalement on considère l'arête wz, soit S un $i(T)$-ensemble contenant le sommet w alors z n'appartient pas à S, cela implique que $L_z \subset S$. D'ou $S - L_z$ est un dominant stable de T_{wz}, donc $i(T_{wz}) \leq |S| - (t+1) < i(T)$, par conséquent $i(T_{wz}) < i(T)$. Alors T est i-point critique

Pour une caractérisation générale des arbres i -point critiques on propose la conjecture suivante. ■

Conjecture 27 *Un arbre T est i -point critique si et seulement si T est un arbre i-excellent.*

References

[1] C.Berge. Graphs, North holland, 1985.

[2] Haynes T.W, Hedetniemi S.T et Slater P.J, " Fundamentals of Domination in graphs", Marcel Dekker, New York, 1998.

[3] Fricke G.H, Haynes T.W, Hedetniemi S.M, Hedetniemi S. T. et Laskar R.C, "Excellent trees", Bull. Inst. Combin. Appl.34,(2002),27-38.

[4] J.Carrington, F.Harary, TW. Haynes, Changing and unchanging the domination number of graph, J.Combin. Math. Combin Comput. 9(1991) 57-63.

[5] T.Burton, et D.P.Sumner, Domination dot-critical graphs. *Discrete Mathematics* 306 (2006) 11-18

[6] M. Chellali, K. Tablennehas et F. Maffray, Les graphes γ_c-point critiques, communication acceptée pour le colloque international COSI08.

[7] M.Blidia, M.Chellali et O.Favaron, Independence and 2-domination in trees. *Austrlasian Journal of combinatorics* 317-327 (2005).

[8] Haynes T.W et Henning M.A, A characterization of i- excellent trees. *Discrete Mathematics* 248 (2002) 69-77.

A problem of optimal control with free initial condition

Louadj Kahina[1], Aidene Mohamed[1],N.V.Balashevich[2]

[1] University Mouloud Mammeri, Faculty of sciences,
Departement of Mathematics, Tizi-Ouzou, Algeria
[2] Institute of Mathematics, Natinal Academy of Science of Belarus
11 Surganov str., 220072 Minsk, Belarus
louadj_kahina@yahoo.fr, aidene@mail.ummto.dz, balash@im.bas-net.by

Abstract. In our paper, we solve a problem of optimal control with free initial condition, the initial state of the optimized system is not known exactly, a priori information on the initial state is exhausted by inclusion $x_0 \in X_0$. This problem is solved by the adaptive method of R.Gabasov.
Key words: Optimal control, suboptimality, support, Adaptive method

1 Introduction

Problems of optimal control (OC) have been intensively investigated in the world literature for over forty years. During this period, a serie of fundamental results have been obtained, among which should be noted maximum principle [1] and dynamic programming [2]. For many of the problems of the optimal control theory (OCT) adequate solutions are found [4, 5]. Results of the theory were taken up in various fields of science, engineering, and economics.

The aim of this paper is to solve a problem of optimal control with free initial state, the problem has the following sense, the initial state of the optimized system is not known exactly, a priori information on the initial state is exhausted by inclusion $x_0 \in X_0$, by analogy with the theory of filtration, we say that the set X_0 is a priori distribution of the initial state of the control system.

The paper has the following structure. In section 2, the canonical OC problem is formulated. And the definition of support is introduced. Primal and dual ways of its dynamical identification are given. In section 3, we defined support control of the problem. In section 4, We calculate value of suboptimality and suboptimality criterion. In section 5, Optimality and ε-optimality criteria. In section 6, Numerical algorithm for solving the problem ;The iteration consists of three procedures: change of control, change of a support, finally final procedure. In section 7, the results are illustrated by a numerical example.

2 Problem statement

Let us consider a problem of optimal control for a linear dynamic system :

$$c'x(t^*) \to max \tag{1}$$

$$\dot{x} = Ax + bu,\ x(0) = z \in X_0 = \{z \in \mathtt{R}^n, Gz = \gamma, d_* \le z \le d^*\}, \tag{2}$$

$$Hx(t^*) = g, \tag{3}$$

$$f_* \le u(t) \le f^*, t \in T = [0, t^*]. \tag{4}$$

Here $x \in \mathtt{R}^n$ is a state of control system $(2)-(4)$; $u(.) = (u(t), t \in T)$, $T = [0, t^*]$, is a piecewise continuous function; $A \in \mathtt{R}^{n\times n}$; $b,\ c \in \mathtt{R}^n$; $g \in \mathtt{R}^m$, $H \in \mathtt{R}^{m\times n}$, $rank\ H = m < n$; f_*, f^* are scalars; $d_* = d_*(J) = (d_{*j}, j \in J)$, $d^* = d^*(J) = (d^*_j, j \in J)$ are n-vectors; $G \in \mathtt{R}^{l\times n}$, $rank\ G = l < n$, $\gamma \in \mathtt{R}^l$, $I = \{1, ..., m\}$, $J = \{1, ..., n\}$, $L = \{1, ..., l\}$ are sets of indices.
By using the Cauchy formula, the solution of the system (2) can be written in the form:

$$x(t) = F(t)(z + \int_0^t F^{-1}(\vartheta)bu(\vartheta)d\vartheta), t \in T, \tag{5}$$

where $F(t) = e^{At}, t \in [0, t^*]$ is defined by the relations :

$$\begin{cases} \dot{F}(t) = AF(t)\ , \\ F(0) = I_n \quad . \end{cases}$$

Substituting (5) into $(1) - (4)$, problem (1)-(4) takes the following equivalent form:

$$\tilde{c}'z + \int_0^{t^*} c(t)u(t)dt \longrightarrow max, \tag{6}$$

$$D(I, J)z + \int_0^{t^*} \varphi(t)u(t)dt = g, \tag{7}$$

$$G(L, J)z = \gamma,\ d_* \le z \le d^*, \tag{8}$$

$$f_* \le u(t) \le f^*, t \in T, \tag{9}$$

where $\tilde{c}' = c'F(t^*),\ c(t) = c'F(t^*)F^{-1}(t)b,\ D(I, J) = HF(t^*),\ \varphi(t) = HF(t^*)F^{-1}(t)b.$

A pair $v = (z, u(.))$ formed of an $n-$vector z and a piecewise continuous function $u(.)$ is called a generalized control.

The constraint (3) is assumed to be controllable, i.e. for any m-vector g, there exists a pair v, for which the equality (3) is fulfilled.

A generalized control $v = (z, u(.))$ is said to be an admissible control if it satisfies the constraints (2)-(4).

An admissible control $v^0 = (z^0, u^0(.))$ is said to be an optimal open-loop control if

$$J(v^0) = \max_v J(v).$$

For a given $\varepsilon \ge 0$, an admissible control $v^\varepsilon = (z^\varepsilon, u^\varepsilon(.))$ is said to be an $\varepsilon-$optimal control if:

$$J(v^0) - J(v^\varepsilon) \le \varepsilon.$$

3 Support control

Let us introduce a discretized time set $T_h = \{0, h, \dots, t^* - h\}$ where $h = t^*/N$, N is an integer. A function $u(t)$, $t \in T$, is called a discrete controls if

$$u(t) = u(\tau),\ t \in [\tau, \tau + h), \tau \in T_h.$$

Let us discretize a problem (6)-(9), we obtain a problem

$$\tilde{c}'z + \sum_{t \in T_h} q(t)u(t) \to max, \tag{10}$$

$$D(I, J)z + \sum_{t \in T_h} d(t)u(t) = g, \tag{11}$$

$$G(L, J)z = \gamma, d_* \le z \le d^*, \tag{12}$$

$$f_* \le u(t) \le f^*, t \in T, \tag{13}$$

where $d(t) = \int_t^{t+h} \varphi(\vartheta)d\vartheta,\ q(t) = \int_t^{t+h} c(\vartheta)d\vartheta,\ t \in T_h$.
First we describe a method of computing the solution of problem (1)-(4) in the class of discrete control, and then we present the final procedure which uses this solution as an initial approximation for solving problem (1)-(4) in the class of piecewise continuous functions. Choose an arbitrary subset $T_B \subset T_h$ of $k \le m$ elements and an arbitrary subset $J_B \subset J$ of $m+l-k$ elements. Form the matrix

$$P_B = \begin{pmatrix} D(I, J_B) & d(t), t \in T_B \\ G(L, J_B) & 0 \end{pmatrix} \tag{14}$$

A set $S_B = \{T_B, J_B\}$ is said to be a support of problem $(1)-(4)$ if $detP_B \neq 0$. A pair $\{v, S_B\}$ of an admissible control $v = (z, u(.))$ and a support S_B is said to be a support control. A support control $\{v, S_B\}$ is said to be not degenerate if $d_{*j} < z_j < d_j^*, j \in J_B, f_* < u(t) < f^*, t \in T_B$.
Let us consider another admissible control $\overline{v} = (\overline{z}, \overline{u}(.)) = v + \Delta v$, where $\overline{z} = z + \Delta z, \overline{u}(t) = u(t) + \Delta u(t), t \in T$, and let us calculate the increment of the cost functional:

$$\Delta J(v) = J(\overline{v}) - J(v) = \tilde{c}'\Delta z + \sum_{t \in T_h} q(t)\Delta u(t).$$

Since

$$D(I, J)\Delta z + \sum_{t \in T_h} d(t)\Delta u(t) = 0,$$

and

$$G(L, J)\Delta z = 0,$$

then the increment of the functional is equal:

$$\Delta J(v) = (\tilde{c}' - \nu' \begin{pmatrix} D(I, J) \\ G(L, J) \end{pmatrix})\Delta z + \sum_{t \in T_h} (q(t) - \nu_u d(t))\Delta u(t),$$

where $\nu = \begin{pmatrix} \nu_u \\ \nu_z \end{pmatrix} \in R^{m+l}, \nu_u \in R^m, \nu_z \in R^l$ is a function of the Lagrange multipliers called potentials, calculated as a solution to the equation: $\nu' = q'_B Q$, where $Q = P_B^{-1}, q_B = (\tilde{c}_j, j \in J_B, q(t), t \in T_B)$. Introduce an n-vector of estimates $\Delta' = \nu' \begin{pmatrix} D(I,J) \\ G(L,J) \end{pmatrix} - \tilde{c}'$, and a function of cocontrol $\Delta(.) = (\Delta(t) = \nu'_u d(t) - q(t), t \in T_h)$.
By using this vectors, the cost of functional increment takes the form:

$$\Delta J(v) = \Delta' \Delta z - \sum_{t \in T_h} \Delta(t) \Delta u(t). \tag{15}$$

A support control $\{v, S_B\}$ is dually not degenerate if $\Delta(t) \neq 0, t \in T_H, \Delta_j \neq 0, j \in J_H$, where $T_H = T_h / T_B,\ J_H = J / J_B$.

4 Calculation of the value of suboptimality

The new control $\overline{v}(t)$ is admissible, if it satisfies the constraints:

$$d_* - z \leq \Delta z \leq d^* - z;\ f_* - u(t) \leq \Delta u(t) \leq f^* - u(t), t \in T. \tag{16}$$

The maximum of functional (15) under constraints (16) is reached for:

$$\begin{cases} \Delta z_j = d_{*j} - z_j & \text{if } \Delta_j > 0 \\ \Delta z_j = d_j^* - z_j & \text{if } \Delta_j < 0 \\ d_{*j} - z_j \leq \Delta z_j \leq d_j^* - z_j, \text{ if } \Delta_j = 0,\ j \in J. \end{cases}$$

$$\begin{cases} \Delta u(t) = f_* - u(t) \text{ if } \Delta(t) > 0 \\ \Delta u(t) = f^* - u(t) \text{ if } \Delta(t) < 0 \\ f_* \leq \Delta u(t) \leq f^*, \text{ if } \Delta(t) = 0,\ t \in T_h, \end{cases}$$

and is equal to:

$$\beta = \beta(v, S_B) = \sum_{j \in J_H^+} \Delta_j (z_j - d_{*j}) + \sum_{j \in J_H^-} \Delta_j (z_j - d_j^*)$$

$$+ \sum_{t \in T^+} \Delta(t)(u(t) - f_*) + \sum_{t \in T^-} \Delta(t)(u(t) - f^*)$$

where

$$T^+ = \{t \in T_H, \Delta(t) > 0\}, T^- = \{t \in T_H, \Delta(t) < 0\},$$

$$J_H^+ = \{j \in J_H, \Delta_j > 0\}, J_H^- = \{j \in J_H, \Delta_j < 0\}.$$

The number $\beta(v, S_B)$ is called a value of suboptimality of the support control $\{v, S_B\}$.
From there, $J(\overline{v}) - J(v) \leq \beta(v, S_B)$. Of this last inequality, the following result is deduced:

5 Optimality and ε-optimality criterion

Theorem 1. *The following relations:*

$$\begin{cases} u(t) = f_*, & \text{if } \Delta(t) > 0 \\ u(t) = f^*, & \text{if } \Delta(t) < 0 \\ f_* \le u(t) \le f^*, & \text{if } \Delta(t) = 0,\ t \in T_h \\ z_j = d_{*j}, & \text{if } \Delta_j > 0 \\ z_j = d_j^*, & \text{if } \Delta_j < 0 \\ d_{*j} \le z_j \le d_j^*, & \text{if } \Delta_j = 0,\ j \in J. \end{cases}$$

are sufficient,and in the cases of non-degeneracy, they are necessary for the optimality of control v.

Theorem 2. *For any $\varepsilon \ge 0$, the admissible control v is $\varepsilon-$optimal if and only if there exists a support S_B such that $\beta(v, S_B) \le \varepsilon$.*

6 Numerical algorithm for solving the problem

Assume $\varepsilon > 0$ is a given number and $\{v, S_B\}$ is a known support control that does not satisfy optimality and $\varepsilon-$ optimality criterion. The method suggested is iterative, its aim is to construct an $\varepsilon-$ solution of problem $(1)-(4)$. As a support will be changing during the iterations together with an admissible control it is natural to consider them as a pair. The iteration of the method is to change the "old" support control $\{v, S_B\}$ for the "new" $\{\overline{v}, \overline{S}_B\}$ so that $\beta(v, S_B) \ge \beta(\overline{v}, \overline{S}_B)$. The iteration consists of two procedures

1. Change of an admissible control $v \to \overline{v}$.
2. Change of support

A construction of the initial support concerns with the first phase and can be solved using the algorithm described below.
At the beginning of each iteration the following information is stored:

1. An admissible control v.
2. A support $S_B = \{T_B, J_B\}$.
3. A value of suboptimality $\beta = \beta(v, S_B)$.

6.1 Change of control.

Consider a beginning support control $\{v, S_B\}$ and $\overline{v} = (\overline{z}, \overline{u})$ a new admissible control constructed by the formulas:

$$\begin{cases} \overline{z}_j = z_j + \theta^0 l_j, & j \in J \\ \overline{u}(t) = u(t) + \theta^0 l(t), & t \in T_h \end{cases} \tag{17}$$

where $l = (l_j, j \in J, l(t), t \in T_h)$ is an admissible direction of changing a control v; θ^0 is the maximum step along this direction.

Construct the admissible direction. Let us introduce a pseudocontrol $\widetilde{v} = (\widetilde{z}, \widetilde{u}(t), t \in T)$. First, we compute the nonsupport values of a pseudocontrol

$$\widetilde{z}_j = \begin{cases} d_{j*} \text{ if } \Delta_j \geq 0 \\ d_j^* \text{ if } \Delta_j \leq 0,\ j \in J_H; \end{cases} \qquad \widetilde{u}(t) = \begin{cases} f^* \text{ if } \Delta(t) \leq 0, \\ f_* \text{ if } \Delta(t) \geq 0,\ t \in T_H. \end{cases}$$

Support values of a pseudocontrol $\{\tilde{z}_j, j \in J_B; \tilde{u}(t), t \in T_B\}$ are computed from the equations

$$\sum_{j \in J_B} D(I,j)\tilde{z}_j + \sum_{t \in T_B} d(t)\tilde{u}(t) = g - \sum_{j \in J_H} D(I,j)\tilde{z}_j - \sum_{t \in T_H} d(t)\tilde{u}(t).$$

$$\sum_{j \in J_B} G(L,j)\tilde{z}_j = \gamma - \sum_{j \in J_H} G(L,j)\tilde{z}_j.$$

With the use of a pseudocontrol we compute the admissible direction l:
$l_j = \tilde{z}_j - z_j$, $j \in J$; $l(t) = \tilde{u}(t) - u(t)$, $t \in T_h$.
Construct the maximal step. Since $\overline{v}$ is to be admissible, the following inequalities are to be satisfied:

$$d_* \leq \overline{z} \leq d^*; \quad f_* \leq \overline{u}(t) \leq f^*,\ t \in T_h,$$

i.e.

$$d_* \leq z_j + \theta^0 l_j \leq d^*,\ j \in J;$$

$$f_* \leq u(t) + \theta^0 l(t) \leq f^*,\ t \in T_h.$$

Thus the maximal step θ^0 is chosen as $\theta^0 = \min\{1; \theta(t_0); \theta_{j_0}\}$. Here $\theta_{j_0} = min\theta_j$:

$$\theta_j = \begin{cases} \frac{d_j^* - z_j}{l_j}, & \text{if } l_j > 0 \\ \frac{d_{*j} - z_j}{l_j}, & \text{if } l_j < 0 \\ +\infty, & \text{if } l_j = 0,\ j \in J_B. \end{cases}$$

$\theta(t_0) = \min_{t \in T_B} \theta(t)$:

$$\theta(t) = \begin{cases} \frac{f^* - u(t)}{l(t)}, \text{ if } l(t) > 0 \\ \frac{f_* - u(t)}{l(t)}, \text{ if } l(t) < 0 \\ +\infty, \quad \text{if } l(t) = 0,\ t \in T_B. \end{cases}$$

Let us calculate the value of suboptimality of the support control $\{\overline{v}, S_B\}$ with $\bar{v}$ computed according to (17): $\beta(\overline{v}, S_B) = (1 - \theta^0)\beta(v, S_B)$.
Consequently
If $\theta^0 = 1$, then $\overline{v}$ is an optimal control.
If $\beta(\overline{v}, S_B) \leq \varepsilon$, then $\overline{v}$ is an ε-optimal control.
If $\beta(\overline{v}, S_B) > \varepsilon$, then we perform a change of support.

6.2 Change of support.

The change of support $S_B \to \overline{S}_B$ will be made so that $\beta(\overline{v}, S_B) > \beta(\overline{v}, \overline{S}_B)$. Here, we have $\theta^0 = \min(\theta(t_0), t_0 \in T_B;\ \theta_{j_0}, j_0 \in J_B)$. We will distinguish between two cases which can occur after the first procedure:

a) $\theta^0 = \theta_{j_0}, j_0 \in J_B$.
b) $\theta^0 = \theta(t_0), t_0 \in T_B$.

Each case is investigated separately.
This change is based on variation of potentials, estimates and cocontrol:

$$\nu^{'} = \nu + \Delta\nu;\ \overline{\Delta}_j = \Delta_j + \sigma^0\delta_j,\ j \in J;\ \overline{\Delta}(t) = \Delta(t) + \sigma^0\delta(t),\ t \in T_h. \qquad (18)$$

where $(\delta_j, j \in J, \delta(t), t \in T_h)$ is an admissible direction of change $(\Delta, \Delta(\cdot))$ and σ^0 is a maximal step along this direction.
Construct an admissible direction $(\delta_j, j \in J, \delta(t), t \in T_h)$**.** First, construct the support values $\delta_B = (\delta_j, j \in J_B, \delta(t), t \in T_B)$ of admissible direction for each case:
a)Case $\theta^0 = \theta_{j0}$. Let us put:

$$\begin{cases} \delta(t) = 0 & \text{if } t \in T_B \\ \delta_j = 0 & \text{if } j \neq j_0, j \in J_B \\ \delta_{j_0} = 1 & \text{if } \overline{z}_{j_0} = d_{*j0} \\ \delta_{j_0} = -1 & \text{if } \overline{z}_{j_0} = d^*_{j_0} \end{cases}$$

b)Case $\theta^0 = \theta(t_0)$. Let us put:

$$\begin{cases} \delta_j = 0 & \text{if } j \in J_B \\ \delta(t) = 0 & \text{if } t \in T_B/t_0 \\ \delta(t_0) = 1 & \text{if } \overline{u}(t_0) = f_* \\ \delta(t_0) = -1 & \text{if } \overline{u}(t_0) = f^* \end{cases}$$

Using the values δ_B, we compute the variation $\Delta\nu = \begin{pmatrix} \Delta\nu_u \\ \Delta\nu_z \end{pmatrix}$ of potentials as $\Delta\nu' = \delta'_B Q$. Finally, we get the variation of non support components of the estimates and the cocontrol:

$$(\delta_j,\ j \in J_H) = \Delta\nu' \begin{pmatrix} D(I,j) \\ G(L,j) \end{pmatrix},$$

$$(\delta(t),\ t \in T_H) = \Delta\nu'_u(d(t),\ t \in T_H).$$

Construct a maximal step σ^0**.** A maximal step equals $\sigma^0 = \min(\sigma^0_j, \sigma^0_t)$ where $\sigma^0_j = \sigma_{j_1} = \min \sigma_j, j \in J_H;\ \sigma^0_t = \sigma(t_1) = \min \sigma(t), t \in T_H$,
where

$$\sigma_j = \begin{cases} -\Delta_j/\delta_j \text{ if } \Delta_j\delta_j < 0, \\ +\infty \quad \text{if } \Delta_j\delta_j \geq 0, \end{cases} \quad j \in J_H,$$

and

$$\sigma(t) = \begin{cases} -\Delta(t)/\delta(t) \text{ if } \Delta(t)\delta(t) < 0, \\ +\infty \qquad\quad \text{if } \Delta(t)\delta(t) \geq 0, \quad t \in T_H. \end{cases}$$

Construct a new support. For constructing a new support, we consider four following cases: 1. $\theta^0 = \theta(t_0),\ \sigma^0 = \sigma(t_1)$.
A new support $\overline{S}_B = \{\overline{T}_B, \overline{J}_B\}$ becomes:

$$\overline{T}_B = T_B/\{t_0\} \cup \{t_1\}, \overline{J}_B = J_B.$$

2. $\theta^0 = \theta(t_0),\ \sigma^0 = \sigma_{j_1}$.
A new support $\overline{S}_B = \{\overline{T}_B, \overline{J}_B\}$ where:

$$\overline{T}_B = T_B/\{t_0\}, \overline{J}_B = J_B \cup \{j_1\}.$$

3. $\theta^0 = \theta_{j_0},\ \sigma^0 = \sigma_{j_1}$.
A new support $\overline{S}_B = \{\overline{T}_B, \overline{J}_B\}$ becomes:

$$\overline{T}_B = T_B, \overline{J}_B = J_B/\{j_0\} \cup \{j_1\}.$$

4. $\theta^0 = \theta_{j_0},\ \sigma^0 = \sigma(t_1)$.
A new support $\overline{S}_B = \{\overline{T}_B, \overline{J}_B\}$ where:

$$\overline{T}_B = T_B \cup \{t_1\}, \overline{J}_B = J_B/\{j_0\}.$$

A value of suboptimality for support control $\beta(\overline{v}, \overline{S}_B)$ is equal to:

$$\beta(\overline{v}, \overline{S}_B) = (1 - \theta^0)\beta(v, S_B) + \alpha\sigma^0$$

where

$$\alpha = \begin{cases} -|\tilde{z}_{j_0} - \bar{z}_{j_0}|, \qquad \text{if } \theta^0 = \theta_{j_0}, \\ -|\tilde{u}(t_0) - \bar{u}(t_0)|, \text{ if } \theta^0 = \theta(t_0). \end{cases}$$

1. If $\beta(\overline{v}, \overline{S}_B) = 0$, then the control $\overline{v}$ is optimal for problem (1)-(4) in the class of discrete controls.
2. If $\beta(\overline{v}, \overline{S}_B) < \varepsilon$, then the control $\overline{v}$ is ε-optimal for problem (1)-(4) in the class of discrete controls.
3. If $\beta(\overline{v}, \overline{S}_B) > \varepsilon$, then we perform the next iteration starting from the support control $\{\overline{v}, \overline{S}_B\}$.

If we would like to get the solution of problem (1)-(4) in the class of piecewise continuous control, we pass to the final procedure when the case 1 or 2 takes place.

6.3 final procedure.

By using a support $\overline{S}_B$, we construct a quasicontrol $\widehat{v} = (\widehat{z}, \widehat{u}(t), t \in T)$:

$$\widehat{z}_j = \begin{cases} d_{j*} & \text{if } \Delta_j > 0 \\ d_j^* & \text{if } \Delta_j < 0 \\ \in [d_{j*}, d_j^*] & \text{if } \Delta_j = 0,\ j \in J \end{cases} \qquad \widehat{u}(t) = \begin{cases} f_*, & \text{if } \Delta(t) < 0 \\ f^*, & \text{if } \Delta(t) > 0, \\ \in [f_*, f^*] & \text{if } \Delta(t) = 0,\ t \in T. \end{cases}$$

If

$$D(I,J)\widehat{z} + \int_0^{t^*} \varphi(t)\widehat{u}(t)dt = g,$$

$$G(L,J)\widehat{z} = \gamma,$$

then $\widehat{v}$ is optimal control, and if

$$D(I,J)\widehat{z} + \int_0^{t^*} \varphi(t)\widehat{u}(t)dt \neq g, \quad G(L,J)\widehat{z} \neq \gamma, \tag{19}$$

then denote $T^0 = \{t_i,\ i = \overline{1,s}\},\ s = |T_B|$. Here, $t_i,\ i = \overline{1,s}$ are zeroes of the optimal cocontrol $\Delta(t) = 0, t \in T;\ t_0 = 0,\ t_{s+1} = t^*$. Assume

$$\dot{\Delta}(t_i) \neq 0,\ i = \overline{1,s}.$$

From system (19)Let us construct the following function:

$$f(\Theta) = \begin{pmatrix} D(I,J_B)z(J_B) + D(I,J_H)z(J_H) + \sum_{i=0}^{s}(\frac{f^*+f_*}{2} - \frac{f^*-f_*}{2} sign\dot{\Delta}(t_i)) \int_{t_i}^{t_{i+1}} \varphi(t)dt - g \\ G(L,J_B)z(J_B) + G(L,J_H)z(J_H) - \gamma \end{pmatrix}$$

where

$$z_j = \frac{d_j^* + d_j*}{2} - \frac{d_j^* - d_j*}{2} sign\Delta_j,\ j \in J_H.$$

$$\Theta = (t_i,\ i = \overline{1,s}; z_j, j \in J_B).$$

The final procedure consists to find the solution

$$\Theta^0 = (t_i^0, i = \overline{1,s}; z_j^0, j \in J_B)$$

of the system of $m + l-$ nonlinear equations

$$f(\Theta) = 0. \tag{20}$$

We solve this system by the Newton method using an initial approximation :

$$\Theta^{(0)} = (\overline{t}_i,\ i = \overline{1,s}; \overline{z}_j, j \in J_B).$$

The $(k+1)^{th}$ approximation $\Theta^{(k+1)}$,at a step $k+1 \geq 1$, is equal:

$$\Theta^{(k+1)} = \Theta^{(k)} + \Delta\Theta^{(k)} \qquad \Delta\Theta^{(k)} = -\frac{\partial f^{-1}(\Theta^{(k)})}{\partial \Theta^{(k)}} \cdot f(\Theta^{(k)}),$$

where

$$\frac{\partial f(\Theta^{(k)})}{\partial \Theta^{(k)}} = \begin{pmatrix} D(I, J_B) & (f_* - f^*) sign\dot{\Delta}(t_i^{(k)})\varphi(t_i^{(k)}),\ i = \overline{1,s} \\ G(L, J_B) & 0 \end{pmatrix}.$$

As $detP_B \neq 0$, we can easily show that

$$det\frac{\partial f(\Theta^{(0)})}{\partial \Theta^{(0)}} \neq 0. \tag{21}$$

For all moment instants $t_i \in T_B$, there exists a small $\mu > 0$ such that $\tilde{t}_i \in [t_i - \mu, t_i + \mu], i = \overline{1,s}$, the matrix $(\varphi(\tilde{t}_i), i = \overline{1,s})$ is not degenerate and the matrix $\frac{\partial f(\Theta^{(k)})}{\partial \Theta^{(k)}}$ are not degenerate. If elements $t_i^{(k)}, i = \overline{1,s}, k = 1, 2, ...$ do not leave the μ-vicinity of t_i, $i = \overline{1,s}$. Vector $\Theta^{(k^*)}$ is taken as solution of equation (20) if

$$\| f(\Theta^{(k^*)}) \| \leq \eta,$$

for a given $\eta > 0$. So we put $\theta^0 = \theta^{(k^*)}$. The suboptimal control for problem (1)-(4) is computed as

$$z_j^0 = \begin{cases} z_j^0,\ j \in J_B \\ \widehat{z}_j,\ j \in J_H; \end{cases}$$

$$u^0(t) = \frac{f^* + f_*}{2} - \frac{f^* - f_*}{2} sign\dot{\Delta}(t_i^0),\ t \in [t_i^0, t_{i+1}^0[,\ i = \overline{1,s}.$$

If the Newton method does not converge, we decrease the parameter $h > 0$ and perform the iterative process again.

7 Example

We illustrate the results obtained in this paper using the following example:

$$\begin{cases} \int_0^{25} u(t)dt \to min, \dot{x}_1 = x_3, \dot{x}_2 = x_4, \\ \dot{x}_3 = -x_1 + x_2 + u, \dot{x}_4 = 0.1x_1 - 1.01x_2, \\ x_1(25) = x_2(25) = x_3(25) = x_4(25) = 0, \\ 0 \leq u(t) \leq 1, t \in [0, 25]. \end{cases} \tag{22}$$

Let be the matrix:
$H = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}, g = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, G = \begin{pmatrix} 0.01 & -1.02 & 0.03 & 1 \\ -0.5 & 0.5 & 0 & 1 \\ -2 & 0.5 & 3.01 & 1.2 \\ 1 & 0.3 & 0.01 & -1 \end{pmatrix}, \gamma = \begin{pmatrix} 0.3 \\ -0.05 \\ 0.2 \\ -0.04 \end{pmatrix}, d_* = \begin{pmatrix} -1.5 \\ -1.5 \\ -1.5 \\ -1.5 \end{pmatrix}, d^* = \begin{pmatrix} 1.5 \\ 1.5 \\ 1.5 \\ 1.5 \end{pmatrix}.$
Let us the initial condition as :$x_1(0) = 0.1, x_2(0) = 0.25, x_3(0) = 2, x_4(0) = 1$.

Problem (22) is reduced to canonical form $(1)-(4)$ by introducing the new variable $\dot{x}_5 = u, x_5(0) = 0$. Then, the control criterion takes the form $-x_5(t^*) \to max$. In the class of discrete controls with quantization period $h = 25/1000 = 0.025$, problem (22) is equivalent to LP problem of dimension 4×1000.
To construct the optimal open-loop control of problem (22). As an initial support, a set $T_B = \{5, 10, 15, 20\}$ was selected. This support corresponds to the set nonsupport zeroes of the cocontrol $T_{n0} = \{3.725, 9.725, 15.3, 21.3\}$. Movements of the cocontrol $\Delta(t)$ in the course of iterations are pictured in Fig.1. The optimal value of the control criterion was equal 6.60205016.

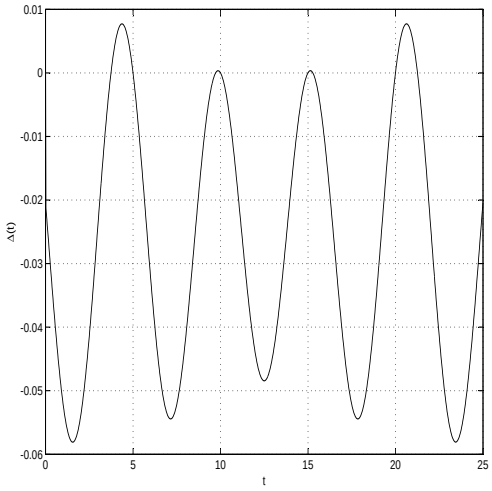


Fig. 1.

The table contains some information on the solution to problem (22) for other quantization periods.

h	number of iterations	value of the control criterion
0.25	11	6.6243433
0.025	18	6.602499
0.0025	26	6.602054
0.001	32	6.602050

Of course, one can solve problem (22) by the LP methods transforming the problem to form $(10)-(13)$. In doing so, one integration of the system is sufficient to form the matrix of the LP problem. However, such "static" approach is concerned with a large volume of required operative memory, and it is fundamentally different from the traditional" dynamical' approaches based on dynamical

models $(1)-(4)$. Then, the problem $(1)-(4)$ was solved.

In Figure 2 , projections of transients of system (22) closed by optimal feedback on planes x_1x_3.In Figure 3 , projections of transients of system (22) closed by optimal feedback on planes x_2x_4 are presented. The realization $u^*(\tau), \tau \in T_h$ is given in Figure 4

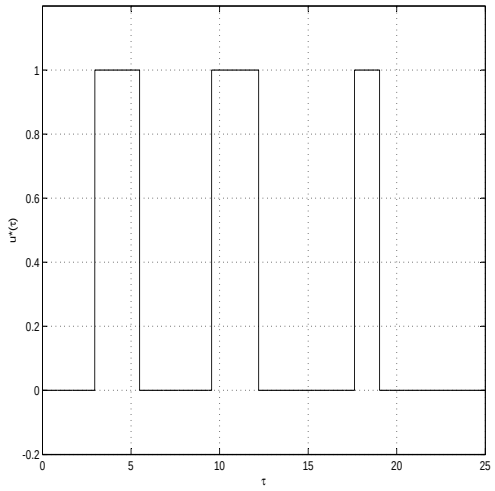


Fig. 2.

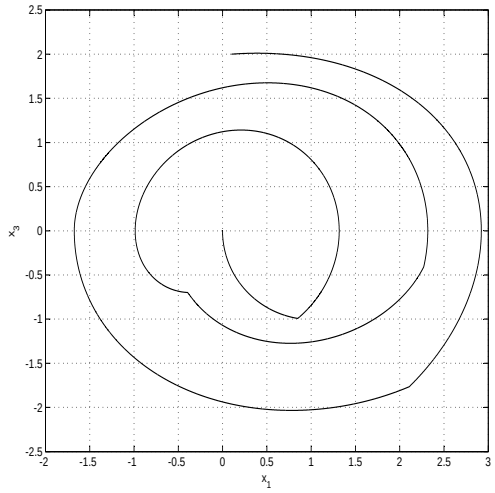


Fig. 3.

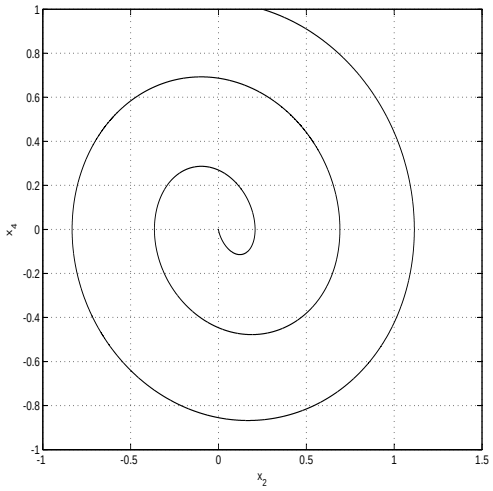


Fig. 4.

8 Conclusion.

The problem of control optimal control with free initial state is studied in this paper. We have used an iterative algorithm to found an optimal support and a solution $\varepsilon-$ optimal. The results presented in the paper have shown by an example.

References

1. L.S. Pontryagin, V.G. Boltyanski, R.V. Gamkrelidze, and E.F. Mischenko, The Mathematical Theory of Optimal Processes, Interscience Publishers, New York, 1962.
2. R.E. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, 1963.
3. R.E. Bellman, I. Glicksberg, and O.A. Gross, Some Aspects of the Mathematical Theory of Control Processes, Report R-313, Rand Corporation, Santa Monica, CA, 1958.
4. A.E. Bryson and Yu-Chi Ho, Applied Optimal Control, Blaisdell, Toronto,Canada, 1969.

5. E.B. Lee and L. Markus, Foundations of Optimal Control Theory, Wiley and Sons, New York, 1967.
6. Balashevich, N.V., Gabasov, R., and Kirillova, F.M., Numerical Methods of Program and Positional Optimization of the Linear Control Systems, Zh. Vychisl. Mat. Mat. Fiz., 2000, vol. 40, no. 6, pp. 838-859.
7. M.Aidene, I.L.Vorob'ev, B.Oukacha, Algorithm for Solving a Linear Optimal Control Problem with Minimax Performance Index, Computational Mathematics and mathematical Physics, Vol.45, No.10, 2005,pp. 1691-1700.
8. R.Gabasov, F.Kirillova, and N.V.Balashevich, On The Synthesis Problem For Optimal Control Systems. SIAM J. Control OPTIM,2000, Vol.39, No.4, pp.1008-1042.

This article was processed using the LaTeX macro package with LLNCS style

Un système Neuro-flou Basé sur les Invariants de Hu pour la Reconnaissance Hors-ligne de Mots Arabes Manuscrits

Leila Chergui[1], Maamar Kef[2], Mohammed Benmohammed[3]

[1]Université Larbi Ben Mhidi, Département d'informatique
Oum El Bouaghi – Algérie
pgliela@yahoo.fr
[2]Université Larbi Ben Mhidi, Département d'informatique
Oum El Bouaghi – Algérie
Lm_kef@yahoo.fr
[3]Université Mentouri Département d'informatique
Constantine – Algérie
Ibnm@yahoo.fr

Résumé. La reconnaissance de l'écriture arabe manuscrite est un domaine de recherche relativement récent et qui a connu ces dernières années des progrès remarquables. Il présente un intérêt indéniable dans l'accomplissement de tâches considérées fastidieuses dans certains domaines comme le tri postal, la lecture de chèques bancaires, la lecture des bordereaux, etc. Ce papier présente la conception, la réalisation et l'évaluation d'un système dédié à la reconnaissance automatique hors-ligne de mots manuscrits arabes représentant des noms de villes tunisiennes tirés de la base IFN/ENIT. Dans ce travail, nous nous pencherons sur une approche basée sur l'utilisation des invariants de Hu, et d'un classifieur neuronal utilisé pour la première fois dans ce domaine, à savoir le réseau Fuzzy ART. Nous montrerons, à travers les différentes étapes considérées, l'apport de notre technique dans la résolution des problèmes liés au traitement de l'écriture arabe. Par ailleurs, nous retenons les limitations enregistrées. Les résultats obtenus sont prometteurs.

Mots clés : Mots arabes manuscrits, Reconnaissance, Squelettisation, Invariants de Hu, Réseau Fuzzy ART.

1 Introduction

La reconnaissance de formes (RF) est un domaine important du monde informatique dans lequel les recherches sont particulièrement actives. Elle est historiquement un chapitre de l'intelligence artificielle qui vise à automatiser le discernement de situations typiques au niveau de la perception. Ses méthodes trouvent des applications nombreuses dans divers domaines tels que : la médecine, le contrôle de procédés de fabrication, la vision robotique, le traitement de données volumineuses d'images, la reconnaissance de la parole et la lecture optique de documents.

La reconnaissance automatique de caractères, au sens large du terme, est une discipline qui a vu le jour dés l'apparition des premiers ordinateurs. Reconnaître de l'écriture manuscrite consiste à associer une représentation symbolique à une séquence de symboles graphiques : on parle aussi de lecture automatique. Le but est de pouvoir utiliser cette représentation dans une application informatique. On distingue deux grands types d'utilisation :

- Traiter automatiquement des documents contenant de l'écriture manuscrite dont l'analyse par des individus prend trop de temps.
- Faciliter l'utilisation des ordinateurs pour des applications où un stylo est plus pratique qu'un clavier et une souris.

Contrairement au Latin, la reconnaissance de l'écriture arabe manuscrite reste encore aujourd'hui au niveau de la recherche et de l'expérimentation. Cependant et depuis quelques années elle a pris un nouvel essor et fait l'objet d'applications de plus en plus nombreuses. Parmi ces applications, nous citons le traitement automatique des dossiers administratifs, des formulaires d'enquêtes, des chèques bancaires, numérisation et sauvegarde du patrimoine culturel écrit, etc.

Notre article est organisé comme suit : la deuxième section abordera les caractéristiques de l'écriture arabe ainsi que les travaux effectués dans ce domaine. La troisième mettra le point sur l'architecture du réseau Fuzzy ART. Le principe du système sera détaillé dans la quatrième section. Le tout sera clôturé par une liste de perspectives.

2 L'écriture Arabe

L'écriture arabe a vu le jour aux alentours du VI ème siècle avant l'apparition de l'écriture cursive nabatéenne, et s'est progressivement répandue avec l'existence de l'Islam et la révélation coranique. L'arabe appartient au groupe des écritures sémitiques consonantiques du fait que seules les consonnes sont représentées. Les principales caractéristiques de la langue arabe sont :

- L'alphabet arabe comprend vingt-huit lettres fondamentales. Contrairement à l'alphabet latin, chacune des lettres arabes peut être écrite sous plusieurs formes suivant sa place dans le mot : début, milieu, fin ou isolée. La plupart des lettres ont ainsi quatre représentations différentes, excepté certaines lettres pour lesquelles les positions de milieu et de fin n'existent pas, ces lettres n'ont alors que deux représentations. La figure 1 donne toutes les formes possibles pour chaque lettre de l'alphabet arabe.
- Quelques caractères arabes incluent dans leur forme un, deux ou trois points diacritiques. Ces points peuvent se situer au-dessus ou au-dessous du caractère mais jamais en haut et en bas simultanément [8].
- L'existence du "hamza" (le zigzag), qui se comporte, soit comme une lettre à part entière, soit comme un diacritique.
- Certaines formes de lettres ne peuvent dans aucun cas être rattachées à la lettre suivante, ce qui fait qu'un mot unique peut être entrecoupé d'un ou plusieurs espaces, lesquels sont aussi utilisés pour séparer les mots.

D	M	Fl	Fs	D	M	Fl	Fs
أ		ـأ	أ	ضـ	ـضـ	ـض	ض
بـ	ـبـ	ـب	ب	طـ	ـطـ	ـط	ط
تـ	ـتـ	ـت	ت	ظـ	ـظـ	ـظ	ظ
ثـ	ـثـ	ـث	ث	عـ	ـعـ	ـع	ع
جـ	ـجـ	ـج	ج	غـ	ـغـ	ـغ	غ
حـ	ـحـ	ـح	ح	فـ	ـفـ	ـف	ف
خـ	ـخـ	ـخ	خ	قـ	ـقـ	ـق	ق
د		ـد	د	كـ	ـكـ	ـك	ك
ذ		ـذ	ذ	لـ	ـلـ	ـل	ل
ر		ـر	ر	مـ	ـمـ	ـم	م
ز		ـز	ز	نـ	ـنـ	ـن	ن
سـ	ـسـ	ـس	س	هـ	ـهـ	ـه	ه
شـ	ـشـ	ـش	ش			ـو	و
صـ	ـصـ	ـص	ص	يـ	ـيـ	ـي	ي

D : En début du mot. M : Au milieu du mot.
Fl : A la fin du mot mais lié à une lettre. Fs : A la fin du mot sans être lié à la lettre.

Fig. 1. Les différentes formes possibles d'apparence des caractères de l'alphabet arabe.

- Les voyelles "a", "i" et "ou" ne sont pas utilisées systématiquement dans l'écriture arabe ; des signes qui correspondent à des voyelles sont employés pour éviter des erreurs de prononciation.
- On trouve également des chevauchements et des ligatures dans l'écriture manuscrite ce qui complique la tâche de reconnaissance (Fig. 2.).

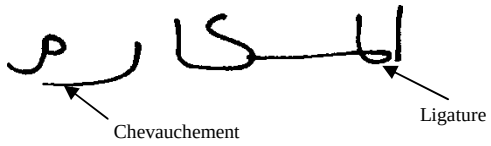


Fig. 2. Les ligatures et les chevauchements dans un mot arabe.

Plusieurs chercheurs ont conçu des systèmes de reconnaissance de l'écriture arabe, ils se diffèrent par le choix de type d'écriture ; imprimé ; manuscrit, en-ligne ou hors-ligne. Abd [1], Aburas [2], Benouareth [5], Burrow [8], Farah [11], Farah [12], Khorsheed [18] et Mozaffari [22] ont préféré l'écriture manuscrite hors-ligne, tandis que Al-Muhtaseb [3], Ben Amor [4] et Khorsheed [17] ont choisi le type imprimé. L'écriture en-ligne quand à elle est discutée dans les systèmes de Biadsy [7], Elanwar [10] et Mezghani [20]. Les Classifieurs les plus utilisés pour l'écriture arabe sont les Chaînes de Markov Cachées (HMM) dans [3], [4], [5], [7], [17], [18] et [20], les réseaux de neurones de type Perceptron Multi-couches (PMC) dans [4], [11] et [19], on trouve également les SVMs dans [1], le classifieur bayésien dans [19] et les k-proches voisins (k-ppv) dans [8] et [19].

3 Architecture de Fuzzy ART

Les réseaux Fuzzy ART qui représentent une classe de la famille des réseaux ART (Adaptive Resonance Theory) est un modèle de réseau de neurones à architecture évolutive développé en 1987 par Carpenter et Grossberg [16]. C'est un réseau compétitif à deux couches de neurones complètement inter-reliées. Une couche de comparaison F_1 sert à coder les entrées avec un encodage dit complémentaire et une couche de compétition F_2 semblable à celle du réseau de Kohonen [15], ces deux couches sont activées par une entrée X (Fig.3.).

Le Fuzzy ART propose une catégorisation originale avec des classes représentées par des prototypes. Un prototype, poids du neurone est défini par un vecteur à N dimensions :

$$W_j = (w_{j1}, w_{j2}, ..., w_{jN}). \quad \textbf{(1)}$$

où j est l'indice du neurone, $1 \le j \le M$, et N est la dimension de l'entrée X [14].

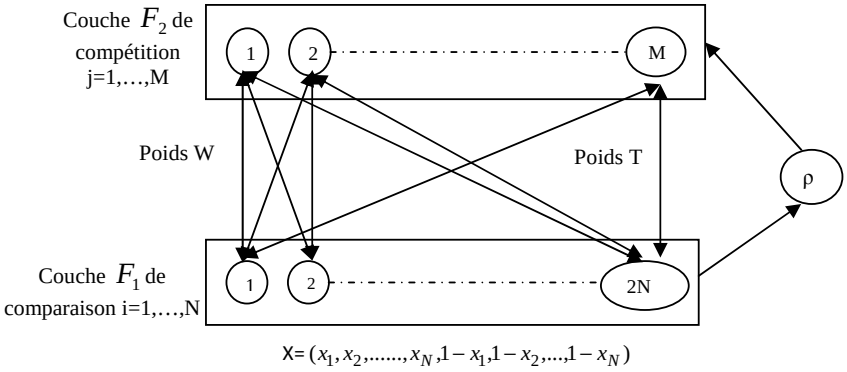


Fig. 3. Réseau de neurones Fuzzy ART.

Cependant, tout comme les autres architectures ART, le Fuzzy ART incorpore un mécanisme de rétroaction permettant de stabiliser les prototypes appris dans les vecteurs de poids qui relient les deux couches [14]. Ce mécanisme dit de résonance est contrôlé par un paramètre qui permet de réinitialiser au besoin la couche compétitive. Pour chaque entrée, les sorties du réseau spécifient une catégorie parmi les classes de sortie.

Le Fuzzy ART est un réseau constructif où de nouveaux neurones sont alloués lors de la phase d'apprentissage. Généralement, on fixe au départ un nombre maximum de neurones S, limitant ainsi le nombre maximum de catégories possibles [6]. Initialement, aucun neurone n'est actif. L'allocation subséquente de nouvelles catégories dépendra à la fois des entrées et des paramètres de l'algorithme. Parmi les avantages de Fuzzy ART on cite :

- L’algorithme de Fuzzy ART propose des calculs simplifiés pour la formation des classes sous forme d’hyper-boites, contrairement à des classes circulaires telles que retrouvées dans la plupart des algorithmes de réseaux de neurones [15].
- Il offre de bons résultats de catégorisation avec une précision modérée sur les poids des neurones.
- L’apprentissage se stabilise dans un nombre fini d’itérations.
- Le modèle Fuzzy ART exploite à fond un des avantages inhérents de l’approche neuronale ; le parallélisme [24].
- La forme prototype d’une classe sera immédiatement reconnue, même si elle n’a jamais été présentée, grâce aux caractéristiques pertinentes.

Ce classifieur n'a pas été exploité dans le domaine de reconnaissance de l'écriture arabe manuscrite ; notre système sera donc le premier a utilisé un réseau de neurones de type Fuzzy ART.

4 Implémentation du Système

Notre système qui réalise la reconnaissance d’écriture arabe manuscrite hors-ligne englobe plusieurs étapes décrites par la figure 4 :

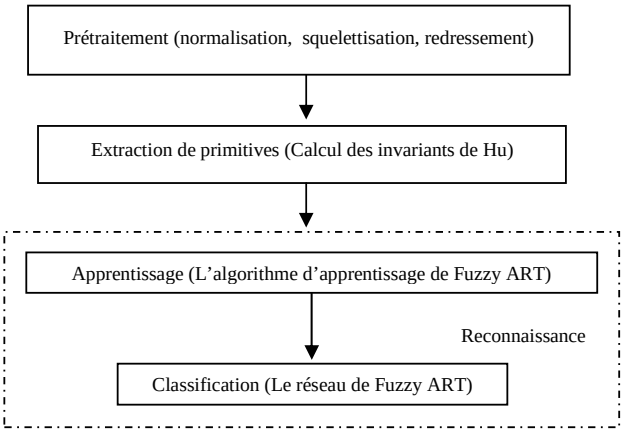


Fig. 4. Architecture du système.

4.1 Prétraitement

Qui est fait afin de réduire le bruit incluant les opérations de normalisation, de squelettisation et de redressement. La normalisation a été réalisée à travers une méthode de normalisation linéaire basée sur trois étapes [9] :

1. Calculer la matrice de dispersion de la forme.
2. Changer l'origine des axes des coordonnées vers le centre de la forme.

3. Changer l'échelle de base.

On a abouti à des images normalisées de taille 370×370 pixels. Pour la squelettisation on a appliqué quatre algorithmes ; de Rutovitz [9], de Zhang et Suen [28], de Deutch [9] et celui de Zhang et Wang [29] ce qui est montré par la figure 5.

Fig. 5. Application des algorithmes de squelettisation pour le mot "تونس القباضة الأصلية".

L'utilisation de plusieurs algorithmes de squelettisation était dans le but de choisir le meilleur entre eux, c'est-à-dire celui préservant le plus la structure du mot.

Le redressement est effectué en utilisant les histogrammes de projection horizontale selon onze angles différents de rotation variant de -5° à +5°, l'angle d'inclinaison correcteur sera celui de l'histogramme le plus dense. En calculant l'entropie de chacun des histogrammes obtenus, on pourra déterminer l'histogramme le plus dense représenté par la plus petite entropie. L'entropie est une mesure de l'information représentée par la formule suivante :

$$E = -\sum_i p_i \log(p_i)\,. \tag{2}$$

$$p_i = \frac{N_i}{N}\,. \tag{3}$$

Où N_i est le nombre de pixels ayant l'ordonnée y_i dans le repère de projection et N est le nombre total des pixels ou de points de contour du mot. La probabilité p_i de l'histogramme désigne la fréquence d'occurrence de l'ordonnée y_i. La figure 6 illustre les histogrammes de rotation ; le plus dense (c'est-à-dire ayant la plus petite entropie) est celui coloré en jaune.

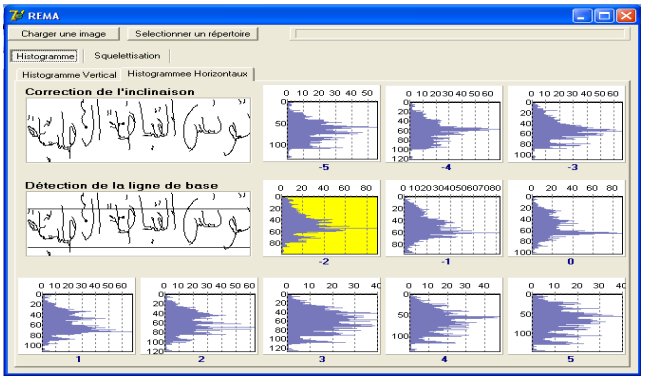


Fig. 6. Histogrammes de rotation du mot "تونس القباضة الأصلية".

4.1 Extraction de primitives

Hu a introduit la famille d'invariants portants son nom en utilisant les moments géométriques. Ces moments sont définis par :

$$m_{pq} = \sum\sum g(x,y)x^p y^q \ . \tag{4}$$

Soit x_0 et y_0 le barycentre de l'image, on définira les moments géométriques centrés par la fonction [21] :

$$\mu_{pq} = \sum\sum g(x,y)(x-x_0)^p (y-y_0)^q \ . \tag{5}$$

Les sept premiers invariants de Hu sont définis dans les équations : 6, 7, 8, 9, 10, 11, et 12.

$$\phi_1 = \eta_{20} + \eta_{02} \, . \tag{6}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \, . \tag{7}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \, . \tag{8}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 . \quad (9)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] . \quad (10)$$

$$\phi_6 = (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) . \quad (11)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2\right] . \quad (12)$$

Où :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} . \quad (13)$$

$$\gamma = \frac{p+q}{2} + 1 \qquad \forall (p+q) \geq 2 . \quad (14)$$

L'avantage principal que présente les moments de Hu est qu'ils sont peu sensibles à la translation, la rotation et au changement d'échelle ce qui permet de préserver l'information contenue dans les images [13]. Leur inconvénient majeur réside dans la présence de redondance de l'information dans la forme du mot [25].

4.1 Reconnaissance

Nous avons utilisé la base IFN/ENIT qui contient 26459 mots arabes représentant 964 classes de nom de villages tunisiens qui sont écrits par 411 scripteurs. Ie processus de reconnaissance inclut les deux phases suivantes :

Apprentissage. Il est fait sur un sous ensemble de la base IFN/ENIT. L'algorithme d'apprentissage non-supervisé utilisé est celui du réseau Fuzzy ART qui est détaillé par l'organigramme ci-dessous.

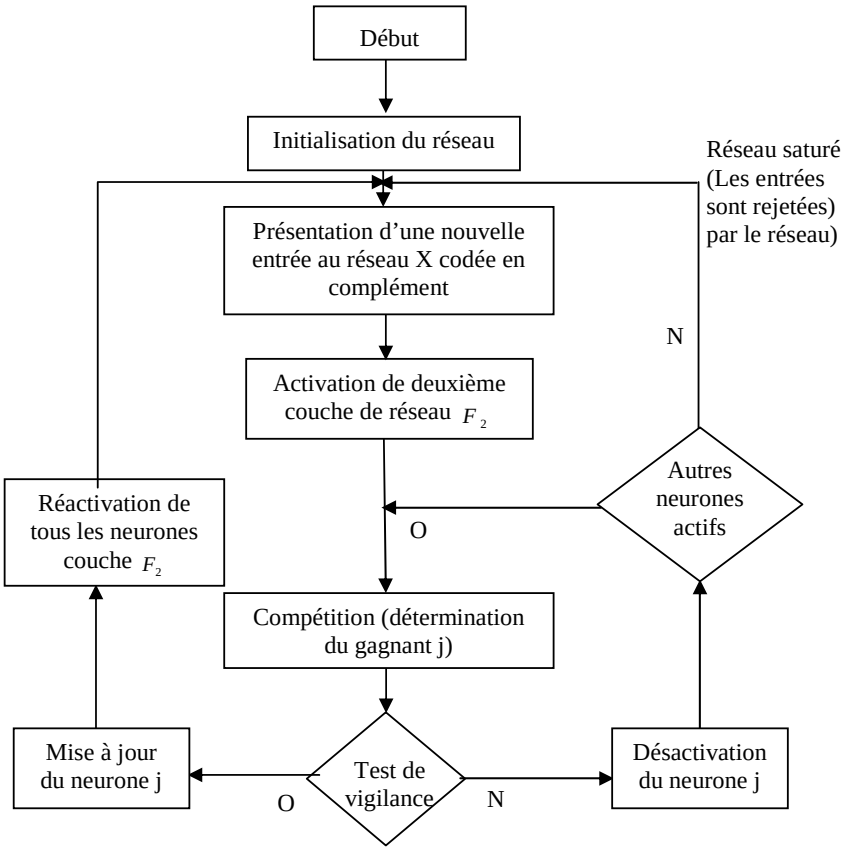


Fig. 7. Algorithme d'apprentissage de Fuzzy ART.

Classification. Elle est exécutée à travers le réseau Fuzzy ART. Le taux de reconnaissance obtenu avoisine les 85%, tandis que le taux de rejet est de 10% ; les 5% restants représentent le taux de mots mal classés. La figure 8 montre des échantillons des mots mal-classés.

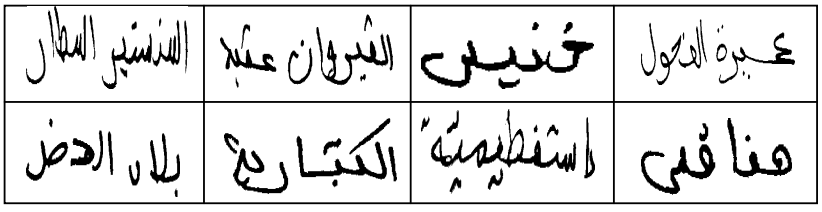


Fig. 8. Echantillons de mots mal-classés.

Afin de mieux mesurer l'efficacité des résultats obtenus, des comparaisons avec des systèmes de reconnaissance de mots arabes manuscrits hors-lignes travaillant sur la base IFN/ENIT ont été effectuées. Les travaux concernés sont ceux de :

- **Le système ICRA (Intelligent Character Recognition for Arabic).** Conçu par Ahmed Abdulkader [19], et qui a utilisé un classifieur neuronal de type PMC. L'apprentissage s'est fait sur les ensembles a, b et c de la base IFN/ENIT, tandis que les tests ont été réalisés sur l'ensemble e de cette dernière.
- **Le système de Burrow [8].** Il a utilisé comme classifieur les k-proches voisins (k-ppv) et les moments de Zernike comme primitives.
- **Le système TH-OCR.** Développé par Pingping Xiu, Hua Wang, J.Jin, Yan Jiang, L.Peng et X.Din [19]. Ce système a utilisé une méthode de reconnaissance statistique basée sur les réseaux de neurones en appliquant trois niveaux de segmentation, à savoir : la segmentation de texte en lignes, la segmentation de lignes en mots et la segmentation de mots en caractères.
- **Le système UOB.** Conçu par Chafik Mokbel et El Hajj [19] de l'université du Liban, où ils ont utilisé les chaînes de Markov cachées (HMM). les quatre premiers ensembles de la base ont été utilisés pour l'apprentissage et le dernier pour le test.
- **Le système REAM (Reconnaissance de l'Ecriture Arabe Manuscrite)**. C'est le résultat d'un travail de groupe de chercheurs tunisiens comprenant S. Masmoudi-Touj, N. Essoukhri-Ben Amara, H. Amiri et N. Ellouz [19], et qui ont utilisé une chaîne de Markov de type planaire dans la phase de reconnaissance.
- **Le système ARAB-IFN.** Conçu par Pechwitz [19], utilisant les chaîne de Markov cachées semi-continus d'ordre 1 comme classifieur. Il a appliqué les opérations de prétraitement de normalisation et de squelettisation afin de détecter la ligne de base. Les tests ont été faits sur l'ensemble e de la base IFN/ENIT.

Les résultats obtenus pour ces cinq systèmes de reconnaissance de l'écriture arabe manuscrite sont indiqués dans le tableau 1, où l'on remarque que le taux de reconnaissance enregistré par notre système est nettement meilleur que celui obtenu par les systèmes; SHOCRAN, TH-OCR ; tandis que les systèmes ICRA, ARABE-IFN et UOB donnent des taux relativement meilleurs, ceci peut être expliqué par l'expérience cumulée à travers la massive exploitation des types de classifieurs utilisés par ces derniers, à savoir les HMMs et les PMCs.

Table 1. Comparaison des résultats.

Système	**Type de Classifieur**	**Taux de reconnaissance**
ICRA	Perceptron Multi Couche	87,75%
Système de Burrow	Les k-proches voisins	76%
TH-OCR	Méthode statistique	50,14%
UOB	Chaînes de Markov Cachées	90,88%
REAM	Chaînes de Markov Cachées	19,86%
ARAB-IFN	Chaînes de Markov Cachées	89,77%
Notre système	Fuzzy ART	85%

5 Conclusion

Dans cet article nous avons mis au point un système de reconnaissance d'écriture arabe manuscrite, utilisant un réseau Fuzzy ART comme classifieur et les invariants de Hu comme primitives. Nous nous somme concentrés sur deux objectifs:

- Appliquer un classifieur Neuro-flou non-utilisé jusqu'à présent dans le domaine de la reconnaissance d'écriture arabe et analyser ses performances par rapport aux classifieurs déjà exploités.
- Démontrer l'efficacité des invariants de Hu et leur qualité en tant que primitives représentant des mots manuscrits arabes.

Les résultats obtenus sont encourageants comparés à d'autres systèmes travaillant eux aussi sur la base IFN/ENIT. Le taux d'erreur enregistré est principalement dû à la mauvaise écriture et aux problèmes de chevauchement des lettres dans les mots classés. Notons ici que le taux de reconnaissance peut être amélioré par une combinaison avec d'autres classifieurs et l'ajout d'une étape de poste-traitement.

References

1. Abd, M.A., Paschos, G.: Effective Arabic Character Recognition Using Support Vector Machines. Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. Springer. 7--11 (2007)
2. Aburas, A.A., Rehiel, S.M.A.: Off-line Omni-Style Handwritten Arabic Character Recognition System based on Wavelet Compression. CS&IT ARISER. 4, 123--135 (2007)
3. Al-Muhtaseb, H.A., Mahmoud, S.A., Qahwaji, R.S.: Recognition of Off-Line Printed Arabic Text Using Hidden Markov Models. Signal Processing. Elsevier. 88, 2902--2912 (2008)
4. Ben Amor, N.: Multi-Fonte Arabic Characters Recognition Using Hough Transform and HMM/ANN. Journal of Multimedia. Academy Publisher. 1, 50--54 (2006)
5. Benouareth, A., Annajy, A., Sellami, M.: Semi-Continuous HMMs with Explicit State Duration for Unconstrained Arabic Word Modelling and Recognition. Pattern Recognition Letters. Elsevier. 29, 1742--1752 (2008)
6. Bezdek, J.C., Killer, J., Krisnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, Springer, New York (2005)
7. Biadsy, F., El-Sana, J., Habash, N.: Online Arabic Handwritten Recognition Using Hidden Markov Models. Pattern Recognition (2004)
8. Burrow, P.: Arabic Handwriting Recognition. University of Edinburgh (2004)
9. Cheriet, M., Kharma, N., Liu, C.L., Suen, C.Y.: Character Recognition Systems. Wiley-Interscience, New Jersey (2007)
10. Elanwar, R.I., ElRashwan, M.A., Mashali, A.: Simultaneous Segmentation and Recognition of Arabic Characters in an Unconstrained On-Line Cursive Handwritten Document. International Journal of Computer and Information Science and Engineering (IJCISE). WASET. 1, 203--206 (2007)
11. Farah, N., Khadir, M.T., Sellami, M.: Artificial Neural Network Fusion: Application to Arabic Word Recognition. ESANN Proceedings-European Symposium on Artificial Neural Networks, pp. 151--156. Bruges (Belgium) (2005)
12. Farah, N., Souici, L., Farah, L. Sellami, M. : Arabic Words Recognition with Classifiers Combination : An Application to Literal Amounts. In Proceeding of Artificial Intelligence: Methodology, Systems and Applications, pp. 420--429. Bulgaria (2004)

13.Flusser, J., Suk, T.: Affine Moment Invariants: A New Tool for Character Recognition. Pattern Recognition Letters. 15, 433--436 (1994)
14.Freeman, J.A., Skapura, D.M.: Neural Networks: Algorithms, Applications and programming Techniques. Addison-Wesley Publishing Company, United States of America (1991)
15.Graupe, D.: Principles of Artificial Neural Networks. World Scientific, Singapore (2007)
16.Hagan, M.T., Demuth, H.B., Beale, M.: Neural Network Design. PWS Publishing Company, United States of America (2002)
17.Khorsheed, M.S.: Offline Recognition of Omni Font Arabic Text using the HMM Toolkit (HTK). Pattern Recognition Letters. Elsevier. 28, 1563--1571 (2007)
18.Khorsheed, M.S.: Recognition Handwriting Arabic Manuscripts Using Hidden Markov Model, Pattern Recognition Letters. Elsevier. 24, 2235--2242 (2003)
19.Lorgio, L.M.: Off-line Arabic Handwriting Recognition: A Survey. IEEE on Pattern Analysis and Machine Intelligence. 28, 712--724 (2006)
20.Mezghani, N., Mitiche, A., Cheriet, M.: A New Representation of Shape and its Use for High Performance in On-Line Arabic Character Recognition by an Associative Memory. International Journal of Document Analysis. Springer. 7, 201--210 (2005)
21.Ming-Kuel, H.: Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory. 179--187 (1962)
22.Mozaffari, S., Faez, K., Margner, V., El-Abed, H. : Lexicon Reduction Using Dots for Off-Line Farsi/Arabic Handwritten Word Recognition. Pattern Recognition Letters. Elsevier. 29, 724--734 (2008)
23.Nixon, M.S., Aguado, A.S..: Feature Extraction and Image processing. Newness, London (2002)
24.Park, S.S., Yoo, H.W., Lee, M.H., Kim, I.Y., Juang, D.S.: Clustering for Image Retrieval via Improved Fuzzy ART. Springer. 743--752 (2005)
25.Reddi, S.S.: Radial and Angular Moments Invariants for Image Identification. IEEE Transactions on Pattern Analysis and Machine Intelligence. 3, 240--242 (1981)
26.Rodtook, S., Makhanov, S.S., Vanderperre, E.J.: A Filter Bank for Rotationally Invariant Image Recognition. ORION. 21, 125--138 (2005)
27.Steinherz, T., Rivlin, E., Intrator, N.: Offline Cursive Script Word Recognition: a Survey. IJDAR, Springer. 2, 90--110 (1999)
28.Zhang, T.Y., Suen, C.Y.: A Fast Parallel Algorithm for Thinning Digital Patterns. CACM. 27, 236--239 (1984)
29.Zhang, L., Wang, G., Wang, W.: A New Fuzzy ART Neural Network Based on Dual Competition and Resonance Technique. Springer. 792--797 (2006)

An Efficient Algorithm for Solving Structured Acyclic Constraint Satisfaction Problems

Mohammed Lalou [1] and Kamal Amroun [2]

[1] University of Bejaia, e-mail : mohammed.lalou@gmail.com
[2] University of Bejaia, e-mail : k_amroun25@yahoo.fr

Abstract. The $\mathcal{CSP}$ formalism ($\mathcal{C}$onstraint $\mathcal{S}$atisfaction $\mathcal{P}$roblem) offers a powerful framework for representing and solving efficiently many problems. Solving $\mathcal{CSP}$ is in general $\mathcal{NP}$-$\mathcal{C}$omplete. The usual method for solving $\mathcal{CSP}$s is based on backtracking search. This approach has an exponential theoretical complexity in $\mathcal{O}(md^n)$ where n is the number of variables, m is the number of constraints and d is the maximum domain cardinality. However, there are various subsets of $\mathcal{CSP}$s that can be solved in polynomial time. Some of them can be identified by analyzing the structure of the $\mathcal{CSP}$. It is well known that if a constraint satisfaction problem $\mathcal{CSP}$ is tree-structured, then it can be solved in polynomial time. However, in practice the known algorithms are not efficient for large instances. In this paper, we propose an efficient algorithm, exploiting these structural proprieties, for solving constraints satisfaction problems.

Key words: Hypergraph, CSP, Hypertree Decomposition, Resolution.

1 Introduction

$\mathcal{C}$onstraints satisfaction problems are known to be $\mathcal{NP}$-$\mathcal{C}$omplete. Considerable efforts have then been made to identify tractable classes. One approach is based on exploiting structural properties of the constraints network. If the $\mathcal{CSP}$ is tree structured then it can be solved in polynomial time; for this reason, many techniques have been made to transform a $\mathcal{CSP}$ to its tree structured equivalent one (witch has the same solutions). The basic principle is to decompose the $\mathcal{CSP}$ into sub-problems that are organized in tree structure. A variety of such decomposition methods have been developed. Examples include methods based on the use of cycle cut-set [3], tree clustering [4], hinges [7], hypertree decomposition, generalized hypertree decomposition [10] and spread cut decomposition [1]. However the algorithms proposed in the literature for solving a $\mathcal{CSP}$ witch is represented by a generalized hypertree decomposition method are not efficient in practice. They require a huge time and a large memory. In this paper, we propose a new approach for solving a $\mathcal{CSP}$ given by a hypertree decomposition.

This paper is organized as follows: section 1 gives preliminaries of constraints satisfaction problems. In section 2, we review well known $\mathcal{CSP}$ decomposition methods, particulary, the hypertree decomposition method which is more general

(except for the spread cut decomposition). In section 3, we present the resolution of a $\mathcal{CSP}$ represented by an hypertree; we particulary give the well known algorithm $\mathcal{A}$cyclic $\mathcal{S}$olving for the resolution of hypertree structured $\mathcal{CSP}$s. In section 4, we present our sequential algorithm for solving the hypertree structured $\mathcal{CSP}$. In section 6, we do some experiments to see the efficiency of our proposition. Finally, in section 7, we draw our conclusions.

1.1 Preliminaries

The notion of $\mathcal{C}$onstraint $\mathcal{S}$atisfaction $\mathcal{P}$roblems ($\mathcal{CSP}$) was introduced by [8].

Definition 1. ***Constraint Satisfaction Problem***
A constraint satisfaction problem is defined as 3-tuple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ *where :*
$\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *is a set of n variables.*
$\mathcal{D} = \{d_1, d_2, ..., d_n\}$ *is a set of finite domains; a variable* x_i *takes its values in its domain* d_i*.*
$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m\}$ *is a set of m constraints. Each constraint* $\mathcal{C}_i$ *is a pair* $(\mathcal{S}(\mathcal{C}_i), \mathcal{R}(\mathcal{C}_i))$ *where* $S(\mathcal{C}_i) \subseteq \mathcal{X}$*, is a subset of variables, called* ***scope*** *of* $\mathcal{C}_i$ *and* $\mathcal{R}(\mathcal{C}_i) \subset \prod_{x_k \in \mathcal{S}(\mathcal{C}_i)} d_k$ *is the constraint relation, that specifies the allowed values combinations.*

A **solution** of a $\mathcal{CSP}$ is an assignment of values to variables which satisfies all constraints.

Definition 2. Hypergraph
The constraint hypergraph [2] of a $\mathcal{CSP}$ $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ *is given by* $\mathcal{H} = \langle \mathcal{V}, \mathcal{S} \rangle$ *where* $\mathcal{S}$ *is a set of hyperedges corresponding to the scopes of the constraints in* $\mathcal{C}$*,* $\mathcal{V}$ *is the set of variables of* $\mathcal{P}$*.*

Definition 3. Hypertree
A hypertree of a hypergraph $\mathcal{H} = \langle \mathcal{V}, \mathcal{S} \rangle$ *is a triple* $\langle \mathcal{T}, \chi, \lambda \rangle$*, where* $\mathcal{T}$ *is a rooted tree,* χ *and* λ *are two functions associating to each node* $p \in \mathcal{T}$ *two sets* $\chi(p) \subset \mathcal{V}$ *and* $\lambda(p) \subset \mathcal{S}$*.*

Proposition 1. *A* $\mathcal{CSP}$ *whose structure of hyper-graph constraints is acyclic can be solved in polynomial time [7].*

2 Structural decomposition of CSP

All structural decomposition methods take an instance of a $\mathcal{CSP}$ and transform it to an equivalent $\mathcal{CSP}$ (which has the same solutions) with an acyclic structure in order to solve the original $\mathcal{CSP}$ in polynomial time. The new $\mathcal{CSP}$ constraints are obtained by a relational joins of some relations in the original $\mathcal{CSP}$. A decomposition method associates to each hypergraph $\mathcal{H}$ a parameter $\mathcal{D}$-width called width of $\mathcal{H}$. The method $\mathcal{D}$ ensures that for fixed k, each $\mathcal{CSP}$ instance for which $\mathcal{D}$-$width \leq k$ is tractable (polynomially solvable).

Gottlob and all, in [6], have shown that the hypertree decomposition method is more general than all the other methods. Recently, it is shown in [1] that Spread Cut decomposition gives a better decomposition for a class of hypergraphs. But in general, we do not know if it is more general than hypertree decomposition. In the next paragraph, we present the hypertree decomposition method [5].

2.1 Hypertree decomposition

Definition 4. *A hypertree decomposition of a hypergraph $\mathcal{H} = \langle V, E\rangle$, is a triple $\mathcal{HD} = \langle \mathcal{T}, \chi, \lambda\rangle$ which satisfies all following conditions:*

1. *For each edge $h \in E$, there exists $p \in vertices(\mathcal{T})$ such that $var(h) \subset \chi(p)$. We say that p covers h.*
2. *For each variable $y \in V$, the set $\{p \in vertices(\mathcal{T}) | y \in \chi(p)\}$ induces a connected subtree of $\mathcal{T}$.*
3. *For each $p \in vertices(\mathcal{T}), \chi(p) \subseteq var(\lambda(p))$.*
4. *For each $p \in vertices(\mathcal{T})$, $var(\lambda(p)) \bigcap \chi(\mathcal{T}_p) \subseteq \chi(p)$. Note that $\mathcal{T}_p$ denotes the subtree rooted at the node p.*

A hyperedge h of a hypergraph $\mathcal{H} = \langle V, E\rangle$ is **strongly covered** in $\mathcal{HD} = \langle \mathcal{T}, \chi, \lambda\rangle$ if there exists $p \in vertices(\mathcal{T})$ such that the vertices in h are contained in $\chi(p)$ and $h \in \lambda(p)$. A hypertree decomposition $\mathcal{HD} = \langle \mathcal{T}, \chi, \lambda\rangle$ of $\mathcal{H} = \langle V, E\rangle$ is called **complete** if every hyperedge h of $\mathcal{H}$ is strongly covered in $\mathcal{HD}$. The **width** of hypertree decomposition $\mathcal{HD} = \langle \mathcal{T}, \chi, \lambda\rangle$ is $max|\lambda(p)|$, $p \in vertices(\mathcal{T})$. The **width** of a hypergraph $\mathcal{H} = \langle V, E\rangle$ is the **minimum** width over all hypertree decompositions of $\mathcal{H}$.

Example 1. of hypertree decomposition Let $\mathcal{H} = \langle V, E\rangle$ be a hypergraph, cf. figure (a) in FIG. 1, where vertices are $V = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s\}$ and the hyperedges are $E = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6, \mathcal{C}_7, \mathcal{C}_8, \mathcal{C}_9, \mathcal{C}_{10}, \mathcal{C}_{11}\}$.
The figure (b) FIG. 1, gives the hypertree decomposition of the hypergraph $\mathcal{H}$. Each node p is a pair of sets, the first is the constraints set $\lambda(p)$ and the second is the variables set $\chi(p)$.

There are two approaches to compute the hypertree decomposition: the exact methods like $\mathcal{O}pt$-k-$decomp$ [10], and heuristic ones: Bucket Elimination and Dual BE [9].

3 Resolution

Given a $\mathcal{CSP}$ and a hypertree decomposition we solve the $\mathcal{CSP}$ as follows:

1. Compute a complete hypertree decomposition.
2. where each vertex p is a sub problem, compute a new constraint relation $\mathcal{R}_p$ which is the projection on the variables in $\chi(p)$ of the join of the constraint relations in $\lambda(p)$, $\mathcal{R}p = \pi_{\chi(p)} \bowtie_{h \in \lambda(p)} h$.

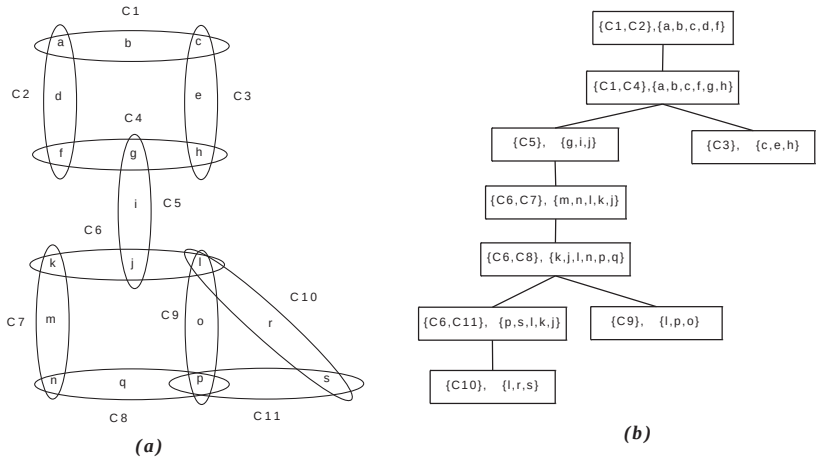


Fig. 1. (a) A hypergraph $\mathcal{H}$, (b) A hypertree decomposition of the hypergraph $\mathcal{H}$.

3. Apply algorithm for solving the obtained equivalent CSP.

After the resolution of each sub problem, we obtained a new join tree which is solved using an appropriate algorithm like $\mathcal{A}$cyclic $\mathcal{S}$olving. This algorithm (cf. algorithm 1) begins by ordering the nodes of the hypertree such that every node parent precedes its sons. Then, it treats successively all the nodes from the leaves to the root. For each node p of $\mathcal{T}$, it realizes a semi jointure on p and its parent in order to filter the parent relation. When the root is treated without making empty any relation, the CSP has solutions.

4 A sequential resolution algorithm S-HBR *(Sequential Hash Based Resolution)*

For the $\mathcal{CSP}$ resolution after a hypertree decomposition, no algorithm is proposed in the literature, according to our knowledge, except the general method presented in *3*. With a simple join operations calculus for the resolution of the sub problems, and an algorithm for the resolution of the whole problem. In this section, we propose a new algorithm that makes a more efficient resolution by optimizing the time for the sub problems and the whole problem.

In this contribution, we use the *hash* technique derivative from the data bases in order to bring its advantages for the resolution of the $\mathcal{CSP}s$ problems. Before presenting our approach, we recall the *hash* definition, on which our algorithm is based.

Definition 5. *A hash function is a particular function easily calculable that, from a data provided in entry $\mathcal{D}$, calculate an imprint $\mathcal{K}$ serving as an index to identify quickly the initial data $\mathcal{D}$. $\mathcal{K}$ designates, generally, the* hash value (key).

Algorithm 1 $\mathcal{A}$cyclic $\mathcal{S}$olving algorithm

1: **Input**:An acyclic constraint network $\mathcal{R} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$, $\mathcal{C} = (\mathcal{R}_1, ..., \mathcal{R}_t)$. $\mathcal{S}_i$ is the scope of $\mathcal{R}_i$. A join tree $\mathcal{T}$ of $\mathcal{R}$.
2: **Output** : Determine consistency and generate a solution.
3: $d = (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t)$ is an ordering such that every relation appears before its descendent relations in the tree rooted at $\mathcal{R}_1$.
4: **for** i= t to 2 **do**
5: **for** each edge (j, k), $k < j$, in the tree **do**
6: $\mathcal{R}_k = (\mathcal{R}_k \bowtie \mathcal{R}_j)[\mathcal{S}_k\ ]$
7: **if** $\mathcal{R}_k = \emptyset$ **then**
8: Exit; the problem is inconsistent.
9: **end if**
10: **end for**
11: **end for**
12: return : the updated relations and a solution : select a tuple in $\mathcal{R}_i$
13: **for** i = 2 to t **do**
14: After instantiating $\mathcal{R}_1$,...,$\mathcal{R}_{i-1}$ select a tuple in $\mathcal{R}_i$ that is consistent with all previous assignments.
15: **end for**

For a constraint $\mathcal{C}_i$, let $\mathcal{R}_i$ be the relation of $\mathcal{C}_i$, ϑ_i the intersection variables of $\mathcal{R}_i$ with all previous relations in the order, and $\mathcal{HT}_i$ the $\mathcal{R}_i$'s hash table. For a node n, Y_n is the set of intersection variables of n with its parent node, and $\mathcal{X}(n, a_i)$ is the intersection variables set of n with its son node a_i.

4.1 S-HBR algorithm

Considering a hypertree of n nodes where each node is a sub problem. The *$\mathcal{S}$-$\mathcal{HBR}$ ($\mathcal{S}$equential $\mathcal{H}$ash $\mathcal{B}$ased $\mathcal{R}$esolution)* algorithm process in two steps : the first one is the sub problems resolution, which is an optimization of the join operations calculus. It is performed using the $\mathcal{S}$-$\mathcal{HBR}$ ($\mathcal{S}$ub $\mathcal{P}$roblem $\mathcal{H}$ash $\mathcal{B}$ased $\mathcal{R}$esolution) algorithm (ligne 4). The substitution of each node by its solutions induce a new hypertree $\mathcal{T}'$ results on an equivalent problem, and contains one and only one constraint in each node. The second step is the resolution of the whole problem which is an optimization of $\mathcal{A}$cyclic $\mathcal{S}$olving algorithm. It is performed using the $\mathcal{A}$-$\mathcal{HBR}$ (*$\mathcal{A}$cyclic $\mathcal{H}$ash $\mathcal{B}$ased $\mathcal{R}$esolution*) algorithm (line 6). We will give more details about the two algorithms.

4.2 SP-HBR *(Sub Problem Hash Based Resolution) algorithm*

To solve sub problems, we apply the join operation, between the constraints relations, for each connected component of the sub hypergraph, using the hash technique. Then we apply the cartesian product between all connected components results. So, if the problem is inconsistent we avoid making cartesian product operation.

Algorithm 2 Sequential hash Based Resolution ($\mathcal{S}$-$\mathcal{HBR}$)

1: **Input**: a hypertree $\langle \mathcal{T}, \chi, \lambda \rangle$ of a hypergraph $\mathcal{H} = (V(\mathcal{H}), E(\mathcal{H}))$, with :$\mathcal{T}$ is a tree, χ associate with each t of $\mathcal{T}$ a set of nodes $\chi(t) \subset V(\mathcal{H})$, and λ, a set of arcs $\lambda(t) \subset E(\mathcal{H})$.
2: **Output** : Determine a hypertree directional arc consistent and generate a solution.
3: **for** each node n of $\mathcal{T}$ **do**
4: $\mathcal{SP}$-$\mathcal{HBR}$ (n), Apply the $\mathcal{SP}$-$\mathcal{HBR}$ algorithm on the node n.
5: **end for**
6: Let $\mathcal{T}'$ be the hypertree which represent the generated acyclic problem. For all $t' \in \mathcal{T}'$, t' contains one and only one constraint.
$\mathcal{A}$-$\mathcal{HBR}$ $(\mathcal{T}')$, Apply $\mathcal{A}$-$\mathcal{HBR}$ algorithm on $\mathcal{T}'$.

The result of the $\mathcal{SP}$-$\mathcal{HBR}$ execution on a node n is, only, a table $\mathcal{R}_n$ which contains the tuples allowed by all sub problem constraints. Leaves are represented as follows: $\mathcal{R}_n = \langle \chi(p), \mathcal{H}rel(p) \rangle$ where $\chi(p)$ is the set of variables of the node p, and $\mathcal{H}rel(p)$ is the constraint relation generated by the join operation. $\mathcal{H}rel(p)$'s tuples are hashed on intersection variables with the parent node of p.

In the $\mathcal{SP}$-$\mathcal{HBR}$ algorithm (cf. the *Algorithm 3*), first we decompose each sub-hypergraph in connected components (line 3). Then, we apply the join operation for each one using the hash join principle (line 4 to 28): for each component $\mathcal{C} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_n\}$, we apply a list of join operations $(((\mathcal{R}_1 \bowtie \mathcal{R}_2) \bowtie \mathcal{R}_3), ... \bowtie \mathcal{R}_n)$. In order to process that, we start by making an order among relations in the same component (line 5). Then, we hash each relation, except the first one (line 6 to 9). The relation hash operation is performed on intersection variables with its all previous relations (line 7). This is because, we cannot ensure that relation join variables are, only, those of the previous one. Also, the obtained tuple from a join operation will be hashed on intersection variables with the relation that constitutes the next operand of the next operation. The join operation is applied using the *Join* procedure (line 10).

The *join* procedure (line 29 to 42) proceeds in a recursive manner. At each iteration, we apply the join operation between the two relations: the result of the last join operation $\mathcal{R}_{tmp}$ (initially $\mathcal{R}_1$, cf. line 29) and the next operand $\mathcal{R}_j$ of the next one, as follows: we hash each tuple t of $\mathcal{R}_{tmp}$ on intersection variables ϑ_j of R_j with $\mathcal{R}_j$'s previous relations (cf. line 35). If the hash value corresponds to a hash table partition p of $\mathcal{R}_j$, we join t with all tuples of p. So, we construct the new relation $\mathcal{R}_{tmp}$ (line 37), and pass to the next operation. If the join result $\mathcal{R}es_i = \phi$ then, the problem has no solution (line 11). Otherwise, we calculate the final relation, denoted $\mathcal{R}es$, using the cartesian product of all relations results of node components (line 14 to 27). $\mathcal{R}es$ is computed progressively with the components join calculus. Indeed, after the join calculus of each component $\mathcal{R}es_i$, we make the cartesian product (line 19) of this one with $\mathcal{R}es$. We illustrate in what follows an example.

Example 2. We consider a node p of a hypertree $\mathcal{HD}$ labled as follows:

Algorithm 3 Sub Problem Hash Based Resolution ($\mathcal{S}$-$\mathcal{HBR}$)

Input : A node n, of a hypertree, represented by the sub hypergraph $\mathcal{G}_n = \langle \chi(t), \lambda(t) \rangle$, where: $\chi(t) \subset V$, $\lambda(t) \subset E$ and $t \in \mathcal{T}$. $\langle \mathcal{T}, \chi, \lambda \rangle$ is the hypertree of the hypergraph $\mathcal{H} = (V, E)$.
Output : Generate solutions for the subproblem n represented by a relation $\mathcal{R}es$.
Extract the connected component set, $\mathcal{C}$, of the hypergraph $\mathcal{G}_n$ (which represent n).
for each component $\mathcal{C}_i = \{\mathcal{R}_1, \ldots, \mathcal{R}_k\} \in \mathcal{C}$ **do**
  Make an order $d = \mathcal{R}_1, \ldots, \mathcal{R}_k$ among constraints of $\mathcal{C}_i$.
  for each relation $\mathcal{R}_i, \forall i \in \{2, \ldots, k\}$ **do**
    $\vartheta_i = \bigcup_{j \in m}(\mathcal{R}_i \cap \mathcal{R}_j)$ with $m = \{$ the $\mathcal{R}_i$ previous relation identities$\}$
    Calculate Hash(t, ϑ_i), $\forall t \in \mathcal{R}_i$
  end for
  $\mathcal{R}es_i = Join(\mathcal{C}_i, n)$, with $\mathcal{R}es_i$ is the join operation result.
  if $\mathcal{R}es_i = \phi$ **then**
    Exit. /* the problem has no solution */
  else
    if $i = 1$, the first component **then**
      $\mathcal{R}es \leftarrow \mathcal{R}es_1$
    else
      $\mathcal{R}es_t \leftarrow \mathcal{R}es$, $\mathcal{R}es \leftarrow \phi$.
      for each tuple t of $\mathcal{R}es_t$ **do**
        $t \leftarrow t \times t'$, $\forall t' \in \mathcal{R}es_i$,
        if $i = |\mathcal{C}|$, and n is a leaf node **then**
          Hash (t, Y_n), hash in intersection variables with the parent node.
        end if
        $\mathcal{R}es \leftarrow \mathcal{R}es \cup t$,
      end for
      $\mathcal{R}es \leftarrow \mathcal{R}es_t$
    end if
  end if
end for

The Join $(\mathcal{C}_i, n)$ **procedure**
$\mathcal{R}_{tmp} \leftarrow \mathcal{R}_1$, $\mathcal{R}_{tmp'} \leftarrow \mathcal{R}_{tmp}$, $\mathcal{R}_{tmp} \leftarrow \phi$.
for j = 2,...,k **do**
  for each tuple $t \in \mathcal{R}_{tmp'}$ **do**
    if j = k, the last join operation **then**
      $\mathcal{R}es_i = \mathcal{R}es_i \cup t$.
    else
      $h = Hash(t, \vartheta_j)$
      if the corresponding partition p to h in $\mathcal{HT}_j \neq \phi$ **then**
        $\forall t' \in p, \mathcal{R}_{tmp} = \mathcal{R}_{tmp} \cup (t \bowtie t')$.
      end if
    end if
  end for
  $\mathcal{R}_{tmp'} \leftarrow \mathcal{R}_{tmp}$, $\mathcal{R}_{tmp} \leftarrow \phi$
end for

$\lambda(p) = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6\}$, with: $\mathcal{S}(\mathcal{C}_1) = \{a, b, c\}$, $\mathcal{S}(\mathcal{C}_2) = \{i, j\}$, $\mathcal{S}(\mathcal{C}_3) = \{a, d\}$, $\mathcal{S}(\mathcal{C}_4) = \{d, b, f\}$, $\mathcal{S}(\mathcal{C}_5) = \{f, g\}$, $\mathcal{S}(\mathcal{C}_6) = \{i, k\}$.

There are two connected components $\mathcal{R} = \{\mathcal{C}_1, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$ and $\mathcal{S} = \{\mathcal{C}_2, \mathcal{C}_6\}$. According to the order of the constraints in each of the components we have:

$\vartheta_3 = \mathcal{S}(\mathcal{C}_3) \cap \{\mathcal{S}(\mathcal{C}_1)\} = \{a\}$, $\vartheta_4 = \mathcal{S}(\mathcal{C}_4) \cap \{\mathcal{S}(\mathcal{C}_1) \cup \mathcal{S}(\mathcal{C}_3)\} = \{b, d\}$, $\vartheta_5 = \mathcal{S}(\mathcal{C}_5) \cap \{\mathcal{S}(\mathcal{C}_1) \cup \mathcal{S}(\mathcal{C}_3) \cup \mathcal{S}(\mathcal{C}_4)\} = \{f\}$, $\vartheta_6 = \mathcal{S}(\mathcal{C}_6) \cap \{\mathcal{S}(\mathcal{C}_2)\} = \{i\}$.

Therefore, the $\mathcal{R}_3$'s tuples will be hashed on the set of intersection variables $\vartheta_3 = \{a\}$, those of $\mathcal{R}_4$ and $\mathcal{R}_5$ on, respectively, $\vartheta_4 = \{b, d\}$ and $\vartheta_5 = \{f\}$. For the second component, the tuples of $\mathcal{R}_6$ will be hashed on $\vartheta_6 = \{i\}$.

After all cartesian products calculus of a node p (line 20), we hash the obtained tuple t on intersection variables with the parent node of p, if p is a leaf. So, t will be ready for the semi join operation in $\mathcal{A}$-$\mathcal{HBR}$ algorithm. We do not add t to its related partition p unless if $p = \phi$. Thus, we will have one and only one tuple in each partition by eliminating double tuples. This elimination has no impact on the resolution because, for the $\mathcal{A}$-$\mathcal{HBR}$ algorithm, we filter each parent node on their sons. [1] Moreover, we are interested to get only one solution for the $\mathcal{CSP}$.

4.3 A-HBR *(Acyclic Hash Based Resolution) algorithm*

The $\mathcal{A}$-$\mathcal{HBR}$ algorithm tests the consistence of a constraints acyclic network and generates a solution if it exists. The algorithm's input is a hypertree, in which each node n contains the $\mathcal{SP}$-$\mathcal{HBR}$ algorithm results $\mathcal{R}es_n$. For leaf nodes, $\mathcal{R}es_n = \langle \chi(n), \mathcal{H}rel(n) \rangle$, where $\chi(n)$ is the set of variables of n, and $\mathcal{H}rel(n)$ is the generated constraint relation using the join operation, which tuples are hashed on intersection variables with the parent node of n.

The $\mathcal{A}$-$\mathcal{HBR}$ algorithm (cf. *Algorithme 4*) proceeds in two steps: the CONTRACT and the S-SEARCH. The first step contracts the deep-rooted hypertree $\mathcal{T}$. A semi join operation is associated to each contraction operation. The result is a directional, from root toward leaves, arc consistent hypertree $\mathcal{T}'$ (line 3). The second step is the solution search operation, which develops the solution from the $\mathcal{T}'$'s root to leaf nodes (line 4).

Algorithm 4 Acyclic Hash Based Resolution algorithm ($\mathcal{A}$-$\mathcal{HBR}$)

1: **Input**: A hypertree $\mathcal{T} = \langle A, E \rangle$, A is the set of nodes, and E is the set of arcs.
2: **Output** : Determine a directional arc consistent and generate a solution.
3: $\mathcal{T}'$ = CONTRACT ($\mathcal{T}$), $\mathcal{T}$ is the $\mathcal{S}$-$\mathcal{HBR}$ result and $\mathcal{T}'$ is the generated hypertree.
4: S - SEARCH ($\mathcal{T}'$),

[1] It is sufficient to have only one tuple by partition in the son node relations.

CONTRACT algorithm The *CONTRACT algorithm* (cf. *Algorithm 5*) strengths the directional arc consistency in the input hypertree, which is the $\mathcal{SP}$-$\mathcal{HBR}$ algorithm result. For each parent node, we apply the contraction operation with each of its son leaf node.

Algorithm 5 CONTRACT algorithm

1: **Input**: A hypertree $\mathcal{T} = \langle A, E \rangle$, A is the set of nodes, and E is the set of arcs.
2: **Output** : Generate a directional arc consistent hypertree $\mathcal{T}' = \langle A', E' \rangle$, where: $\forall a' \in A', a' = \langle \chi(p), \mathcal{H}rel(p) \rangle$.
3: **for** each node n of $\mathcal{T}$, which have a leaf son nodes **do**
4: **for** each tuple t of $\mathcal{R}_n$ **do**
5: **for** each leaf son node a_i of n **do**
6: Calculate Hash $(t, \mathcal{X}(n, a_i))$.
7: **if** the corresponding partition in the $\mathcal{HT}_{a_i} = \phi$ **then**
8: Eliminate t, $(\mathcal{R}_n \leftarrow \mathcal{R}_n - t)$.
9: **else**
10: **if** it is the last son node of n **then**
11: Hash (t, Y_n), hash in intersection variables with the parent node.
12: **end if**
13: $\mathcal{H}rel(n) \leftarrow \mathcal{H}rel(n) \cup t$.
14: **end if**
15: **end for**
16: **end for**
17: Mark all sons nodes a_i of n.
18: **if** $\mathcal{R}_n = \phi$ **then**
19: Exit /* the problem has no solution */
20: **end if**
21: **end for**

For each contraction operation (line 3 to 9), we hash each tuple t of a parent node (p) relation (line 4 to 14) on intersection variables with its sons leaves nodes a_i (line 6). We denote h_i the hash values. The a_i's relations are hashed on intersection variables with p after the $\mathcal{S}$-$\mathcal{HBR}$ algorithm execution, this make the join test more quickly using a direct access to a_i's hash table. If a_i's hash tables partitions related to h_i are not empty (line 7), we hash t on intersection variables of p with its parent node (line 10) and stock it. Otherwise, we eliminate t (line 8). The leaves nodes a_i are marked and considered as pruned. We make the same with the new hypertree, and so on, until all the nodes, except the root, will be marked.

The final result of the contraction operation of a parent node p with its sons leaves nodes, is a node $p' = \langle \chi(p'), \mathcal{H}rel(p') \rangle$, whose relation is arc consistent with those of the son nodes relations, and hashed on the intersection variables of p with its parent node. If the hashed relation $\mathcal{H}rel(p')$ is empty, the problem has no solution (line 17).

S-SEARCH algorithm For searching a solution to the $\mathcal{CSP}$, we apply a *Backtrack free* algorithm to the directional arc consistent hypertree result from the *CONTRACT* algorithm execution.

Algorithm 6 S - SEARCH algorithm

1: **Input**: A hypertree tree $\mathcal{T}' = \langle A', E' \rangle$, where: $\forall a' \in A', a' = \langle \chi(a'), \mathcal{H}rel(a') \rangle$.
2: **Output** : Generate a solution to the $\mathcal{CSP}$ problem.
$\mathcal{S}ol \leftarrow \emptyset$, the solution of the problem. *// the set of variables values.*
3: Mark the root node,
4: Take a tuple $t \in \mathcal{H}rel(root)$, $\mathcal{S}ol \leftarrow t$.
5: **for** each node a of $\mathcal{T}'$ **do**
6: **if** the parent node of a is marked **then**
7: Hash($\mathcal{S}ol$, Y_a), and take a tuple $t \in \mathcal{H}rel(a)$ from the related partition in the $\mathcal{HT}_a$.
8: $\mathcal{S}ol \leftarrow \mathcal{S}ol \cup t$,
9: Mark the node a,
10: **end if**
11: **end for**
12: return $\mathcal{S}ol$.

In *S-SEARCH* algorithm (cf. *Algorithme 6*), first we mark the root node r (line 3) and take a tuple t from its relation. For each r's son node a (line 5 to 11), we hash t on intersection variables of r with a (line 7) and take a tuple from the related partition in the a's hash table. We do the same, in recursion, for all nodes of which the parent node is marked. After each iteration, the sons nodes are marked (line 9). The solution is up dated for each search to a related tuple in a node (line 8).

5 S-HBR algorithm Complexity

The $\mathcal{S}$-$\mathcal{HBR}$ algorithm complexity is $\mathcal{O}(r\mathcal{P}^{h-1})$ for the $\mathcal{SP}$-$\mathcal{HBR}$ algorithm, and $\mathcal{O}(nl)$ for the $\mathcal{A}$-$\mathcal{HBR}$ algorithm, where h is the hypertree width, $\mathcal{P}$, r and l are the maximal tuples number in, respectively, the hash partition, the $\mathcal{CSP}$ relation and the result of $\mathcal{S}$-$\mathcal{HBR}$ algorithm. n is the constraints number in the $\mathcal{CSP}$ result of $\mathcal{SP}$-$\mathcal{HBR}$ algorithm. If $r = d^s$ where d is the maximum domain cardinality and s is the maximal number of variables by tuple, then $\mathcal{P} = d^{s-|\vartheta|}$, where $|\vartheta|$ is the hash variables number. So, there is an exponential gain in complexity.

In practice, the algorithm efficiency depends on the chosen hash function. If this one has no collision, the $\mathcal{A}$-$\mathcal{HBR}$ algorithm complexity is $\mathcal{O}(n)$.

6 Experiments

To show the convenient interest of our approach and evaluate practically the results relatively to other algorithm, we have implemented the $\mathcal{S}$-$\mathcal{HBR}$ algorithm, and a simple resolution after a hypertree decomposition ($\mathcal{SIRH}$). In this section, we will give the experimental results. We have tested the two solvers on a set of $\mathcal{CSP}$ benchmarks. This benchmarks collection is represented using the new *XCSP 2.1* format (the format spread for the constraints networks representation using the *XML*) proposed by *The organizational committee of the* 3^{th} *international competition of the* $\mathcal{CSP}s$ *solvors* [2][11]. Solvers are constitued of a classes set which permit the $\mathcal{CSP}$ acyclic structure manipulation. Theirs inputs are $\mathcal{CSP}s$ instances and the corresponding hypertrees in *GML* format. For this, We have exploited a *GML* Parser proposed by *M. Raitner and Mr. Himsol*[3] and an other one, proposed by *O. Roussel*[4], for *XCSP* file.

The experiments are processed on *Linux 6.0.52 version* using an *Intel Pentium IV*, with *2.4 GHz* of *CPU* and *600 Mb* of *RAM*. The table Tab. 1 summarizes the tests results obtained, which permits us to conclude that:

$\mathcal{CSP}$						$\mathcal{SIRH}$ (sec.)	$\mathcal{S}$-$\mathcal{HBR}$ (sec.)
Name	|V|	|E|	$\mathcal{N}d$	$\mathcal{HTW}$	|r|		
3-insertions-3-3	56	110	86	7	3	>1000	127
domino-100-100	100	100	50	2	100	27	7
domino-100-200	100	100	50	2	200	103	28
geom-30a-4	30	81	70	4	4	5	2
domino-100-300	100	100	50	2	300	241	61
hanoi-6	62	61	59	1	2148	8	2
series-7	13	42	39	4	42	>1000	106
hanoi-7	126	125	125	1	6558	71	9
haystacks-06	36	95	88	3	8	9	4
haystacks-07	49	153	144	4	9	514	66
langford-2-4	8	32	31	4	8	11	3
pigeons-7	7	21	19	3	6	12	3
Renault	101	134	81	2	48721	780	20

Table 1. Experiments results of $\mathcal{S}$-$\mathcal{HBR}$ algorithm

1. The instances whose hypertree width is big take an enormous time of resolution in spit of small size of the relations (*3 - Insertion 3 3*), where the objective of minimizing the decomposition with for all methods.
2. For the instances of which the number of tuples by relation is relatively small, a simple resolution after a decomposition hypertree can give the results

[2] http://www.cril.univ-artois.fr/ lecoutre/research/benchmarks/benchmarks.html
[3] http://www.cs.rpi.edu/ puninj/XGMML/GML-XGMML/gml-parser.html
[4] http://www.cril.univ-artois.fr/ roussel/CSP-XML-parser/

proximate of those of $\mathcal{S}$-$\mathcal{HBR}$ (for *domino-100-100*, because of a hypertree width of 2, and *hanoi-6* because of the small number of nodes that is 27), to reason of the hash values calculation, that is negligible as much more that the instance is big.
3. The $\mathcal{S}$-$\mathcal{HBR}$ algorithm gets the more profit for the instances which have large relations, because it browses a part of the relation rather than the totality as in $\mathcal{SIRH}$ (there is an exponential complexity gain). The relation size is big for one of three reasons, a big hypertree width (*3 - Insertion 3 3*), a big maximum number of tuples by relation (*Renault* and *hanoi-7*) or a big number of nodes in the hypertree (*hanoi-7* and *haystacks-07*).

Remark 1. $|V|$, $|E|$ and $|r|$ designate, repectively, the varaibles number, the constraints number of the $\mathcal{CSP}$ and the maximum number of tuples by relation. $\mathcal{N}d$ and $\mathcal{HTW}$ are, repectively, the nodes number and the hypertree width. $\mathcal{S}$-$\mathcal{HBR}(sec)$ illustrate the $\mathcal{S}$-$\mathcal{HBR}$ algorithm execution time in second, and $\mathcal{SIRH}(sec)$ the one of the simple resolution after a hypertree decomposition.

7 Conclusion

In this paper, we have presented a sequential algorithm to solve a $\mathcal{CSP}$ by exploiting its structural proprieties. This algorithm is based on the hash technique which optimizes the solving algorithm complexity. We have done some experiments on benchmarks from the literature and compare the results to those given by the well known algorithm $\mathcal{A}$cyclic $\mathcal{S}$olving. The results obtained are promising. However, we must do more experiments on large benchmarks. Future work include the parallelization of this approach.

References

1. Gyssens, M., Cohen D., Jeavons, P.: A unified theory of structural tractability for constraint satisfaction problems. J. Comp. and Sys. Sci. 74, 721–743 (2007)
2. Dechter, D.: Constraint Processing. Morgan Kaufmann, (2003)
3. Dechter, R., Pearl, J.: The cycle-cutset method for improving search performance in AI applications. In: 3th IEEE Proceedings on Artificial Intelligence Applications, pp. 224–230. Orlando (1987)
4. Dechter, D., Pearl, J.: Tree clustering for constraint networks. Art. Int. 38, 353–366, (1989)
5. Gottlob, G., Leone, N., Scarcello, F.: Hypertree decompositions and tractable queries. In: Preceedings of PODS'99, pp. 21–32, (1999)
6. Gottlob, G., Leone, N., Scarcello, F.: A comparison of structural csp decomposition methods. Art. Int. 124, 243–282 (2000)
7. Gyssens, M., Jeavons, P.G., Cohen, D.A.: Decomposing constraint satisfaction problems using database techniques. Art. Int. 66, 57–89 (1994)
8. Montanari, U.: Networks of constraints: Fundamental properties and applications to pictures processing. Inf. Sci. 7, 95–132 (1974)
9. Dermaku, A., Ganzow, T., Gottlob, G., McMahan, B., Musliu, N., Samer, M.: Heuristic Methods for hypertree decompositions. DBAI-R (2005)

10. Gottlob, G., Leone, N., Scarcello, F.: On tractable queries and constraints. In: Proceedings of DEXA'99 (1999)
11. XML Representation of Constraint Networks Format XCSP 2.1, http://www.cril.univ-artois.fr/CPAI08

Looking for the Best and the Worst

Souhila Kaci and Cédric Piette

Université Lille-Nord de France, Artois
CRIL, CNRS UMR 8188 - IUT de Lens
F-62307, France
{kaci,piette}@cril.fr

Abstract. Representing preferences and reasoning about them are important skills in many real-life applications. Several formalisms have been developed for this purpose ranging from quantitative to qualitative representations. In this paper, we focus on qualitative formalisms modeling comparative preference statements of the form "I prefer p to q". Although comparative statements offer a simple and natural way for expressing preferences, they come however with difficulties regarding their interpretation. Several (more or less strong) semantics have been proposed leading to different (pre)orders on outcomes. Researchers have argued so far for a separate handling of two categories of semantics each obeying a principle from non-monotonic reasoning. We show that the separate use of the principles may be debatable. We propose an algorithm based on both principles for rank-ordering outcomes given a set of comparative preference statements. Our algorithm handles all semantics at hand at the same time. A connection with interval qualitative algebra used in spatial and temporal reasoning is also provided.

1 Introduction

Representing preferences and reasoning about them are important skills in many real-life applications. Preferences are of interest to economists, operations researchers, logicians, philosophers, etc. They are also of greater interest in many areas such as multi-agent systems, constraint satisfaction problems, decision making, social choice theory, etc. Several formalisms have been developed for representing preferences ranging from quantitative to qualitative representations. However it has been early recognized that value functions (i.e., quantitative representations) or even complete/partial ordering cannot be explicitly defined because of the great number of possible outcomes. Even more, in some applications, the user is not able to explicitly rank-order a small number of outcomes. Instead, users generally express partial qualitative preferences such as "I like Paris", "I prefer London over Rome", etc. Therefore we need qualitative compact representations of preferences which support such partial preferences.

The compact representation languages for preference representation have been extensively developed in Artificial Intelligence in the last decade. In this paper, we focus on representations based on (conditional) comparative preference statements for e.g. "I prefer fish to meat", "If red wine is served then I prefer meat to fish", etc. Although comparative statements offer a simple and natural way for expressing preferences, they come however with difficulties regarding their interpretation. Several (more or less strong) semantics have been proposed leading to different (pre)orders on outcomes [8,

6, 4, 2, 12]. Researchers have argued so far for a separate handling of two categories of semantics each obeying a principle from non-monotonic reasoning. We show that the separate use of the principles may be debatable. We propose an algorithm based on both principles for rank-ordering outcomes given a set of comparative preference statements. Our algorithm handles all semantics at hand at the same time. A first connection with interval qualitative algebra used in spatial and temporal reasoning is also provided.

The remainder of this paper is organized as follows. After providing necessary definitions, we recall in Section 3 the different semantics of comparative preferences. In Section 4 we recall algorithms to rank-order outcomes for each semantics. In Section 5 we show that the way to rank-order outcomes based on the separate use of two principles may be debatable. In order to overcome this limitation, we propose a new algorithm which integrates both principles. Section 6 provides a connection of our framework with qualitative interval algebra. Lastly, we conclude with some perspectives.

2 Notations

Let $V = \{X_1, \cdots, X_h\}$ be a set of h variables. Each variable X_i takes its values in a domain $Dom(X_i)$. A possible outcome, denoted ω, is the result of assigning a value in $Dom(X_i)$ to each variable X_i in V. Ω is the set of all possible outcomes. We write $\omega \models \varphi$ when ω makes the formula φ true. We say that ω satisfies φ.

An ordering relation $\succeq$ on $\mathcal{X} = \{x, y, z, \cdots\}$ is a reflexive binary relation such that $x \succeq y$ stands for x is at least as preferred as y. $x \approx y$ means that both $x \succeq y$ and $y \succeq x$ hold, i.e., x and y are equally preferred. Lastly $x \sim y$ means that neither $x \succeq y$ nor $y \succeq x$ holds, i.e., x and y are incomparable. A strict ordering relation on $\mathcal{X}$ is an irreflexive binary relation such that $x \succ y$ means that x is strictly preferred to y. We also say that x dominates y. A strict ordering relation $\succ$ can be defined from an ordering relation $\succeq$ as $x \succ y$ if $x \succeq y$ holds but $y \succeq x$ does not. When neither $x \succ y$ nor $y \succ x$ holds, we also write $x \sim y$. $\succeq$ (resp. $\succ$) is a preorder (resp. order) on $\mathcal{X}$ if and only if $\succeq$ (resp. $\succ$) is transitive, i.e., if $x \succeq y$ and $y \succeq z$ then $x \succeq z$ (resp. if $x \succ y$ and $y \succ z$ then $x \succ z$). $\succeq$ (resp. $\succ$) is a complete preorder (resp. order) if and only if $\forall x, y \in \mathcal{X}$, we have either $x \succeq y$ or $y \succeq x$ (resp. either $x \succ y$ or $y \succ x$).

The set of the best (or undominated) elements of $A \subseteq \mathcal{X}$ w.r.t. $\succ$, denoted $\max(A, \succ)$, is defined by $\max(A, \succ) = \{x | x \in A, \nexists y \in A, y \succ x\}$. The set of the worst elements of $A \subseteq \mathcal{X}$ w.r.t. $\succ$, denoted $\min(A, \succ)$, is defined by $\min(A, \succ) = \{x | x \in A, \nexists y \in A, x \succ y\}$. The best (resp. worst) elements of A w.r.t. $\succeq$ is $\max(A, \succ)$ (resp. $\min(A, \succ)$) where $\succ$ is the strict ordering relation associated to $\succeq$.

A complete preorder $\succeq$ can also be represented by a well ordered partition of Ω. This is an equivalent representation, in the sense that each preorder corresponds to one ordered partition and vice versa. A sequence of sets of outcomes of the form $(E_1, \ldots, E_n)$ is a *partition* of Ω if and only if (i) $\forall i$, $E_i \neq \emptyset$, (ii) $E_1 \cup \cdots \cup E_n = \Omega$, and (iii) $\forall i, j$, $E_i \cap E_j = \emptyset$ for $i \neq j$. A partition of Ω is *ordered* if and only if it is associated with a preorder $\succeq$ on Ω such that ($\forall \omega, \omega' \in \Omega$ with $\omega \in E_i, \omega' \in E_j$ we have $i \leq j$ if and only if $\omega \succeq \omega'$).

Complete preorders can be compared following specificity principle [13]:

Definition 1 (Minimal/Maximal specificity principle). *Let $\succeq$ and $\succeq'$ be two complete preorders on a set of outcomes Ω represented by ordered partitions $(E_1, \cdots, E_n)$ and $(E'_1, \cdots, E'_{n'})$ respectively. We say that $\succeq$ is less specific than $\succeq'$, written as $\succeq \sqsubseteq \succeq'$, iff $\forall \omega \in \Omega$, if $\omega \in E_i$ and $\omega \in E'_j$ then $i \leq j$. $\succeq$ belongs to the set of the least (resp. most) specific preorders among a set of preorders $\mathcal{O}$ if there is no $\succeq'$ in $\mathcal{O}$ such that $\succeq' \sqsubset \succeq$ (resp. $\succeq \sqsubset \succeq'$), i.e., $\succeq' \sqsubseteq \succeq$ (resp. $\succeq \sqsubseteq \succeq'$) holds but $\succeq \sqsubseteq \succeq'$ (resp. $\succeq' \sqsubseteq \succeq$) does not.*

3 Comparative preference statements

We denote comparative statements of the form "I prefer p to q" as $p > q$ and denote conditional (also called contextual) comparative statements of the form "if r is true then I prefer p to q" as $r : p > q$, where p, q and r are any propositional formulas. Note that $r : p > q$ is equivalent to $r \wedge p > r \wedge q$. Indeed we will simply focus on statements of the form $p > q$. Comparative statements, apparently very simple, come with difficulties regarding their interpretation. How should we interpret such statements? For example, given the preference statement "I prefer fish to meat", how do we rank-order meals based on fish and those based on meat? Four semantics have been proposed in literature:

- *strong preferences:* [6, 12]
  any fish-based meal is preferred to any meat-based meal.
- *optimistic preferences:* [3, 6, 8]
  at least one fish-based meal is preferred to all meat-based meals.
- *pessimistic preferences:* [2]
  at least one meat-based meal is less preferred to all fish-based meals.
- *opportunistic preferences:* [10]
  at least one fish-based meal is preferred to at least one meat-based meal.

Comparative preference statements are generally expressed in a strict form (i.e., $>$). In this paper, we use both strict and non-strict comparative statements, denoted $>$ and $\geq$ respectively, as already defined in [7]. This makes our framework more general.
We define preference of the formula p over the formula q as preference of $p \wedge \neg q$ over $\neg p \wedge q$. This is standard and known as von Wright's expansion principle [11]. Additional clauses may be added for the cases in which sets of outcomes are nonempty, to prevent the satisfiability of preferences like $p > \top$ and $p > \bot$. We do not consider this borderline condition to keep the formal machinery as simple as possible.

We denote the preference of p over q following strong semantics (resp. optimistic, pessimistic, opportunistic) by $p >_{st} q$ (resp. $p >_{opt} q$, $p >_{pes} q$, $p >_{opp} q$). Non-strict preferences are denoted $p \geq_{st} q$, $p \geq_{opt} q$, $p \geq_{pes} q$ and $p \geq_{opp} q$.

Definition 2. *Let p and q be two propositional formulas and $\succeq$ be a preorder on Ω.*

- *$\succeq$ satisfies $p >_{st} q$ (resp. $p \geq_{st} q$), denoted $\succeq \models p >_{st} q$ (resp. $\succeq \models p \geq_{st} q$), iff $\forall \omega \models p \wedge \neg q$, $\forall \omega' \models \neg p \wedge q$ we have $\omega \succ \omega'$ (resp. $\omega \succeq \omega'$).*
- *$\succeq$ satisfies $p >_{opt} q$ (resp. $p \geq_{opt} q$), denoted $\succeq \models p >_{opt} q$ (resp. $\succeq \models p \geq_{opt} q$), iff $\exists \omega \models p \wedge \neg q$, $\forall \omega' \models \neg p \wedge q$ we have $\omega \succ \omega'$ (resp. $\omega \succeq \omega'$).*
- *$\succeq$ satisfies $p >_{pes} q$ (resp. $p \geq_{pes} q$), denoted $\succeq \models p >_{pes} q$ (resp. $\succeq \models p \geq_{pes} q$), iff $\exists \omega' \models \neg p \wedge q$, $\forall \omega \models p \wedge \neg q$ we have $\omega \succ \omega'$ (resp. $\omega \succeq \omega'$).*
- *$\succeq$ satisfies $p >_{opp} q$ (resp. $p \geq_{opp} q$), denoted $\succeq \models p >_{opp} q$ (resp. $\succeq \models p \geq_{opp} q$), iff $\exists \omega \models p \wedge \neg q$, $\exists \omega' \models \neg p \wedge q$ we have $\omega \succ \omega'$ (resp. $\omega \succeq \omega'$).*

The following lemma gives the mathematical description of Definition 2 [7]:

Lemma 1. *Let p and q be two propositional formulas and $\succeq$ be a preorder on Ω.*

- $\succeq$ *satisfies* $p >_{st} q$ *(resp.* $p \geq_{st} q$*) iff* $\forall \omega \in \min(p \wedge \neg q, \succeq)$, $\forall \omega' \in \max(\neg p \wedge q, \succeq)$ *we have* $\omega \succ \omega'$ *(resp.* $\omega \succeq \omega'$*).*
- $\succeq$ *satisfies* $p >_{opt} q$ *(resp.* $p \geq_{opt} q$*) iff* $\forall \omega \in \max(p \wedge \neg q, \succeq)$, $\forall \omega' \in \max(\neg p \wedge q, \succeq)$ *we have* $\omega \succ \omega'$ *(resp.* $\omega \succeq \omega'$*).*
- $\succeq$ *satisfies* $p >_{pes} q$ *(resp.* $p \geq_{pes} q$*) iff* $\forall \omega \in \min(p \wedge \neg q, \succeq)$, $\forall \omega' \in \min(\neg p \wedge q, \succeq)$ *we have* $\omega \succ \omega'$ *(resp.* $\omega \succeq \omega'$*).*
- $\succeq$ *satisfies* $p >_{opp} q$ *(resp.* $p \geq_{opp} q$*) iff* $\forall \omega \in \max(p \wedge \neg q, \succeq)$, $\forall \omega' \in \min(\neg p \wedge q, \succeq)$ *we have* $\omega \succ \omega'$ *(resp.* $\omega \succeq \omega'$*).*

Definition 3 (Preference set). *A preference set of type $\rhd$, denoted $\mathcal{P}_{\rhd}$, is a set of preferences of the form $\{p_i \rhd q_i | i = 1, \cdots, n\}$, where $\rhd \in \{>_{st}, >_{opt}, >_{pes}, >_{opp}, \geq_{st}, \geq_{opt}, \geq_{pes}, \geq_{opp}\}$. A complete preorder $\succeq$ is a model of $\mathcal{P}_{\rhd}$ if and only if $\succeq$ satisfies each preference $p_i \rhd q_i$ in $\mathcal{P}_{\rhd}$.*

A set $\mathcal{P}_{\rhd}$ is consistent if and only if it admits a model, namely if there exists at least one preorder satisfying each preference $\mathcal{P}_{\rhd}$.

4 From comparative preference statements to complete preorders

Two kinds of queries can be drawn when we deal with a set of comparative preference statements: (1) which are the preferred outcomes? (2) is an outcome preferred to another outcome? In many applications (for e.g. database queries), users are more concerned with the preferred outcomes. However preferred outcomes are not always feasible. For example the best meals w.r.t. a user's preferences may be no longer available so we have to look for meals that are immediately less preferred w.r.t. user's preferences. In such a case a complete preorder on meals is needed to answer user's preferences. Therefore we focus on procedures that derive complete preorders on outcomes. In the next subsections, we recall algorithms which derive a unique complete preorder given a set of preferences of the same type following minimal/maximal specificity principle.

Proposition 1. *[7]*

- *The least specific model of* $\mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$ *(resp.* $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$*) is unique.*
- *The most specific model of* $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$ *(resp.* $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$*) is unique.*
- *The most specific model of* $\mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$ *(resp.* $\mathcal{P}_{>_{opp}} \cup \mathcal{P}_{\geq_{opp}}$*) is not unique.*
- *The least specific model of* $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$ *(resp.* $\mathcal{P}_{>_{opp}} \cup \mathcal{P}_{\geq_{opp}}$*) is not unique.*

Note that the notion of minimal/maximal specificity principle for preference sets consisting of non-strict preferences only is not interesting. Indeed the model is trivial in which all outcomes are equivalent.
A unique complete preorder for opportunistic preferences does not exist w.r.t. these principles. Indeed we do not consider such preferences in the remainder of this section.

Let $\mathcal{P}_{\rhd} = \{s_i : p_i \rhd q_i | i = 1, \cdots, n\}$ be a preference set with $\rhd \in \{>_x, \geq_x | x \in \{st, opt, pes\}\}$. Given $\mathcal{P}_{\rhd}$, we define a set of pairs on Ω as follows:

$$\mathcal{C}(\mathcal{P}_{\rhd}) = \{C_i = (L(s_i), R(s_i)) | s_i \in \mathcal{P}_{\rhd}, i = 1, \cdots, n\},$$

where $L(s_i) = \{\omega | \omega \in \Omega, \omega \models p_i \wedge \neg q_i\}$ and $R(s_i) = \{\omega | \omega \in \Omega, \omega \models \neg p_i \wedge q_i\}$.

Example 1. Let *dish*, *wine* and *dessert* be three variables such that $Dom(dish) = \{fish, meat\}$, $Dom(wine) = \{white, red\}$ and $Dom(dessert) = \{cake, ice\_cream\}$.

We have $\Omega = \{$ $\omega_0 = fish - white - ice\_cream$, $\omega_1 = fish - white - cake$,
$\omega_2 = fish - red - ice\_cream$, $\omega_3 = fish - red - cake$,
$\omega_4 = meat - white - ice\_cream$, $\omega_5 = meat - white - cake$,
$\omega_6 = meat - red - ice\_cream$, $\omega_7 = meat - red - cake\}$.

Let $\mathcal{P}_{\rhd} = \{s_1 : fish \rhd meat,\ s_2 : red \wedge cake \rhd white \wedge ice\_cream,\ s_3 : fish \wedge white \rhd fish \wedge red\}$. We have $\mathcal{C}(\mathcal{P}_{\rhd}) = \{C_1 = (\{\omega_0, \omega_1, \omega_2, \omega_3\}, \{\omega_4, \omega_5, \omega_6, \omega_7\})$, $C_2 = (\{\omega_3, \omega_7\}, \{\omega_0, \omega_4\}), C_3 = (\{\omega_0, \omega_1\}, \{\omega_2, \omega_3\})\}$.

4.1 Optimistic preferences

As stated in Proposition 1, a unique model of optimistic preferences can be characterized following minimal specificity principle, used in System Z [8]. Following this principle, each outcome is put in the highest possible rank in the model. Therefore the principle is built under the assumption that an outcome is preferred unless there is a reason to state the contrary. Algorithm 1.1 gives the way this preorder is computed. At each step l of the algorithm, we put in E_l outcomes that are not dominated by any other outcome. These outcomes are those which do not appear in the right-hand side of any pair $(L(s_i), R(s_i))$ of $\mathcal{C}(\mathcal{P}_{>_{opt}}) \cup \mathcal{C}(\mathcal{P}_{\geq_{opt}})$.

Example 2. (Example 1 con'd) Let $\mathcal{P}_{>_{opt}} = \{s_1 : fish >_{opt} meat,\ s_2 : red \wedge cake >_{opt} white \wedge ice\_cream,\ s_3 : fish \wedge white >_{opt} fish \wedge red\}$.
We have $E_1 = \{\omega_1\}$. We remove C_1 and C_3 since $s_1 = fish >_{opt} meat$ and $s_3 : fish \wedge white >_{opt} fish \wedge red$ are satisfied. We get $\mathcal{C}(\mathcal{P}_{>_{opt}})$ = $\{C_2 = (\{\omega_3, \omega_7\}, \{\omega_0, \omega_4\})\}$. Now $E_2 = \{\omega_2, \omega_3, \omega_5, \omega_6, \omega_7\}$. We remove C_2 since $s_2 : red \wedge cake >_{opt} white \wedge ice\_cream$ is satisfied. So $\mathcal{C}(\mathcal{P}_{>_{opt}}) = \emptyset$. Lastly, $E_3 = \{\omega_0, \omega_4\}$. Indeed $\succeq$= $(\{\omega_1\}, \{\omega_2, \omega_3, \omega_5, \omega_6, \omega_7\}, \{\omega_0, \omega_4\})$. We can check that each outcome has been put in the highest possible rank in $\succeq$. Therefore, if we push any outcome to a higher rank then the preorder would not satisfy the preference set. For instance, $\succeq'$= $(\{\omega_1, \omega_5\}, \{\omega_2, \omega_3, \omega_6, \omega_7\}, \{\omega_0, \omega_4\})$ does not satisfy $s_1 : fish >_{opt} meat$.

4.2 Pessimistic preferences

The unique model of pessimistic preferences obeys maximal specificity principle. Accordingly, each outcome is put in the lowest possible rank in the preorder. The principle is built under the assumption that an outcome is not preferred unless there is a reason to state the contrary. Algorithm 1.2 gives the way this preorder is computed.

Example 3. (Example 1 con'd) Let $\mathcal{P}_{>_{pes}} = \{s_1 : fish >_{pes} meat,\ s_2 : red \wedge cake >_{pes} white \wedge ice\_cream,\ s_3 : fish \wedge white >_{pes} fish \wedge red\}$. We have $E_1 = \{\omega_4, \omega_5, \omega_6\}$. We remove C_1 and C_2 since $s_1 : fish >_{pes} meat$ and $s_2 : red \wedge$

Data: A preference set $\mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$.
Result: A complete preorder $\succeq$ on Ω.
begin
  $l = 0$;
  while $\Omega \neq \emptyset$ **do**
    $l = l + 1, k = 1$;
    $E_l = \{\omega | \omega \in \Omega, \nexists (L(s_i), R(s_i)) \in \mathcal{C}(\mathcal{P}_{>_{opt}}), \omega \in R(s_i)\}$;
    /** Handling non-strict optimistic preferences **/
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{opt}})$ **do**
        if $L(s_i) \cap E_l = \emptyset$ *and* $R(s_i) \cap E_l \neq \emptyset$ **then**
          $E_l = E_l \backslash R(s_i), k = 1$
    /** if preferences are inconsistent, the algorithm stops and the current preorder is returned **/
    if $E_l = \emptyset$ **then return** $\succeq = (E_1, \cdots, E_{l-1})$;
    $\Omega = \Omega \backslash E_l$;
    /** remove satisfied preferences **/
    remove $(L(s_i), R(s_i))$ in $\mathcal{C}(\mathcal{P}_{>_{opt}}) \cup \mathcal{C}(\mathcal{P}_{\geq_{opt}})$ where $L(s_i) \cap E_l \neq \emptyset$;
  return $\succeq = (E_1, \cdots, E_l)$;
end

Algorithm 1.1: A complete preorder associated to $\mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$.

$cake >_{pes} white \wedge ice\_cream$ are satisfied. We repeat the same reasoning and get $E_2 = \{\omega_2, \omega_3, \omega_7\}$ and $E_3 = \{\omega_0, \omega_1\}$. So $\succeq = (\{\omega_0, \omega_1\}, \{\omega_2, \omega_3, \omega_7\}, \{\omega_4, \omega_5, \omega_6\})$. We can check that each outcome has been put in the lowest possible rank in the preorder.

4.3 Strong preferences

Strong preferences induce a unique partial order on outcomes. We can use both construction principles used in optimistic and pessimistic preferences to linearize the partial order and compute a unique complete preorder. Algorithms 1.1 and 1.2 can be adapted to deal with strong preferences. Due to space limitation, we only give the algorithm adapting Algorithm 1.1. See Algorithm 1.3.

Example 4. (Example 1 con'd) Let $\mathcal{P}_{>_{st}} = \{s_1 : fish >_{st} meat,\ s_2 : red \wedge cake >_{st} white \wedge ice\_cream,\ s_3 : fish \wedge white >_{st} fish \wedge red\}$. There is no complete preorder which satisfies $\mathcal{P}_{>_{st}}$ so $\mathcal{P}_{>_{st}}$ is inconsistent. This is due to s_1 and s_2. Following s_1, ω_0 is preferred to ω_7 whereas ω_7 is preferred to ω_0 following s_2.

Example 5. (Consistent strong preferences)
Let $\mathcal{P}_{>_{st}} = \{fish \wedge white >_{st} fish \wedge red, red \wedge cake >_{st} red \wedge ice\_cream, meat \wedge red >_{st} meat \wedge white\}$. Following Algorithm 1.3, we have $\succeq = (\{\omega_0, \omega_1, \omega_7\}, \{\omega_3\}, \{\omega_2, \omega_6\}, \{\omega_4, \omega_5\})$. Now following the adaptation of Algorithm 1.2 to deal with strong preferences, we have $\succeq = (\{\omega_0, \omega_1\}, \{\omega_3, \omega_7\}, \{\omega_6\}, \{\omega_2, \omega_4, \omega_5\})$.

Data: A preference set $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$.
Result: A complete preorder $\succeq$ on Ω.
begin
  $l = 0$;
  while $\Omega \neq \emptyset$ **do**
    $l = l + 1, k = 1$;
    $E_l = \{\omega | \omega \in \Omega, \nexists (L(s_i), R(s_i)) \in \mathcal{C}(\mathcal{P}_{>_{pes}}), \omega \in L(s_i)\}$;
    /** Handling non-strict pessimistic preferences **/
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{pes}})$ **do**
        if $R(s_i) \cap E_l \neq \emptyset$ *and* $L(s_i) \cap E_l = \emptyset$ **then**
          $E_l = E_l \backslash L(s_i)$, $k = 1$
    /** if preferences are inconsistent, the algorithm stops and the current preorder is returned **/
    if $E_l = \emptyset$ **then return** $\succeq = (E_{l-1}, E_{l-2}, \cdots, E_1)$;
    $\Omega = \Omega \backslash E_l$;
    /** remove satisfied preferences **/
    remove $(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$ where $R(s_i) \cap E_l \neq \emptyset$
  return $\succeq = (E_l, E_{l-1}, \cdots, E_1)$;
end

Algorithm 1.2: A complete preorder associated to $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$.

It may happen that a user expresses heterogeneous preferences, i.e., preferences obeying different semantics. The following result states that the different semantics can be divided into two categories following minimal and maximal specificity principles.

Proposition 2. *[7]*

- *The least specific model of* $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}} \cup \mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$ *is unique.*
- *The most specific model of* $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}} \cup \mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$ *is unique.*

5 Our framework

Minimal and maximal specificity principles have been extensively used in literature [8, 6, 4, 5, 7]. However one may argue that the separate use of the two principles is debatable. Let us consider the following example:

Example 6. Let *dish* and *wine* be two propositional variables such that $Dom(dish) = \{fish, meat\}$ and $Dom(wine) = \{white, red\}$. Let $\mathcal{P}_{>_{st}} = \{s_1 : fish >_{st} meat, s_2 : meat \wedge red >_{st} fish \wedge red, s_3 : fish \wedge white >_{st} fish \wedge red\}$. We have $\mathcal{C}(\mathcal{P}_{>_{st}}) = \{(\{fish - white,\ fish - red\},\ \{meat - white, meat - red\}),\ (\{meat - red\},\ \{fish - red\}),\ (\{fish - white\},\ \{fish - red\})\}$.
Let us first compute the model of $\mathcal{P}_{>_{st}}$ following the minimal specificity principle. We have $E_1 = \{fish - white\}$. We update $\mathcal{C}(\mathcal{P}_{>_{st}})$ and get $\mathcal{C}(\mathcal{P}_{>_{st}}) = \{(\{fish - red\},$

Data: A preference set $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$.
Result: A complete preorder $\succeq$ on Ω.
begin
  $l = 0$;
  while $\Omega \neq \emptyset$ **do**
    $l = l + 1, k = 1$;
    $E_l = \{\omega | \omega \in \Omega, \nexists (L(s_i), R(s_i)) \in \mathcal{C}(\mathcal{P}_{>_{st}}), \omega \in R(s_i)\}$;
    /** Handling non-strict strong preferences **/
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{st}})$ **do**
        if $L(s_i) \not\subseteq E_l$ *and* $R(s_i) \cap E_l \neq \emptyset$ **then**
          $E_l = E_l \backslash R(s_i), k = 1$
    /** if preferences are inconsistent, the algorithm stops and the current preorder is returned **/
    if $E_l = \emptyset$ **then return** $\succeq = (E_1, \cdots, E_{l-1})$;
    $\Omega = \Omega \backslash E_l$;
    substitute $(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$ by $(L(s_i) \backslash E_l, R(s_i))$;
    /** remove satisfied preferences **/
    remove $(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$ where $L(s_i) = \emptyset$
  return $\succeq = (E_1, \cdots, E_l)$;
end

Algorithm 1.3: A complete preorder associated to $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$.

$\{meat - white, meat - red\})$, $(\{meat - red\}, \{fish - red\})$, $(\{fish - white\}, \{fish - red\})\}$. The algorithm now stops since the preference statements are inconsistent. Indeed we have $fish - red$ is preferred to $meat - red$ following s_2 and $meat - red$ is preferred to $fish - red$ following s_3. Besides, let us note that even though the preference statements are inconsistent, $meat - white$ should be the least preferred outcome since it is dominated and does not dominates any other outcome. However such a result cannot be obtained following the minimal specificity principle since the algorithm stops as soon as inconsistency is met. This is called the drowning problem [3].
Let us now compute the model of $\mathcal{P}_{>_{st}}$ following the maximal specificity principle. We have $E_1 = \{meat - white\}$. We update $\mathcal{C}(\mathcal{P}_{>_{st}})$ and get $\mathcal{C}(\mathcal{P}_{>_{st}}) = \{(\{fish - white, fish - red\}, \{meat - red\})$, $(\{meat - red\}, \{fish - red\})$, $(\{fish - white\}, \{fish - red\})\}$. The algorithm now stops since the preference statements are inconsistent. Again we have both $fish - red$ preferred to $meat - red$ and $meat - red$ preferred to $fish - red$. We can notice that $fish - white$ dominates other outcomes but is not dominated by any other outcome. Therefore it should be stated as the most preferred outcome. However this is not possible following the maximal specificity principle since the algorithm also stops as soon as an inconsistency is met.

The limitation of minimal and maximal specificity principles stated in the above example is due to their separate use. In this section, we develop an algorithm which integrates both principles. The basic idea of our algorithm is that outcomes are not ordered following the semantics of comparative preference statements at hand. Instead,

we look for (1) outcomes that dominate other outcomes w.r.t. at least one statement, but are not dominated w.r.t. any statement, and (2) outcomes that do not dominate any other outcomes. The first set of outcomes is the set of the best (i.e., preferred) outcomes and the second is the set of the least preferred outcomes. Once these two sets are computed, the semantics of preference statements is used to check whether the statement is satisfied or not yet. Our proposed approach to compute a preorder using both maximal *and* minimal specificity principles is depicted in Algorithm 1.4. This algorithm starts each step, or iteration, l by computing two sets of outcomes E_l and E'_l that contain the outcomes that dominate and the ones that do not dominate other outcomes, respectively. Then, the special case of non strict preferences (under the form $\geq_x$ with $x \in \{st, opt, pes, opp\}$) is handled by possibly removing outcomes from E_l and E'_l. If both E_l and E'_l sets are empty, the algorithm is stopped and returns the current preorder, since preferences are inconsistent. Otherwise, satisfied preferences are removed. If the current set of remaining outcomes is not empty, the algorithm starts an $l+1$ iteration. Otherwise, the preorder is complete and is returned.

Example 7. (con'd) Let $\mathcal{P} = \mathcal{P}_{>_{pes}} \cup \mathcal{P}_{>_{opt}}$ with $\mathcal{P}_{>_{pes}} = \{fish >_{pes} meat\}$ and $\mathcal{P}_{>_{opt}} = \{red \wedge cake >_{opt} white \wedge ice\_cream, fish \wedge white >_{opt} fish \wedge red\}$.
We have $\mathcal{C}(\mathcal{P}) = \mathcal{C}(\mathcal{P}_{>_{pes}}) \cup \mathcal{C}(\mathcal{P}_{>_{opt}}) = \{(\{\omega_0, \omega_1, \omega_2, \omega_3\}, \{\omega_4, \omega_5, \omega_6, \omega_7\})\} \cup \{(\{\omega_3, \omega_7\}, \{\omega_0, \omega_4\}), (\{\omega_0, \omega_1\}, \{\omega_2, \omega_3\})\}$.
We first have $E'_1 = \{\omega_4, \omega_5, \omega_6\}$ and $E_1 = \{\omega_1\}$. We update $\mathcal{C}(\mathcal{P})$ and get $\{(\{\omega_3, \omega_7\}, \{\omega_0, \omega_4\})\}$. Now we have $E'_2 = \{\omega_0, \omega_2\}$ and $E_2 = \{\omega_2, \omega_3, \omega_7\} \backslash E'_2 = \{\omega_3, \omega_7\}$. So, the returned preorder is $(E_1, E_2, E'_2, E'_1) = (\{\omega_1\}, \{\omega_3, \omega_7\}, \{\omega_0, \omega_2\}, \{\omega_4, \omega_5, \omega_6\})$. We can check that this preorder satisfies each preference in $\mathcal{P}$.

Our algorithm does not provide a unique model following minimal or maximal specificity principle since it combines both principles. However it is unique following the basic idea of its construction which consists in computing the sets of outcomes that dominate other outcomes and those which do not dominate any other outcomes. Therefore at each step of the algorithm we ensure that E_l (resp. E'_l) contains all outcomes that dominate (resp. do not dominate) other outcomes. It is worth noticing that our algorithm also handles opportunistic preferences for which a unique model does not exist following minimal and maximal specificity principles when considered separately. Let us also emphasize that although our algorithm can deal with the four presented semantics of preferences, all of them are not necessary and our approach can also be used with any subset of those semantics.

6 From preference representation to spatial or temporal representation

Representing and reasoning about time and space is an important task in many domains. For this purpose, numerous qualitative approaches have been proposed to represent spatial or temporal entities and their relations and reason about them (see e.g. [1, 9]).

Data: A preference set $\mathcal{P}$.
Result: A complete preorder $\succeq$ on Ω.
begin
  $l = 0$;
  while $\Omega \neq \emptyset$ **do**
    $l = l + 1$;
    $E'_l = \{\omega | \omega \in \Omega, \nexists (L(s_i), R(s_i)) \in (\mathcal{C}(\mathcal{P}_{>_{st}}) \cup \mathcal{C}(\mathcal{P}_{>_{opt}}) \cup \mathcal{C}(\mathcal{P}_{>_{pes}}) \cup \mathcal{C}(\mathcal{P}_{>_{opp}})), \omega \in L(s_i)\}$;
    $E_l = \{\omega | \omega \in \Omega, \nexists (L(s_i), R(s_i)) \in (\mathcal{C}(\mathcal{P}_{>_{st}}) \cup \mathcal{C}(\mathcal{P}_{>_{opt}}) \cup \mathcal{C}(\mathcal{P}_{>_{pes}}) \cup \mathcal{C}(\mathcal{P}_{>_{opp}})), \omega \in R(s_i)\} \backslash E'_l$;
    /*** non-strict strong preferences ***/
    $k = 1$;
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{st}})$ **do**
        if $L(s_i) \not\subseteq E_l$ *and* $R(s_i) \cap E_l \neq \emptyset$ **then** $E_l = E_l \backslash R(s_i), k = 1$;
    /*** non-strict optimistic and non-strict opportunistic preferences ***/
    $k = 1$;
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{opt}}) \cup \mathcal{C}(\mathcal{P}_{\geq_{opp}})$ **do**
        if $L(s_i) \cap E_l = \emptyset$ *and* $R(s_i) \cap E_l \neq \emptyset$ **then** $E_l = E_l \backslash R(s_i), k = 1$;
    /*** non-strict pessimistic preferences ***/
    $k = 1$;
    while $k = 1$ **do**
      $k = 0$;
      foreach $(L(s_i), R(s_i))$ *in* $\mathcal{C}(\mathcal{P}_{\geq_{pes}})$ **do**
        if $L(s_i) \cap E'_l \neq \emptyset$ *and* $R(s_i) \cap E'_l = \emptyset$ **then** $E'_l = E'_l \backslash L(s_i), k = 1$
    /** if inconsistent preferences, the algorithm stops and the current preorder is returned **/
    if $E_l = \emptyset$ *and* $E'_l = \emptyset$ **then**
      $E_l = \Omega$, **return** $\succeq = (E_1, \cdots, E_l, E'_{l-1}, \cdots, E'_1)$;
    $\Omega = \Omega \backslash (E_l \cup E'_l)$;
    substitute $(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$ by $(L(s_i), R(s_i) \backslash E'_l)$;
    /** remove satisfied preferences **/
    remove$(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{st}} \cup \mathcal{P}_{\geq_{st}}$ where $R(s_i) = \emptyset$;
    remove$(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{opt}} \cup \mathcal{P}_{\geq_{opt}}$ where $L(s_i) \cap E_l \neq \emptyset$;
    remove$(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{pes}} \cup \mathcal{P}_{\geq_{pes}}$ where $R(s_i) \cap E'_l \neq \emptyset$;
    remove$(L(s_i), R(s_i))$ in $\mathcal{P}_{>_{opp}} \cup \mathcal{P}_{\geq_{opp}}$ where $L(s_i) \cap E_l \neq \emptyset$ or $R(s_i) \cap E'_l \neq \emptyset$
  return $\succeq = (E_1, \cdots, E_l, E'_l, \cdots, E'_1)$ *after removing empty* E_i *and* E'_i ;
end

Algorithm 1.4: A complete preorder associated to $\mathcal{P}$.

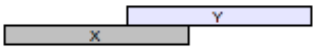
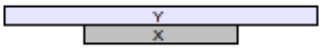
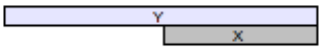
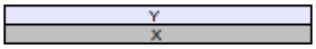
Relation	Mathematical encoding	Encoding in our framework
X precedes Y 	$\max(X) < \min(Y)$	$Y >_{st} X$
X meets Y 	$\max(X) = \min(Y)$	$X \geq_{opp} Y$ $Y \geq_{st} X$
X overlaps Y 	$\max(X) > \min(Y)$ $\min(X) < \min(Y)$ $\max(X) < \max(Y)$	$Y >_{opp} X$ $Y >_{pes} X$ $Y >_{opt} X$
X starts Y 	$\min(X) = \min(Y)$ $\max(X) < \max(Y)$	$X \geq_{pes} Y$ $Y \geq_{pes} X$ $Y >_{opt} X$
X during Y 	$\min(X) > \min(Y)$ $\max(X) < \max(Y)$	$X >_{pes} Y$ $Y >_{opt} X$
X finishes Y 	$\max(X) = \max(Y)$ $\min(X) > \min(Y)$	$X \geq_{opt} Y$ $Y >_{opt} X$ $X >_{pes} Y$
X equals Y 	$\min(X) = \min(Y)$ $\max(X) = \max(Y)$	$X \geq_{pes} Y$ $Y \geq_{pes} X$ $X \geq_{opt} Y$ $Y \geq_{opt} X$

Table 1. Relations between intervals and their encoding in our framework

As a case study, this section is concerned with interval algebra proposed in [1]. We consider an interval as a set of discrete and finite elements. Roughly, a (time) interval can have thirteen possible positions with respect to another interval. Seven of them are presented in the first column of Table 1. Obviously enough, reversing the positions of X and Y, defines other relations, except for "equals" that stays identical. The beginning of a time interval I is denoted $\min(I)$, and its end is denoted $\max(I)$. All relations between two intervals are expressed with those notations in the second column of Table 1. For instance, "an interval X *precedes* an interval Y" can be expressed under the form $\max(X) < \min(Y)$. Most of relations are encoded under several conditions. For example, "X *during* Y" needs both $\max(Y) > \max(X)$ *and* $\min(X) > \min(Y)$ to hold.

Considering intervals as preference statements, the mathematical encoding of the relations enables us to build an equivalent encoding in our framework. The latter is presented is the third column of Table 1. Each relation between intervals is translated into different (sets of) semantics. For instance, "X *during* Y" is encoded using two preference semantics at the same time. Indeed, Y must be preferred to X in an optimistic way to ensure that at least one outcome in X is preferred to all outcomes in X. Moreover, X

must be pessimisticly preferred to Y to translate that there exists at least one outcome in Y that is less preferred to all outcomes in Y. The encoding of the different relations of our framework shows the possibility to relate two different frameworks, namely preference representation and qualitative spatial-temporal reasoning, which could be a great opportunity to take advantage of one of the many constraint-based reasoning calculi proposed in the last decade.

7 Conclusion

In this paper, we present a framework to deal with both maximal and minimal specificity principles in preference representation. Whereas most of existing approaches focus on only one of those principles, we argue that combining both principles may be of interest. More precisely, given a set of comparative preference statements, we consider both the outcomes that dominate other ones *and* those that do not dominate other outcomes. We also propose an algorithm that returns a unique complete preorder from a set of preference statements on the basis of this principle. Then, some links between our framework and spatial-temporal frameworks are discussed. Our first results pled for further investigations.

References

1. J.F. Allen. An interval-based representation of temporal knowledge. In *7th International Joint Conference on Artificial Intelligence (IJCAI'81)*, pages 221–226, 1981.
2. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar possibilistic representations. In *UAI'02*, pages 45–52, 2002.
3. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings of 3rd International Conference of Principles of Knowledge Representation and Reasoning (KR'92)*, pages 673–684, 1992.
4. S. Benferhat, D. Dubois, and H. Prade. Towards a possibilistic logic handling of preferences. *Applied Intelligence*, 14(3):303–317, 2001.
5. S. Benferhat and S. Kaci. A possibilistic logic handling of strong preferences. In *International Fuzzy Systems Association (IFSA'01)*, pages 962–967, 2001.
6. C. Boutilier. Toward a logic for qualitative decision theory. In *4th International Conference on Principles of Knowledge Representation, (KR'94)*, pages 75–86, 1994.
7. S. Kaci and L. van der Torre. Reasoning with various kinds of preferences: Logic, nonmonotonicity and algorithms. *Annals of Operations Research*, 163(1):89–114, 2008.
8. J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'90)*, pages 121–135, 1990.
9. D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *KR'92*, pages 165–176, 1992.
10. L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory. *Applied Intelligence*, 14(3):285–301, 2001.
11. G. H. von Wright. *The Logic of Preference*. University of Edinburgh Press, 1963.
12. N. Wilson. Extending CP-nets with stronger conditional preference statements. In *AAAI*, pages 735–741, 2004.
13. R.R. Yager. Entropy and specificity in a mathematical theory of evidence. *International Journal of General Systems*, 9:249–260, 1983.

Proposition d'un modèle hiérarchique et coopératif pour la segmentation d'image

Mansouri Ziad[1], Hayet Farida Merouani[1],

[1] Département d'informatique
Laboratoire LRI/Equipe SRF
Université Badji Mokhtar Annaba Algérie
Mansouri Ziad, manzedinf@yahoo.fr.com
Hayet Farida Merouani, hayet_merouani@yahoo.fr

Résumé: Une nouvelle approche hybride de segmentation d'images couleurs ou en niveau de gris est proposée dans ce travail. C'est une approche hiérarchique et adaptative basée sur une coopération région-contour. La segmentation procède par l'élaboration d'un ensemble de régions et de contours initiaux qui vont être améliorés mutuellement et hiérarchiquement dans un environnement multi-agents offrant une vision à la fois globale et locale au processus de segmentation.

Mots clés: segmentation, coopération région-contour, Color Structure Code, traitement d'image.

1 Introduction

La segmentation d'images est une étape cruciale dans tout processus d'analyse d'image. Elle consiste à préparer l'image afin de la rendre mieux exploitable par un processus automatique telle que l'interprétation.

L'approche de segmentation par contour consiste à localiser les frontières des objets, et qui opère d'une manière purement locale, complique donc la délimitation et la précision de ces objets [1].

Les approches de segmentation par région quant à elles agissent en partitionnant l'image en un ensemble de régions où chacune désigne un ou plusieurs objets connexes, mais ils ont tendance à déformer les frontières naturelles des objets.

Dans la pratique les meilleurs résultats de segmentation sont obtenus en combinant conjointement des méthodes distinctes. En faisant cela nous obtenons des approches hybrides plus solides et plus efficaces, car la limite d'une méthode peut être surpassée par une autre, ou bien sa force peut être renforcée [8].

Dans ce travail, on propose un système de segmentation qui offre une coopération région-contours au sein d'un système multi agent. Notre objectif est de concevoir un système qui engendrera des résultats de bonne qualité pour la segmentation d'images couleurs et en niveau de gris tout en ayant un temps d'exécution acceptable.

Notre approche de segmentation requiert l'utilisation d'une topologie hexagonale spéciale pour coder l'image afin que nous puissions utiliser l'algorithme de

segmentation région CSC qui sera présentée en 2. Le principe de cette segmentation est donné en section 3, en précisant le seuil adaptatif, la coopération contour-contour en 3.2, la coopération région-contour en 3.3 et la correction des régions en 3.4. La plate forme d'agents est donnée en section 4.
Quelques précisions concernant l'implémentation sont données en section 5, on termine cet article par une conclusion et discussion.

2 Segmentation par Color Structure Code

CSC est une méthode de segmentation par région, introduit par Rehrmann [12]. Son fonctionnement requiert l'utilisation d'une structure hexagonale hiérarchique pour coder l'image.
Au début, l'image est structurée en un ensemble de petits ilots contenant chacun 7 pixels. Ces ilots initiaux se recouvrent où chaque deux ilots adjacents partagent un seul pixel. Avec cette structure chaque ilot a exactement 7 ilots voisins comme illustre la figure1 [3] suivante:

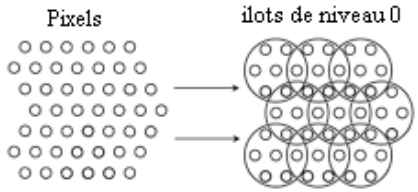


Figure 1 : Structuration des pixels dans des îlots.

Ces ilots initiaux forment le niveau 0 de la hiérarchie. Pour générer le niveau suivant on considère que les ilots initiaux sont des pixels et on réitère le processus. Donc chaque ilot de niveau 1 est formé par l'assemblement de 7 ilots de niveau 0.
Le processus est itéré de façons que chaque îlot de niveau *n* sera constitué de 7 ilots de niveau *n-1* jusqu'à l'obtention d'un seul îlot qui englobe toute l'image [3].

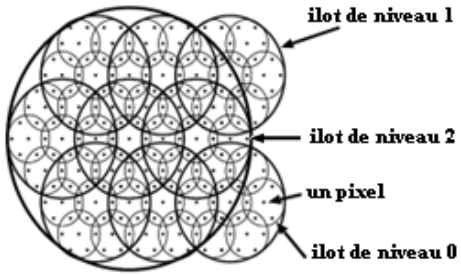


Figure 2 : les îlots de différents niveaux [3]

La problématique triviale de cette topologie hexagonale est que la plupart des outils d'acquisition et d'affichage des images adoptent une topologie orthogonale. Pour cela on propose de simuler une topologie hexagonale sur un grillage orthogonal, comme ce qui est résumé dans la figure 3 proposée par [6] :

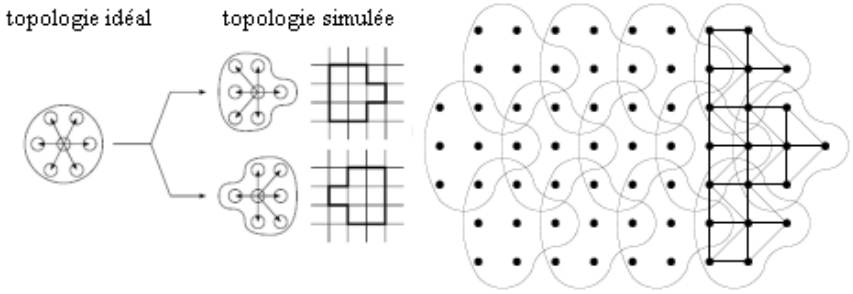


Figure 3 : adaptation de la topologie hexagonale sur une topologie orthogonale

Généralement pour une image de taille : $(2^m + 1) \times (2^m + 1)$ on aura *m* niveaux.
Cette manière de grouper les zones de l'image d'une manière élégante et hiérarchiques facilite et encourage la répartition et le partage du processus de segmentation.
Cette hiérarchie hexagonale représente un squelette qui va être utilisée par le processus de segmentation CSC. Ce dernier procède en 3 étapes distinctes, à savoir l'initialisation, le groupage et le découpage.

2.1 L'initialisation

Cette phase traite les îlots de niveau 0 seulement, elle consiste à appliquer dans chaque îlot un algorithme de croissance de régions, ceci va donner entre une et sept régions -de niveau 0- dans chaque îlot. Puisque ce processus traite les îlots d'une manière indépendante, cette étape peut être exécutée en parallèle sur les différents îlots de niveau 0.
Pour mesurer la similarité entre régions il est préférable d'utiliser la représentation HSV des couleurs au lieu du RGB car il s'accorde mieux au système visuel humain ce qui conduit à un meilleur résultat de segmentation.
Les régions obtenues dans cette phase sont encapsulées dans des code-éléments (ou code-région). Un code-élément est une structure qui décrit une région ainsi que toutes les informations la concernant (couleur moyennes, taille,...etc.). Ainsi la phase d'initialisation figure 4 [3], consiste à créer les code-éléments de niveau 0.

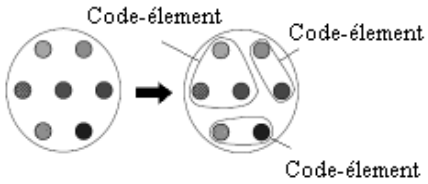


Figure 4 : Phase d'initialisation

2.2 Le groupement

Dans la phase d'initialisation (figure 4) le processus de segmentation a eu une vision limitée aux ilots initiaux traité indépendamment, si on passe au niveau suivant on aura des ilots de niveau 1 qui englobent chaqu'un 7 ilots de niveau 0 avec leurs code-elements respectifs. Donc on aura une vision plus globale qui nous permettra de fusionner les code-elements homogènes si nécessaire.
Donc cette phase consiste à créer les code-elements des niveaux supérieurs à 0 comme suit :
Dans chaque ilot de niveau *n* on génère les codes-éléments de niveau *n* en groupant les code-éléments de niveau *n-1* qui sont à la fois connectés et similaires.
La phase de groupement ressemble a la phase d'initialisation, seulement elle ne groupe pas des pixels mais des sous régions indiquées par des code-éléments. Et la encore le processus peut être complètement parallèle pour chaque îlot de niveau *n*.
Le résultat de cette étape est un ensemble d'arbre de code-elements. Lorsqu'un code-élément d'un niveau quelconque *i* ne sera plus fusionné avec aucun autre code-élément de même niveau, alors ce dernier devient la racine d'un arbre qui désigne une région proprement dite. Donc l'image segmentée sera représentée par une liste de segments où chaque segment est désigné par une racine d'un arbre de code-élément.
Pour clarifier la structure de cet arbre on signale que chaque code-element a au maximum deux parents et un nombre *n* d'enfants.
Le fait de rechercher si deux codes éléments sont connectés devient une chose aisée avec l'utilisation de la topologie hexagonale vu que les îlots se recouvrent partiellement. Car deux code-éléments de niveau *i* sont connectés s'ils ont au moins un code d'éléments de niveau *i-1* en commun (2 code-éléments de niveau *0* sont connectés s'ils ont au moins un pixel en commun). Cette caractéristique offre plus de rapidité et moins de complexité, contrairement aux techniques split & merge qui utilisent des graphes d'adjacences (temps de calcul, coût de MAJ,...etc.) [14].

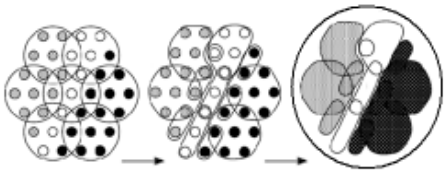


Figure 5 : Phase de groupage

2.3 Le découpage

Comme on peut le constater, le processus de segmentation est jusqu'ici purement local, s'il existe un groupe de pixels entre deux régions qui changent de couleurs finement entre-elle (le problème d'enchaînement successif et aveugle des pixels voisins des méthodes locales) on peut fusionner ces deux régions même s'ils ne sont pas assez homogènes. Dans ce cas on va avoir des petites sous régions homogènes, en les fusionnant on obtient une grande région non homogène.
Ce problème dont souffre la plupart des méthodes de segmentation utilisant uniquement l'information locale, peut être résolu en ajoutant une vision globale au résultat obtenue, afin de corriger toute fusion non adéquate.
Ainsi, la phase de découpage opère simultanément avec la phase de groupement, en vérifiant le respect de similarité entre les différents code-éléments nouvellement connectés dans chaque niveau. Si la phase de groupement à mal grouper des code-éléments, créant ainsi un nouveau code-élément non homogène, alors la phase de découpage détecte automatiquement les sous code-éléments responsable et les privent de faire partie du code-éléments englobant (elle découpe ce dernier), ce qui donne des régions homogènes.
On signale que découper deux code-éléments déjà connectés opère d'une manière récursive dans toutes les parties communes de ces deux code-éléments. N'oublions pas que dans l'algorithme CSC tous les ilots s'interposent (structure hexagonale), donc même si ces deux codes-éléments vont être séparés, ils auront quand même une zone commune. Ce qui nécessite qu'on descend vers le niveau de cette zone commune et qu'on la découpe elle aussi d'une manière récursive jusqu'au niveau des pixels.
On peut conclure que la phase de découpage est couteuse en temps de calcule, mais fort heureusement pour nous qu'elle ne s'exécute que très rarement dans la pratique (en réalité, il y aura que des groupages et très peu de découpages).
On peut améliorer le résultat de segmentation en changeant le seuil ou les critères d'homogénéité dans chaque niveau. Par exemple les critères seront plus stricts dans les niveaux supérieurs qu'aux niveaux inferieurs.
La phase de découpage peut engendrer quelques problèmes dans certains cas très rares où on peut avoir des éléments non connexes (des régions disjointes), ou des régions vides. On peut corriger ces erreurs en ajoutant une phase de vérification qui

contrôle l'état des régions après chaque découpage (dans notre système, l'agent contrôleur s'occupe de ça) [6].

Finalement, on peut remarquer que la méthode CSC avec son organisation hiérarchique et parallèle et sa vision locale et globale s'adapte bien à un système multi-agent, d'où notre choix s'est porté sur elle. Pour avoir plus d'information sur la structure des ilots et l'implémentation de cette méthode consulter [13].

3 Principe général de l'approche de segmentation

En partant du principe que les deux primitives régions et contours sont complémentaires et qu'une coopération ou une coordination entre ces 2 approches peut combler les lacunes dont souffrent les méthodes de segmentation classiques, et en considérant que notre approche doit être adaptative au contenu de l'image, on a constater que notre système de segmentation en se basant sur la structure hiérarchique imposée par la méthode CSC va nous permette de :

- D'avoir des zones de focalisation (ou de traitement) : la segmentation ne va pas procéder dans toute l'image aveuglement, mais l'image va être découpée en plusieurs petites zones (les ilots dans chaque niveau), et le processus de segmentation consiste à créer des sous processus de segmentation dans chaque zone.
- D'avoir une coopération mutuelle entre les contours et les régions dans chaque zone : dans notre système les contours évoluent de niveau en niveau en utilisant l'information région dans chaque ilot et l'information contours des ilots des niveaux inferieurs, mais les régions seront construites indépendamment par la méthode CSC, et à la fin une phase de raccordement des régions sur les contours finales sera lancée.
- D'affiner la qualité de segmentation : puisque notre approche est hiérarchique, alors à chaque fois qu'on monte vers un niveau supérieur on aura une vision plus globale (cela est dû au système des ilots hiérarchiques du CSC), et une coopération avec les contours et les régions trouvés dans les sous zones de la zone concernée avec les contours et les régions de cette zone englobante peut améliorer la segmentation comme on va détailler par la suite (fermeture de contours, aménagement des régions, changement de seuil,…etc.).
- De pouvoir exécuter le processus de segmentation d'une manière parallèle et distribuée, car une image peut être découpée en plusieurs parties où chacune peut être affectée a un sous processus de segmentation, et a la fin on groupe le résultat (afin de pouvoir obtenir l'ilot globale) pour obtenir l'image segmentée finale. Cette vision peut être implémentée dans les systèmes ayant une architecture parallèle pour bénéficier de l'accélération du traitement.

Notre approche repose sur 4 concepts, on va les détailler dans ce qui suit :

3.1 Seuil adaptatif des contours

Pour détecter un contour nous avons utilisé le filtre de Deriche qui offre bonne précision et détection. Cependant et comme on l'a cité plus haut, la détection d'un contour se fait dans un ilot précis de niveau n. Alors le seuil choisi pour ce filtre afin de retenir les points de contours dépend de l'état de l'ilot dans lequel ce contour appartient [11].
L'état d'un ilot pour nous est désigné par le niveau d'homogénéité des régions contenues dans cet ilot.
Si un contour se retrouve dans un ilot contenant des régions très homogènes (couleurs proche), alors le contour peut être négligé afin de fusionner ces régions dans le futur (dans l'ilot de niveau supérieur), donc pour ignorer le contour il faut augmenter le seuil.
Si un contour se retrouve dans un ilot contenant des régions très hétérogènes, alors le contour doit être renforcé afin de bien distinguer les frontières des régions dans le futur, donc pour renforcer le contour il faut diminuer le seuil.

On aura donc une relation linéaire entre le seuil des contours et le niveau d'homogénéité des régions, et qui peut être donnée par la formule :

$$Seuil = \propto + \beta . homogénéité \qquad (1)$$

Avec :

α : une variable qui control le nombre de contours (nous l'avons choisie égale à la moyenne locale des valeurs de la norme du gradient).

β : variable qui contrôle le niveau de prise en compte de l'homogénéité, dans les niveaux inferieurs β sera plus petit que dans les niveaux supérieurs (l'homogénéité de gros régions est plus significative que l'homogénéité de petites régions). Nous l'avons choisie égale au nombre de niveau concerné.

Pour calculer l'homogénéité des régions dans un ilot on a choisie le calcul de la variance des couleurs moyennes des régions (code-éléments) où :

$$Homogénéité = \begin{Bmatrix} \frac{1}{Variance} & si\ Variance \neq 0 \\ Val_{Max} & si\ Variance = 0 \end{Bmatrix} \qquad (\mathbf{2})$$

Avec: $\boldsymbol{Val_{Max}}$: une constante définie par l'application qui définie l'homogénéité maximale.

Avec un seuil dynamique, la détection des contours sera adaptative selon le contenue de l'image et des contours dans des zones a contraste faible peuvent être détectés.

3.2 La coopération contour-contour

Initialement les contours initiaux (se situant sur des ilots de niveau 0) seront construits comme on a expliqué en choisissant un seuil spécifié, ensuite et au niveau suivant on aura un ilot englobant de niveau 1 contenant 7 ilot de niveau 0. Dans cet ilot on va recommencer le même processus (designer un nouveau seuil et créer une nouvelle carte de contours).
Donc on aura une carte de contour de niveau 1 nouvellement créé, et 7 sous-cartes de contours résultant des 7 sous ilots de niveau 0.
Une coopération contour-contour se fasse entre les différentes cartes de contours des deux niveaux afin de mieux suivre et fermer les contours comme illustre la figure 6.
On signale que les cartes de contours de différents niveaux ne seront pas nécessairement les mêmes vu que les seuils utilisés pour les calculer diffèrent d'un niveau a l'autre car l'homogénéité change assurément.

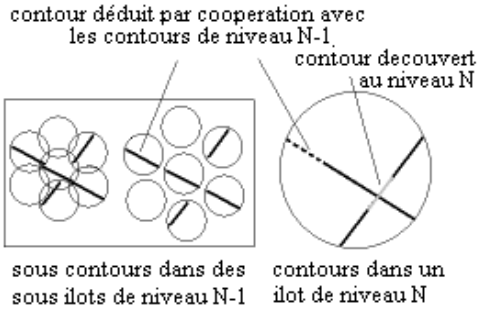


Figure 6 : coopération des contours

Ce processus de négociation entre les contours se poursuit de niveau en niveau jusqu'au dernier où on obtient la carte de contour finale.

3.3 La coopération région-contour (les contours utilisent les régions)

Deux cas se présentent :

Si un contour est inclut dans une région (c'est à dire tous les points du contour sont dans la zone de la région) et que ce contour n'a pas évolué (changer de taille) depuis au moins 2 niveaux de la hiérarchie malgré que la région englobante a amplifié, alors ce contour sera supprimé car il s'agit d'un faux contour ou d'un contour négligeable.

Si un contour se situe entre 2 régions distinctes, alors en coopérant avec les deux régions avoisinantes il peut se complété en suivant les frontières des deux régions.

On signale que cette coopération ne se produit que dans les niveaux supérieurs de la hiérarchie où les régions seront assez grandes, car si on procède par compléter les contours dans les niveaux inferieurs on aura plus de faux contours et la performance de la coopération contours-contours sera diminuée

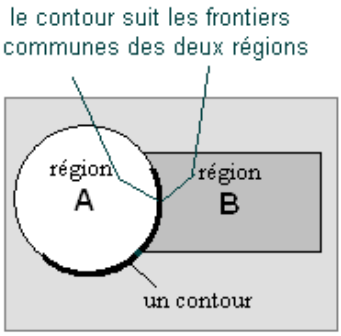


Figure 7 : coopération contours-régions

3.4 Correction des régions

La segmentation par CSC donne parfois des objets ayant des contours qui ne collent pas bien avec les frontières exactes des objets (elle fusionne des parties des objets avoisinantes), mais avec une coopération contour on peut corriger cela on coupant les régions pour leur donner un contour plus naturel.
Ce processus de découpage produit de nouveaux régions sur les frontières des objets qui doivent être fusionnés avec les régions avoisinantes de l'autre coté comme illustre la figure 8.
Donc pour avoir une bonne segmentation on doit fusionner ces petites régions avec leurs régions correspondantes après leurs découpages (une coopération région-région).
On signale que la phase de correction des régions ne s'applique qu'à la fin du processus de segmentation quand toutes les régions et les contours seront définis car le découpage d'une région de niveau inferieur est très couteux en temps de calcule et ça nous donne rien de meilleur de diviser une centaine de sous régions vu que le résultat sera le même.

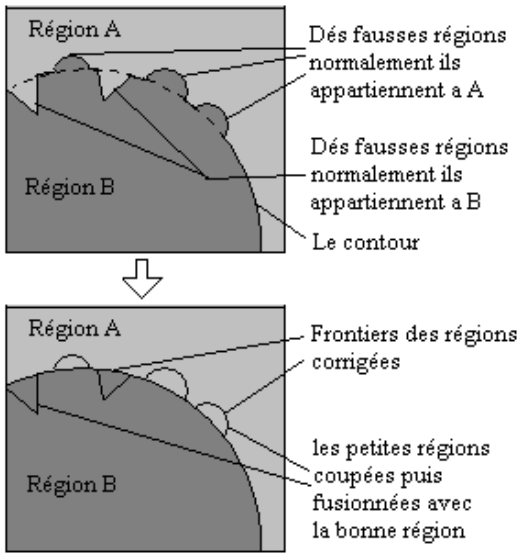


Figure 8 : correction des régions

4 La plate forme d'agents

La plateforme d'agents proposée est constituée de 4 types d'agent: l'agent région, l'agent contours, l'agent coordinateur et l'agent contrôleur (figure 10).

1. ***L'agent région :*** représente une région (chaque agent région pour chaque racine d'un arbre de code-élément), il gère les informations de sa région (sa taille, son homogénéité, sa forme,…etc.), ainsi que la coopération avec les contours via l'agent coordinateur.
2. ***L'agent contours :*** représente une carte de contours dans un ilot définie, il gère la communication et la coopération avec les autres cartes de contours des niveaux supérieurs et inferieurs ainsi que la coopération avec les régions.
3. ***L'agent coordinateur :*** coordonne et arrange les transactions entre les agents contours et régions. Il fait les calcules concernant le seuil des contours et l'homogénéité des régions. Pour chaque ilot on aura un seul agent coordinateur.
4. ***L'agent contrôleur :*** qui vérifier progressivement le résultat et la qualité du processus de segmentation (si une incohérence se produit, il l'a détecte et tente de la corriger en communicant avec les agents contours et régions). On aura un seul agent contrôleur par niveau (figure 9).

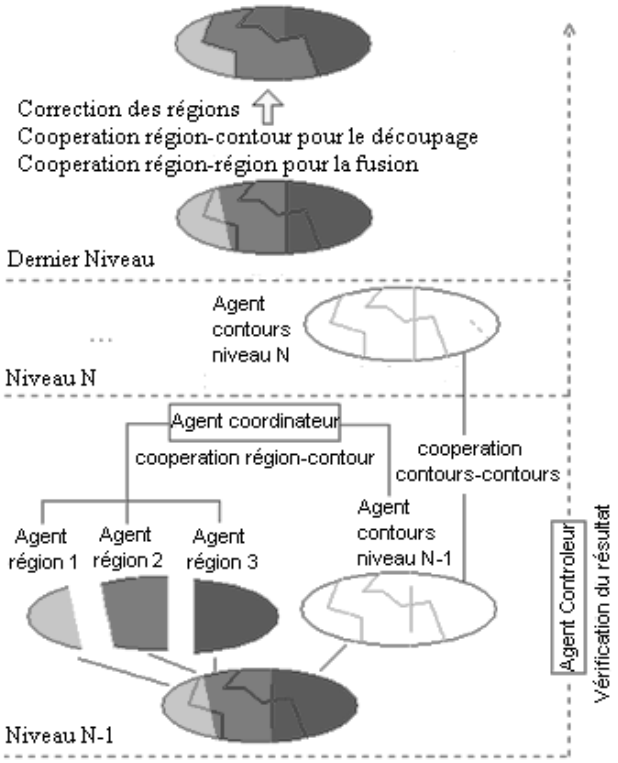


Figure 9 : Principe du système de segmentation à travers l'évolution d'un ilot

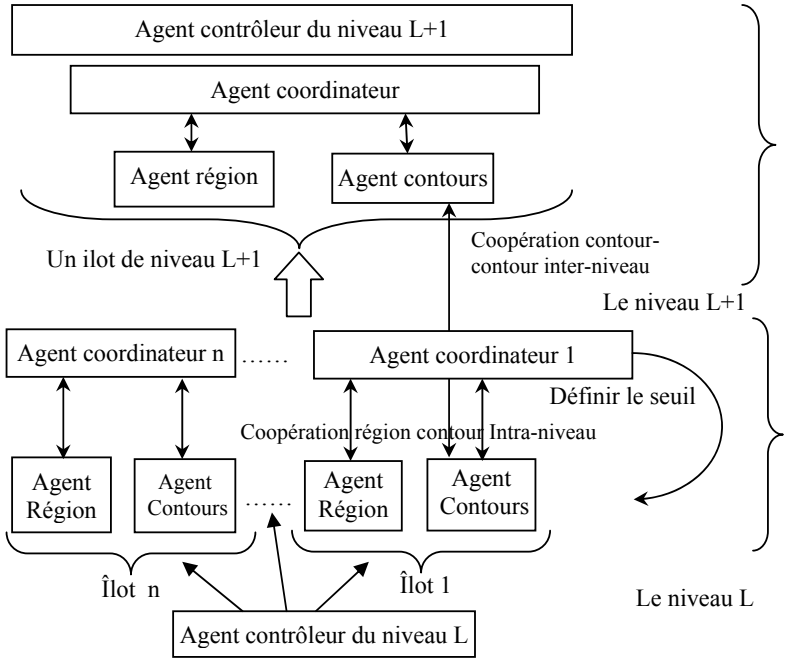


Figure 10 : les interactions entre les agents

5 Implémentation

Nous avons implémenté en JAVA la méthode CSC ainsi que l'algorithme de détection de contours de Deriche.
On a utilisé JADE de FIPA pour implémenter notre SMA, mais nous n'avons pas encore réalisé toutes les types de coopérations cités.
On a comparé notre système incomplet (prototype) avec les méthodes de segmentation par histogramme, et par split & merge, et il était clair que le résultat de segmentation de notre système est bien meilleur visuellement, et plus rapide que l'algorithme split & merge.
Donc, en attendant l'achèvement de notre SMA, l'étape de comparaison et d'estimation de la qualité de l'approche est à prévoir dans un futur proche.

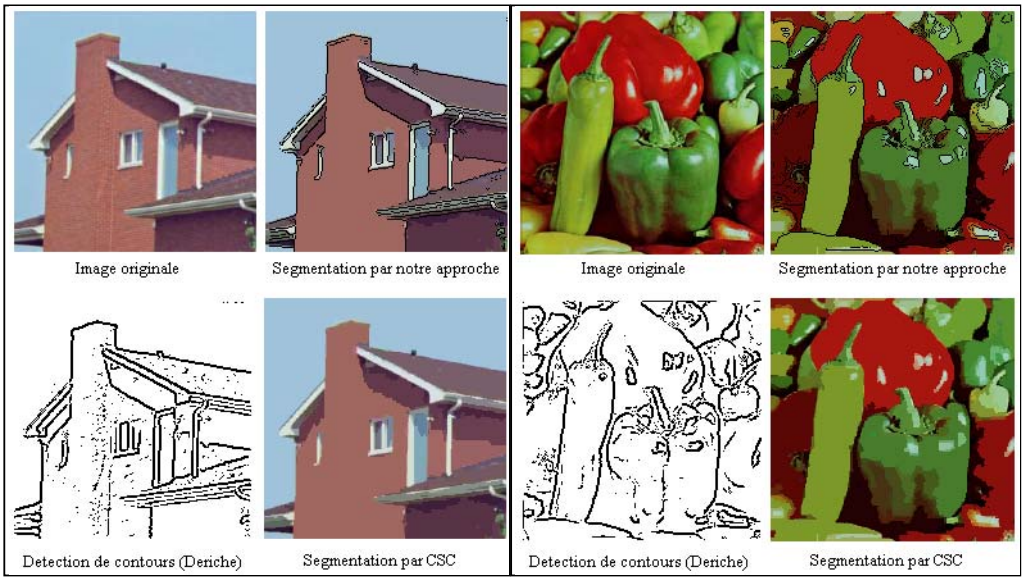


Figure 11 : Segmentation par notre système (incomplet).

6 CONCLUSION

L'architecture basée agents présentée offre une flexibilité et une adaptabilité supérieure à la plupart des méthodes de segmentation classique, elle exploite le maximum d'information en combinant les deux approches région et contour profitant ainsi des avantages de chacune d'elles, et donne une vision locale et globale appuyée par un environnement hiérarchique et coopératif, cette manière de faire comble les lacunes de ces deux approches.

References

1. T. Acharya, Ajoy K. Ray, "Image Processing, Principles and Applications", ouvrage, "A Wiley-Interscience Publication", chapitre 7, 2005.
2. Z. AI Aghbari, R. AI-Haj, "Hill-manipulation: An effective algorithm for color image segmentation", Département d'informatique, université de Sharjah, Emirates, 2006.
3. J.C Baillie, , "Segmentation", Cours Traitement d'Image et Vision Artificielle, 2003.
4. A. Chehikian, "Image segmentation by contours and regions cooperation", Laboratoire des Images et des Signaux, Institut National Polytechnique de Grenoble et l'université Joseph Fourier Grenoble, INPG, 38031 Grenoble Cedex, France, 1999.
5. H.D. Cheng, X.H. Jiang, Y.Sun, Jingli Wang, "Color image segmentation: advances and prospects", Département d'informatique, Utah state university, USA, 2000.
6. Gy. Dorko, D. Paulus, U. Ahlrichs. "Color segmentation for scene exploration". Université d'Erlangen-Nurenberg, Institut d'informatique, 2000.
7. J. Lecoeur, C. Barillot, "Segmentation d'images cérébrales : État de l'art", Théme BIO — Systèmes biologiques, Projet VisAGeS Rapport de recherche n°6306 — Juillet 2007.
8. Z. Mansouri, F. H. Merouani, 'Un modèle d'interaction multi-agents région-contour pour la segmentation d'images', Conférence JSIA 3-4 mars 2009-Guelma-Algérie.
9. [Moghaddamzadeh, 1996] A. Moghaddamzadeh, N. Bourbakis, "A Fuzzy Region Growing Approch For Segmentation Of Color Images", Departement de EE, AAAI Lab, université de Binghamton, Binghamton, USA, 1996.
10. O. Monga, "Segmentation d'images : ou en somme nous ?" Support de cours pour le congres PIXIM 89, 1990.
11. E. Navon, O. Miller, A. Averbuch, "Color image segmentation based on adaptative local thresholds", Département d'informatique, université de Tel-Aviv 69978, Israel, 2004.
12. L. Priese, V. Rehrmann, "A Fast Hybride Color Segmentation Method", Institut d'informatique, Université de Koblenz-Landau, Rheinau 1, D-56075 Koblenz, Allemagne, 1993.
13. L. Priese, V. Rehrmann. "Introduction to the Color Structure Code and its Implementation", 2003.
14. V. Rehrmann, L.Priese. "Fast and Robust Segmentation of Natural Color Scenes", Image Recognition Lab, Université de Koblenz-Landau. Rheinau 1, 56075 Koblenz. Allemagne, 1997.
15. A. Tremeau, N. Borel, "A region growing and merging algorithm to color segmentation", Institut d'ingénierie de la vision, 42007 Saint-Etienne, France, 1996.
16. D. Zugai, V.Lattuati, "A new approach of color images segmentation based on fusing region and edge segmentation outputs", Laboratoire d'automatique des arts et métiers/LAAM, F-75013 Paris, France, 1997.

Construct & Reduce : une heuristique pour calculer l'hypertree decomposition

Ait Amokhtar AbdelMalek[1] and Amroun Kamel[1] and Habbas Zineb[2]

[1] Université de Béjaïa, Ecole doctorale en informatique, Algérie,
malek.aitamokhtar@gmail.com, k_amroun25@yahoo.fr
[2] Université de Metz, Laboratoire d'Informatique Théorique et Appliquée, France,
zineb@univ-metz.fr

Résumé Plusieurs problèmes du monde réel peuvent être modélisés sous la forme d'un problème de satisfaction de contrainte (CSP). Même si la résolution d'un CSP est un problème NP-complet en général, certaines classes de CSP peuvent être résolues en un temps polynomial à l'aide d'algorithmes spécifiques. Une de ces classes est la classe des CSP acycliques. Dans ce papier nous proposons une nouvelle heuristique pour le calcul de l'hypertree decomposition, une méthode permettant de transformer n'importe quel CSP en un CSP acyclique équivalent en terme de solution. Cette nouvelle heuristique dénommée Construct & Reduce se compose de trois étapes : l'extraction des hypertrees decomposition, leurs concaténation et la réduction de la largeur, les deux premières étapes permettent de construire une hypertree decomposition alors que la dernière tente de réduire la largeur de la décomposition.

Mots clés : Problème de satisfaction de contraintes (CSP) , décomposition structurelle, hypertree decomposition. largeur de décomposition, heuristique.

1 Introduction

Le formalisme CSP (pour Constraint Satisfaction Problem) permet de représenter et de résoudre un grand nombre de problèmes réels ou académiques. L'approche classique de résolution des CSP est la recherche énumérative qui se base sur des algorithmes de la famille Backtrack. Ces algorithmes sont généralement très efficaces en pratique pour des problèmes de taille limitée. Néanmoins, leur complexité théorique étant exponentielle en la taille du problème à résoudre, ces algorithmes s'avèrent inefficaces pour résoudre des problèmes de grande taille.

L'approche de résolution par décomposition permet de borner cette complexité par un paramètre de mesure de cyclicité appelé largeur de la décomposition qui est indépendant de la taille de problème. Cette approche se base sur des méthodes dites de décomposition structurelle qui permettent de transformer n'importe quel CSP en un CSP acyclique équivalent en terme de solutions et pour lesquelles des algorithmes spécifiques de résolution polynomiale existent [1]. La méthode de décomposition structurelle la plus intéressante actuellement est l'hypertree décomposition car elle généralise la majeure partie des méthodes existantes et surtout sa définition permet le développement de nombreux algorithmes pour la construire.

Les premiers algorithmes de calcul de l'hypertree décomposition avaient généralement pour but de calculer pour chaque paramètre en entrée k, une hypertree decomposition de largeur inférieure ou égale à k si elle existe ou d'infirmer son existence. Ce type d'algorithmes dit exacts permettent de calculer directement ou indirectement une décomposition de largeur optimale mais ils restent utilisables en pratique uniquement pour des problèmes de petite taille et ce à cause de leurs coût important en terme de temps et d'espace mémoire. Ainsi, pour pouvoir décomposer des problèmes de grande taille, des méthodes heuristiques ont été proposées pour calculer des hypertree decomposition de largeur raisonnable en un temps réaliste mais sans garantie d'optimalité. Ces méthodes se basent généralement sur des heuristiques pour minimiser la largeur de décomposition en cours de construction de l'hypertree decomposition. Dans ce papier, Nous proposons une nouvelle méthode de calcul de l'hypertree décomposition appelée ***Construct & Reduce*** qui s'appuie sur des heuristiques pour réduire la largeur après une première construction de l'hypertree decomposition. L'expérimentation que nous avons réalisée révèle que cette façon de faire permet d'obtenir rapidement des décompositions de largeur très proche des meilleures largeurs obtenues par les autres méthodes heuristiques de la littérature.

Nous organisons le reste du papier de la façon suivante : La section 2 précise les notions de base concernant les CSP et leurs représentations graphiques. Dans la section 3, nous introduisons la notion d'hypertree decomposition. La section 4 est consacrée à la description de notre heuristique de calcul de l'hypertree decomposition, nommée ***Construct & Reduce***. Dans la section 5, nous présentons les résultats d'expérimentation de ***Construct & Reduce*** et nous la comparerons aux heuristiques ***BE*** et ***DBE***. Enfin, dans la section 7 nous concluons notre contribution par quelques remarques et des perspectives pour nos travaux futurs.

2 Préliminaires

La notion de problèmes de satisfaction de contraintes a été introduite par Montanari dans [2] selon la définition 1 suivante.

Definition 1. *Un Problème de Satisfaction de Contraintes est un quadruplet* $P = (X, D, C, R)$, *où* $X = \{X_1, ..., X_n\}$ *est un ensemble de n variables,* $D = \{D_1, ..., D_n\}$ *est un ensemble de n domaines finis. Chaque domaine* D_i *est associé à une variable* X_i, $C = \{C_1, ..., C_m\}$ *est un ensemble de m contraintes. Chaque contrainte* C_i *est définie par un ensemble de* n_i *variables* $\{X_{i_1}, ..., X_{i_{ni}}\} \in X$. $R = \{R_1, ..., R_m\}$ *est un ensemble de m relations. Chaque relation* R_i *définit l'ensemble des tuples sur* $D_{i_1} \times ... \times D_{i_{ni}}$ *autorisés par la contrainte* C_i.

Une solution d'un CSP est une affectation de valeur à chaque variable du problème qui ne viole aucune contrainte.

Definition 2. ***(Graphe et Hypergraphe [3])*** *Un hypergraphe est un couple H= (V,E) où V est l'ensemble des sommets et E est l'ensemble des hyperarêtes. Une hyperarête est un sous ensemble de V. Si toutes les hyperarêtes sont de dimension deux alors l'hypergraphe est réduit à un graphe simple.*

Remarque 1 *La structure d'un CSP peut être capturée par un hypergraphe tel que chaque variable du CSP sera représentée par un sommet et une hyperarête relie un ensemble de sommets si les variables correspondantes appartiennent à la même contrainte.*

Definition 3. ***(Graphe dual [3])***

Un hypergraphe H = (V, E) peut êêtre représenté par un graphe dual HDual défini comme suit : Les noeuds du graphe dual sont les hyperarêtes de H. Deux noeuds sont reliés dans le graphe dual si les hyperarêtes correspondantes partagent au moins un sommet.

Definition 4. ***(Connectivité, join-graphe et Join-tree [3])***

Un sous graphe d'arc d'un graphe est un graphe qui contient le même ensemble de sommets que le graphe d'origine et dont l'ensemble des arêtes est un sous ensemble de l'ensemble d'arêtes du graphe d'origine.

Un sous graphe d'arc d'un graphe dual d'un hypergraphe respecte la propriété de ***connectivité*** *si pour chaque couple de nœuds qui partagent un sommet il existe un chemin les reliant tel que les nœuds traversés contiennent tous le sommet partagé. Dans ce cas, le sous graphe d'arc est appelé un* ***join-graphe****. Un join-graph qui est un arbre est appelé un* ***join-tree****.*

Definition 5. ***(CSP acyclique et hypertree [3])*** *Un hypergraphe dont le graphe dual possède un join-tree est appelé un* ***hypertree.*** *Un CSP dont l'hypergraphe est une hypertree est dit* ***acyclique.***

3 Hypertree decomposition

L'hypertree decomposition est la méthode de décomposition structurelle la plus intéressante actuellement car d'une part, elle généralise [4] la majeure partie des méthodes existantes et d'autre part, sa définition permet le développement d'algorithmes efficaces pour la calculer.

3.1 Définitions de base

Definition 6. ***(Hypertree decomposition** [5])* *Une hypertree decomposition d'un hypergraphe H = (V,E), est un triplet HD = $< T, \chi, \lambda >$ tel que T = (V(T), E(T)) est un arbre, χ et λ sont des fonctions d'étiquetage définies comme suit :*

χ : V(T) $\rightarrow 2^V$ qui associe à chaque nœud t de T un ensemble de sommets $\chi(t) \subseteq V$

λ : V(T) $\rightarrow 2^E$ qui associe à chaque nœud t de T un ensemble d'arêtes $\lambda(t) \subseteq E$

Et qui satisfait les conditions suivantes :

1. $\forall h \in E, \exists p \in V(T)$ *tel que* $var(h) \subseteq \chi(p)$.
2. $\forall v \in V$ *, l'ensemble* $\{p \in V(T) | v \in \chi(p)\}$ *induit un sous arbre(connecté) de T.*
3. $\forall p \in V(T), \chi(p) \subseteq var(\lambda(p))$.
4. $\forall p \in V(T), [var(\lambda(p)) \cap \chi(T_p)] \subseteq \chi(p)$ *où* T_p *est le sous arbre de T enraciné en p.*

Si seule les 3 premières conditions sont vérifiées alors il s'agit d'une hypertree decomposition généralisée

Definition 7. *(Largeur)*
La largeur d'une hypertree decomposition < T, χ, λ > est $max_{(p \in V(T))}|\lambda(p)|$. La largeur d'un hypergraphe H (hw(H)) est la largeur minimum de toutes ses hypertrees decompositions.

Dans la suite de papier, nous utilisons l'abréviation HTD au lieu de hypertree decomposition.

3.2 Calcul d'une Hypertree decomposition

Un des atouts majeurs de l'HTD est l'existence d'algorithmes permettant de la calculer. Ceux-ci se décomposent en deux groupes : les algorithmes exacts et les algorithmes heuristiques.

Les algorithmes exacts calculent, pour un paramètre en entrée k donné, une HTD de largeur inférieure ou égale à k ou infirme son existence. Certains de ces algorithmes permettent de calculer directement une décomposition de largeur optimale tel que Opt-k-decomp [6] et Red-k-decomp [7] alors que d'autre doivent être utilisés de manière itérative pour calculer une décomposition optimale tel que Det-k-decomp [8].

Les algorithmes exacts étant très coûteux en temps et en espace, des méthodes heuristiques ont été proposées pour calculer des HTD de bonne largeur en un temps raisonnable pour des problèmes de grande taille mais sans garantie d'optimalité. Nous citons notamment la Bucket Elimination (BE [9]) et la Dual Bucket Elimination (DBE [10]). Celles-ci utilisent des heuristiques pour ordonner les sommets ou les hyperarêtes tels que les ordres Minimum Cardinality Search (MCS) et le Minimum Induced Width (MIW).

4 Construct & Reduce

Dans cette section, nous présentons une nouvelle méthode heuristique de calcul de l'HTD nommée Construct & Reduce. Notre méthode, après une construction rapide d'une HTD, tente de réduire la largeur de la décomposition.

La phase de construction se déroule en deux étapes. La première consiste à extraire à partir d'un hypergraphe de contrainte des HTD de profondeur 1 alors que la deuxième étape consiste à les fusionner afin d'obtenir l'HTD de l'hypergraphe d'origine. Enfin, lors de la dernière phase, on transforme successivement l'HTD afin de réduire la largeur. Dans la suite de cette section, nous allons détailler les étapes de notre méthode avant de présenter les résultats de notre expérimentation afin de montrer son intérêt pratique.

4.1 Extraction des hypertree decomposition

La première phase de notre démarche consiste à extraire les sous ensembles d'arêtes qui forment des hypertree decomposition (de largeur et de profondeur 1) composées uniquement d'une racine et de ses fils. L'ensemble λ de chaque nœud contiendra exactement une arête et l'ensemble χ se composera des variables de l'arête appartenant à λ. Après le choix d'une racine, les fils seront choisis de manière à ne pas violer la 2éme condition de l'hypertree décomposition ceci en s'assurant que deux fils d'un même père ne partagent jamais de variables (de l'ensemble χ) qui ne soient pas contenus dans l'ensemble χ de leur père (Voir condition 3 du théoreme 1). L'étape d'extraction est décrite formellement par l'algorithme1 où :

Hypertree Nouvelle_Hypertree() : crée une hypertree vide

Noeud Nouveau_Noeud(**Hypertree** HT) : crée un nouveau noud pour l'hypertree HT

Arête choisir (**Ensemble** E, **Arête** e) : Retourne une arête e' $\in$ E tel que var(e) $\cap$ var(e') $\neq \emptyset$ et var(e) $\cap$ Nogoods = $\emptyset$

Arête choisir (**Ensemble** E) : Retourne une arête e' $\in$ E

Algorithm 1 Extraction des hypertrees decomposition

1: **Entrées** : H = (V, E) : hypergraphe de CSP original
2: **Sortie** : {HTi = (Ti, χi, λi) / i = 1 à k} ensemble de k hypertrees decomposition
3: i = 0
4: **while** $E \neq \emptyset$ **do**
5: i = i + 1
6: Racine = **choisir**(E) ; N = **Nouveau_Noeud**(HTi)
7: HTi = **Nouvelle_Hypertree()**
8: λi(N) = Racine ; χi(N) = var(Racine)
9: Nogoods = $\emptyset$
10: **Décompose** (Racine, HTi)
11: **end while**

Algorithm 2 Décompose (**Arête** Racine, **Hypertree** HT)

1: E = E – Racine
2: **while** $\exists$ e $\in$ E / (var(e) $\cap$ var(Racine)) $\neq \emptyset$ et (var(e) $\cap$ Nogoods) = $\emptyset$ **do**
3: Courrant = **Choisir**(E, Racine) ; N = **Nouveau_Noeud**(HT)
4: λ (N) = courant ; χ (N) = var(Courrant)
5: Nogoods = Nogoods $\cup$ var(Courrant) - var(racine)
6: E = E – Courrant
7: **end while**

Remarque 2 *la première fonction choisir sert à sélectionner les fils de la racine d'une hypertree. L'heuristique que nous préconisons consiste à choisir l'arête qui partage le plus de variables avec la racine pour avoir des HTDs fortement liés.*

La deuxième fonction choisir sert à choisir la racine d'une hypertree. La nouvelle racine choisie sera celle qui aura le moins de variables en commun avec les racines des hypertree précédents. Ceci pour avoir des HTDs plus indépendants et faciliter la 2éme étape de notre démarche comme nous le verrons ultérieurement. L'ensemble Nogoods contient les sommets des nœuds fils déjà choisis qui ne sont pas couverts par la racine, il est utilisé pour choisir des arêtes qui ne violent pas la condition 3 de théorème 1.

Exemple 1 *soit le CSP P = <X,D,C > suivant :*
$X = \{x_1, \ldots, x_8\}$ $C = \{c_1, \ldots, c_5\}$ *tel que*
$c_1 = \{x_1, x_2, x_3\}$, $c_2 = \{x_2, x_4, x_5\}$, $c_3 = \{x_4, x_5, x_8\}$, $c_4 = \{x_1, x_6, x_7\}$, $c_5 = \{x_6, x_7, x_8\}$

L'extraction des hypertree donne le résultat illustré dans la figure 1.

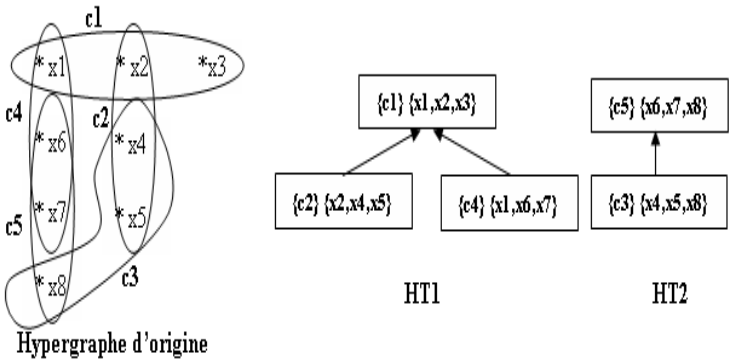


FIG. 1. L'extraction des hypertree decomposition

4.2 Contraction des hypertree decomposition

Apres avoir extrait les HTDs {HTi}, celles-ci sont contractées afin d'obtenir l'HTD HT de CSP. La fusion est composée des deux étapes principales suivantes qui se basent sur le théorème 1.

1. Création de la racine : la racine est l'union des racines des hypertrees extraites par l'étape d'extraction.
2. Rajout des autres nœuds : les autres nœuds sont rajoutés comme nœuds fils un à un. S i des nœuds ne respectent pas la condition 3 du théorème 1, on les fusionne

Théorème 1 *Soit un hypergraphe H=< V, E >. Un triplet HT = < T, χ, λ > tel que T=<V(T),E(T)> est de profondeur 1 et qui vérifie les conditions suivantes :*

1. $\forall h \in E, \exists p \in V(T)$ *tel que* $var(h) \subseteq \chi(p)$.
2. $\forall p \in V(T), \chi(p) = var(\lambda(p))$.
3. $\forall p \in V(T) : [\chi(p) - \chi(père(p))] \cap [\cup \chi(f) / f \in frère(p)] = \emptyset$

Alors HT est une hypertree decomposition

Preuve 1 *La condition 1 et la même que la première condition de l'HTD*

La condition 2 implique trivialement la condition 3 de l'HTD

La condition 2 du théorème 1 implique que $\forall p \in V(T), \chi(p) = var(\lambda(p))$

De plus, $\forall p \in V(T) : (var(\lambda(p)) \cap \chi(T_p)) \subseteq var(\lambda(p))$ *est trivialement vraie. Donc* $\forall p \in V(T) : (var(\lambda(p)) \cap \chi(T_p)) \subseteq \chi(p)$ *. Ce qui implique que la condition 4 de l'HTD est vérifiée.*

La condition 2 de l'HTD est aussi vérifiée car le seul cas invalide quand T est de profondeur 1 consiste à avoir des nœuds feuilles partageant des sommets non couvert par la racine ce qui est contraire à la condition 3 de théorème 1.

L'étape de contraction est décrite formellement par l'algorithme3 qui se base sur le théorème1.

Algorithm 3 Contraction des hypertrees decomposition

Entrées : {HTi} = {<Ti, χi, λi> / i = 1 à k} : ensemble d'hypertrees decomposition
Sortie : HT = <T, χ, λ > hypertrees decomposition de CSP
HT = **Nouvelle_Hypertree()**
racine = **Nouveau_Noeud**(HT)
for i =1 to k **do**
  λ(racine) = λ(racine) $\cup$ λi(racine(HTi))
  χ (racine) = χ (racine) $\cup$ χi(racine(HTi))
end for
for i =1 to k **do**
  for all N tel que N $\in$ V(Ti) **do**
    N' = **Nouveau_Noeud**(HT)
    λ(N') = λ(N)
    χ (N') = χ (N)
    for all N" $\in$ V(T) tel que N" $\neq$ N' **do**
      if χ (N") $\cap$ (χ (N') - χ (racine)) $\neq \emptyset$ **then**
        χ (N') = (χ (N') $\cup$ χ (N"))
        λ (N') = (λ (N') $\cup$ λ (N"))
        Supprimer(N")
      end if
    end for
  end for
end for

Exemple 2 *La fusion des hypertrees de l'exemple précédent donne le résultat illustré dans la figure suivante (Figure.2).*

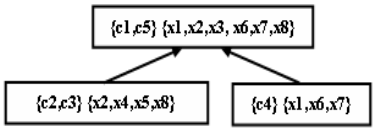


FIG. 2. Fusion des hypertree decomposition

4.3 Réduction de la largeur

C'est l'étape la plus coûteuse en temps CPU de notre démarche. Elle s'appuie sur le théorème 2 qui permet de réduire la largeur des nœuds en respectant les conditions de l'HTD. Le principe de ce théorème est de retirer une arête de l'ensemble λ d'un nœud si celle-ci permet de créer un nœud dont la largeur est inférieure à celle du nœud réduit multiplié par un certain facteur en privilégiant les arêtes qui induisent la création d'un nœud de plus petite largeur. Ce processus est répété plusieurs fois jusqu'à ce que plus aucun nœud ne soit réduit.

Théorème 2 *Soit une hypertree decomposition, HT = <T=(V(T),E(T)), χ, λ > d'un hypergraphe H=(V,E) tel que :*

1. $\forall p \in V(T) : \chi(p) = var(\lambda(p))$
2. *Et soit* $N \in V(T)$ *et* $h \in \lambda(N)$ *tel que :* $[var(h) - var(\lambda(N) - \{h\})] \cap \chi(père(N)) = \emptyset$
3. *soit* $F \subseteq Fils(N)$ *tel que :* $\forall p \in F, [\chi(p) - var(\lambda(N) - \{h\})] \cap var(h) \neq \emptyset$

soit N' le nouveau nœud qui sera créé à partir de h et le triplet HT' = < T'=(V(T),E(T)),χ', λ'> tel que :

4. $V(T') = V(T) \cup \{N'\} - F$
5. $E(T') = E(T) \cup \{(N,N')\} \cup \{(N',p') / (p,p') \in E(T) \text{ et } p \in F\} - \{(p,p') \in E(T) \text{ et } p \in F\} - \{(N,p) \in E(T) \text{ et } p \in F\}$
6. $\forall p \in \{V(T') - \{N'\}\} : \lambda'(p) = \lambda(p)$
7. $\lambda'(N') = h \cup \{\cup\lambda(p) / p \in F\}$
8. $\forall p \in V(T') : \chi'(p) = var(\lambda(p))$

Alors HT' est une HTD de H

Preuve 2 *La condition 8 implique trivialement la condition 3 de l'HTD*

La condition 8 implique la condition 4 de l'HTD(voir preuve théorème1)

Les conditions 4, 6, 7 et 8 impliquent que chaque arête de H est couverte => condition 1 de l'HTD.

Pour la condition 2 de l'HTD :

HT est une HTD donc par définition, aucun nœud de T ne partage un sommet avec h qui ne soit couvert par le père ou un fils de N (condition 2 de l'HTD) donc pour que l'ajout de nouveau fils N' de N dans HT', qui est créé à partir de h, garde la condition 2 vrai il suffit que

1. *N' ne partage pas de variable avec le père de N qui ne soit couverte par p ce qui est impliqué par la condition 2 de théorème*
2. *N' ne partage pas de variable avec les autres fils de N ce qui est évité en fusionnant les fils de N qui ne vérifie pas cette condition (Ensemble F) ceci est induit par les conditions 3, 4, 5, et 7 de théorème.*

La réduction de la largeur se base sur le théorème 2 et est décrite formellement dans l'algorithme suivant :

Algorithm 4 Réduction de la largeur de l'hypertree

Entrées : HT = (T, χ, λ) : hypertree decomposition
Sortie : HT = (T, χ, λ) : hypertree decomposition
largeur = 0 ; largeur_finale = m
Continuer = vrai
while Continuer = vrai **do**
  Continuer = faux
  for all N ∈ T **do**
    for i = 0 to |λ (N)| * facteur **do**
      min_nb = 0 ;
      for all e ∈ λ (N) tel que [var(e) ∩ [χ(Pere(N)) – var(λ (N) - e)]] = ∅ **do**
        E' = {F∈ Fils(N)/ χ(F) ∩ [var(e) – var(λ (N) -e)] ≠ ∅}
        if $\sum_{N' \in E'}$(|λ (N')|) ≤ i **then**
          Continuer = vrai
          min_nb = min ($\sum_{N' \in E'}$(|λ(N')|) , min_nb)
          N" = **Nouveau_Noeud**(HT)
          χ (N") = var(e) ∪ χ (E')
          λ (N") = {e} ∪ λ (E')
          Pere(N") = N
          for all N' ∈ E' **do**
            Supprimer(N')
          end for
        end if
        i = min_nb – 1
      end for
    end for
  end for
end while

Remarque 3 *Le facteur utilisé permet de faire descendre les nœuds de plus grande largeur plus en profondeur dans l'arbre et permettre ainsi de minimiser la largeur globale de la décomposition. Dans notre expérimentation, nous l'avons fixé à 1,2.*

Exemple 3 *La réduction de la largeur de l'exemple précédent donne le résultat illustré dans la figure 3*

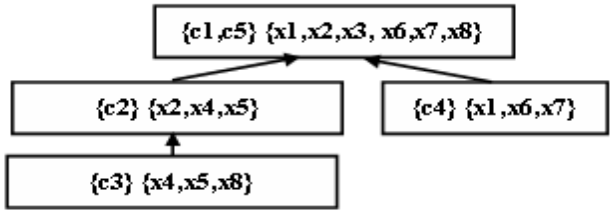


FIG. 3. Reduction de la largeur de l'hypertree decomposition

5 Résultats expérimentaux

Pour démontrer l'intérêt pratique de notre heuristique, nous avons implémenté et expérimenté notre approche sur une série de benchmarks de la littérature. Nous avons effectué une étude comparative entre notre méthode et les méthodes Bucket Elimination (BE) [9] et Dual Bucket Elimination (DBE) [10] qui font partie des méthodes les plus performantes de la littérature. Ces méthodes se basent sur des heuristiques d'ordonnancement de variables et nous avons considéré les ordres *minimum cardinality search* (MCS) et *minimum induced width* (MIW) pour leur large utilisation et leurs performances.

Les problèmes testés sont extraits des benchmarks proposés par l'équipe DBAI [11] de l'université de Vienne, et de ceux proposés par Lecoutre dans le cadre d'une compétition de solveurs de CSP[1].

Les tests sur les méthodes BE et DBE ont été effectués à l'aide de l'outil proposé par les auteurs[2]. Comme ces méthodes sont aléatoires, nous avons effectué cinq tests de suite pour chaque exemple et c'est la moyenne des résultats qui est prise en compte pour chaque CSP. Les résultats des tests sont résumés dans le tableau suivant.

[1] http ://www.cril.univ-artois.fr/ lecoutre/research/benchmarks/benchmarks.html.
[2] http ://www.dbai.tuwien.ac.at/proj/hypertree/downloads.html.

CSP			BE(MCS)		DBE(MCS)		BE(MIW)		DBE(MIW)		Construct & Reduce	
Nom	**|V|**	**|E|**	**W**	**T**	**W**	**T**	**W**	**T**	**W**	**T**	**W**	**T**
Nasa	579	680	79	18	69.6	138	28.2	10	57.2	160	44	13
Reines	20	190	10	0	16.8	11	10	0	10.8	25	10	0
Renault	101	113	2.4	1	8	3	3	0	7.6	6	5	0
Schure	19	81	7	0	9.2	2	7	0	8.2	4	7	0
Fischer1	308	284	43.6	3	36.4	17	19.6	1	30.4	8	28	2
Fischer6	1523	1419	188.8	177	210.6	947	75.2	33	118	154	115	224
aim50	50	80	10.8	0	12.2	1	10.2	0	13.6	1	11	0
Aim100	100	160	20.8	1	22.4	6	21.8	1	28.8	6	24	0
grid5	25	40	4.4	0	5.4	0	5	0	6.2	0	6	0
Ii8a1	66	186	9	0	9	5	9.4	0	9	5	12	0
Par8-1-c	64	254	12.2	0	12	1	7	0	14.4	0	9	0
Par8-2-c	68	270	14.2	0	12.6	1	7	0	14.8	0	10	0
complex	414	849	22.8	2	15.4	11	10	2	12	10	17	12
uf20-050	20	91	6.8	0	9.4	2	6.4	0	8.4	4	8	0
uf75-099	75	325	22.4	1	32.6	42	20	2	31	139	24	0

TAB. 1. Comparaison Construct & Reduce, BE, DBE

Remarque 4 *|V| et |E| désignent respectivement le nombre de variables et le nombre de contraintes ; W désigne la largeur de la décomposition et T le temps pris par la décomposition en secondes.*

La première remarque qu'on peut tirer du tableau est que notre méthode obtient de meilleurs résultats que ceux des deux versions de la DBE pour une bonne partie des problèmes testés que se soit pour la largeur ou pour le temps de la décomposition. Concernant la BE, nous observons des résultats équivalents avec ceux de la version MCS, et même si la version MIW de la BE reste la meilleure méthode existante, notre méthode obtient des résultats très proches de ceux donnés par celle-ci.

6 Complexité de la méthode

Un des avantages des méthodes heuristiques et leurs faibles coûts comparativement aux méthodes exactes. De ce faite, nous avons particulièrement porté notre attention sur le coût de notre méthode notamment aux travers de sa complexité théorique polynomiale comme le résume les résultats suivant :

1. La complexité spatiale des algorithmes est en $O(m^2 + mn)$ ce qui correspond à la taille maximale de l'hypertree decomposition maintenu.
2. La complexité temporelle de l'extraction est en $O(m^4 + m^3n^2)$ en prenant en compte le coût des heuristiques sur le choix des arêtes, la complexité de la fusion est en $O(m^4 + m^2n^2)$ et enfin, La complexité de la réduction est en $O(m^6 + m^5n^2)$.

7 Conclusion

Dans ce papier nous avons présenté une nouvelle heuristique Construct & Reduce pour construire l'HTD. Nous l'avons comparé à quelque unes des méthodes les plus performantes et les plus citées de la littérature. Les résultats de la comparaison ont montré l'intérêt pratique de notre méthode notamment face à *la Dual Bucket Elimination* (DBE) et à la version MCS de la B*ucket Elimination* (BE). Comme perspective, nous envisageons de combiner notre algorithme de réduction de la largeur avec d'autre méthodes de même nature que la notre (de type query decompopsition) puis de l'adapter afin d'être utilisé par des méthodes plus générales teles que la BE et la DBE.

Références

1. Freuder, E.C. : A sufficient condition for backtrack-bounded search. Journal of the Association for Computing Machinery **32** (1985) 755–761
2. Montanari, U. : Networks of constraints : Fundamental properties and applications to pictures processing. Information Sciences **7** (1974) 95–132
3. Dechter, R. : Constraint Processing. Morgan Kaufmann (2003)
4. Gottlob, G., Leone, N., Scarcello, F. : A comparison of structural csp decomposition methods. Artificial Intelligence **124** (2000) 243–282
5. Gottlob, G., Crohe, M., Musliu, N. : Hypertree decomposition : structure, algorithms and applications. In : Proceeding of 31 st International workshop WG, Metz (2005)
6. Gottlob, G., Leone, N., Scarcello, F. : A comparison of structural csp decomposition methods. In : Proceedings of IJCAI'99. (1999) 394–399
7. Harvey, P., Ghose, A. : Reducing redundancy in the hypertree decomposition scheme. In : Proceding of ICTAI'03, Montreal (2003) 548–555
8. Gottlob, G., Samer, M. : A backtraching based algorithm for computing hypertree decompositions. arXivics.DS 0701083vl 14 Jan 2007 (2007)
9. McMahan, B. : bucket elimination and hypertree decompositions. Technical report, Implementation report, institute of information systems (DBAI), TU, Vienna (2003)
10. Dermaku, A., Ganzow, T., Gottlob, G., McMahan, B., Musliu, N., Samer, M. : Heuristic methods for hypertree decompositions. Technical report, DBAI-R (2005)
11. Ganzow, T., Gottlob, G .and Musliu, N., Samer, M. : A csp hypergraph library. Technical report, DBAI-TR-2005-50, Technische Universität Wien. (2005)

A Study of (0,λ)-Graph Type

Abdelhafid BERRACHEDI and Nawel KAHOUL

Faculté de Mathématiques, USTHB,
BP 32 El-Alia, Bab-Ezzouar 16111, Alger, Algérie
aberrachedi@usthb.dz,
kahoul.nawel@yahoo.fr

Abstract. *I.Havel* gave a conjecture which asserts that there is a Hamilton cycle for the graph induced by the middle levels of any hypercube of an odd degree. This graph verify the property that each path of length three belongs to a single cycle of length six. Graphs verifying this property are denoted [3,1,6]-cycle regular graphs, they are studying in this paper. We are interested in giving some new characterizations for these graphs. We present also a class of graphs which generalizes them and give new characterizations for these graphs.

Key Words : Hypercubes, Odd graphs, semi-regular graphs, cycle regular graphs, (0,λ)-graphs

1 Introduction

Unless specified otherwise, all graphs in this paper are finite, simple, undirected and connected. A graph G will have a vertex set $V(G)$ and an edge set $E(G)$. In the sequel, we always write V (resp. E) instead of $V(G)$ (resp. $E(G)$), except in the case where two or more graphs are considered. Thus, we simply write $G = (V, E)$. The order of G is the number of its vertices. The graph on n pairwise adjacent vertices is denoted by K_n. The *neighborhood* of a vertex $u \in V$ will be denoted by $N(u)$. The *degree* of a vertex u of G and the *minimum degree* of vertices of G will be respectively denoted by $d(u)$ and $\delta(G)$. A bipartite graph is *semiregular* if the vertices in the same part of the bipartition have the same degree.

In this paper, an *elementary path* $P_{\mu+1}$ (also called a (u_0, u_μ)-*path*) of length μ (in G) is a sequence $u_0, \cdots, u_\mu$ of pairwise distinct vertices except possibly u_0 and u_μ, where $u_iu_{i+1} \in E$ for $i = 0, \cdots, \mu-1$. An *elementary cycle* of length μ (in G) is a (u_0, u_μ)-path with $u_0 = u_\mu$ and is called a μ-*cycle*. Both a (u_0, u_μ)-path and a μ-cycle are *induced* if any *two* non-consecutive vertices are not adjacent. The *girth* of a graph G is the length of the shortest cycle in G. A (u_0, u_μ)-path *belongs* to an elementary cycle $v_0, \cdots, v_{\nu-1}, v_0$ if $\mu \leq \nu$ and $u_i = v_i$ for some $0 \leq i \leq \nu - 1$.

The *distance* between two vertices u and v in G is the length of the shortest (u, v)-path and is denoted by $d(u, v)$. The *diameter* of the graph G is diam(G) =

$\max\{d(u,v) : u,v \in V\}$. For any vertex $u \in V$, we denote by $N_i(u) = \{v \in V : d(u,v) = i\}$. For a given $u \in V$ and a positive integer n such that $n = \max_{v \in V} d(u,v)$, the partition of V into $\{N_i(u) : i = 0, \cdots, n\}$ is a *level decomposition of G from u*. The set $N_i(u)$ is called the i^{th} *level*. In this paper we are mostly interested in some specific level decomposition, where the vertex in the bottom level is not of interest. Then we will write N_i for the i^{th} *level*. In such a decomposition, edges connect vertices in consecutive levels or in the same level. Given $u \in V$ and a level decomposition $\{N_i : i = 0, \cdots, n\}$ from u, we define for $v \in N_i$ the number $d^-(v) = |N(v) \cap N_{i-1}|$ (resp. $d^+(v) = |N(v) \cap N_{i+1}|$).

The *Categorical product* $G \otimes H$ of two graphs G and H has a vertex-set $V(G \times H) = V(G) \times V(H)$ and two vertices (u,v), (u',v') in $G \times H$ are adjacent if and only if $uu' \in E(G)$ and $vv' \in E(H)$.

The *hypercube* Q_n has $V = \{A : A \subseteq \{1,2,\cdots,n\}\}$ as a vertex-set and two vertices A and B are adjacent if and only if $|A \Delta B| = |(A \setminus B) \cup (B \setminus A)| = 1$. Q_n is regular of degree n and has diameter n.

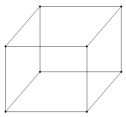


Fig. 1. Cube Q_3

The subgraph of Q_n induced by two consecutive levels N_{k-1} and N_k and denoted by L_n^k is semiregular of degrees $n-k+1$ and k, and it has order $\binom{n}{k} + \binom{n}{k-1}$. In particular, the subgraph induced by the two middle levels N_{k-1} and N_k of Q_{2k-1} and denoted by L_{2k-1}^k or more frequently H_k is regular of degree k. For $k = 3$, we obtain the Desargues graph H_3.

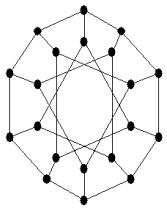


Fig. 2. Desargues Graph H_3

The Odd graph O_n has the set $\{A : A \subseteq \{1, 2, \cdots, 2n-1\}; |A| = n-1\}$ as a vertex-set and two vertices are adjacent if their corresponding subsets are disjoint. The odd graph O_n is regular of degree n and is of girth six for $n \geq 4$.

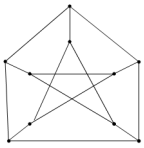


Fig. 3. Petersen Graph

2 Preliminary Results.

Mulder [11] introduced (0,λ)-graphs for which each two distinct vertices have either 0 or λ common neighbours. Furthermore, he proved that maximum (0,λ)-graphs are hypercubes. One way of generalizing this concept is to consider cycle-regular graphs which have some regularity properties and whose maximum graph for particular cases is related to hypercubes.

Definition 1 (Mollard [9]). *A graph $G = (V, E)$ of girth at least μ ($\mu \geq 2$) is a $[\mu, \lambda]$-cycle-regular graph ($\lambda \geq 1$) if there is a non-empty subset C of elementary cycles in G such that every path $P_{\mu+1}$ in G belongs to exactly λ cycles in C. In the particular case when C is the set of elementary cycles of a given length η ($\eta \geq 2\mu$), we say that G is a* $[\mu, \lambda, \eta]$-cycle-regular graph *(also called a cycle-regular graph).*

We can say now that the (0,λ)-graphs are the $[2, \lambda-1, 4]$-*cycle-regular graphs*. Our study focuses on the [3,1,6]-*cycle-regular graphs*, like the odd graphs O_n ($n > 2$). The [3,1,6]-cycle-regular graphs are triangle-free, since the triangle cannot belong to an elementary cycle of length greater than three. So for any vertex u, the set $N(u)$ is stable. In this class, we can also find the subgraph L_n^k.

Mulder [11] and Laborde and Rao Hebbare [6] showed separately that for a given degree and among all the $[2, 1, 4]$-cycle-regular graphs, the hypercube is of maximum order. On the other hand, Mollard [8] showed that for a given degree, the hypercube is of maximum diameter among these graphs. Furthermore for a given degree n, he showed that H_n is of maximum order among the $[3, 1, 6]$-cycle-regular graphs [9]. In this paper, we give some new characterizations of H_n in the class of graphs which are $[3, 1, 6]$-cycle-regular graphs. Moreover, we give other properties of $[3, 1, 6]$-cycle-regular graphs.

Proposition 1 (Mollard [9]). *If $G = (V, E)$ is a $[\mu, \lambda]$-cycle-regular graph of minimal degree $\delta(G) \geq 3$, then G is regular or semi-regular.*

3 [3,1,6]-Cycle-Regular Graphs

Mollard [9] gave an upper bound for the order of a [3,1,6]-cycle-regular graph of a given degree. Moreover, he gave a characterization of the subgraph H_n.

Proposition 2 (Mollard [9]). *Let $G = (V, E)$ be a $[3, 1, 6]$-cycle-regular graph and for an arbitrary level decomposition $\{N_0, N_1, \cdots, N_p\}$ of G, let $u \in N_i$. Then $d^-(u) \geq \lceil \frac{i}{2} \rceil$.*

Proposition 3 (Mollard [9]). *Let $G = (V, E)$ be a $[3, 1, 6]$-cycle-regular graph of maximum degree n and $\{N_0, N_1, \cdots, N_p\}$ be a level decomposition from a vertex of degree n. Then for $k = 0, \cdots, n-2$,*

$$|N_{2k+1}| \leq \frac{n}{k+1}(\binom{n-1}{k})^2 \quad \text{and} \quad |N_{2k+2}| \leq \frac{n(n-k-1)}{(k+1)^2}(\binom{n-1}{k})^2.$$

From 2 and 3, Mollard [9] can deduce that:

Proposition 4 (Mollard [9]). *Let $G = (V, E)$ be a $[3, 1, 6]$-cycle-regular graph of maximum degree n. Then*
1. $|V| \leq \binom{2n}{n}$,
2. $|V| = \binom{2n}{n}$ *if and only if G is H_n.*

By Proposition 2, Mollard [9] noticed that the diameter of a [3,1,6]-cycle-regular graph is at most $2n-1$ for a given maximum degree n. In other hand, we show that H_n is of maximum diameter among these graphs.

Theorem 1. *Let $G = (V, E)$ be a $[3, 1, 6]$-cycle-regular graph of maximum degree $n \geq 2$. Then*
1. $\mathrm{diam}(G) \leq 2n - 1$,
2. $\mathrm{diam}(G) = 2n - 1$ *if and only if G is H_n.*

Proof of Theorem 1. The first assertion is established by Mollard in [9] and [7].
The [3,1,6]-cycle regular graph H_n is a graph of degree n and diameter $2n-1$. Now, let $G = (V, E)$ be a [3,1,6]-cycle-regular graph of maximum degree n, diameter $2n-1$ and not isomorphic to H_n. According to Proposition 3,

$$|V| < \binom{2n}{n}. \tag{1}$$

Without loss of generality, consider a level decomposition of G with respect to a vertex p that has a diametral vertex, i.e a vertex such that $d(p, q) = 2n-1$. It is obvious that $|N_{2n-1}| \geq 1$. Using Proposition 4, we deduce that $|N_{2n-1}| \leq 1$, so $|N_{2n-1}| = 1$. Then $|N_{2n-2}|$ is the degree of the unique vertex in N_{2n-1}. So $|N_{2n-1}| \leq n$. According to Proposition 3 and the inequality 1, there is an index k, $0 \leq k \leq n-2$, such that either

$$|N_{2k+1}| < \frac{n}{k+1}(\binom{n-1}{k})^2 \tag{2}$$

or

$$|N_{2k+2}| < \frac{n(n-k-1)}{(k+1)^2}(\binom{n-1}{k-1})^2. \quad (3)$$

Suppose that 2 is valid. Proposition 2 yields:

$$d^-(u) \geq k+1 \quad \text{for any} \quad u \in N_{2k+2}.$$

By counting the edges between N_{2k+1} and N_{2k+2}, we obtain

$$|N_{2k+2}| \leq \frac{n-k-1}{k+1}|N_{2k+1}|.$$

After using the expression 2, we have:

$$|N_{2k+2}| < \frac{n(n-k-1)}{(k+1)^2}(\binom{n-1}{k})^2.$$

By iteration, we obtain $\forall h \geq k$:

$$|N_{2h+1}| < \frac{n}{h+1}(\binom{n-1}{h})^2$$

and

$$|N_{2h+2}| < \frac{n(n-h-1)}{(h+1)^2}(\binom{n-1}{h})^2.$$

On the one hand for $u \in N_{2n-1}$: $m(u, N_{2n-2}) \geq \lceil \frac{2n-1}{2} \rceil = n$. On the other hand, $|N_{2n-2}| < n$, a contradiction.$\diamondsuit$

Theorem 2. *Let $G = (V, E)$ be a bipartite $[3, 1, 6]$-cycle-regular graph of maximum degree $n > 3$. Then G is regular if and only if G is H_n.*

Proof. It is obvious that H_n is a bipartite regular [3,1,6]-cycle-regular graph of degree n.
Let $G = (V, E)$ be a regular bipartite [3,1,6]-cycle regular graph of degree $n > 3$ and not isomorphic to H_n. From Proposition 4 and Theorem 1, $|V(G)| < \binom{2n}{n}$ and $\text{diam}(G) < 2n-1$. So, for a level decomposition $\{N_0, ..., N_p\}$ of G, there is $k \in \{0, 1, ..., p\}$ such that

$$|N_{2h+1}| < \frac{n}{h+1}(\binom{n-1}{h})^2 \quad \text{and} \quad |N_{2h+2}| < \frac{n(n-h-1)}{(h+1)^2}(\binom{n-1}{h})^2, \forall h \geq k.$$

On the one hand for $u \in N_p$: $m(u, N_{p-1}) = n \geq \lceil \frac{p}{2} \rceil$. On the other hand, $|N_{p-1}| < n$, a contradiction. $\diamondsuit$

Proposition 5. *Let u and v be two vertices having at least two common neighbours in a $[3, 1, 6]$-cycle-regular graph $G = (V, E)$ with $\delta(G) \geq 2$. Then u, v and $N(u) \cap N(v)$ are on the same 6-cycle.*

Proof. Let u and v be *two* vertices of a [3,1,6]-cycle-regular graph having at least *two* common neighbours denoted a and b. Then the path v, a, u, b belongs to a single 6-cycle $\beta = v, a, u, b, c, d, v$. If there is $e \in N(u) \cap N(v)$ not on β, the path d, c, b, u would belong to at least *two* 6-cycles: β and d, c, b, u, e, v, d, a contradiction.

Proposition 6. *Let $A = \{x, y, z, t, u, v\}$ be the vertex set of one 6-cycle in a $[3, 1, 6]$-cycle-regular graph $G = (V, E)$ such that d(x,t) = 1. Then the subgraph G_A induced by A is isomorphic to $K_{3,3}$.*

Proof. Let $G = (V, E)$ be a [3,1,6]-cycle-regular graph and A the vertex set of a 6-cycle $\beta = x, y, z, t, u, v, x$ such that $d(x, t) = 1$. The path z, t, x, v belongs to a single 6-cycle z, t, x, v, x', t', z. If $x' \notin A$ and $t' \notin A$ then the path z, y, x, v belongs to at least *two* 6-cycles in $G : z, y, x, v, x', t', z$ and β, a contradiction. The cases ($x' = u$ and $t' \notin A$) and ($x' \notin A$ and $t' = y$) do not occur according to Proposition 5. Then $x' = u$ and $t' = y$. By symmetry, we can conclude that $zv \in E(G)$.

Proposition 7. *Let $G = (V, E)$ be a $[3, 1, 6]$-cycle-regular graph of maximum degree $n \geq 2$. If every two vertices have exactly three common neighbours then G is $K_{3,3}$.*

Proof. Let $G = (V, E)$ be a [3,1,6]-cycle-regular graph of maximum degree $n \geq 2$. If every *two* vertices have exactly *three* common neighbours, then each pair of vertices are of distance *one* or *two*. Hence $\text{diam}(G) = 2$ and G would be regular of degree 3. It is the bipartite graph $K_{3,3}$.

Proposition 8. *For each pair of vertices u and v, in any $[3, 1, 6]$-cycle regular graphs, one of the three cases arises:*
- *there is no P_4 joining u and v;*
- *there are two P_4 vertex-disjoint between u and v;*
- *there are four P_4 joining u and v, but exactly two of them are vertex-disjoint.*

Proof. Let G be a [3,1,6]-cycle regular graph and u, v two vertices in G. Four cases arise:
- If $d(u, v) \geq 4$, there is no P_4 joining them;
- If $d(u, v) = 3$, so there is a P_4 having u and v as ends. From the [3,1,6]-cycle regularity of G, there are a second P_4 having u and v as ends having no common vertex with the first one. From Proposition 5, we can deduce that there is no other P_4 joining u and v.
- If $d(u, v) = 2$ and there is a P_4 joining them, so there is a second P_4 between u and v, disjoint with the first one. Since G is triangle free, there is no other one.
- If $d(u, v) = 1$ and there is a P_4 joining them, so there is a second P_4 between u and v, disjoint with the first one. From Proposition 6, u and v are on $K_{3,3}$. So there are four P_4 between u and v, for which there are two P_4 vertex-disjoint.

4 The [3,1,6]-Cycle-Induced Regular Graphs

Consider a class of [3,1,6]-cycle-induced regular graphs, where in any graph, each induced P_4 with distinct ends, belongs to a single induced 6-cycle.

In this class, we can find the hypercubes, the odd extended graphs E_k ($k \geq 3$) and the [3,1,6]-cycle regular graphs.

Proposition 9. *If G is a $[3,1,6]$-cycle regular graph then G is a [3,1,6]-cycle-induced regular graph.*

Proof. Let G be a [3,1,6]-cycle regular graph and x, y, z, t be an induced P_4 joining two distinct vertices x and t. As G is a [3,1,6]-cycle regular graph then there is a single 6-cycle x, y, z, t, u, v, x containing the path x, y, z, t. If this 6-cycle is not induced, there is at least one of the edges zv, yu, vt or xu in G. If one of the two first edges appears, then x and t are adjacent, absurd. Whereas the edges vt and xu do not take place, since G is without triangle.

The reciprocal is not always true, since the hypercube Q_n, for $n \geq 2$, is a [3,1,6]-cycle-induced regular graphs but it is not a [3,1,6]-cycle regular graph.

Proposition 10. *If G is a $[3,1,6]$-cycle-induced regular graph, triangle-free and 4-cycle free, then G is a [3,1,6]-cycle regular graph.*

Proof. Let x, y, z, t be a P_4 joining two vertices in a [3,1,6]-cycle-induced regular graph. As G admits neither the triangle, or 4-cycle like induced subgraph, x, y, z, t is induced and its two ends are distinct, the path x, y, z, t belongs to an induced 6-cycle $\beta = x, y, z, t, u, v, x$. β is the single 6-cycle which contains the path x, y, z, t. If not, there is an other induced P_4, t, u_0, v_0, x with $u_0 \neq u$ or $v_0 \neq v$, forming not induced 6-cycle $\beta_0 = x, y, z, t, u_0, v_0, x$, since G is [3,1,6]-cycle-induced regular graph. However, this does not take place according to the assumptions, from where the result is obtained.

The graph $K_{3,3}$ is a [3,1,6]-cycle regular graph and [3,1,6]-cycle-induced graph belongs too which contains 4-cycle.

Lemma 1. *Let $\beta = x, y, z, t, u, v, x$ be an induced 6-cycle, in a $[3,1,6]$-cycle-induced regular graph, triangle free. If $|N(y) \cap N(v)| > 1$, then any vertex α of $N(y) \cap N(v) \setminus \{x\}$ is not on β. Moreover, $\alpha \in N(t)$.*

Proof. Let $\beta = x, y, z, t, u, v, x$ be an induced cycle. We have $x \in N(y) \cap N(v)$. If α is a vertex in $(N(y) \cap N(v)) \setminus \{x\}$, it is not on β, as β is induced. As G is triangle free, $\alpha \notin N(z) \cup N(u)$. In other hand, α and t are adjacent, because if not, the induced path y, z, t, u belongs to at least two induced 6-cycles β and y, z, t, u, v, a, y. Absurd.

Let u and v be two vertices of distance 2, in a [3,1,6]-cycle-induced regular graph, triangle free. Set:
$N(u) = (N(u) \cap N(v)) \cup A \cup B$, with :
$A = \{a \in N(u)/\exists b \in N(v) : d(a,b) = 1\}$ and $B = N(u) \setminus (N(v) \cup A)$
$N(v) = (N(v) \cap N(u)) \cup A' \cup B'$, with:
$A' = \{b \in N(v)/\exists a \in N(u) : d(a,b) = 1\}$ and $B' = N(v) \setminus (N(u) \cup A')$.

Proposition 11. *Let G be a $[3,1,6]$-cycle-induced regular graph, triangle-free. Then for any two vertices u and v of distance 2: $d(u) = d(v)$.*

Proof. Let u and v be two vertices in a [3,1,6]-cycle-induced regular graph G, triangle free. As G is triangle free, $N(u) \cap N(v)$, $A \cup B$, $A' \cup B'$ are stables. Furthermore, as G is in [3,1,6]-cycle-induced regular graph, for any $x \in B$ (resp. $a \in A$), the path x, u, z, v (resp. a, u, z, v), which is an induced P_4, belongs to exactly one induced 6-cycle. Therefore, the path u, z, v belongs to $|B| + |A|$ induced 6-cycles. In the same way, while reasoning on each vertex $y \in B'$ (resp. $b \in A'$), the path u, z, v belongs to $|B'| + |A'|$ induced 6-cycles. Thus, $|B| + |A| = |B'| + |A'|$. Then: $d(u) = |N(u) \cap N(v)| + |A| + |B| = |N(u) \cap N(v)| + |A'| + |B'| = d(v)$.

Proposition 12. *If G is a $[3, 1, 6]$-cycle-induced regular graph, triangle-free, then G is regular or semi-regular.*

Proof. Let G be a [3,1,6]-cycle-induced regular graph, triangle free.
1. If G is bipartite. Let u and v be two vertices in the same bipartition. As G is connected, there is a path with u and v as ends. Consider the geodesic $P_{\mu+1} = u, u_1, \cdots, u_{\mu-1}, v$. Since G is bipartite, μ is even. Therefore according to, Proposition 11, $d(u) = d(v)$. So that G is semi-regular or regular.
2. If G is not bipartite. There is a cycle of odd length, at least equal to 5, since G is triangle free. Let $C = a_0, a_1, \cdots, a_k, a_{k+1}, \cdots, a_{2k}, a_0$ an induced cycle of odd length $2k + 1$. For any i, consider the path $a_{i-2}, a_{i-1}, a_i, a_{i+1}, a_{i+2}$.
If i is even, $d(a_i) = d(a_0) = d(a_{i+1})$. If i is odd, $d(a_i) = d(a_0) = d(a_{i-1})$.

Thus, all the vertices of C have the same degree. If C is Hamiltonian, G is regular. If not, for any vertex u not belonging to C. From the connexity of G, there is a shorter P_{+1} having an end v in C and the other on u. If is even, it is finished. If not, the neighbor of v in C has the same degree of u. So that G is regular.

By analogy of the result established by Mollard, for a [3,1,6]-cycle-induced regular graph, triangle free, the number of edges between an arbitrary vertex and the vertex set of the first level which is former for him, is delimited in a level decomposition.

Lemma 2. *If G is a [3,1,6]-cycle-induced regular graph, triangle free and $N_0, N_1, \cdots, N_p$ an arbitrary level decomposition of G and u a vertex in N_i. Then $d^-(u) = m(u, N_{i-1}) \geq \lceil \frac{i}{2} \rceil$.*

Proof. This is true for $i = 0, 1$ and 2. Suppose that the property is true for all the vertices in N_i and let u in N_{i+2} and v in N_i on a path of length 2: u, y_0, v. Let $x_1, \cdots, x_p$ the neighbours of v in N_{i-1} ($p \geq \lceil \frac{i}{2} \rceil$) and $y_1, \cdots, y_q$ the neighbours of u (distinct from y_0) in N_{i+1}. Let $N_{i,j}$ ($i = 1, \cdots, p; j = 1, \cdots, q$) the number of induced 6-cycles containing the path x_i, v, y_0, u, y_j. As G is a [3,1,6]-cycle-induced regular graph triangle free, by using the two paths x_i, v, y_0, u (induced) and v, y_0, u, y_j (induced if $v \notin N(y_j)$), we obtain
$\forall i \sum_j N_{i,j} = 1$ and $\forall j \sum_i N_{i,j} \leq 1$.

Then $\forall i \forall j \sum_i \sum_j N_{i,j}$ is equal to p and is at most q; but $d^-(u) = q+1$ and $d^-(v) = p$. So that the result is obtained, by the recurrence assumption.

By using Lemma 2, we can deduce the following Lemma:

Lemma 3. *Let G be a [3,1,6]-cycle-induced regular graph, triangle free of maximum degree n and $N_0, N_1, \cdots, N_p$ a level decomposition from a vertex of degree n. Then, for $k = 0, \cdots, n-2$*

$$|N_{2k+1}| \leq \frac{n}{k+1}\left(\binom{n-1}{k}\right)^2 \text{ and } |N_{2k+2}| \leq \frac{n(n-k-1)}{(k+1)^2}\left(\binom{n-1}{k}\right)^2$$

Proof. This is true for $i = 0$. Also we have: $|N_1| = n$ and $|N_2| \leq n(n-1)$.

Suppose by induction that $|N_{2k}| \leq \frac{n(n-k)}{k^2}\left(\binom{n-1}{k-1}\right)^2$. Let n' be the degree of the vertices in the odd levels (n' may be equal to n). By counting the edges between N_{2k} and N_{2k+1}, we obtain, according to lemma 2, that this number is at least $|N_{2k+1}|\lceil\frac{2k+1}{2}\rceil$ and at most $|N_{2k}|\left(n - \lceil\frac{2k}{2}\rceil\right)$. Thus, we obtain:
$|N_{2k+1}| \leq |N_{2k}|\frac{n-k}{k+1} \leq \frac{n(n-k)^2}{k^2(k+1)} = \frac{n}{k+1}\left(\binom{n-1}{k}\right)^2$.
In the same way, by counting the edges between N_{2k+1} and N_{2k+2}, we obtain:
$|N_{2k+2}|leq|N_{2k+1}|\frac{n'-k-1}{k+1} \leq \frac{n(n-k)^2}{k^2(k+1)} = \frac{n(n-k-1)}{(k+1)^2}\left(\binom{n-1}{k}\right)^2$.

According to lemma 2, for a [3,1,6]-cycle-induced regular graph G, triangle free, of maximum degree n and for each vertex of N_p, there is $\lceil\frac{p}{2}\rceil \leq n$. So that, $p \leq 2n$.
But after counting the edges between N_{2n-1} and N_{2n}, we can deduce that $N_{2n} = \emptyset$. Thus G is of diameter at most $2n-1$. Always by the same process, the inequality $|N_{2n-1}| \leq 1$ hold. From this remark we could limit the order of G superlatively, as follows: $|V| \leq |N_0| + |N_1| + \cdots + |N_{2n-1}| \leq 2 + \sum_{i=0}^{n-2}\left(\binom{n-1}{k}\right)^2\left(\frac{n}{i} + \frac{n(n-i-1)}{(i+1)^2}\right) \leq \sum_i^n = 0\left(\binom{n}{i}\right)^2$
Thus any [3,1,6]-cycle-induced graph, triangle free, is of maximum order $\left(\binom{2n}{n}\right)$.
Let us show now that H_n, the subgraph induced by the central levels N_{n-1} and N_n of the hypercube of odd degree Q_{2n-1}, is the only n-regular graph, of order $\left(\binom{2n}{n}\right)$ and diameter $2n-1$.
Let G be a [3,1,6]-cycle-induced regular graph, triangle free, of maximum degree n and order $\left(\left(\binom{2n}{n}\right)\right)$. Then, for $k = 1, \cdots, n-1$,

$$|N_{2k+1}| = \frac{n}{k+1}\left(\binom{n-1}{k}\right)^2 \text{ and } |N_{2k+2}| = \frac{n(n-k-1)}{(k+1)^2}\left(\binom{n-1}{k}\right)^2$$

Then G is regular and it is clear that by using these equalities in the proof of Lemma 2, we obtain for any u in N_i $(i = 0, \cdots, 2n-1)$ $d^-(u) = m(u, N_{i-1}) = \lceil\frac{i}{2}\rceil$ and $d^+(u) = m(u, N_{i+1}) = n - \lceil\frac{i}{2}\rceil$.

Under these assumptions, there is the following Lemma:

Lemma 4. *G is 4-cycle free.*

Proof. Suppose that G admits at least one 4-cycle. According to the assumptions, $\text{diam}(G) = 2n - 1 \geq 3$. Moreover, we find no 4-cycle on the first two levels of G, nor even a 4-cycle meeting N_0, N_1 and N_2.

1^{st} **Case:** If this 4-cycle, denoted u, v, x, w meets three consecutive levels N_i, N_{i-1} and N_{i-2}, such as $u \in N_i$, v and w are in N_{i-1} and $x \in N_{i-2}$ ($i \geq 3$), there is $y^i nN(x) \cap N_{i-3}$. The induced path y, x, v, u belongs to a single induced 6- cycle y, x, v, u, t, t', y. $t' \in N(w) \cap N_{i-2}$. Since the 4-cycle y, x, w, t' meets the three levels N_{i-1}, N_{i-2} and N_{i-3}, there is $z \in N(y) \cap N_{i-4}$. The induced path z, y, x, v belongs to the single induced 6-cycle z, y, x, w, s, s', z. Following in the same way, we rich to one 4-cycle which would meet the three first levels N_0, N_1 and N_2, absurd.

2^{nd} **Case:**If u, v, x, w meets two consecutive levels N_i, N_{i-1} ($i \geq 2$). Four cases arise.

1. If u and x are in N_i, v and w on N_{i-1}. There is $y \in N(v) \cap N_{i-2}$. As $y \notin N(w)$, the induced path y, v, u, w belongs to a single induced 6-cycle y, v, u, w, t, t', y.

- If ($t \in N_{i-2}$ and $t' \in N_{i-3}$) or (t and t' are in N_{i-2}). Since G is triangle free, the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

- If ($t \in N_{i-1}$ and $t' \in N_{i-2}$) or ($t \in N_{i-2}$ and $t' \in N_{i-1}$) or ($t \in N_i$ and $t' \in N_{i-1}$) or (t and t' are in N_{i-1}). Since G is triangle free and according to the first case, the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

2. If u, w, x are in N_i, v is in N_{i-1}. There is y in $N(v) \cap N_{i-2}$. The induced path y, v, u, w belongs to a single induced 6-cycle y, v, u, w, t, t', y.

- If ($t \in N_{i-1}$ and $t' \in N_{i-2}$). Since G is triangle free, the induced path t, t', y, v belongs to at least two induced 6-cycles t, t', y, v, u, w, t and t, t', y, v, x, w, t, absurd.

- If (t and t' are in N_{i-1}) or ($t \in N_i$ and $t' \in N_{i-1}$). Since G is triangle free and according to the 1^{st} case, the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

3. If u and w are in N_i, v and x are in N_{i-1}. There is $y \in N(v) \cap N_{i-2}$. The induced path y, v, u, w belongs to a single induced 6-cycle y, v, u, w, t, t', y.
- If ($t \in N_i$ and $t' \in N_{i-1}$) or (t and t' are in N_{i-1}). As G is triangle free and according to item 2 of the 2^{nd} case, the induced path t, t', y, v belongs to at least two induced 6-cycles t, t', y, v, u, w, t and t, t', y, v, x, w, t, absurd.

- If ($t \in N_{i-1}$ and $t' \in N_{i-2}$). Since G is without triangle and according to the 1^{st} case, the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

4. If v, w, x are in N_{i-1}, $u \in N_i$ ($i \geq 2$). There is y in $N(v) \cap N_{i-2}$. The induced path y, v, u, w belongs to a single induced 6-cycle y, v, u, w, t, t', y.

- If ($t \in N_{i-2}$ and $t' \in N_{i-3}$). Since G is triangle free, the induced path t, t', y, v belongs to at least two induced 6-cycles t, t', y, v, u, w, t and t, t', y, v, x, w, t, absurd.

- If (t and t' are in N_{i-2}) or (t and t' are in N_{i-1}). Since G is triangle free and according to items 2 and 3 of the 2^{nd} case, the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

- If ($t \in N_{i-1}$ and $t' \in N_{i-2}$) or ($t \in N_{i-2}$ and $t' \in N_{i-1}$) or ($t \in N_{i-1}$ and $t' \in N_i$). Since G is triangle free and according to item 2 of the 2^{nd} case,.then the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

3^{rd} **Case:** If u, v, x, w is in N_i ($i \geq 2$). There is $y \in N(v) \cap N_{i-1}$, $y \notin N(x)$, because G is triangle free. The induced path y, v, u, w belongs to a single induced 6-cycle y, v, u, w, t, t', y. Several cases occur.

1. If (t and t' are in N_{i-1}) or ($t \in N_i$ and $t' \in N_{i-1}$) or ($t \in N_{i-1}$ and $t' \in N_i$) or (t and t' are in N_i). Since G is triangle free and from the 2^{nd} case, hence the induced path t, t', y, v belongs to at least two induced 6-cycles t, t', y, v, u, w, t and t, t', y, v, x, w, t, absurd.

2. If $t \in N_{i-1}$ and $t' \in N_{i-2}$. Since G is triangle free, so that the induced path w, t, t', y belongs to at least two induced 6-cycles w, t, t', y, v, u, w and w, t, t', y, v, x, w, absurd.

Theorem 3. *If G is a $[3, 1, 6]$-cycle-induced regular graph, triangle free, of maximum degree n. Then:*
1) $|V(G)| \leq \binom{2n}{n}$
2) $|V(G)| = \binom{2n}{n}$ *if and only if G is the subgraph H_n.*

Proof. **1)** The result is established above.
2) H_n is a [3,1,6]-cycle-induced regular graph, triangle free, regular of degree n and order $\binom{2n}{n}$.
If $|V(G)| = \binom{2n}{n}$, By Lemma 4, G is without 4-cycle. Then, the equivalence arise from Propositions 4 and 10.

Theorem 4. *Let G be a $[3, 1, 6]$-cycle-induced regular graph, triangle free, of maximum degree n, ($n \geq 2$). Then:*
1) $\text{diam}(G) \leq 2n - 1$
2) $\text{diam}(G) = 2n - 1$ *if and only if G is H_n.*

Proof. **1)** According to lemma 2, for a [3,1,6]-cycle-induced regular graph G, triangle free, of maximum degree n, for each vertex of N_p, there is $\lceil \frac{p}{2} \rceil \leq n$. So that, $p \leq 2n$. But after counting the edges between N_{2n-1} and N_{2n}, we can deduce that $N_{2n} = \emptyset$. Thus G is of diameter at most $2n - 1$.
2) H_n the subgraph induced by the two central levels N_{n-1} and N_n of the hypercube of odd degree Q_{2n-1}, is a [3,1,6]-cycle-induced regular graph, n-regular and of diameter $2n - 1$. Let G be a [3,1,6]-cycle-induced regular graph, of diameter $2n - 1$, so it would be of order $\binom{2n}{n}$. So, it is a 4-cycle free graph. So, from the above result, it is H_n.

References

[1] Bondy,J.A., Murty,U.S.R.: Graph Theory with applications, Macmillan & Co, London 1976.

[2] Havel,I.: Semipaths in directed cubes, in: Graphs and other Combinatorial Topics, **B. 59**(Teubneur Texte zumMathematik, Teubneur,Leipzig) (3° Symposium Tchescoslovaque de Th. Graphs), (1982).

[3] Berrachedi,A., Kahoul,N. : Graphs with special induced 6-cycle. Actes du colloque sur l'optimisation et les systèmes d'information *COSI'04*, Tizi-ouzou, Algérie (7-9 juin 2004).

[4] Berrachedi,A., Kahoul,N.: Cyclic Regularity in Some Particular Graphs. Acte du colloque sur l'optimisation et les sytmes d'information *COSI'05*, Bejaïa, Algérie (12-14 juin 2005).

[5] Berrachedi,A., Kahoul,N.: Special Cyclic Construction in Some Particular Graphs. Colloque International sur l'optimisation et les systmes d'information *COSI'06*, Alger-Algérie (11-13 juin 2006).

[6] Laborde,J.M., Rao Hebbare,S.P.: Another chararacterization of hypercubes. Discrete. Math, **39**(82) 161–166

[7] Mollard, M.: Cycle-Regular Graphs. Discrete. Math, **89** (1991) 29–41.

[8] Mollard,M.: Les invariants du n-cube, Thèse 3ème cycle, Université Joseph Fourier, Grenoble 1981.

[9] Mollard,M.: Quelques problèmes combinatoires sur l'Hypercube et les graphes de Hamming, Thèse Doctorat es-Science, Université Joseph Fourier, Grenoble 1989.

[10] Mulder,H.M.: $(0,\lambda)$-graphs and n-cubes, Discrete . Math, **28** (1979) 179–188.

[11] Mulder,H.M.: The interval function of a graph, Mathematics Centre Tracts 132, Mathematisch Centrum, Amesterdam 1980.

Optimisation parallèle et mathématiques financières

Pierre Spiteri[1]

IRIT – ENSEEIHT, UMR CNRS 5505 – 2 rue Charles Camichel, B.P. 7122
F-31 071 Toulouse, France
Pierre.Spiteri@enseeiht.fr

Résumé. Pour un problème issu des mathématiques financières, on établit un lien entre la formulation de ce problème et un problème de minimisation sur un espace approprié en fonction de la nature économique de l'application. L'expression des conditions d'optimalité conduit alors à la résolution d'équations ou d'inéquations aux dérivées partielles qu'on résout numériquement par des algorithmes parallèles asynchrones. Après avoir analysé la convergence des algorithmes parallèles asynchrones, on compare les résultats de ces derniers aux méthodes synchrones sur diverses architectures.

Mots clés: Options américaines et européennes, optimisation convexe, équations et inéquations aux dérivées partielles, problèmes complémentaires, méthode du gradient projeté, algorithmes parallèles synchrones et asynchrones.

1 Introduction et modèle mathématique

Le modèle mathématique étudié intervient dans de nombreuses applications mécaniques ou économiques. Compte tenu de la diversité des applications on a l'habitude de nommer ce problème, problème de l'obstacle.

Pour fixer la problématique, exposons une situation intervenant en mécanique des structures, qu'on peut comprendre intuitivement [3]. On considère, par exemple, une structure occupant un domaine fermé borné Ω, soumise à un certain nombre de contraintes notées f, correspondant à diverses forces appliquées sur cette structure. On souhaite alors connaître le déplacement noté u de la structure. On peut envisager deux cas distincts

- le déplacement n'est soumis à aucune contrainte, auquel cas le modèle mathématique décrivant cette étude est une équation aux dérivées partielles, en général linéaire, qui est soit stationnaire auquel cas l'opérateur est elliptique, ou qui dépend de la variable temps, et on doit résoudre un problème d'évolution,
- le déplacement est soumis à une contrainte, par exemple, on suppose que $u \leq \psi$, où ψ est une quantité connue ; dans ce cas le modèle mathématique décrivant cette étude est une inéquation aux dérivées partielles, appelée aussi par les mathématiciens inéquation variationnelle, problème fortement non linéaire, qui est soit stationnaire auquel cas l'opérateur associé est elliptique, ou qui dépend de la variable

[1] Cette étude a bénéficié du soutien du CNRS dans le cadre du projet ANR-07-CIS7-011-03.

supplémentaire temps, auquel cas l'opérateur associé est parabolique où hyperbolique du second ordre.

En fait qu'il s'agisse d'un problème avec ou sans contrainte, la formulation découle de celle d'un problème d'optimisation. Dans le cas le plus simple du problème stationnaire, considérons la fonctionnelle à minimiser suivante

$$J(v) = \frac{1}{2} a(v,v) - L(v), \ \forall v \in V \ ; \tag{1}$$

lorsque la structure est homogène, la forme bilinéaire a(u,v) et de la forme linéaire L(v) sont définies comme suit

$$a(u,v) = \int_\Omega (\nabla u.\nabla v + \theta.u.v).dx \ \text{ et } \ L(v) = \int_\Omega f.v.dx, \ \theta \geq 0 ; \tag{2}$$

V est un espace fonctionnel choisi en fonction des conditions aux limites ainsi que de la nature du problème ; en pratique V est soit un espace vectoriel normé complet dans le cas de l'optimisation sans contrainte soit un ensemble convexe fermé dans le cas de l'optimisation avec contrainte. Une position d'équilibre correspond au problème de minimisation suivant

$$\begin{cases} \text{Déterminer } u \in V \text{ tel que} \\ J(u) \leq J(v), \ \forall v \in V \end{cases} . \tag{3}$$

Dans le cas sans contrainte, lorsque la structure admet un déplacement nul le long de la frontière $\partial\Omega$, l'espace V est l'espace $H_1^0(\Omega)$ des fonctions de carré sommable dont les dérivées (en toute rigueur au sens des distributions) sont de carré sommable et nulles au bord. Sous des hypothèses convenables, il est facile de vérifier que la condition d'Euler qui traduit la nullité de la dérivée de la fonctionnelle J, s'écrit

$$<J'(u),v> = a(u,v) - L(v) = 0, \ \forall \ v \in H_1^0(\Omega), \tag{4}$$

expression qui conduit moyennant l'utilisation de la formule de Green, à la résolution de l'équation aux dérivées partielles suivante

$$\begin{cases} -\Delta u + \theta.u = f, \text{ presque partout dans } \Omega \\ u = 0, \text{ presque partout sur } \partial\Omega \end{cases} ; \tag{5}$$

le problème (5) correspond donc à la résolution d'un problème de Poisson stationnaire, avec conditions aux limites de Dirichlet homogènes. Toujours dans le cas sans contrainte, si de plus $\theta > 0$, et lorsque la dérivée normale du déplacement est nulle le long de la frontière $\partial\Omega$, l'espace V est l'espace $H_1(\Omega)$ des fonctions de carré sommable dont les dérivées sont de carré sommable ; dans ce cas, la condition d'Euler, appliquée dans les mêmes conditions que précédemment conduit à la résolution de l'équation aux dérivées partielles stationnaire suivante

$$\begin{cases} -\Delta u + \theta.u = f, \text{ presque partout dans } \Omega \\ \dfrac{\partial u}{\partial n} = \vec{\nabla} u.\vec{n} = 0, \text{ presque partout sur } \partial\Omega \end{cases} , \tag{6}$$

où n est la normale orientée vers l'extérieur au domaine Ω ; le problème (6) correspond à la résolution d'un problème de Poisson, avec conditions aux limites de

Neumann homogènes. De la même façon, lorsque la structure est soumise à d'autres conditions aux limites classiques, le problème de minimisation se traduit par la résolution d'un problème de Poisson, avec conditions aux limites appropriées, moyennant la définition correcte de l'espace V et celle de la forme bilinéaire a(u,v) et de la forme linéaire L(v).

Si on considère à présent le cas avec contrainte, on doit considérer le problème de minimisation, non pas sur un espace vectoriel normé complet V, mais sur un ensemble convexe fermé $\mathscr{C}$, défini par

$$\mathscr{C} = \left\{ v \in V \middle| v \le \psi, \text{ presque partout sur } \Omega \right\}, \tag{7}$$

V étant choisi comme précédemment en fonction des conditions aux limites ; l'expression de la condition d'optimalité, c'est à dire ici de l'inéquation d'Euler, s'exprime classiquement par

$$\langle J'(u), v \rangle = a(u, v-u) - L(v-u) \ge 0, \ \forall\, v \in \mathscr{C} ; \tag{8}$$

en considérant plusieurs choix distincts de la fonction test v, et si de plus V est l'espace $H_1^0(\Omega)$, on montre classiquement que l'inéquation d'Euler s'exprime par

$$\begin{cases} -\Delta u + \theta.u - f \le 0 \text{ et } u \le \psi, \text{ presque partout sur } \Omega, \\ (-\Delta u + \theta.u - f).(u - \psi) = 0, \text{presque partout sur } \Omega, \\ u = 0, \text{presque partout sur } \partial\Omega, \end{cases} \tag{9}$$

qui correspond à la résolution d'une inéquation aux dérivées partielles stationnaire.

Il est à noter que, dans le cas avec ou sans contrainte, la fonctionnelle J(v) est strictement convexe et de plus $\lim_{v \to \infty} J(v) = +\infty$; cette dernière propriété découle du fait que d'une part, la forme bilinéaire a(v,v) est continue et coercive et d'autre part que la forme linéaire est continue. Donc, dans les deux cas avec ou sans contrainte, le problème de minimisation admet une solution unique.

Ce type d'équations se retrouve également de manière analogue en mathématiques financières où on a à résoudre soit des équations soit des inéquations aux dérivées partielles [7]. Cependant l'analogie se borne uniquement à l'expression formelle des équations ou des inéquations aux dérivées partielles à résoudre. La modélisation du problème se conçoit de manière nettement moins intuitive que le problème précédent de mécanique. A ce stade, pour situer les choses, il est nécessaire d'introduire quelques éléments de terminologie. On appelle actif, une action, une obligation, une devise, un taux de change ou encore une matière première ; dans le jargon financier, on parle d'actif sous-jacent sur lequel porte l'option. Une option d'achat (ou call) donne à son détenteur le droit et non l'obligation d'acheter un actif financier (ou risqué) à une date future convenue d'avance, appelée échéance et notée T (ou date d'expiration qui limite la durée de vie de l'option) et à un prix fixé d'avance à la signature du contrat, conventionnellement appelé contrat interne. Une option de vente (ou put) donne à son détenteur le droit et non l'obligation de vendre un actif financier (ou risqué) à une date future convenue d'avance et à un prix fixé d'avance à la signature du contrat. Une option est également caractérisée par son montant, c'est-à-dire la quantité d'actif sous-jacent à acheter ou à vendre. Il y a deux

types d'option. Une option européenne est une option dont la date de maturité (c'est-à-dire la date future d'expiration) est fixée d'avance à la date d'échéance ; c'est-à-dire qu'il est interdit d'exercer le droit d'acheter ou de vendre avant la date d'échéance. Une option américaine est telle que le droit d'acheter ou de vendre l'actif considéré peut être exercé à n'importe quel moment avant la date d'échéance. Le prix d'exercice (ou strike) , est le prix fixé d'avance. L'option considérée (call ou put) a un prix appelé habituellement prime.

Pour fixer les idées, considérons le cas d'un call européen (option d'achat européen), d'échéance T, sur une action dont le cours à la date t, est donnée par S_t. Soit K le prix d'exercice. De deux choses l'une

- si, à l'échéance T, le prix K est supérieur au cours S_T, le détenteur de l'option n'a pas intérêt à exercer,
- si, par contre, $S_T > K$, l'exercice de l'option permet à son détenteur de réaliser un profit égal à $S_T - K$, en achetant l'action au prix K et en la revendant sur le marché au cours S_T .

On voit qu'à l'échéance la valeur du call est donnée par la quantité $Max(S_T - K,0)$. Pour le vendeur de l'option (appelé aussi trader), il s'agit, en cas d'exercice, d'être en mesure de fournir une action au prix K, et par conséquent de pouvoir produire à l'échéance une richesse égale à $Max(S_T - K,0)$. Au moment de la vente de l'option, qu'on choisira comme origine des temps, le cours S_T est inconnu et il se pose deux questions :

- combien faut-il payer à l'acheteur de l'option, autrement dit comment évaluer à l'instant t = 0 une richesse $Max(S_T - K,0)$ disponible à la date T ? C'est le problème du pricing,
- comment le vendeur qui touche la prime à l'instant t = 0, parviendra t'il à produire la richesse $Max(S_T - K,0)$ à la date T ? C'est le problème de la couverture, qui correspond à la détermination de la prime du call.

Notons que pour un put, la valeur à échéance est donnée par $Max(K - S_T ,0)$.

La réponse à ces deux questions ne peut se faire qu'à partir d'un minimum d'hypothèse de modélisation. L'hypothèse retenue est le principe d'absence d'opportunité d'arbitrage (A.O.A.) qui s'énonce comme suit : « *il est impossible de réaliser une série d'opérations financières qui procure un profit certain sans mise de fonds initiales c'est à dire sans prendre de risque* ».

Pour déterminer le montant de cette prime d'achat ou de vente, Black et Scholes d'une part, Merton d'autre part, ont effectué en 1973, une modélisation stochastique qui conduit à la résolution d'un problème aux limites. Par exemple, pour déterminer le prix d'un call européen, Black et Scholes ont proposé un modèle permettant de déterminer la prime par une formule explicite. Cependant pour le calcul d'un put américain, il n'existe pas de formules explicites et l'utilisation de méthodes de calcul numérique est indispensable. En fait, on montre que la prime du put ou du call américain, peut se calculer en résolvant une inéquation aux dérivées partielles. Dans la suite on note A un opérateur aux dérivées partielles, $\psi(x)= Max(S-K,0)$ pour un call ou $\psi(x) = Max(K-S,0)$ pour un put, suivant le cas, K est le prix d'exercice, r est le taux d'intérêt et S_t est le cours d'exercice à la date t. Avec ces notations, l'inéquation aux dérivées partielles s'écrit

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} + A.u(x,t) - r.u(x,t) \geq 0,\ \mathrm{u}(x,t) \geq \psi(x),\ \text{dans } [\mathrm{o},\mathrm{T}]xR^n \\ (\dfrac{\partial u(x,t)}{\partial t} + A.u(x,t) - r.u(x,t))(\mathrm{u}(x,t) - \psi(x)) = 0,\ \text{dans } [\mathrm{o},\mathrm{T}]xR^n \\ u(x,T) = \psi(x) \end{cases} , \tag{10}$$

Dans cette équation (10) u correspond au supremum de l'espérance mathématique des flux actualisés que rapporte la mise de fonds initiale. A noter que dans le cas d'option européenne, on a à résoudre une équation aux dérivées partielles d'évolution, alors que dans le cas du calcul d'option américaine on a à résoudre une inéquation aux dérivées partielles d'évolution, à la différence près que les conditions aux limites ne sont pas ici précisées puisqu'on travaille dans un milieu non borné ; cependant aussi bien dans le cas du calcul d'option européenne que dans celui d'option américaine, des difficultés supplémentaires subsistent.

Dans la suite le papier est organisé comme suit : au second paragraphe est exposée la résolution numérique du problème de mathématiques financières, tout particulièrement les algorithmes parallèles avec échanges asynchrones entre les processeurs ; s'agissant de méthode itérative, la convergence est analysée. Le dernier paragraphe présente les résultats des expérimentations parallèles d'une part sur un clusteur et d'autre part sur une grille de calcul.

2 Résolution numérique du calcul d'options

2.1 Remarques préliminaires

A partir de la formulation (10) pour le calcul d'option américaine, ou de manière analogue pour le calcul d'option européenne, on peut effectuer plusieurs remarques

- le problème de mathématique financière est défini dans un domaine non borné, soit ici R^n ; sur le plan pratique et numérique, on résout le problème dans un domaine fermé borné Ω inclus dans R^n et, par des arguments d'analyse très sophistiqués, on montre que la solution du problème défini dans le domaine Ω tend vers celle du problème à résoudre, c'est-à-dire dans R^n, quand le mesure du domaine Ω tend vers l'infini [7],
- aussi bien dans le cas de calcul d'options américaine qu'européenne, on doit résoudre un problème d'évolution. Il est à noter que dans ce cas, on ramène habituellement la résolution numérique d'un tel problème dépendant du temps, à celle d'une suite de problèmes stationnaires par discrétisation convenable et classique de la variable temporelle qui sera abordée au paragraphe 2.2,
- en général l'opérateur A est l'opérateur de convection – diffusion ; cet opérateur n'est pas auto – adjoint. Cependant si les coefficients des dérivées sont constants, on peut toujours effectuer un changement de variable, tel que dans la nouvelle expression de l'opérateur, le coefficient du terme de gradient est nul ; ainsi l'opérateur intervenant dans le modèle est auto – adjoint. Considérons par exemple, le problème unidimensionnel suivant

$$\begin{cases} \dfrac{\partial v(x,t)}{\partial t} + c\dfrac{\partial v(x,t)}{\partial x} - \dfrac{\partial^2 v(x,t)}{\partial x^2} = f,\ [0,1]x[0,T] \\ v(x,0) = v_0(x) \\ v(0,t) = v(1,t) = 0 \end{cases} ;$$

si on pose v(x,t) = exp((αx+βt)).u(x,t), avec $\alpha = \dfrac{c}{2}$ et $\beta = -\alpha^2 = -\dfrac{c^2}{4}$, on vérifie que ce changement de variable conduit à résoudre le problème suivant

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} - \dfrac{\partial^2 u(x,t)}{\partial x^2} = \exp(-(\alpha x + \beta t)).f,\ [0,1]x[0,T] \\ u(x,0) = \exp(-\alpha x).v_0(x) \\ u(0,t) = u(1,t) = 0 \end{cases} .$$

Cette remarque est particulièrement importante, car il est alors facile de montrer que la résolution du problème aux limites est équivalente à celle d'un problème de minimisation. Si les coefficients des dérivées ne sont pas constants, on n'a pas besoin de résoudre le problème d'optimisation associé et on résout directement le problème aux limites,

- dans le cas du calcul d'options européennes, en fonction des conditions aux limites, l'espace de travail est un espace vectoriel normé complet, et en exprimant la condition d'Euler, on aboutit à la résolution d'une équation aux dérivées partielles du type (5) ou (6) dont la résolution ne pose aucune difficulté. Dans le cas du calcul d'options américaines, on travaille dans un convexe fermé $\mathcal{C}$ et en exprimant la condition d'optimalité d'Euler, on aboutit à la résolution d'une inéquation aux dérivées partielles du type (10). Précisons que le problème de minimisation est strictement équivalent à celui de la résolution des équations ou des inéquations aux dérivées partielles, c'est à dire que toute solution du premier problème est solution du second et inversement,
- précisons enfin qu'on prend pour origine des temps, le moment de la vente ou de l'achat de l'option. Comme souligné lors des questions abordant les notions de pricing et de couverture, cela découle du fait que la valeur de la prime à l'échéance est connue comme une des données du problème et qu'en fait le problème revient à déterminer la valeur de la prime à l'instant t=0.

2.2 Discrétisation du problème de l'obstacle

Le calcul d'options européennes revenant, comme on l'a rappelé, à résoudre une équation aux dérivées partielles, ne pose aucun problème méthodologique. C'est pourquoi nous ne développerons pas cette problématique classique.

Par contre, nous développerons plus en détail le calcul d'options américaines. Sur le plan numérique, comme déjà indiqué, on doit résoudre ce même

problème dans un domaine Ω ce qui nécessite d'adjoindre, en plus de la condition finale u(x,T) = ψ(x) (rappelons que T représente l'échéance), des conditions aux limites ; en pratique on choisit soit des conditions aux limites de Dirichlet spécifiant la valeur de u sur la frontière du domaine Ω, soit des conditions aux limites de Neumann fixant la valeur de la dérivée normale à la frontière du domaine Ω (c.f. [7]).

La discrétisation des problèmes précédents ne pose pas de difficulté majeure.

Pour la variable temporelle on utilise les différences finies. En pratique, on préfère utiliser des schémas implicites ou semi – implicites, car ces derniers sont insensibles aux propagations d'erreurs de troncature liées à la discrétisation des opérateurs aux dérivées partielles ainsi qu'aux erreurs de chute liées à la représentation des nombres réels en machine ; ce qui assure par conséquent la stabilité numérique des schémas utilisés et évite un effet papillon. A chaque pas de temps, on est donc conduit à la résolution d'un système algébrique.

En ce qui concerne la variable d'espace, on peut utiliser soit des différences finies, soit des éléments finis soit même des volumes finis. En pratique, on utilise habituellement une discrétisation en espace par différences finies classiques, ce qui assure que les matrices de discrétisations vérifient des propriétés spécifiques, assurant la convergence des méthodes itératives parallèles synchrones ou asynchrones utilisées dans la suite. Ainsi, la dérivée seconde de u par rapport à la variable x sera approximée par le quotient différentiel suivant

$$\frac{\partial^2 u(x,y,z)}{\partial x^2} \approx \frac{u(x+\delta x,y,z) - 2.u(x,y,z) + u(x-\delta x,y,z)}{(\delta x)^2} + O(\delta x^2) \ , \qquad (11)$$

où δx représente le pas de discrétisation spatial. On utilise des formules analogues pour obtenir une approximation des dérivées secondes par rapport aux variables y et z. De plus, pour discrétiser les dérivées premières à 0(δx) près, on considère un schéma décentré forward si le coefficient de convection est négatif ou un schéma décentré backward si le coefficient de convection est positif, défini respectivement par

$$\frac{\partial u(x,y,z)}{\partial x} \approx \frac{u(x+\delta x,y,z) - u(x,y,z)}{\delta x} \text{ ou } \frac{\partial u(x,y,z)}{\partial x} \approx \frac{u(x,y,z) - u(x-\delta x,y,z)}{\delta x} \qquad (12)$$

On est donc conduit à résoudre à chaque pas de temps, des systèmes algébriques de grandes tailles, pour lesquels, compte tenu de la propagation des erreurs de chute, l'utilisation de méthodes itératives est fortement recommandée.

Dans la suite, nous noterons par des lettres majuscules les analogues discrets des inconnues et des données du problème ; par exemple U représentera l'analogue discret de u intervenant dans (11)–(12). On notera aussi **A** la matrice de discrétisation du problème aux limites, obtenue en utilisant les formules de type (11)-(12), et F le second membre du système discrétisé.

2.3 Linéarisation du problème de l'obstacle

Comme déjà indiqué, le problème de calcul d'options américaines est fortement non linéaire. On doit donc à chaque itération, linéariser par un procédé approprié le système à résoudre. On dispose de deux méthodes de linéarisation
- soit l'utilisation d'une méthode de Richardson projetée sur le convexe $\mathcal{C}$,
- soit la méthode d'Howard à partir de l'écriture sous forme complémentaire (voir [2]) du problème de l'obstacle.

Nous présentons d'abord la méthode de Richardson projetée puis plus brièvement la méthode d'Howard.

On a vu que la résolution du problème d'options européennes ou américaines revient à résoudre un problème de minimisation soit dans un espace vectoriel, soit dans un convexe. Cette équivalence entre les deux types de problèmes joue un rôle important au plan algorithmique. On va donc considérer une variante parallèle synchrone ou asynchrone de la méthode de Richardson projetée sur le convexe $\mathcal{C}$, pour le calcul d'option américaine, ce qui revient à résoudre numériquement, une équation de point fixe.

La parallélisation d'un algorithme nécessite la décomposition du problème en plusieurs sous - problèmes interconnectés. Sur la plan mathématique, on travaille donc sur des espaces produits. Si E désigne l'espace de travail en dimension finie, on décompose donc E en un produit fini de p espaces E_i comme suit $E = \prod_{i=1}^{p} E_i$, où ici p désigne le nombre de processeurs utilisés. De plus, dans la mesure où on souhaite résoudre un problème d'optimisation, il est clair que l'espace E ainsi que les espaces E_i, i=1,..p, sont des espaces de Hilbert. Pour tout $V \in E$, soit $P_C(V)$ la projection orthogonale de E sur le convexe discrétisé C; on considère une décomposition compatible de $P_C(V)$ avec la décomposition en espace produit définie par $P_C(V) = \{....,P_{C_i}(V_i),...\}$, où $P_{C_i}(V_i)$ est le projecteur orthogonal de E_i sur C_i, pour i = 1,..,p. Soit δ un nombre réel positif donné ; définissons l'application de point fixe suivante

$$U = P_C(V - \delta(AV - F)) = F_\delta(V) \ ; \tag{13}$$

De façon analogue, on peut décomposer l'application de point fixe précédente, comme suit $F_\delta(V) = \{....,F_{i,\delta}(V),...\}$; pour i =1,..,p, on écrit donc

$$U_i = P_{C_i}(V_i - \delta(\sum_{j=1}^{p} A_{i,j}V_j - F_i)) = F_{i,\delta}(V). \tag{14}$$

L'approximation initiale $U^{(0)}$ étant donnée, on définit les itérations asynchrones par

$$U_i^{(q+1)} = \begin{cases} U_i^{(q)} & \text{si i } \notin \text{ s(q)} \\ F_{i,\delta}(....,U_j^{\rho_j(q)},....) & \text{si i } \in \text{ s(q)} \end{cases} \tag{15}$$

où s(q) représente l'ensemble des composantes i relaxées et vérifie $s(q) \subset \{1,....,p\}$, condition prenant en compte la mise à jour des composantes en parallèle ; de plus s(q) vérifie également la condition

$$\text{pour tout i} \in \{1,\ldots,p\}, \text{ l'ensemble } \{q \mid i \in s(q)\} \text{ est dénombrable,} \tag{16}$$

ce qui traduit que, théoriquement, on relaxe une infinité de fois chaque composante. De plus les exposants $\{\rho_j(q)\}$ modélisent les échanges asynchrones entre les processeurs; ces exposants sont tels que pour tout j = 1,…,p, $\rho_j(q) \in N$, pour tout nombre entier q ; de plus, ces exposants vérifient les propriétés suivantes

$$0 \leq \rho_j(q) \leq q \quad \text{et} \quad \lim_{q \to \infty} (\rho_j(q)) = +\infty, \tag{17}$$

cette dernière hypothèse prenant en compte une hypothétique panne des processeurs.

Remarque 1 : lorsque pour tout j = 1,…, p et pour tout $q \in N$, on a $\rho_j(q) = q$, alors (15) modélise un algorithme parallèle synchrone, qui apparaît comme un cas particulier d'algorithme parallèle asynchrone. De plus, toujours dans ce contexte synchrone, si $s(q) = \{1, \ldots, p\}$ ou s(q) = q modulo(p) +1, respectivement, alors (15) modélise la méthode séquentielle de Jacobi ou de Gauss – Seidel, respectivement.

Pour résumer, lorsqu'on utilise les méthodes itératives parallèles asynchrones, les processeurs effectuent en parallèle la résolution de chaque sous - problème en utilisant les données d'interactions disponibles produites par les autres processeurs. L'analyse de la convergence de ces méthodes s'effectue en utilisant la notion de M-matrice dont on rappelle la définition

Définition 1 : une matrice ayant des coefficients hors diagonaux négatifs, est une M – matrice si celle-ci est non singulière et d'inverse non négative.

En application du théorème de Perron Frobenius (c.f. [13]), il est à noter que si J est la matrice de Jacobi d'une M – matrice, alors J est une matrice non négative admettant un rayon spectral $\mu \in R$, tel que $0 < \mu < 1$, auquel est associé un vecteur propre Γ de composantes Γ_l strictement positives. Cette propriété joue un rôle essentiel pour l'analyse de la convergence des algorithmes parallèles asynchrones.

On suppose que la décomposition en blocs de la matrice A soit telle que les blocs diagonaux sont fortement définis positifs, c'est à dire

$$<A_{i,i}.\, V_i,\, V_i> \geq\ n_{i,i}.\ \|V_i\|^2_{i,2}, \tag{18}$$

où $\|V_i\|^2_{i,2}$ dénote la norme euclidienne du sous – vecteur $V_i \in E_i$; de plus, on suppose que les normes induites des sous – matrices $A_{i,j}$ sont bornées par des nombres - $n_{i,j}$ (les réels $n_{i,j}$ étant des nombres négatifs), ce qui est toujours vrai. Conformément à un résultat de [9], ces deux hypothèses sont équivalentes à l'hypothèse globale suivante

$$< A_{i,i}.V_i + \sum_{j \neq i} A_{i,j}.V_j, V_i > \geq \sum_{j=1}^{p} n_{i,j}.\|V_i\|_{i,2}.\|V_j\|_{j,2},\ \forall\ i \in \{1,...,p\},\ \forall\ V \in E. \tag{19}$$

Il convient de remarquer que lorsque les discrétisations sont effectuées convenablement et conformément à (11)-(12), la condition (18) est vérifiée, ce qui assure la validité de l'hypothèse (19).

Théorème 1 : L'hypothèse (19) étant vérifiée, et si de plus la matrice N de coefficients $(n_{i,j})$ est une M – matrice, il existe un nombre réel strictement positif δ_0, tel que pour tout $\delta \in]0, \delta_0[$, les variantes parallèles synchrone et asynchrone (14) - (15) de l'algorithme de Richardson projeté converge vers la solution du problème.

Principe de la démonstration : elle est basée sur le fait que, d'une part l'opérateur de projection orthogonal est une contraction et d'autre part sous les hypothèses considérées, l'application de point fixe est une contraction dans l'espace E normé par

la norme uniforme avec poids W → $\|W\|_{\mu,J} = \max_l (\frac{\|W_l\|_{l,2}}{\Gamma_l})$, μ étant la constante de contraction strictement inférieure à l'unité [2] (c.f. [1]).

Comme indiqué précédemment, la méthode d'Howard utilise la formulation discrète correspondant à une formulation du problème d'obstacle mis sous forme complémentaire, soit

$$\text{Max}(\mathbf{A}.U - F, U - \Psi) = 0, \tag{20}$$

où Ψ correspond à l'analogue discret de l'obstacle ψ intervenant dans la discrétisation du problème (10). L'écriture précédente sous forme complémentaire fournit un procédé de résolution numérique du problème. En effet, pour linéariser ce problème (20), on va considérer une méthode analogue à la classique méthode de Newton utilisée pour résoudre un système algébrique non linéaire. Soit $U^{(0)}$ une approximation initiale de la solution du problème à résoudre au $k^{ième}$ pas de temps ; dans le cadre de l'intégration en temps du problème de l'obstacle, $U^{(0)}$ donnée initiale de la méthode de linéarisation, peut par exemple être choisie comme la valeur obtenue à la $(k-1)^{ième}$ itération en temps. Au départ, lors de l'intégration au premier pas de temps, dans la mesure où on intègre le problème aux limites dans le sens rétrograde, on choisit la condition finale ψ(x) comme valeur d'initialisation de $U^{(0)}$ de l'algorithme de Howard . A (19), on associe un système linéarisé.

$$B^{(k)} . U = G^{(k)} ; \tag{21}$$

l'algorithme de Howard est schématisé par le diagramme suivant

Boucle en temps

$U^{(0)}$ donné;

Norme ← 1 + précision ;

Tant que norme > précision **faire** :

Construction de la matrice linéarisante $B^{(k)}$ par Howard

Construction du second membre $G^{(k)}$ par Howard

Résolution du système $B^{(k)}.U^{(1)} = G^{(k)}$

Norme ← $|| U^{(1)} - U^{(0)} ||$

$U^{(0)} \leftarrow U^{(1)}$

fait

fin boucle en temps

la matrice linéarisante $B^{(k)}$ et le second membre $G^{(k)}$ étant définis comme suit

- si la $l^{ième}$ composante de (**A**.U – F) est supérieure à la $l^{ième}$ composante de (U - Ψ), alors la $l^{ième}$ ligne de la matrice $B^{(k)}$ sera la $l^{ième}$ ligne de la matrice **A** et la $l^{ième}$ composante de $G^{(k)}$ sera la $l^{ième}$ composante de F,
- sinon, dans le cas contraire, la $l^{ième}$ ligne de la matrice $B^{(k)}$ sera constituée par une ligne de zéros, sauf l'élément diagonal qui sera égal à un et la $l^{ième}$ composante de $G^{(k)}$ sera égale à la $l^{ième}$ composante de Ψ.

[2] μ rayon spectral de la matrice de Jacobi de la M-matrice $N = (n_{i,j})$

Pour la résolution du système linéarisé, on peut utiliser n'importe quelle méthode de résolution, y compris les méthodes parallèles synchrones et asynchrones par sous domaines avec ou sans recouvrement (méthode alternée de Schwarz). Compte tenu du procédé de discrétisation choisi, qui conduit à des matrices de discrétisations **A** qui sont des M – matrices, on vérifie que les matrices linéarisantes $B^{(k)}$ sont à chaque pas des M – matrices, ce qui assure la convergence des méthodes parallèles synchrones et asynchrones par sous domaines.

Théorème 2 : la matrice de discrétisation **A** étant une M – matrice, les méthodes parallèles synchrones et asynchrones par sous domaines avec ou sans recouvrement, convergent vers la solution du problème linéarisé (21).

Démonstration : on utilise des techniques d'ordre partiel (c.f. [11]).

Remarque 2 : Il est à noter qu'intuitivement, la méthode de Howard utilisée dans le cas d'option américaine, revient en fait à effectuer une projection sur le convexe.

3 Résultats expérimentaux

3.1 Résultats code séquentiel

Pour effectuer les calculs d'options américaines on a mis en œuvre la méthode de Richardson projetée; pour faciliter la portabilité des codes, ces derniers sont écrits en langage C. On limite essentiellement la présentation des résultats expérimentaux au cas du calcul d'options américaines; en effet le calcul d'options européennes revient à résoudre une équation aux dérivées partielles linéaire. Il est à noter que l'algorithme utilisé a de bonnes performances.

Le solveur de calcul d'options américaines a été testé avec une taille de problème correspondant au nombre de points de discrétisation dans le domaine Ω; soit ici 96 x 96 x 96 = 884 736 points de discrétisation. Sur le clusteur utilisé le temps de calcul est de 708 secondes. La vitesse de convergence est bonne.

2.4 Résultats code parallèle

Les essais numériques ont été effectués sur un réseau de processeurs POWER 3 organisé en grappe de type SMP (clusters) dans lequel chaque nœud possède 16 processeurs partageant une mémoire commune. Lors des expérimentations 2, 4, 8, 16 et 32 processeurs ont été utilisés. La parallélisation des codes séquentiels est effectuée au moyen de MPI. Il est à noter qu'au-delà de 16 processeurs, on utilise 2 nœuds, alors que jusqu'à 16 processeurs un seul nœud est utilisé.

La table 1 donne les temps de restitutions de l'algorithme de Richardson projeté sur le réseau de processeurs. La table 2 et 3, respectivement donnent un récapitulatif de l'accélération et de l'efficacité de l'algorithme de Richardson projeté et parallélisé. La figure n°1 permet de comparer l'efficacité des méthodes synchrones (x) et asynchrones (o) entre elles, dans le cas du problème considéré.

Sur le plan numérique, on constate que le nombre de relaxations nécessaires pour converger, effectuées par l'algorithme synchrone est identique à celui nécessaire en mode séquentiel. Par contre en mode asynchrone il y a un surcoût minime en nombre de relaxations, de l'ordre de 5%, du au comportement chaotique des communications inter processeurs. Le nombre de relaxation a donc tendance à augmenter légèrement

Table 1. Temps de restitution de l'algorithme parallélisé (96 x 96 x 96).

Nb de processeurs	Mode synchrone	Mode asynchrone
1	708 secondes	708 secondes
2	358 secondes	349 secondes
4	161 secondes	163 secondes
8	80 secondes	76 secondes
16	46 secondes	40 secondes
32	35 secondes	23 secondes

Table 2. Accélération de l'algorithme parallélisé (temps parallèle sur temps sequential).

Nb de processeurs	Mode synchrone	Mode asynchrone
2	1.98	2.03
4	4.4	4.34
8	8.85	9.32
16	15.39	17.7
32	20.23	30.78

Table 3. Efficacité de l'algorithme parallélisé (Accélération sur nb. de proc.).

Nb de processeurs	Mode synchrone	Mode asynchrone
2	0.99	1.01
4	1.10	1.08
8	1.10	1.16
16	0.96	1.10
32	0.63	0.96

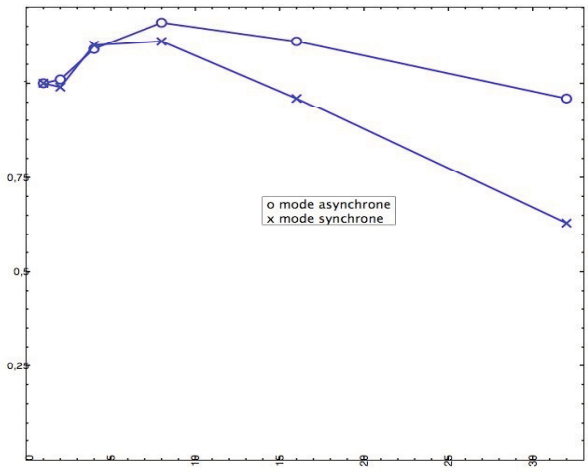


Fig. 1. *Efficacité en fonction du nombre de processeurs pour le problème 96 x 96 x 96*

sans que pour autant le temps de calcul soit pénalisé. En effet ce surcoût n'est pas pénalisant dans la mesure où le temps de restitution reste inférieur à celui obtenu avec l'algorithme en mode synchrone. De plus, un nombre supplémentaire de relaxations en mode asynchrone a une incidence positive sur la précision obtenue et a donc pour effet d'améliorer la qualité numérique de la solution obtenue.

Il est à noter que les algorithmes qu'ils soient en mode synchrone ou en mode asynchrone ont des performances très proches. Ceci est dû au fait que le nombre de points de discrétisation est relativement petit. En dessous de 8 processeurs, l'algorithme en mode synchrone peut avoir de meilleures performances ; en fait, on a pu remarquer que le découpage en blocs du vecteur solution a une influence sur les performances. Par contre, à partir de 8 processeurs, l'algorithme en mode asynchrone donne toujours de meilleures performances que l'algorithme en mode synchrone. Dans ce dernier cas, cette chute de performance est due au poids des synchronisations dans les méthodes synchrones car les temps d'attente entre processeurs pénalisent ce type d'algorithme. Soulignons que les communications en mode asynchrone ont un grand intérêt lorsque le nombre de processeurs augmente car les phénomènes de deadlock sont largement minimisés ce qui n'est pas le cas des communications en mode synchrone. Signalons que lors de la mise au point de l'implémentation, on a comparé les performances entre méthode synchrone et asynchrone, pour la résolution d'un problème de convection – diffusion correspondant par exemple au calcul d' options européennes. Pour un nombre de points de discrétisation de l'ordre de 3 750 000, avec 128 processeurs, les communications asynchrones permettent de diminuer le temps de restitution de l'ordre d'un tiers.

L'étude des valeurs de l'accélération et de l'efficacité résumée respectivement en table 2 et 3 confirme l'intérêt des méthodes utilisées. Pour le problème considéré, l'algorithme asynchrone se démarque de l'algorithme synchrone à partir de 8 processeurs. Globalement, les deux versions de l'algorithme ont des performances équivalentes dans le cas où un seul nœud du cluster est utilisé. Une différence très significative entre les deux modes d'algorithmes parallèles apparaît lorsque 32 processeurs sont utilisés. La perte d'efficacité de l'algorithme en mode synchrone peut s'expliquer de la manière suivante

- lorsque 32 processeurs sont utilisés, chaque processus ne prend en charge que 3 blocs. Comme les messages et les blocs sont pratiquement de même taille, le surcoût de communications atteint une proportion significative,
- deux processus sont impliqués dans les communications faisant intervenir le réseau d'interconnexion entre les nœuds. Ce type de communication étant plus lent, cela entraîne un déséquilibrage de charge qui retarde les autres processeurs.
- Il est clair que l'algorithme parallèle asynchrone est moins sensible aux problèmes liés aux communications et à l'équilibrage des charges que l'algorithme parallèle synchrone. Il est à noter que l'algorithme asynchrone subit aussi une chute de performance normale lorsque 32 processeurs sont utilisés. Toutefois, dans ce cas, l'efficacité reste bonne, de l'ordre de 0.96.
- On remarque que lorsqu'il y a peu de processeurs, on obtient des efficacités supérieures à l'unité. Ce phénomène est dû au fait que dans ce cas, le poids des synchronisations est faible par rapport au temps de calcul pur ; ce phénomène découle aussi de défauts des mémoires caches, générés par le volume important de données à

échanger. Par contre, lorsque le nombre de processeurs augmente, les efficacités obtenues sont inférieures à l'unité, ce qui est normal.

Pour terminer et montrer l'intérêt du calcul parallèle asynchrone dans le cas d'applications de grande taille, signalons quelques tests en cours sur une grille de calcul. Si on considère des essais effectués sur un premier cluster comportant 20 processeurs, on obtient un temps de restitution synchrone de 53.164 secondes et un temps de restitution asynchrone de 34.627 secondes. Si on considère un second clusteur de 20 processeurs, on obtient un temps de restitution synchrone de 56.00 secondes et un temps de restitution asynchrone de 39.491 secondes. Donc, sur chacun des clusteurs, les algorithmes synchrones et asynchrones ont des comportements quasiment identiques. Si à présent on effectue les mêmes calculs en utilisant 10 processeurs sur chacun de ces clusters très distants l'un de l'autre, on obtient un temps de restitution synchrone de 5 minutes 12.574 secondes et un temps de restitution asynchrone de 39.989 secondes. On constate clairement que sur deux clusteurs la version synchrone est pénalisée avec une perte de performance d'un facteur de l'ordre de 8 et que les synchronisations dégradent nettement les performances des algorithmes.

Références

1. Chau, M., Spiteri, P., Parallel asynchronous Richardson method for the solution of obstacle problem. In : Proceedings of HPCS 2002, pp. 133 – 138, IEEE Press, Los Alamitos (2002)
2. Cottle, R.W., Golub, G.H., Sacher, R.S., On the solution of large structured linear complementary. The block partitioned case. Appl. Math. Optim., 347 – 363 (1978)
3. Duvaut, G., Lions, P.L., Les inéquations en mécanique et en physique. Dunod (1972)
4. Giraud, L., Spiteri, P., Résolution parallèle de problèmes aux limites non linéaires. M^2AN, 25 – 5, 579 – 606 (1991)
5. Glowinski, R., Lions, P.L., Tremolières, R., Analyse numérique des inéquations variationnelles. Dunod, Tome 1 & 2 (1976)
6. Hull, J., Options, futures et autres actifs dérivés. 5ème édition, Pearson Education (2004)
7. Lamberton, D., Lapeyre, B., Introduction au calcul stochastique appliqué à la finance. Ellipses (1991)
8. Miellou, J.C., Spiteri, P., Two criteria for the convergence of asynchronous iterations. Dans Computers and Computing, Chenin, P., et all ed., Masson & Wiley, 90 – 95 (1985)
9. Miellou, J.C., Spiteri, P., Un critère de convergence pour des méthodes générales de point fixe. M^2AN, 19 – 4, 645 – 669 (1985)
10. Spiteri, P., Simulations d'exécution parallèle pour la résolution d'inéquations variationnelles stationnaires. Revue EDF, Informatique et Mathématiques Appliquées, série C, 1, 149 – 158 (1983)
11. Spiteri, P., Miellou, J.C., El Baz, D., Asynchronous Schwarz alternating methods with flexible communications for the obstacle problem. Résaux et systèmes répartis, 13 – 1, 47 – 66 (2001)
12. Wilmott, P., Howison, S., Dewyne, J., The mathematics of financial derivatives : a studient introduction. Cambridge university press (1995)
13. Varga, R.S., Matrix iterative analysis, Prentice all (1962)

Simultaneously lifting several sets of variables into a cover inequality for binary knapsack polytope

Méziane Aïder and Chafia Boughani

LAID3, Faculté de Mathématiques, U.S.T.H.B.
BP 32 El Alia, 16111 Bab Ezzouar, Alger, Algeria.
{m-aider@usthb.dz, chafia7806@yahoo.fr}
http://www.usthb.dz

Abstract. The cover inequality induces a face for the knapsack polytope. Nevertheless, it can be lifted to define a facet. Lifted cover inequalities often prove useful as cutting planes for solving linear integer programming problems.
This paper deals with developing polynomial time algorithms for simultaneously lifting sets of variables into a cover inequality for the binary knapsack polytope. Our aim is to suggest a generalization of the simultaneous lifting on several sets of variables into a cover inequality for the binary knapsack polytope. Necessary conditions for the non dominated inequalities, generated by our algorithms, to define facets for the knapsack polytope are presented.

Key words: Simultaneous lifting, Knapsack problem, Polyhedral approach.

1 Introduction

The knapsack problem is an NP-hard combinatorial optimization problem, but it can be solved in pseudo-polynomial time by dynamic programming [2]. This problem has been intensively studied, particulary over the two past decades [7] [9] [10]. It arises in many business and industrial applications, and also in many complex problems as a subproblem.

The knapsack problem is the problem of selecting from a set $N = \{1, \ldots, n\}$ of n items, each with a specified weight w_j and a profit p_j, a subset of items whose value is maximum and whose weight sum does not exceed the capacity of the knapsack, d.

This paper deals with the 0-1 knapsack problem defined as follows:

$$\max\{p^T x | w^T x \leq d, x \in \{0,1\}^n\} \tag{1}$$

where $p \in \mathbb{Z}_+^n$, $w \in \mathbb{Z}_+^n$ and $d \in \mathbb{Z}_+$.

A $0-1$ knapsack polytope is the convex hull of the feasible solutions, i.e. the polytope $\mathcal{P} = conv(\mathbb{P})$ where:

$$\mathbb{P} = \left\{ x \in \{0,1\}^n, \quad \sum_{j=1}^{n} w_j x_j \leq d, \right\} \tag{2}$$

An inequality $\pi x \leq \pi_0$ is valid for a polyhedron $P \subseteq \mathbb{R}^n$ if it is satisfied by every point in P, i.e., $P \subseteq \{x \in \mathbb{R}^n \mid \pi x \leq \pi_0\}$. The polyhedron $F = \{x \in P \mid \pi x = \pi_0\}$ is called a face of P. The dimension of a polyhedron P is the maximum number of its affinely independent points minus 1. If $dim(F) = dim(P) - 1$ then F is a facet and the corresponding valid inequality is said to be a facet-defining.

Without loss of generality, let e^j be the j^{th} unit vector, i.e. $e^j_j = 1$ and $e^j_i = 0$ for $i \neq j$. By the assumption $w_j \leq d$, on the weights of the items, the elements of the set $V = \{e^1, \ldots, e^n\}$ are affinely independent solutions vectors of the knapsack problem. Consequently, P is full dimensional $dim(P) = n$.

The $0-1$ knapsack polytope has been widely studied by several authors [1] [6] [12], whom use in general the cover of the knapsack problem.

A set C is called a cover if $\sum_{j \in C} w_j > d$.

A cover is minimal, if for each $l \in C$, $\sum_{j \in C \setminus l} w_j \leq d$.

Every cover induces a valid inequality, called a cover inequality, of the form:

$$\sum_{j \in C} x_j \leq |C| - 1 \tag{3}$$

If C is minimal the inequality is said minimal cover inequality which is a facet-defining for a lower dimensional knapsack polytope $conv(\mathcal{P}_C)$, where

$$\mathcal{P}_C = \{x \in \{0,1\}^{|C|} : w_C^T x \leq d\} \tag{4}$$

Nevertheless, this inequality can be lifted to be a facet-defining for $\mathcal{P}$.

Lifting, introduced by Gomory [4], increases the dimension of a valid inequality by adding variables that have the maximum coefficient that maintains the validity of the new inequality. Recently, some works, based upon lifting over superadditive functions, have been done on sequence independent lifting [12] [5].

In 1978, Zemel [13] introduced the simultaneous lifting for binary integer variables into a cover inequality. This method finds the extreme points of the polar from the solutions of exponentially many linear integer programs. Since, this area of research is not explored. In 2005, Easton and Kevin [3] proposed a linear time algorithm for simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. Their work left an open question, on how to choose the minimal cover C and the set of variables to lift E, solved by Sharma [11], in 2007, who develop a polynomial algorithm named MSLA (Maximal Simultaneously Lifted Algorithm).

Sharma has proposed to use the idea developed in MSLA to lift several sets, say E, F, G, and H, to get lifted cover inequalities which can be a facet-defining over $\mathcal{P}_{C\cup E\cup F\cup G\cup H}$.

The motivation of our work is to suggest a generalization of the simultaneous lifting on several sets of variables into a cover inequality for the binary knapsack problem using a small modification of Sharma's procedure. We develop an algorithm for simultaneously lifting several sets of variables E_1, E_2, ..., E_m into a cover inequality. The elements of the sets C and E_i, for $i = 1, \ldots, m$, are arranged in decreasing order of their w coefficients.

2 Simultaneously lifting several sets of variables

In this section, we suggest a generalization of the simultaneous lifting on several sets, E_i where $i = 1, \ldots, m$, of variables into a cover inequality by developing a general algorithm, Algorithm 1 which operates as follows: at each iteration of the running through of the set E_i we test if the weight sum of the selected variables do not exceed the capacity of the knapsack, then, allowing the running through of the $(i+1)^{th}$ set E_{i+1} in order to lift a number of variables, q_{i+1}. The form of the inequalities, generated by algorithm 1, is the following:

$$\sum_{j\in C} x_j + \sum_{l=1}^{i} \alpha^l \sum_{j\in E_l} x_j \leq \mid C \mid -1 \qquad i = 1, \ldots, m \tag{5}$$

where:

$\alpha^i = \frac{|C|-1-p}{q_i} \times \lambda_i$, for $i = 1, \ldots, m$.

$\lambda_i = \dfrac{q_i}{\sum_{l=1}^{i} q_l}$ is the rate of the elements of E_i taken from $N \setminus C$, with

$\lambda_i \in]0, 1[$ for $i = 1, \ldots, m$.

Thus the lifting coefficients are:

$$\alpha^i = \alpha^{i+1} = \frac{\mid C \mid -1-p}{\sum_{l=1}^{i} q_l} \qquad i = 1, \ldots, m \tag{6}$$

The inequalities generated by Algorithm 1 have the following form:

$$\sum_{j\in C} x_j + \alpha \sum_{l=1}^{i} \sum_{j\in E_l} x_j \leq \mid C \mid -1 \qquad i = 1, \ldots, m \tag{7}$$

Where $\alpha = \alpha^i = \alpha^{i+1} = \ldots$

Algorithm 1: Algorithm for simultaneously lifting several sets of variables

Data: A knapsack constraint: $\sum_{j\in N} w_j x_j \le d$.

The set C and m sets ($m < n$): $E_i = \{j_1^i, j_2^i, \ldots, j_{|E_i|}^i\}$, $i = 1, \ldots, m$.

Result: Inequalities of the form (7)

Variables:

p : The number of elements taken from C corresponding to largest coefficients w

$E_{\#}^i$: The number of indices chosen from the set E_i to be lifted

$C_{\Sigma p}$: The sum of the p elements taken from the minimal cover C

q_i : : The maximum number of elements taken from $\{j_1^i, \ldots, j_{E_{\#}^i}^i\}$ to be lifted

$E_{\Sigma q_i}^i$: The sum of the q_i elements of the set E_i

$\alpha_{i,j}$: The lifting coefficient at the iteration (i, j)

$p := |C| - 1$;

While $p \ge 0$ **Do**

$C_{\Sigma p} := \sum_{r=|C|-p+1}^{|C|} w_{i_r}$; $\alpha'_{p,0} := \infty$; $q_1 := 1$; $E_{\#}^1 := 1$;

While $E_{\#}^1 \le |E_1|$ **Do**

$E_{\Sigma q_1}^1 := \sum_{r=E_{\#}^1-q_1+1}^{E_{\#}^1} w_{j_r^1}$; $T_1 = C_{\Sigma p} + E_{\Sigma q_1}^1$;

If $T_1 \le d$ **Then**

$\alpha'_{p,E_{\#}^1} := \frac{|C|-1-p}{q_1}$; $q_1 := q_1 + 1$;

$\vdots$

$q_m := 1$; $E_{\#}^m := 1$; $\alpha_{E_{\#}^{m-1},0} := \alpha'_{E_{\#}^{m-1},E_{\#}^m}$;

While $E_{\#}^m \le |E_m|$ **Do**

$E_{\Sigma}^m := \sum_{r=E_{\#}^m-q_m+1}^{E_{\#}^m} w_{j_r^2}$; $T_m = C_{\Sigma p} + E_{\Sigma q_1}^1 + \ldots + E_{\Sigma q_m}^m$;

If $T_m \le d$ **Then**

$\alpha_{E_{\#}^{m-1},E_{\#}^m} := \frac{|C|-1-p}{q_1+\ldots+q_m}$; $q_m := q_m + 1$;

Else

$\alpha_{E_{\#}^{m-1},E_{\#}^m} := \alpha_{E_{\#}^{m-1},E_{\#}^m-1}$;

End if

$E_{\#}^m := E_{\#}^m + 1$;

End while

$\vdots$

Else $\alpha'_{p,E_{\#}^1} := \alpha'_{p,E_{\#}^1-1}$;

End if

$E_{\#}^1 := E_{\#}^1 + 1$;

End while

$p := p - 1$;

End while

The running time of Algorithm 1 is

$$O(|C|\prod_{i=1}^{m}|E_i| + |C|log|C| + \sum_{i=1}^{m}|E_i|\log|E_i|) \simeq O(n^{m+1}).$$

Therefore, Algorithm 1 runs in polynomial time.

Example 1. Consider the following knapsack constraint:

$$29x_1+29x_2+27x_3+25x_4+14x_5+14x_6+12x_7+11x_8+11x_9+9x_{10}+9x_{11} \leq 90$$

The minimal cover is $C = \{1, 2, 3, 4\}$ and the corresponding minimal cover inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$.

For MSLA, $E = \{5, 6, 7, 8, 9, 10, 11\}$.
For Algorithm 1, $E_1 = \{5, 6, 7, 8\}$ and $E_2 = \{9, 10, 11\}$

Inequalities generated by MSLA:

$x_1 + x_2 + x_3 + x_4 + 1x_5 \leq 3$

$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$

$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$

Inequalities generated by Algorithm 1:

$x_1 + x_2 + x_3 + x_4 + 1x_5 \leq 3$

$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$

$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$

Contrarily to MSLA, Algorithm 1 generates maximal non-dominated inequalities even if the instance has the w_i coefficients of the set $N \setminus C$ less than or equal to $d - \sum_{j=2}^{|C|} w_j$. The condition is that these coefficients do not belong to the set E_1 but they can belong to the set E_2.

In the remaining of this paper, we keep the same signification for the variables p, q_i, $E^i_\#$ and $\alpha_{(i,j)}$ given in Algorithm 1. We will also use the following notations:

- $E^{E^i_\#}$: The set of $E^i_\#$ variables actually lifted.
- $E^{E^i_\#}_{q_i}$: The set of $E^i_\#$ variables actually lifted from which we can take only q_i variables
- e: Unitary vector
- i_r: The r^{th} element of the minimal cover C
- j^i_r: The r^{th} element of the set E_i

Theorem 1. *Any non-dominated inequality reported from Algorithm 1 is a facet-defining over $P_{C\cup(\bigcup_{i=1}^{m} E_i)}$ as long as all of the following conditions are satisfied:*

i) The inequality dominates another inequality reported by Algorithm 1;

ii) If $E^i_{\#} - q_i \geq 1$ *then the following set is not a cover:*

$$\{i_{|C|-p+1}, \ldots, i_{|C|}\} \cup \{j^1_{|E_1|-q_1+1}, \ldots, j^1_{|E_1|}\} \cup \ldots \cup \{j^1_i, j^i_{|E_i|-q_i+3}, \ldots, j^i_{|E_i|}\} \cup \ldots \cup \{j^m_{|E_m|-q_m+1}, \ldots, j^m_{|E_m|}\}$$

where $i = 1, \ldots, m$.

iii) If $E^m_{\#} - q_m \geq 2$ *then the following set is not a cover:*

$$\{i_{|C|-p+1}, \ldots, i_{|C|}\} \cup \{j^1_{|E_1|-q_1+1}, \ldots, j^1_{|E_1|}\} \cup \ldots \cup \{j^{m-1}_{|E_{m-1}|-q_{m-1}+1}, \ldots, j^{m-1}_{|E_{m-1}|}\} \cup \{j^m_1, j^m_{|E_m|-q_m+2}, \ldots, j^m_{|E_m|}\}$$

iv) If $E^{E^m_{\#}} \neq E_m$, *then* $\alpha_{E^{m-1}_{\#}, E^m_{\#}} > \alpha_{E^{m-1}_{\#}, E^m_{\#}+1}$.

Proof. We will show that the inequalities of the form (7), generated by Algorithm 1 and which verify the conditions of Theorem 1, define facets using the direct technic of facet proof, i.e. by showing that the dimension of the faces induced by the non-dominated inequalities generated by Algorithm 1 is $dim(P) - 1$. Algorithm 1 proves validity by exhaustively checking every single relevant point that may make the inequality invalid.

Clearly, the dimension of $conv(P_{C \cup (\bigcup_{i=1}^m E_i)})$ is $|C| + (\sum_{i=1}^m |E_i|)$. Furthermore, the origin does not meet any simultaneously lifted cover inequality at equality so the dimension of the face induced by any reported inequality

$$\sum_{j \in C} x_j + \alpha_{E^i_{\#}, E^{i+1}_{\#}} (\sum_{i=1}^m \sum_{j \in E^{E^i_{\#}}} x_j) \leq |C| - 1 \tag{8}$$

can be at most $|C| + (\sum_{i=1}^m |E_i|) - 1$.

The $|C| + \sum_{i=1}^n |E_i|$ affinely independent points are determined as follows:

- Algorithm 1 requires that C is minimal and so the point

$$\sum_{i \in C \setminus \{h\}} e_i,$$

for each $h \in C$, is feasible and meets the inequality (8) at equality. We obtain $|C|$ points.

- Since the inequality (8) dominates at least one inequality, so

$$\alpha^i_{E^i_\#, E^{i+1}_\# - 1} = \alpha^i_{E^i_\#, E^{i+1}_\#}$$

Therefore, the point:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E^{E^i_\#}_{q_i}} e_j + \sum_{k \in E^{E^m_\# - 1}_{q_m}} e_k$$

is feasible. Since the set E_i is sorted, the points:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E^{E^i_\#}_{q_i}} e_j + \sum_{k \in E^{E^m_\#}_{q_m+1} \setminus \{k\}} e_k$$

are feasible for all $k \in E^{E^m_\#}_{q_m+1}$. We obtain $q_m + 1$ points.
- Moreover the inequality (8) dominates another inequality, the set:

$$\{i_{|C|-p+1}, \ldots, i_{|C|}\} \cup \Big(\bigcup_{i=1}^{m-1} \{j^i_{|E_i|-q_i+1}, \ldots, j^i_{|E_i|}\} \Big) \cup \{j^m_{|E_m|-q_m-x}, \ldots, j^m_{|E_m|-x}\}$$

is a minimal cover. So the set:

$$\{i_{|C|-p+1}, i_{|C|-p+2}, \ldots, i_{|C|}\} \cup \Big(\bigcup_{i=1}^{m-1} \{j^i_{|E_i|-q_i+1}, \ldots, j^i_{|E_i|}\} \Big) \cup \{j^m_{|E_m|-q_m-x}, \ldots, j^m_{|E_m|-x}\} \setminus \{l\}$$

is not a minimal cover, where $l \in L = \{j^i_{|E_i|-q_i+1}, \ldots, j^i_{|E_i|}\}$ and $x \in \{1, \ldots, q_m - 1\}$, then the points:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-2} \sum_{j \in E^{E^i_\#}_{q_i}} e_{j^i} + \sum_{j \in E^{E^k_\#}_{q_k} \setminus \{l\}} e_{j^k} + \sum_{j^m \in E^{E^m_\#}_{q_m+1}} e_{j^m}$$

are feasible for all $k = 1, \ldots, m-1$. We obtain q_i points.
- By assumption, if $E^i_\# - q_i \geq 1$ then:

$$\{i_{|C|-p+1}, \ldots, i_{|C|}\} \cup \Big(\bigcup_{i=1 i \neq k}^{m-2} \{j^i_{|E_i|-q_i+1}, \ldots, j^i_{|E_i|}\} \cup \{j^k_1, j^k_{|E_k|-q_k+3}, \ldots, j^k_{|E_k|}, \} \Big) \cup \{j^m_{|E_m|-q_m+1}, \ldots, j^m_{|E_m|}\}$$

is not a cover, where $k = 1, \ldots, m-1$, and since the inequality (8) dominate an other inequality then the points:

$$\sum_{i \in C_p} e_i + e_l + \sum_{j \in E^k_{\#_{q_k-2}}} e_{j^k} + \sum_{i=1}^{m-2} \sum_{j \in E^i_{\#_{q_i}}} e_{j^i} + \sum_{j^m \in E^m_{\#_{q_m+1}}} e_{j^m}$$

are feasible, where $k = 1, \ldots, m-1$ and $l = 1..E^k_\# - q_k$. So we have $E^k_\# - q_k$ points.

- By assumption, if $E^m_\# - q_m \geq 2$ then:

$$\{i_{|C|-p+1}, i_{|C|-p+2}, \ldots, i_{|C|}\} \cup \Big(\bigcup_{i=1}^{m-1} \{j^i_{|E_i|-q_i+1}, \ldots, j^i_{|E_i|}\} \Big) \cup \{j^m_1, j^m_{|E_m|-q_m+2}, j^m_{|E_m|-q_m+3}, \ldots, j^m_{|E_m|}\}$$

is not a cover. So the points:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E^i_{\#_{q_i}}} e_{j^i} + e_l + \sum_{j^m \in E^{E^m_\#}_{q_m-1}} e_{j^m}$$

are feasible, where $l = 1..E^m_\# - q_m - 1$. Thus, we obtain $E^m_\# - q_m - 1$ points.

- If $E^{E^m_\#} \neq E_m$ et $\alpha_{E^{m-1}_\#, E^m_\#} > \alpha_{E^{m-1}_\#, E^m_\#+1}$, then Algorithm 1 changes $\alpha_{E^{m-1}_\#, E^m_\#}$ values during the $(E^{m-1}_\#, E^m_\# + 1)^{th}$ step of the p^{th} iteration. Therefore, the point:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E^i_{\#_{q_i}}} e_{j^i} + \sum_{j^m \in E^{E^m_\#+1}_{q_m+1}} e_{j^m}$$

is feasible. Thus, the points:

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E^i_{\#_{q_i}}} e_{j^i} + \sum_{j^m \in E^{E^m_\#}_{q_m}} e_{j^m}$$

are feasible for all $l \in E_m \setminus E^{E^m_\#}$. So we have $|E_m \setminus E^{E^m_\#}|$ points.

Clearly, these points are affinely independent as the matrix, representing these points, has cyclically permuted matrices with only one 0 over these permutations. In addition, there are several rows containing only a single 1.

3 Conclusions and future research

In this paper, we have suggest generalization of the simultaneous lifting on several sets of variables into a cover inequality for binary knapsack problems. Some theoretical results provide necessary conditions when the non-dominated inequalities, generated by these algorithms, define facets.

Computational research with CPLEX can be done to strengthen our theoretical results. They are also some open questions on how many sets can be lifted into a cover inequality and what will be their cardinal. It is also interesting to do a study on the values of λ_i in $]0, 1[$.

References

1. E. Balas. Facets of the knapsack polytope. Mathematical Programming, 8:146–164, 1975.
2. G. Dantzig. Discrete variable extremum problems. Operations Research, 5:266–277, 1957.
3. T. Easton and K. Hooker. Simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. Discrete Optimization, 5(2):254–261, 2008.
4. R. E. Gomory. Some polyhedra related to combinatorial problems. Linear Algebra and its Applications, 2:451–588, 1969.
5. Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. Combinatorial Optimization, 4:109–129, 2000.
6. P. L. Hammer, E. L. Johnson, and U. N. Peled. Facets of regular 0-1 polytopes. Mathematical Programming, 8:179–206, 1975.
7. S. Martello and P. Toth. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Chichester, 1990.
8. G.L Nemhauser and L.A Wolsey. Integer and Combinatorial Optimization. John Wiley & Sons, New York, 1999.
9. D. Pisinger and P. Toth. Knapsack problems. In D.-Z. Du and P.M. Paralos, editors, Handbook of Combinatorial Optimization, volume 1, pages 299–428. Kluwer Academic Publishers, 1998.
10. H. M. Salkin and C. A. De Kluyver. The knapsack problem: a survey. Naval Research Logistics Quarterly, 22(1):127–144, 2006.
11. K. Sharma. Simultaneously lifting sets of varibales in binary knapsack problems. Master of science, Kansas State University, 2007.
12. L.A. Wolsey. Valid inequalities and superadditivity for 0-1 integer programs. Mathematic of Operations Research, 2(1):66–77, February 1977.
13. E. Zemel. Lifting the facets of zero one polytopes. Mathematical programming, 15:268–277, 1978.

Propriété duale de König pour l'hypergraphe des intervalles d'un poset sans N

Fatma KACI

Faculté des Sciences et des Sciences de l'ingénieur,
Université Mohamed Khider, Biskra, Algérie

kaci_fatma2000@ yahoo.fr

Résumé : Soit P un poset fini rangé. On considère l'hypergraphe $H(P)$ dont les sommets sont les éléments de P et dont les arêtes sont les intervalles maximaux de P. Il est connu que $H(P)$ a la propriété de König et duale de König pour la classe des posets séries-parallèles. Notre étude porte sur la classe des posets sans N qui représente une extension de celle des posets séries-parallèles. Nous prouvons que $H(P)$ possède la propriété duale de König.

Mots clés: hypergraphe, poset, intervalle, propriété duale de König.

Abstract: Let P be a finite poset. We consider the hypergraph $H(P)$ whose vertices are the elements of P and whose edges are the maximal intervals of P. It is known that $H(P)$ has the König and dual König properties for the class of series-parallel posets. In this paper we study the class of N-free posets which contains series-parrallel posets, we prove that $H(P)$ has the dual König property.

Keywords: Hypergraph, poset, interval, dual König property.

---

1 Introduction:

Soit P un ensemble partiellement ordonné (dit brièvement *poset)* rangé fini. On dit que p couvre q et on note $p \prec q$ si $p \prec v \leq q$ implique $v = q$. Un sous ensemble I de P, de la forme $I = \{v \in P, p \leq v \leq q\}$, noté par $[p,q]$, est appelé *intervalle*. Si en plus, p est un élément minimal et q est un élément maximal alors I est appelé intervalle *maximal*. Nous notons par $\mathcal{I}(P)$ l'ensemble des intervalles maximaux de P. L'hypergraphe $\mathcal{H}(P) = (P, \mathcal{I}(P))$, brièvement noté $\mathcal{H} = (P, \mathcal{I})$, est appelé *hypergraphe des intervalles* du poset P. Les sommets de $\mathcal{H}$ sont les éléments de P et ses arêtes sont les intervalles maximaux de P.

Un sous ensemble A (resp. T) d'éléments de P est appelé stable (resp. recouvrement par sommets) de $\mathcal{H}$ si toute arête de $\mathcal{H}$ contient au plus un élément de A (resp. au moins un élément de T). Une famille d'arêtes $\mathcal{R}$ (resp. $\mathcal{M}$) de $\mathcal{H}$ est appelée un recouvrement par arêtes (resp. couplage) de $\mathcal{H}$ si tout élément de P appartient à au moins (resp. au plus) une arête de $\mathcal{R}$ (resp. $\mathcal{R}$).
Soient

$\alpha(\mathcal{H}) = \max\{|A| : A \text{ est un ensemble stable de } \mathcal{H}\}$,
$\nu(\mathcal{H}) = \max\{|M| : M \text{ est un couplage de } \mathcal{H}\}$,
$\tau(\mathcal{H}) = \min\{|T| : T \text{ est un recouvrement par sommets de } \mathcal{H}\}$,
$\rho(\mathcal{H}) = \min\{|\mathcal{R}| : \mathcal{R} \text{ est un recouvrement par arêtes de } \mathcal{H}\}$,

respectivement les *nombres de stabilité, de couplage, de recouvrement par sommets* et *de recouvrement par arêtes* de $\mathcal{H}$. Il est facile de voir que $\nu(\mathcal{H}) = \tau(\mathcal{H})$ et $\alpha(\mathcal{H}) = \rho(\mathcal{H})$. On dit que $\mathcal{H}$ a *la propriété de König* si $\nu(\mathcal{H}) = \tau(\mathcal{H})$ et duale *de König* si $\alpha(\mathcal{H}) = \rho(\mathcal{H})$ (ou de manière équivalente $\nu(\mathcal{H}^*) = \tau(\mathcal{H}^*)$ puisque $\alpha(\mathcal{H}) = \nu(\mathcal{H}^*)$ et $\rho(\mathcal{H}) = \tau(\mathcal{H}^*)$ pour le dual $\mathcal{H}^*$ de $\mathcal{H}$).

Cette classe d'hypergraphes a été étudiée dans [1]-[5] et [8]. D'intéressants résultats existent aussi bien d'un point de vue algorithmique que des relations min-max.

Soient P_1 et P_2 deux posets. La *somme directe* $P_1 + P_2$ de P_1 et P_2 est le poset défini sur l'ensemble $P_1 \cup P_2$ tel que $x \leq y$ dans $P_1 + P_2$ ssi $(x, y \in P_1$ et $x \leq_{P_1} y)$ ou $(x, y \in P_2$ et $x \leq_{P_2} y)$. La *somme linéaire* $P_1 \oplus P_2$ de P_1 et P_2 est le poset défini sur l'ensemble $P \cup Q$, tel que $x \leq y$ dans $P_1 \oplus P_2$ ssi $(x, y \in P_1$ et $x \leq_{P_1} y)$ ou $(x, y \in P_2$ et $x \leq_{P_2} y)$ ou $(x \in P_1$ et $y \in P_2)$.

Un poset est dit *série-parallèle* s'il peut être construit à partir des singletons en utilisant seulement la somme directe et la somme linéaire.

Soit $Q_1 = \{x_1, x_2, ..., x_n\}$ un sous ensemble d'éléments maximaux de P_1 et $Q_2 = \{y_1, y_2, ..., y_n\}$ un sous ensemble d'éléments minimaux de P_2. La construction de *gluig* $P_1 * P_2(Q_1, Q_2)$ est le poset obtenu en identifiant les couples $(x_i, y_i), i = 1,...,n$ [9].

Un poset est dit *sans N* si et seulement si son diagramme de Hasse ne contient pas le poset N (Fig. 1) comme sous poset induit [9], c'est-à-dire s'il n'existe pas des sommets v_1, v_2, v_3 et v_4 tels que $v_1 \prec v_2 \succ v_3 \prec v_4$ et v_1 incomparable à v_4. La classe des posets sans N contient celle des posets séries-parallèles.

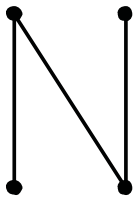


Fig 1

2 Principaux résultats

Soit *P* un poset sans N, *P* peut être construit à partir de deux posets en utilisant seulement la somme linéaire et la construction de gluig [6]: si *MinP* est l'ensemble des éléments minimaux de *P*, alors *P* contient un élément minimal *a* tel que $A(a) \subseteq MinP$ et *P* se décompose sous la forme :

$$P = (A(a) \oplus B(a)) * (P - A(a)) \ \ (B(a), B(a))$$

Où $B(a)$ est l'ensemble des couvertures supérieures de *a* et $A(a) = \{x \in P : \exists y \in B(a), x \prec y\}$.

A partir de cette définition, on peut déduire qu'un poset connexe *P* de rang *r* est sans N s'il est de *type 1* c'est-à-dire que *P* peut s'écrire comme somme linéaire de deux posets sans N, ou de *type 2* (resp.*type3*), s'il existe un entier $n \in IN^*$ tel que N_n est le premier niveau dont le sous poset induit par l'union consécutive des niveaux $N_n \cup ... \cup N_r$ noté $P_{n,r}$ est un poset non connexe de la forme $P_{n,r} = P_1 + P_2 + ... + P_l$, $l \in IN^*$ où $\forall i : P_i$ connexe, de type 1(resp. de type 1 ou 2).

Nous avons les résultats suivants :

Théorème 1 :

Soit P un poset sans N. Si P peut s'écrire comme somme linéaire de deux posets, alors H (P) a la propriété duale de König et on a :

$$\alpha(H(P)) = \rho(H(P)) = Max\{|MaxP|, |MinP|\}$$

Preuve :

Soit P un poset rangé sans N, $P = P_1 \oplus P_2$ où P_1 et P_2 deux posets rangés sans N.
Il est clair que : $MinP = MinP_1$ et $MaxP = MaxP_2$

On note par : $MinP_1 = \{a_1, a_2, ..., a_k\}$, $MaxP_1 = \{b_1, b_2, ..., b_{k'}\}$, $MinP_2 = \{c_1, c_2, ..., c_l\}$ et $MaxP_2 = \{d_1, d_2, ..., d_{l'}\}$.

On considère les familles des intervalles suivantes :

$$\mathcal{J}_1 = \begin{cases} \{[a_i, c_i], i = 1, ..., k\} \cup \{[a_k, c_j], j = k+1, ..., l\} & \text{si } k \le l \\ \{[a_i, c_i], i = 1, ..., l\} \cup \{[a_j, c_l], j = l+1, ..., k\} & \text{si } l < k \end{cases}$$

$$\mathcal{J}_2 = \begin{cases} \{[b_i, d_i], i = 1, ..., k'\} \cup \{[b_{k'}, d_j], j = k'+1, ..., l'\} & \text{si } k' \le l' \\ \{[b_i, d_i], i = 1, ..., l'\} \cup \{[b_j, d_{l'}], j = l'+1, ..., k'\} & \text{si } l' < k' \end{cases}$$

La famille $\mathcal{J}_1$ (resp. $\mathcal{J}_2$) représente bien une recouvrement par arêtes de l'hypergraphe $H(P_1 \oplus MinP_2)$ (resp. $H(MaxP_1 \oplus P_2)$).

Alors la famille des intervalles suivante :

$$\mathcal{J} = \begin{cases} \{[a_i, d_i], i = 1, ..., k\} \cup \{[a_k, d_j], j = k+1, ..., l'\} & \text{si } k \le l' \\ \{[a_i, d_i], i = 1, ..., l'\} \cup \{[a_j, d_{l'}], j = l'+1, ..., k\} & \text{si } l' < k \end{cases}$$

représente une famille de recouvrement par arêtes de l'hypergraphe $H(P)$.

Car tout intervalle $I_i = [a_i, d_i]$ de $\mathcal{J}$ est de la forme : $I_i = [a_i, b_i] \cup [b_i, c_i] \cup [c_i, d_i]$

Soit l'ensemble A suivant :

$$A = \begin{cases} MaxP & \text{si} \quad k \le l' \\ MinP & \text{si} \quad l' < k \end{cases}$$

Comme A est un ensemble stable de $\mathcal{H}(P)$ et de même cardinalité que $\mathcal{J}$, on obtient :

$$\alpha(\mathcal{H}(P)) = \rho(\mathcal{H}(P)) = Max\{|MaxP|, |MinP|\}$$

Pour les posets rangés sans N, nous avons

Théorème 2:

L'hypergraphe des intervalles d'un poset rangé sans N a la propriété duale de König.

Preuve :

Soit P un poset rangé de rang *r* sans N. On distingue 3 cas :

Cas 1 :

Si P est de type 1, d'après le théorème 1, on trouve :

$$\alpha(\mathcal{H}(P)) = \rho(\mathcal{H}(P)) = Max\{|MaxP|, |MinP|\}$$

Cas 2 :

Si P est de type 2 ; c'est-à-dire : $\exists n \in IN : P_{n,r} = P_1 + P_2 + ... + P_l$ où $\forall i \in L = \{1,...,l\} : P_i$ connexe, de type 1.

On aura besoin de ces quelques notations :

Notations :

1.a) On répartit N_{n-1} comme suit :

$$N_{n-1} = \left(\bigcup_{i \in I_1} R_{1i}\right) \cup \left(\bigcup_{j \in J_1} R'_{1j}\right)$$

Tels que :

- $\forall i \in I_1 ; R_{1i}$ forme un bloc avec K_{0i} un sous ensemble de $MinP_i$ (c'est-à-dire on trouve des sommes de la forme $R_{1i} \oplus K_{0i}$).

- $\forall j \in J_1; R'_{1j}$ forme un bloc avec K'_{0j} un sous ensemble de N_n. On dit que R'_{1j} relie les posets P_k, pour $k \in L$.

En plus, les ensembles K_{0i} et K'_{0j} forment une partition de $N_n, \forall i \in I_1, \forall j \in J_1$.

b) On note par α_{1j} le nombre des posets P_k qu'ils sont reliés dans N_{n-1} par l'ensemble R'_{1j}.

2.a) On répartit $N_{n-p}, \forall p = 2,...,n$ comme suit :

$$N_{n-p} = \left(\bigcup_{i \in I_p} R_{pi} \right) \cup \left(\bigcup_{j \in J_p} R'_{pj} \right)$$

Tels que :

- $\forall i \in I_p; R_{pi}$ forme un bloc avec $K_{(p-1)i}$ un sous ensemble de $N_{n-(p-1)}$ et qui ne relie aucun poset avec un autre.
- $\forall j \in J_p; R'_{pj}$ forme un bloc avec $K'_{(p-1)j}$ un sous ensemble de $N_{n-(p-1)}$ et qui relie les mêmes posets P_k , $k \in L$.

  Pour les ensembles R'_{pj} qui relient des posets non déjà reliés, on répartit $K'_{(p-1)j}$ comme suit :

$$K'_{(p-1)j} = \bigcup_{s \in S} B^s_{p-1,j}$$

Où : $B^s_{p-1,j}$ est un sous ensemble de $K'_{(p-1)j}$ qui forme un bloc avec R'_{pj} et qui ne relie aucun poset avec un autre.

En plus, les ensembles $K_{(p-1)i}$ et $K'_{(p-1)j}$ forment une partition de $N_{n-(p-1)}$
$\forall i \in I_{p-1}, \forall j \in J_{p-1}$.

b) On dit que P_k est un poset voisin de $P_{k'}$ s'il existe un ensemble R'_{pj} qui relie les deux.

Par récurrence, on montre que pour $p = 1,...,n$; $\alpha(H(P_{n-p,r})) = \rho(H(P_{n-p,r}))$.

A. Dans l'hypergraphe $H(P_{n-1,r})$:

On commence par la construction d'un ensemble stable maximale de $H(P_{n-1,r})$.

Soit $K^0 \subseteq L$ tel que $\forall k^0 \in K^0$; $\left| MaxP_{k^0} \right| \geq \left| \bigcup_{i \in I_1^{k0}} R_{1i} \right| + \left| \bigcup_{j \in J_1^{k0}} R'_{1j} \right|$.

Où R_{1i} et R'_{1j} forment des blocs avec des sous ensembles de $MinP_{k^0}$, $\forall i \in I_1^{k0}$, $\forall j \in J_1^{k0}$.

On note par $M_0 = \bigcup_{k^0 \in K^0} MaxP_{k^0}$

Soit l'ensemble $M_1 = \bigcup_{k^1 \in K^1} MaxP_{k^1}$, $K^1 \subseteq L$ tel que:

$\forall k^1 \in K^1 : P_{k^1}$ est un poset voisin de P_{k^0} pour $k^0 \in K^0$ et qui vérifie la propriété $(*)$ suivante:

$$\left|MaxP_{k^1}\right| \geq \sum_{i \in I_1^{k1}} \left|R_{1i}\right| + \sum_{j \in J_1^2} \left|R_{1j}^{'}\right| + \left|J_1^1\right| + \left|J_1^3\right|$$

Tels que :

- $\forall k^1 \in K^1$, R_{1i} et $R_{1j}^{'}$ forment des blocs avec des sous ensembles de $MinP_{k^1}$, $\forall i \in I_1^{k^1}$, $\forall j \in J_1^{k^1}$
- L'ensemble $\bigcup_{j \in J_1^{k^1}} R_{1j}^{'}$ est écrit sous forme:

$$\bigcup_{j \in J_1^{k^1}} R_{1j}^{'} = \left(\bigcup_{j \in J_1^1} R_{1j}^{'}\right) \cup \left(\bigcup_{j \in J_1^2} R_{1j}^{'}\right) \cup \left(\bigcup_{j \in J_1^3} R_{1j}^{'}\right)$$

Où :

- $\forall j \in J_1^1 : R_{1j}^{'}$ est l'ensemble qui relie le poset P_{k^0} avec le poset P_{k^1}, $k^0 \in K^0$ et $k^1 \in K^1$.
- $\forall j \in J_1^2 : \left|R_{1j}^{'}\right| \geq \alpha_{1j}$ et $\forall j \in J_1^3 : \left|R_{1j}^{'}\right| < \alpha_{1j}$.

De la même manière, on détermine les ensembles $M_t : M_t = \bigcup_{k^t \in K^t} MaxP_{k^t}$; $t = 2,..,T$

Où : $\forall k^t \in K^t : P_{k^t}$ est un poset voisin de $P_{k^{t-1}}$ pour $k^{t-1} \in K^{t-1}$ et qui vérifie la propriété $(*)$.

Soit l'ensemble suivant :

$$A_1 = \left(\bigcup_{i \in I_1^{'}} R_{1i}\right) \cup \left(\bigcup_{j \in J_1^{'}} A_{1j}\right) \cup \left(\bigcup_{j \in J_1^{"}} R_{1j}^{'}\right) \cup \left(\bigcup_{j \in J_1^{'"}} A_{1j}\right) \cup \left(\bigcup_{k \in K} MaxP_k\right).$$

Tels que :

- $\bigcup_{t=0}^{T} M_t = \bigcup_{k \in K} MaxP_k$
- $I_1^{'} = I_1 - \left(\bigcup_{k \in K} I_1^k\right)$.
- $\forall j \in J_1^{'}, R_{1j}^{'}$ est l'ensemble qui relie un poset P_k pour $k \in K$ avec un poset $P_{k^{'}}$ pour $k^{'} \notin K$.

  On prend $A_{1j} = \bigcup_{s \in S} \left\{b_{0,s}^j / b_{0,s}^j \in B_{0,s}^j\right\}$ où $B_{0,s}^j \not\subset MinP_k ; \forall k \in K$.

- $\forall j \in J_1^{"}, \left|R_{1j}^{'}\right| \geq \alpha_{1j}$.
- $\forall j \in J_1^{'''}, \left|R_{1j}^{'}\right| < \alpha_{1j}$, on prend $A_{1j} = \bigcup_{s \in S} \left\{ b_{0,s}^{j} / b_{0,s}^{j} \in B_{0,s}^{j} \right\}$.
- $\forall j \in J_1^{"} \cup J_1^{'''}; R_{1j}^{'}$ relie des posets $P_{k'}$ pour $k' \notin K$.

Par construction de A_1, A_1 est un stable maximale de H $\left(P_{n-1,r}\right)$, et sa cardinalité:

$$\left|A_1\right| = \alpha \text{ (H } (\left(P_{n-1,r}\right)) = \sum_{k \in K} \left|MaxP_k\right| + \sum_{i \in I_1^{'}} \left|R_{1i}\right| + \sum_{j \in J_1^{"} \cup J_1^{'''}} Max\left(\left|R_{1j}^{'}\right|, \alpha_{1j}\right) + \sum_{j \in J_1^{'}} \left|A_{1j}\right| .$$

D'un autre coté, on note par $\mathcal{J}^1$ la famille des intervalles de H $\left(P_{n-1,r}\right)$ formée par :

- $\mathcal{J}_1^1 = \bigcup_{k \in K} \left\{ \left\{ \left[r_m^k, a_m^k\right], m = 1,..., \beta \right\} \cup \left\{ \left[r_\beta^k, a_m^k\right], m = \beta + 1,.., \left|MaxP_k\right| \right\} \right\}$

Où : $a_m^k \in MaxP_k$, $r_m^k \in (\bigcup_{i \in I_1^k} R_{1i}) \cup (\bigcup_{j \in J_1^k} R_{1j}^{'})$ et $\beta = \left| \bigcup_{i \in I_1^k} R_{1i} \right| + \left| \bigcup_{j \in J_1^k} R_{1j}^{'} \right|$ où R_{1i} et $R_{1j}^{'}$ forment des blocs avec des sous ensembles de $MinP_k$.

- $\mathcal{J}_2^1 = \bigcup_{i \in I_1^{'}} \left\{ \left[r_m^i, a_m^i\right], m = 1,..., \left|R_{1i}\right| / r_m^i \in R_{1i} \text{ et } a_m^i \in MaxP_i \right\}$.
- $\mathcal{J}_3^1 = \bigcup_{j \in J_1^{"}} \left\{ \left[r_m^j, a_m^j\right], m = 1,..., \left|R_{1j}^{'}\right| / r_m^j \in R_{1j}^{'} \text{ et } a_m^j \in MaxP \right\}$.

- $\mathcal{J}_4^1$ la famille de tout les intervalles T_j d'extrémité initiale un élément de $R_{1j}^{'}$, $j \in J_1^{'}$ et qui passe par l'élément $b_{0,s}^{j}$, pour $s \in S$.
- $\mathcal{J}_5^1$ la famille de tout les intervalles T_j d'extrémité initiale un élément de $R_{1j}^{'}$, $j \in J_1^{"}$ et qui passe par l'élément $b_{0,s}^{j}$, pour $s \in S$.

Il est clair que la construction minimale de $\mathcal{J}^1$ est une recouvrement par arêtes de H $\left(P_{n-1,r}\right)$ dont tout intervalle de $\mathcal{J}^1$ contient exactement un élément de A_1.

On déduire que : α (H $\left(P_{n-1,r}\right)$) = ρ(H $\left(P_{n-1,r}\right)$).

Supposons que la propriété duale de König est vérifiée pour l'hypergraphe H $\left(P_{n-(p-1),r}\right)$.

B. Dans l'hypergraphe H $\left(P_{n-p,r}\right)$:

On détermine l'ensemble M : $M = \bigcup_{k \in K} MaxP_k$

Où P_k vérifie : $\left|MaxP_k\right| \geq \sum_{i \in I_p^k} \left|R_{pi}\right| + Max\left(\sum_{j \in J_p^k} \left|R_{pj}^{'}\right|, \sum_{j \in J_p^k} \left|C_{pj}\right| \right) \quad \forall k \in K$.

- $\forall i \in I_p^k; R_{pi}$ vérifie: $\forall x \in R_{pi}; x < y$ pour $y \in MaxP_k$.
- $\forall j \in J_p^k; R_{pj}^{'}$ est l'ensemble qui relie le poset P_k aux posets $P_l, l \in L$.

$$C_{pj} = \bigcup_{s \in S} \left\{ b_{p^{'},j}^{s} / b_{p^{'},j}^{s} \in B_{p^{'},j}^{s} \text{ et } b_{p^{'},j}^{s} \not< x ; \forall x \in MaxP_k \text{ et } p^{'} \in \{1,...,p\} \right\}$$

On détermine les $\bigcup_{j \in J_p^k} C_{pj}$ en appliquant la propriété $(**)$ suivante :

« $\forall a, b \in \bigcup_{j \in J_p^k} C_{pj}$; a et b ne sont pas dans le même intervalle » .

On répartit l'ensemble $\bigcup_{j \in J_p} R_{pj}^{'}$ comme suit :

$$\bigcup_{j \in J_p} R_{pj}^{'} = \left(\bigcup_{j \in J_p^1} U_{pj} \right) \cup \left(\bigcup_{j \in J_p^2} U_{pj} \right)$$

Tels que :

- $\forall j \in J_p^1 : U_{pj}$ est l'union des ensembles $R_{pj}^{'}$ qui relient les mêmes posets $P_{k^{'}}, k^{'} \notin K$.
- $\forall j \in J_p^2 : U_{pj}$ est l'union des ensembles $R_{pj}^{'}$ qui relient les mêmes posets $P_{k^{'}}, k^{'} \notin K$ avec au moins un poset $P_k, k \in K$.

Soit l'ensemble suivant:

$$A_p = \left(\bigcup_{k \in K} MaxP_k \right) \cup \left(\bigcup_{i \in I_p^{'}} R_{pi} \right) \cup \left(\bigcup_{j \in J_p^1} A_{pj} \right) \cup \left(\bigcup_{j \in J_p^2} A_{pj} \right)$$

Où :

- $I_p^{'} = I_p - \left(\bigcup_{k \in K} I_p^k \right)$
- $\forall j \in J_p^1 : A_{pj} = \begin{cases} U_{pj} & \text{si } \left|U_{pj}\right| \geq \left|C_{pj}^{'}\right| \\ C_{pj}^{'} & \text{si } \left|C_{pj}^{'}\right| > \left|U_{pj}\right| \end{cases}$

Où $C'_{pj} = \bigcup_{s\in S} \left\{ b^s_{p',j} / b^s_{p',j} \in B^s_{p',j}; p' \in \{1,...,p\} \right\}$ qui vérifie $(**)$ sur J^1_p

- $\forall j \in J^2_p : A_{pj} = C_{pj}$ qui vérifie $(**)$ sur J^2_p.

Par construction, A_p est un stable maximale dans l'hypergraphe H $(P_{n-p,r})$.

D'un autre coté :

On construit une famille des intervalles $\mathcal{J}^p$ en utilisant $\mathcal{J}^{p-1}$ de la manière suivante:

On prolonge tout intervalle de $\mathcal{J}^{p-1}$ par un intervalle au minimum qui contient un élément de A_p.

Il est clair que la construction minimale de $\mathcal{J}^p$ est une recouvrement de H $(P_{n-p,r})$ dont tout intervalle de $\mathcal{J}^p$ contient exactement un élément de A_p.

On déduire que : α (H $(P_{n-p,r})$) = ρ(H $(P_{n-p,r})$).

Cas 3 :

Si P est de type 3 ; c'est-à-dire : $\exists n \in IN : P_{n,r} = P_1 + P_2 + ... + P_l$ où $\forall i : P_i$ connexe, de type 1 ou 2.
Le poset $P_{n,r}$ peut s'écrire alors :

$$P_{n,r} = (\sum_{v\in V} P_v) + (\sum_{w\in W} P_w).$$

Où : $\forall v \in V; P_v$ est de type 1 et $\forall w \in W; P_w$ est de type 2.

C'est-à-dire : $\forall w \in W; \quad P_w = \sum_{s=1}^{l_w} P_s^w$

On aura : $P_{n,r} = \left(\sum_{v\in V} P_v \right) + \left(\sum_{w\in W} \left(\sum_{s=1}^{l_w} P_s^w \right) \right)$.

Par récurrence et avec les mêmes notations que le cas 2, on construit le même ensemble stable maximale pour montrer que α (H $(P_{n-p,r})$) = ρ(H $(P_{n-p,r})$), $\forall p = 1,...,n$.
Sauf dans ce cas on travail par tout les $|V| + \sum_{w\in W} |L_w|$ posets où $L_w = \{1,...,l_w\}, \forall w \in W$.

C'est-à-dire : $\forall p = 1,...,n$.

$$A_p = \left(\bigcup_{k \in K} MaxP_k \right) \cup \left(\bigcup_{i \in I_p^*} R_{pi} \right) \cup \left(\bigcup_{j \in J_p^1} A_{pj} \right) \cup \left(\bigcup_{j \in J_p^2} A_{pj} \right).$$

Où : P_k est un poset de type 1 ou un poset de la forme P_s^w, pour $w \in W$ et $s \in L_w$.

Références

[1] Bouchemakh. I. and Engel. K. : *Interval stability and interval covering property in finites posets.* Order 9, 163-175 (1992).

[2] I.Bouchemakh and K.Engel. *The order-interval hypergraph of a finite poset and the König property.* Discrete Math., 170:51-61 (1997).

[3] I.Bouchemakh. *On the chromatic number of order-interval hypergraphs.* Rostock. Math. Kolloq.,54 (2000).

[4] I.Bouchemakh. *On the König and dual König properties of the order interval hypergraphs of series-parallel posets.* Rostock. Math. Kolloq., 56 (2001).

[5] I.Bouchemakh. *On the dual König property of the order-interval hypergraph of a new class of posets.* Rostock. Math. Kolloq., 59 19-27 (2005).

[6] Duffin. R. J.: *Topology of series-parrallel networks.* J. Math. Analysis Appl. 10, 303-318 (1965).

[7] EL-Zahar. M.H. and Rival. R. : *Examples of jump-critical ordered sets.* SIAM J. Alg. Disc. Math. 6, 713-720 (1985).

[8] Engel. K. : *Interval packing and covering in the Boolean lattice.* Comb. Probab. Comput. 5, N°4, 373-384(1996).

[9] Rival. I. and Zaguia. N. : *Constructing N-free, jump-critical ordered sets.* Congressus Numerantium 55, 199-204 (1986).

[10] Rival. I. : *Optimal linear extensions by interchanging chains.* Proc. Amer. Soc. 89, 387-394 (1982).

Un Système Interactif d'Aide à la Décision de Groupe En Aménagement du Territoire: Couplage SMA-SIG

Sarah Oufella, Djamila Hamdadou, Karim Bouamrane

Département d'informatique, Faculté de Sciences, Université d'Oran Es-Senia,*BP 1524, El-M'Naouer, 31000, Oran, Algérie*

{osarah84,dzhamdadoud, ,kbouamranedz }@yahoo.fr

Résumé. Les problèmes territoriaux, par leur nature complexe à caractère spatial, nécessitent la définition de plusieurs critères et font intervenir plusieurs acteurs aux intérêts conflictuels, dont les différents points de vue doivent être pris en compte pour la décision publique. Dans cette optique, il nous a semblé intéressant de concevoir un modèle décisionnel basé sur un couplage SMA- SIG susceptible d'apporter une aide aux décideurs du territoire, en exploitant simultanément les avantages qu'offrent les SMA très adaptés pour modéliser des entités complexes pouvant coopérer, collaborer ou négocier et ceux des SIG caractérisés par leur capacité de gestion, d'analyse, et d'affichage de données à référence spatiale.

Mots clé : Aide à la Décision de Groupe, Système d'Information Géographique (SIG), Aménagement du Territoire, (AT), Système multi Agents (SMA), Négociation, ELECTRE III, Diagramme UML.

1 Introduction

Les problèmes territoriaux (spatiaux) sont de nature complexe nécessitant la définition de plusieurs critères conflictuels dont l'importance n'est pas la même, et manipulant une quantité considérable de données quantitatives et/ou qualitatives. Ce type de problématique fait intervenir plusieurs acteurs aux intérêts conflictuels, cette multiplicité d'objectifs découle directement de la nature multidimensionnelle des problèmes spatiaux [6]. En effet, la même étendue spatiale est perçue, différemment par un environnementaliste, un politicien, un économiste, etc. Chacun de ces intervenants détient une perception différente de l'espace selon ses objectifs et ses préoccupations, de ce fait il est indispensable d'apporter une aide efficace aux décideurs du territoire pour une performance globale du projet urbain. Dans cet article, on s'intéresse à l'élaboration d'un modèle décisionnel dédié aux problématiques de localisation en AT et basé sur un couplage SMA-SIG. La problématique abordée ainsi que notre contribution sont décrites en section 2. Par la suite, en section 3 et 4, nous présentons successivement les systèmes d'information géographique (SIG) ainsi que les systèmes multi agents (SMA).

Le modèle proposé est présenté, en détails, en section 5 et est accompagné d'une étude de cas discutée en section 6. Enfin la section 7 présente la conclusion de cette étude et propose des travaux futurs.

2 Problématique et Contribution

L'AT est un domaine vaste, dont les problématiques sont nombreuses et diverses, nous nous intéressons, dans cette étude, plus particulièrement à la problématique de localisation qui consiste en la recherche d'une surface satisfaisant au mieux certains critères pour une construction donnée[6] .Les problèmes décisionnels territoriaux sont à caractère spatial de nature multidimensionnelle, interdisciplinaires et mal définis, nécessitant la définition de plusieurs critères souvent conflictuels dont l'importance n'est pas la même, impliquent plusieurs personnes et institutions, ayant généralement des préférences et des objectifs divergeant, cela implique que le processus de décision est distribué entre les différentes entités impliquées et impactées par cette décision de groupe. La résolution de ce problème consiste alors à trouver une décision commune à tous les décideurs Notre contribution porte sur la proposition d'une approche originale pour le développement d'outils d'aide à la décision de groupe pour la conduite des projets urbains et traitant, principalement, la problématique de localisation en AT. Cette approche est basée sur l'utilisation d'un système d'information géographique couplé à un système multi agents doté d'un protocole de négociation afin d'aboutir à un consensus qui satisfait les acteurs territoriaux. L'outil suggéré permet de prendre en compte à la fois la dimension spatiale et les intérêts spécifiques et divergents des différentes parties prenantes.

3 Les Systèmes d'information Géographique

A partir du milieu des années 90, les décideurs ont commencé à utiliser les SIG comme des outils d'aide à la décision dans le but de mettre en évidence des faits spécialisés, de chercher des solutions à des problématiques, de réaliser des analyses et de comparer des scénarios [7]. Les SIG diffèrent selon leurs domaines d'applications et les demandes qu'ils doivent satisfaire. Toutefois, ils ont en commun des fonctionnalités nommées les « 5A » [7]: Abstraction, Acquisition, Archivage, Affichage et Analyse.

-L'Abstraction : c'est la modélisation des données géographiques et de leurs spécifications afin de représenter le monde réel. Ces représentations cherchent à reproduire le plus fidèlement possible la réalité d'une manière compréhensible par les utilisateurs et utilisable informatiquement dans le but de répondre à des objectifs donnés.

-L'Acquisition des données : concerne la récupération de l'information existante (données qui peuvent provenir de fournisseurs extérieurs, de numérisation directe ou de traitements particuliers comme des images satellites par exemple) et son intégration dans le système

-L'Archivage : c'est la fonctionnalité qui permet le stockage des données de façon à les retrouver et les utiliser facilement par des applications variées.
-L'Analyse : concerne la manipulation et l'interrogation des données géographiques. Elle est considérée comme étant le cœur du SIG.
-L'Affichage : vise à assurer la fonction de visualisation et de mise en forme dans le SIG. Cette opération est très importante car elle assure la convivialité et l'ergonomie des applications. Un SIG performant doit permettre la visualisation rapide et directe du résultat d'un traitement, ainsi que la visualisation intelligente de certains attributs.

4 Les Systèmes Multi Agents (SMA)

Un système multi agents (SMA) est généralement défini comme étant un ensemble d'agents, en interaction les uns avec les autres, pouvant coopérer négocier ou collaborer [4]. Ils évoluent dans un environnement qu'ils perçoivent, dans lequel ils peuvent se déplacer et qu'ils peuvent modifier. De la notion de système multi agent, se dégage immédiatement l'idée d'un système constitué de plusieurs agents, le concept d'agent reste donc le pivot de ce domaine. En effet, les agents sont des entités physiques (capteurs, processeurs, etc.) ou abstraites (tâches à réaliser, déplacements, etc.), qui sont capables d'agir sur leur environnement et sur elles-mêmes, c'est-à-dire de modifier leur propre comportement. Elles disposent, pour se faire, d'une représentation partielle de cet environnement et de moyens de perception et de communication [2].

5 Approche proposée

L'originalité de notre approche tient à l'utilisation simultanée d'un système multi agents (SMA) et d'un système d'information géographique (SIG). La littérature offre peu d'exemples de couplage de ces deux types de représentations de la réalité.
Les SMA sont très adaptés pour modéliser les phénomènes dans lesquels les interactions entre diverses entités sont assez complexes pour être appréhendées par les outils de modélisation classiques. Ils sont, de plus en plus, utilisés dans les problèmes de gestion de l'environnement et d'AT car ils permettent de représenter des entités autonomes, dotées de comportements, pouvant coopérer, négocier et communiquer avec les autres.
Les SIG sont de plus en plus mis en avant dans les projets d'AT , ils offrent la possibilité de saisir, stocker, gérer, visualiser et manipuler des informations sur des objets géo référencés, ils offrent également de nombreux outils d'analyse spatiale. Par soucis de simplicité, nous choisissons un "couplage lâche" entre les modules SMA et SIG qui restent indépendants et communiquent uniquement en s'échangeant des données. Ainsi, les fonctionnalités des SMA et des SIG sont bien distinctes. Le choix d'un "couplage étroit" aurait nécessité l'insertion des fonctionnalités SMA au sein du SIG et des fonctionnalités SIG au sein du SMA, dupliquant de fait une partie du code et rendant l'évolution du système très lourde. Le modèle décisionnel proposé (SMAG) est composé de deux modules : un module SIG et un module SMA doté d'un

protocole de négociation. L'architecture du modèle SMAG est illustrée par la figure (Fig.1).

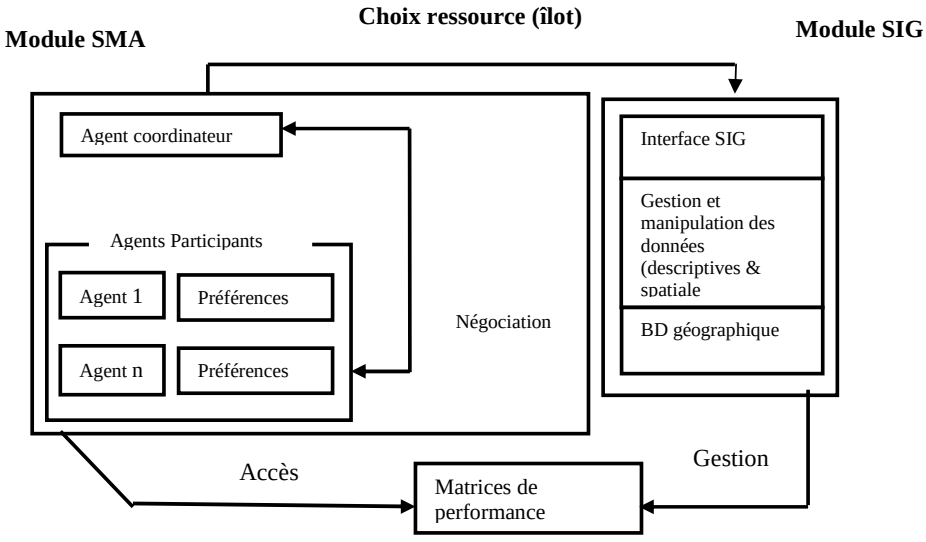


Fig.1. Le modèle décisionnel proposé (SMAG)

5.1 Le module SIG

Le SIG aura pour fonction essentielle de permettre la gestion des connaissances du territoire. Grâce à ses fonctionnalités, il est possible de :
- Gérer la base de données géographiques ;
- Archiver des informations ;
- Manipuler et interroger les bases des données géographiques ;
- Fournir une représentation spatiale des systèmes étudiés ;
- Mettre en forme et visualiser les données.

Nous exploitons les fonctionnalités du SIG pour préparer les entrées (inputs) nécessaires pour la prise de décision. Lorsque les décideurs parviennent à identifier les actions et les critères, grâce aux capacités analytiques du SIG, une valeur (note) est affectée à chaque critère. L'ensemble des actions et de leurs notes relativement aux différents critères constitue la matrice d`évaluation (Tableau de performances). Cette matrice est gérée par le SIG.

5.2 Le module SMA

La technologie Multi Agents à déjà fait ses preuves dans de nombreux domaines par leur capacité de modélisation, ils permettent de représenter les interactions entre diverses entités pouvant coopérer négocier et communiquer.

Les intervenants du système, que nous étudions, sont les différents décideurs ou experts qui disposent de leurs propres objectifs .Cela implique que le processus de décision est distribué entre les différentes entités impliquées et impactées par cette décision de groupe. Le module SMA aura pour mission de représenter les différents acteurs qui disposent de leurs propres objectifs et préférences. Afin de faire face à cette décision de groupe où différents points de vue doivent être pris en considération, il est indispensable de passer par une phase de négociation pour arriver à un consensus bénéfique aux groupes. A cet effet, nous dotons le module SMA d'un protocole de négociation basé sur la médiation, mettant en scène un agent coordinateur et un ensemble d'agents participants représentant les différents acteurs concernés par la décision en AT.

5.2.1 La Modélisation des agents

L'agentification d'un problème est un aspect important de la conception d'un SMA, elle influence fortement les performances et l'efficacité du Système à résoudre un problème. Dans la littérature, il existe une multitude de méthodologies offrant un intérêt certain pour l'étude des SMA d'un point de vue organisationnel telle que Gaia, Voyelles, Ingenias, Aalaadin, etc. Le rôle de ces méthodologies est d'assurer une aide efficace durant toutes les phases de modélisation du cycle de vie d'une application basée sur les agents.

Notre modélisation agent se base sur la méthodologie Aalaadin [4], qui s'appuie sur les concepts d'agent, groupe et rôle pour définir une organisation réelle.

-Un agent est défini comme étant une entité autonome et communicante jouant des rôles au sein de différents groupes ;

- Un groupe est composé de différents agents ;

-Un rôle représente une fonction, un service ou une identification d'un agent appartenant à un groupe particulier. Les rôles des agents, dans le cadre de nos travaux, sont de deux types : coordinateur et participants ;

- L'agent coordinateur : est responsable de la création de tous les agents participants concernés par la décision en AT , du bon déroulement de la négociation ainsi que du choix finale concernant la ressource(îlot) élue.

- Les agents participants : sont les agents concernés par la décision en AT chacun de ces agents a ces propres préférences et objectifs concernant les ressources (îlots), le but de chacun des ces agents et que sa ressource (îlot) préférée soit choisie lors de la décision finale.

5.2.2 Détermination des préférences des agents participants

A fin de représenter les préférences des agents participants nous optons pour l'utilisation de la méthode multicritère **ELECTRE III** qui relève de la problématique γ (procédure de classement) [11]: son but est de classer les actions (ressources) potentielles, depuis les "meilleures" jusqu'aux "moins bonnes ». Pour se faire, ELECTRE III traite une matrice d'évaluation contenant des actions et des pseudo critères. Les traitements de surclassement munis sur cette matrice permettront d'établir un préordre final partiel [9]. ELECTRE III a apporté des évolutions remarquables par rapport aux autres méthodes appartenant a la famille (ELECTRE) notamment l'introduction de la notion de préférence faible : zone intermédiaire où le décideur hésite entre la préférence et l'indifférence. Ceci est assuré à travers

l'utilisation de deux seuils : seuil d'indifférence et seuil de préférence stricte. Ces seuils ont été définis de manière à tenir compte directement de l'incertitude qui entache plus ou moins les valeurs de la matrice des évaluations. Aussi, un troisième seuil, le seuil de veto, est utilisé pour la concrétisation de la notion de discordance.
Grâce à ELECTRE III, chacun des agents participants va pouvoir exprimer ses préférences concernant les ressources (îlots) et construire ce qu'on appelle un vecteur de préférence où il va classer les ressources (îlots) de la meilleure à la moins bonne. Cependant pour prendre une décision publique, il est indispensable de passer par une phase de négociation.

5.2.3 Phase de négociation

La phase de négociation va permettre de trouver un accord commun qui satisfait la plus part des agents , pour cela nous adoptons un protocole de négociation multilatéral basé sur la médiation mettant en scène l'agent coordinateur et l'ensemble d'agents participants.

5.2.3.1Le protocole de négociation proposé

Le protocole de négociation que nous proposons se caractérise par une suite de messages échangés entre un agent coordinateur et des agents participants. Il s'inspire largement du Contrat Net Protocol [3] qui est pratiquement le protocole de négociation le plus utilisé dans les SMA. Dans ce qui suit, nous décrivons en détail les différentes caractéristiques du protocole de négociation que nous proposons[10].

- Les ressources de la négociation

Les ressources sont les objets de la négociation Dans notre cas, ce sont des ressources communes (les îlots vierges destinés pour une construction donnée).

- Le seuil d'acceptation : représente le nombre de réponses positifs pour que la négociation soit un succès.

- Les primitives de négociation

Pour mener à terme un processus de négociation entre agents, il est nécessaire de définir des primitives spécifiques au coordinateur et d'autres primitives spécifiques aux participants.

a). Les primitives du coordinateur

Les messages envoyés par le coordinateur sont destinés à tous les agents participants nous lui associons, par conséquent, trois primitives de négociation :

- Request () : l'agent coordinateur envoie un message aux participants pour leur indiquer le début de la négociation chacun des agents doit associer à chaque ressource de son vecteur de préférence un rang, la ressource classée première au niveau de chaque participant aura le rang le plus grand (elle représentera la préférence du participant lors du premier tour) ce rang est a chaque fois décrémenté de 1 pour les ressources suivantes

- Propose () : l'agent coordinateur propose un contrat aux agents participants concernant une ressource (îlot) donnée ;

- Confirm () : l'agent coordinateur envoie un message à tous les agents pour les informer que la négociation a été un succès et que la ressource (îlot) a été trouvé ;

b). Les primitives du participant

Les messages envoyés par les participants sont uniquement destinés à l'agent coordinateur. L'agent participant dispose de trois primitives de négociation :

-Inform () : les agents participant indiquent à l'agent coordinateur qu'il peut leur faire une première proposition ;
- Accept () : ce message répond à la proposition du contrat faite par le coordinateur. Le participant indique par ce message au coordinateur qu'il accepte le contrat ;
- Refuse () : le participant indique au coordinateur qu'il refuse sa proposition.
Afin de représenter les différentes interactions entre l'agent coordinateur et les agents participants, nous optons pour l'utilisation du diagramme de séquence d'UML, très souvent employé pour décrire l'interaction des agents [1]. La figure suivante (Fig.2) représente les différentes primitives associées à l'agent cordinateur et participantes via un diagramme UML.

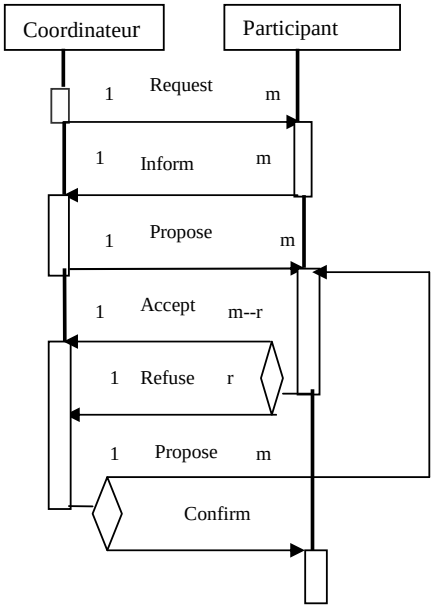


Fig. 2. Diagramme de séquence du protocole de négociation proposé

-Les stratégies des agents
La négociation se déroule en plusieurs tours jusqu'à ce qu'un compromis qui satisfait la majorité des agents soit trouvé. Le coordinateur fait une proposition aux participants concernant une ressource (îlot) donné ces derniers vont soit accepter soit refuser la proposition. La stratégie du coordinateur lui permet de modifier sa proposition dans le cas où les participants n'ont pas été assez nombreux à l'accepter tandis que la stratégie associée aux participants leur permet d'accepter la proposition du coordinateur ou de la refuser

a) Stratégies du participant
La négociation peut se dérouler en plusieurs tours, jusqu'à ce que un compromis soit trouvé, à chaque nouveau tour le participant reçoit une nouvelle proposition, il

consulte alors son vecteur de préférences, si la proposition correspond à sa préférence au tour t, il l'accepte. Si non il vérifie si la proposition correspond à l'une de ses préférences antérieures, si c'est le cas, il accepte la proposition tout en indiquant sa préférence actuelle.
Lorsque le participant reçoit une proposition et que celle-ci ne correspond ni à sa préférence au tour t, ni à aucune de ses préférences antérieures, il la refuse et fait une contre proposition qui correspond à sa préférence au tour t.

b) Stratégie du coordinateur

On associe à l'agent coordinateur une seule stratégie qu'il utilisera lors de la phase de négociation, si les agents participants n'ont pas été assez nombreux à accepter sa proposition, il est obligé de la modifier son contrat pour le prochain tour et ceci en s'inspirant des réponses des participants au tour précédent, afin de trouver une nouvelle possibilité pour le prochain tour. Pour cela, il associe un score à chaque ressource en prenant en compte le poids de l'agent participant ainsi que le rang de la ressource**.** Pour calculer le score de chaque ressource lors d'un certain tour t, nous avons utilisé la formule suivante :

$$SCORE(Ri)=\sum_{J=1}^{N} POID(participant[j])*RANG(Ri,participant[j]) \quad (\mathbf{1})$$

Tel que :

POID(participant[j]) **:** A chaque participant est associé un poids différent puisque dans la réalité, les représentants politiques, par exemple, n'ont pas du tout le même poids que les associations de protection de l'environnement lors d'une décision en AT.

RANG(Ri, participant[j]) : Le rang associé à la ressource par le participant j dans son vecteur de préférence

Comme dans les méthode de scorages [8], la ressource qui a obtenu le score le plus élevé lors du tour t, sera la ressource gagnante et l'initiateur la proposera dans le prochain tour. Ce score est remis à jour à chaque fois que les participants n'ont pas été assez nombreux à accepter la proposition.

6. Etude de Cas

Le développement d'un module multi agents est un problème complexe, donc il est préférable d'utiliser une plateforme multi agents existante que nous adaptons à nos besoins .Notre choix s'est porté sur la plate forme **JADE** [5] pour servir de base au module multi agents. Cette plateforme de développement est gratuite, implémentée en java, son code source et celui de son environnement de développement sont ouverts et modifiables permettant ainsi de l'utiliser selon nos besoins.

Pour le développement du module SIG, notre choix s'est porté sur le logiciel **MAPINFO :** C'est un outil de type Système d'Information Géographique. Il a permis, dans notre étude, de visualiser et de modifier les différentes bases de données géographiques utilisées selon le besoin. Cependant, les deux logiciels demeurent

indépendants et les modules SMA et SIG communiquent, ainsi, entre eux par l'intermédiaire des données.

- Délimitation de la région d'étude

La région d'étude consiste en un ensemble de sites situés à l'est de la ville d'Oran. Le choix de cette région est dû, principalement, à sa multitude de projets d'aménagement. La région choisie pour réaliser notre étude est le quartier d' El Hamar situé dans la commune de Gdyel, wilaya d'Oran. Il s'avère qu'il y a absence d'un secteur sanitaire, activité essentielle pour répondre aux besoins des habitants en matière d'urgence et de premiers soins. Ainsi, notre action portera sur le choix d'un site parmi plusieurs pour l'implantation d'un secteur sanitaire. Nous disposons de six (06) îlots vierges pouvant convenir pour notre construction. Cependant, chacun des participants a ses propres préférences concernant ces îlots. Afin de pouvoir visualiser les îlots vides, nous exploitons les diverses avantages qu'offrent les SIG [6] en terme d'affichage (Fig.3).

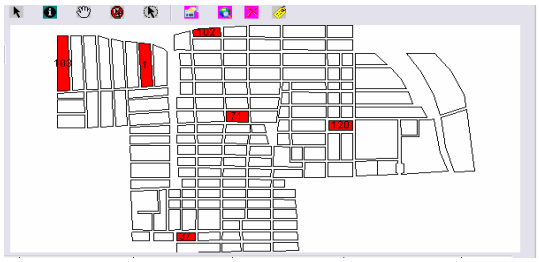


Fig .3. Affichage des îlots vides

Puis nous utilisons les autres fonctionnalités du SIG pour préparer les entrées (*inputs*) nécessaires pour la prise de décision. L'ensemble des actions et de leurs notes relativement aux différents critères (Tableau de performances). Cette matrice de performance est gérée par le SIG. Nous avons pu identifier les critères suivants :

- Nombre de population avoisinant aux actions (C1) ;
- Eloignement par rapport au site industriel (C2);
- Nuisance sonore (C3) ;
- Proximité au réseau d'assainissement (C4) ;
- Proximité au réseau de la moyenne tension (C5)

La définition ainsi que l'évaluation des critères identifiés permettent d'élaborer la matrice des performances, illustrée par la Table1 Cette matrice est gérée par le SIG.

Id_Ilots	Nombre_de_Population	Eloignement_site_indu	Nuisance_Sonore	Proximité_réseau_assainiss	Proximit
1	2670	820	8	10	30
37	1145	710	14	30	460
71	3510	530	24	110	230
102	2180	700	6	20	1
103	1450	1040	4	20	1
120	1145	240	18	40	570

Table1 Le tableau d'évaluation (matrice de performance)

Les participants concernés par la décision en AT sont les associations d'environnement, les politiciens, les économistes et le public. Chacun de ces acteurs est représenté par un agent. La génération des agents est réalisée à l'aide de la plateforme SMA **JADE** (Fig.4).

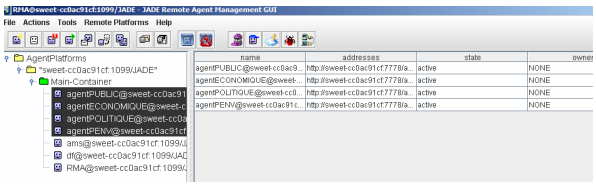


Fig. 4. Création de l'agent initiateur et des agents participants

On associe, à chacun de ces agents participants, un poids pour exprimer son importance lors du déroulement de la négociation.

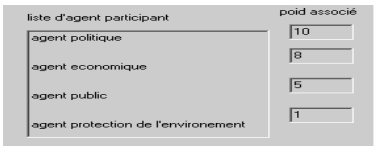


Fig. 5. Liste des agents participants

Chacun des participants va établir son vecteur de préférence où il classe les ressources de la meilleure à la moins bonne et ceci en se basant sur les cinq critères du tableau d'évaluation. Pour parvenir à cet objectif, il utilise la méthode multicritère ELECTRE III [9] (Fig.6).

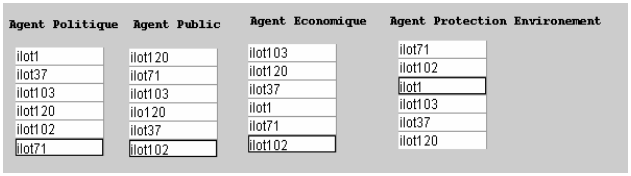


Fig. 6. Etablissement des préférences de chaque participant

3.5.3 Simulation de la négociation

Avant de lancer la négociation, il est indispensable de définir le seuil d'acceptation (le nombre d'accords nécessaires pour l'acceptation d'un contrat). Dans notre étude, il est fixé à 70%. Il est possible de visualiser les différents messages échangés entre l'agent coordinateur et les agents participants. Dés que les agents participants reçoivent le message **Confirm**, synonyme de la fin de la négociation, la ressource

finale a bien été trouvée. Les différents messages échangés lors de la phase de négociation sont présentés par la figure (Fig.7)

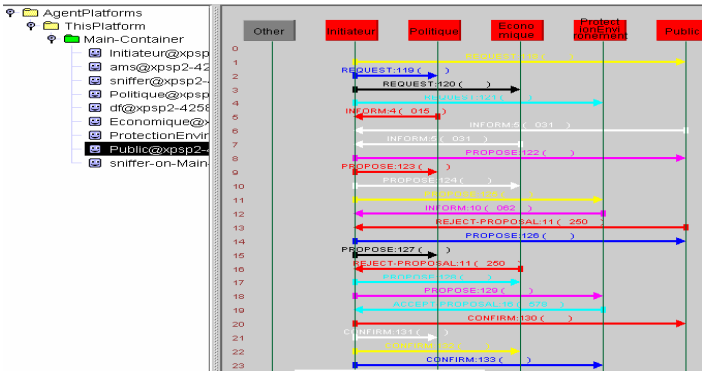


Fig. 7. Visualisation des messages échangés lors de la négociation

Après plusieurs modifications du contrat et au bout du troisième tour, les participants arrivent à un consensus, la ressource choisie est la ressource (ilot103) Avec un taux d'acceptation de 79% (Fig.8).

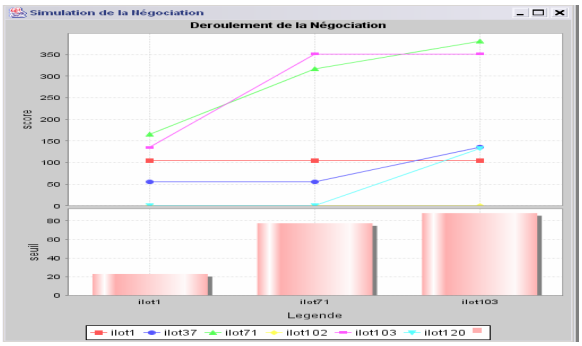


Fig. 8. Choix de la ressource (îlot 103) après négociation

La ressource choisie (îlot 103) après négociation est acceptée par la majorité écrasante des participants et est visualisée grâce aux SIG illustrée par la figure (Fig.9).

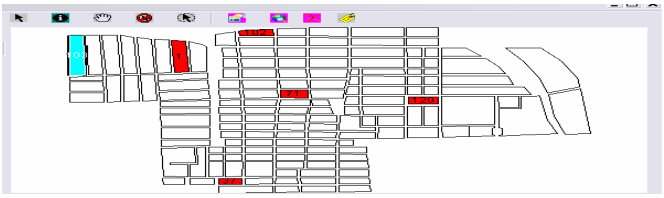


Fig. 9. Visualisation de la ressource (îlot) gagnante

7 Conclusion

Les problèmes territoriaux, par leur nature complexe, nécessitent la définition de plusieurs critères et font intervenir plusieurs acteurs aux intérêts conflictuels, dont les différents points de vue doivent être pris en compte pour la décision publique. A cet effet, nous avons proposé un modèle décisionnel basé sur un couplage SMA- SIG susceptible d'apporter une aide aux décideurs du territoire, en exploitant simultanément les avantages qu'offrent les SMA très adaptés pour modéliser des entités complexes pouvant coopérer, collaborer ou négocier et ceux des SIG caractérisés par leur capacité de gestion, d'analyse, et d'affichage de données à référence spatiale. Le modèle proposé permet de :

- Représenter le territoire grâce aux fonctionnalités du SIG ;
- Représenter la multiplicité et la diversité des acteurs grâce aux capacités des SMA ;
- Représenter les préférences de chacun des participant grâce aux avantages qu'offre la méthode multicritère ELECTRE III.
- Prendre une décision collective en se basant sur une stratégie de négociation.

Pour cela, nous avons doté le module SMA d'un protocole de négociation multilatéral basé sur la médiation qui met en scène deux types d'agents : un agent coordinateur et un ensemble d'agents participants négociants. Le protocole de négociation proposé offre aux participants la possibilité d'exprimer leurs préférences et de formuler des contre-propositions pour converger vers une solution plus rapidement aboutissant à un accord acceptable au regard des contraintes et des préférences de chacun. Dans nos travaux futurs, nous prévoyons l'enrichissement de notre architecture (SMA-SIG) à laquelle nous ajouterons de nouveaux modules et de nouvelles classes qui permettront de modéliser plus facilement les systèmes réels et de développer d'autres systèmes informatiques d'aide à la décision.

References

1. B.Bauer, P.Müller, J.Odell, « Agent UML: A formalism for specifying multiagent interaction»,International Journal of Software Engineering and Knowledge Engineering, 2001
2. B.Chaib-Draa "Hierarchical models and communication in multi-agent environments", dans Proceedings of the 6th European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds, p. 119-134, Odense, Danemark,1994
3. R.Davis, R.Smith, «Negotiation as a Metaphor for Distributed Problem Solving», Reading in Distributed Artificial Intelligence, 1980.
4. J.Ferber, O.Gutknecht, «A meta-model for the analysis and design of organizations in multi agent systems», Third International Conference on Multi-Agent Systems, ICMAS, 1998
5. Fipa, «Fipa interaction protocol specifications. Technical report, Foundation for Intelligent Physical Agents», 2002.
6. D.Hamdadoud, KBouamrane, «Multicirterion SDSS for the espace Control: Towards a Hybrid Approach», MICAI 2007: Advances in Artifical Intelligence,LNCS,Springer ISSB 0302-9743,2007
7. E.Maillé., "Du couplage de systèmes à l'intégration spatio-temporelle dans les systèmes d'aide à la décision spatiale" dans CABM-HEMA-SMAGET05, Bourg Saint–Maurice, France, 2005.
8. T.Marchant : «Agrégation de relation valuées par la méthode borda en vue d'un rangement axiomatique », Thèse Doctorat Université libre de Bruxelles, 1997.
9. L.Maystre, J.Pictet, J.Simos «Méthodes multicritères Electre », Presses Polytechniques et uiversitaires Romandes, Lausanne, Suisse, 1994.
10. S.Oufella, D.Hamdadou, KBouamrane «Proposition d'un modèle d'aide à la négociation pour les problèmes d'aménagement du territoire», Première journée scientifique sur l'informatique et ces applications, Guelma, Algérie, 2009.
11. B.Roy, «Electre III, un algorithme de classement fondé sur une représentation floue des préférences en présence de critères multiples», rapport de recherche, 1977

Extracting Conceptual Schema From Domain Ontology: A Web Application Reverse-Engineering Approach

Sidi Mohamed Benslimane, Mimoun Malki, and Djelloul Bouchiha

Computer Science Department, University of Sidi Bel Abbes
BP 89, Sidi Bel Abbes, 22000, Algeria
{Benslimane,Malki,Bouchiha}@univ-sba.dz

Abstract. The heterogeneous and dynamic nature of components making up a Web application, the lack of effective programming mechanisms for implementing basic software engineering principles in it, and undisciplined development processes induced by the high pressure of a very short time-to-market, make Web application maintenance a challenging problem. A relevant issue consists of reusing the methodological and technological experience in the sector of traditional software maintenance, and exploring the opportunity of using reverse engineering to support effective Web application maintenance. This paper presents reverse engineering approach that help to understand existing undocumented Web applications to be maintained or evolved, through the extraction from domain ontology of conceptual schema describing a Web application. The advantage of using ontology for conceptual data modelling is the reusability of domain knowledge.

Key words: Reverse-engineering, Web application, Ontology, Conceptual schema, Conceptual modeling.

1 Introduction

Web Applications have become one of the most important means of communication for commercial enterprises of all kinds. They provide the underlying engines that not only improve a company's image, but also act as a useful resources for increasing a company's overall market share. However, the heterogeneous and dynamic nature of components making up a Web application, the lack of effective programming mechanisms for implementing basic software engineering principles in it, and undisciplined development processes induced by the high pressure of a very short time-to-market, make Web application maintenance a challenging problem. A relevant issue consists of reusing the methodological and technological experience in the sector of traditional software maintenance, and exploring the opportunity of using reverse engineering to support effective Web application maintenance [1]. To reconstruct already existing Web applications that do not respect the development life cycle, the reverse engineering is essential. The reverse engineering of Web application has been addressed in various

ways. Some research works take an interest in the evolution of the presentation [2] while others focus on restructuring HTML static pages into dynamic ones [3]. Recently, some approaches consider the ontology creation as the goal for Web application reverse engineering [4–6].

This paper deals with a new approach to that development, consisting on analyzing HTML Web pages to derive a first cut version of the conceptual schema modelling the Web application based on domain ontology. The important requirement for developing conceptual data models is to reduce efforts, costs and time.

The remainder of the paper is as follows. In the next section, we briefly review related researches. In section 3 we explain the overall of our reverse engineering approach. Section 4 presents a portal prototype implementation of the proposed approach and details some experimental result. Finally, some discussions and conclusion are respectively presented in sections 5 and 6.

2 Our Approach

In this section, we describe how conceptual schema modeling a Web application, can be derived from domain ontology using useful information extracted from HTML pages. The approach consists of four phases. The following paragraphs describe each of these phases.

2.1 Extraction of candidate elements

The extraction phase aims to retrieve the pertinent elements coded on each Web application's HTML page. It is performed in four steps:

Pre-processing. This step takes as input HTML pages, corrects them, proceed to some cleaning by removing stop words and eliminating useless tags such as those of layout (e.g., `<b>, <i>`), and preserving useful tags, which carry information to be processed in the following stages (e.g., `<form>, <table>, <td>`, `<tr>, <ul>, <li>`). The result of this step is a coded sequence describing the structure of the HTML page.

DOM construction. The second step permits the generation of the DOM (Document Object Model) [7] representations of cleaned HTML pages in order to facilitate their manipulation. DOM representations describe the physical views of the Web application HTML pages (one physical view per HTML page).

Candidate elements identification. In the third step, DOM representations are parsed to obtain a set of elements. An element is either: forms, tables, or lists. Each element has a name and a set of attributes (fields).

Morphological analysis. In the last step, a morphological analysis is applied to the obtained elements and their attributes. It consists in performing word stemming (lemmatization). Auxiliary information like stop words list, English lexicon (WordNet in this particular case) are used to perform the necessary linguistic transformation (e.g., morphological analysis of 'running-away' is 'run away').

2.2 Ontological constructs identification

During this phase a set of ontological constructs are detected while matching candidate elements to the constructs of domain ontology. The matching aims to quantify how much two entities are alike by calculating semantic distance between them.

Matching strategies The semantic distance calculation in our approach is based on different similarity measures. The matching is achieved at three levels: name-based matching, lexical-based matching, and structure-based matching.

a. Name-based matching. name-based matching is to compare elements with equal names. At this level string similarity $SimN$ measure is used based on edit distance formulated as:

$$SimN(e_1, e_2) = max(0, \frac{min(|e_1|, |e_2|) - ed(e_1, e_2)}{min(|e_1|, |e_2|)}) \in [0, 1]$$

Where ed is the edit distance formulated by Levenshtein [8]. It measures the minimum number of token insertions, deletions, and substitutions required to transform a string e_1 into another string e_2.

b. Lexical-based matching. For two entities e_1 and e_2, the lexical semantics similarity measure $SimL$ can be given using the WordNet synsets (i.e. term for a sense or a meaning by a group of synonyms) based on the formula:

$$SimL(e_1, e_2) = 1/length(e_1, e_2)$$

Where $length$ is the length of the shortest path between two entities e_1 and e_2 using node-counting.

c. Structure-based matching. At this level, the attributes of elements are matched according to a strategy of calculation. If the attributes of two elements are equal, the elements are also equal.

$$SimS(E, F) = \frac{\sum_{e \in E} \boldsymbol{e}}{|\sum_{e \in E} \boldsymbol{e}|} \times \frac{\sum_{f \in F} \boldsymbol{f}}{|\sum_{f \in F} \boldsymbol{f}|}$$

With entity set $E = \{e_1, e_2, ...\}$,
$\boldsymbol{e} = (sim(e, e_1), sim(e, e_2), ..., sim(e, f_1), sim(e, f_2), ...)$,
F and f are defined analogously.

Matching process A single similarity measure may be unlikely to be successful. Hence, combining different similarity measures is an effective way. We have developed an algorithm (see Algorithm 1) that takes as input a set of candidate elements, and produces as output a set of relevant ontological constructs. A matching process is carried out between three vectors.

- *The vector of candidate elements (V_E)* consists of useful information extracted from HTML pages. A vector element can be either: form, table, or list. Each element E_i in V_E is described by a name and a set of attributes.
- *The vector of ontological concepts (V_C)* contains the concepts of domain ontology. Each concept E_c in V_C is described by its name and a set of reattached properties.
- *The vector of ontological relations (V_R)* represents the relations of domain ontology. Each relation E_r in V_R is described by its name and the original and the result concepts that it relates. E_r can be either: taxonomic (Is-as), or no-taxonomic relation.

Rule *i1*: Concept identification. Let $E_i \in V_E$ a candidate element, $E_c \in V_C$ an ontological concept, $E_r \in V_R$ an ontological relation, E_{c1}, $E_{c2} \in V_C$ the concepts that E_r relates, and K a threshold.

- If match result between E_i and E_c is greater than or equal to K, then E_c is marked as identified concept.
- If match result between E_i and E_r is greater than or equal to K, then E_{c1} and E_{c2} are marked as identified concepts in V_C.

Rule *i2*: Relation identification. Let $E_r \in V_R$ an ontological relation, E_{c1}, $E_{c2} \in V_C$ the concepts that E_r relates.

- If E_{c1} and E_{c2} are marked as identified concepts, then E_r is marked as identified relation in V_R.

2.3 Enrichment

Enrichment phase consists in inferring new constructs (concepts and/or relations) before generating the conceptual schema describing the Web application. The following rules summarize the mechanisms that permit the deduction of new constructs.

Rule *e1*: Taxonomic enrichment. Let $E_c \in V_C$ an ontological concept.

- If E_c is marked as identified concept in the previous phase then all the direct sub-concepts/super-concepts of E_c are marked as identified concepts in V_C.

Rule *e2*: No-taxonomic enrichment. Let $E_r \in V_R$ an ontological relation, E_{c1}, $E_{c2} \in V_C$ the original (domain) and the result (range) concepts that E_r relates.

– If only the concept E_{c1} is marked as identified concept in V_C, then the concept E_{c2} is also marked as identified concept in V_C, and vice-versa.

Algorithm: ***(Ontological constructs identification)***.

```
Input:
Vector VE of the extracted useful-information;
Vector VC of domain ontology concepts;
Vector VR of domain ontology relations;
String similarity measure between elements names SimN;
Lexical similarity measure between elements names SimL;
Structure similarity measure between elements groups SimS;
Threshold k.
Output:
Set of relevant ontological constructs.
BeginAlgo
 // Concepts identification
 For each element Ei in VE do
  For each element Er in VR do
    If (SimN(Ei.name, Er.name) > k) or
        (SimL(Ei.name, Er.name) > k) then
              the tow concepts Ec1 and Ec2 related by Er
              are marked as identified concepts in VC
    Endif
  Enddo
 Enddo
 For each element Ei in VE do
  For each element Ec in VC do
   If (SimN(Ei.name, Ec.name) > k)  or
       (SimL(Ei.name, Ec.name) > k )  then
              Ec is marked as identified concept in VC
   ElseIf SimS(Ei.group, Ec.group) > k then
              Ec is marked as identified concept in VC
   Endif
  Enddo
 Enddo
   // Relations identification
  For each element  in VR do
  If the concepts that Er relates are marked as identified
   concepts then Er is marked as identified relation in VR
 Endif
 Enddo
 // NO-Taxonomic enrichment
 For each element Er  do
  If (Ec1 Xor Ec2) are marked as identified concepts then
               Er is marked as identified relation in VR
 Enddo
 // Taxonomic enrichment
 For each element Ec in VC do
  If Ec is marked as identified concept then
   all the direct sub-concepts/super-concepts of Ec
   are marked as identified concepts in VC.
 Enddo
EndAlgo
```

2.4 Conceptualization

In this section, we propose a set of mapping rules to generate a conceptual model from the domain ontology. However, the concepts used in knowledge representation languages in a machine readable form, i.e., ontology, are very close to those used to represent data in conceptual models; both models have much in common. Although the aim of each model is different. In general, any conceptual

model (CM) can be considered as a 4-tuple: $CM = (CE, CR, CA, CS)$, where CE, CR, CA, CS stand respectively for Entities, Relationships, Attributes, and Constraints.

Whereas, ontological structure is a 5-tuple $O = (C, A^C, R, H^C, X)$, where:
- C is a finite set of concepts;
- A^C is a collection of attribute sets about concepts;
- R is a finite set of no-taxonomic relations; each relation has a pair of concepts (domain and range);
- H^C is called concept hierarchy or taxonomy, which is a directed relation $H^C \subseteq C \times C$;
- X is a set of axioms that describe additional constraints on the ontology.

Reverse engineering the domain ontology assists in developing the conceptual model. The rules below briefly summarize the transformation rules used in this research and are part of conceptualization phase. Each rule is followed by an example illustrating the mapping between OWL (Web Ontology Language) elements [**?**], and UML (Unified Modelling Language) constructs [9]. However, we believe that our approach can be applied to any similar language.

Rule *c1***.** Ontology concept C becomes CM Entity. The ontology concept name becomes the CM Entity name.

$OWLClass$ element is transformed into Entity element in UML. All classes in OWL are identified by URI. An entity in UML is identified by name.
Example: According to Rule c1, the OWL class "Destination":
`<owl: class rdf: ID="Destination"/>` Is translated as shown in Fig. 1.

Fig. 1. UML class

Rule *c2***.** The collection of attribute sets A^C about concept C becomes a CM Attribute of the corresponding CM Entity.

$OWLDataTypeProperty$ element is transformed into an attribute element in UML. An attribute in UML represents a common characteristic of some entity instances. OWL data type properties are used to link individuals to data values. A data type property is defined as an instance of the built-in OWL class $owl : DatatypeProperty$.
Example: According to Rule c2, the $OWLDataTypeProperty$ elements of the OWL class "Destination":

```
<owl:Class rdf:about="# Destination" />
<owl:DatatypeProperty rdf:about="#DestinationID">
     <rdfs:domain rdf:resource="#Destination"/>
     <rdfs:Datatype rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#Name">
     <rdfs:domain rdf:resource="#Destination"/>
     <rdfs:Datatype rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
</owl:Class>
```

Are translated as shown in Fig. 2:

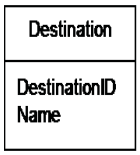


Fig. 2. UML Class with properties.

Rule *c3***.** Taxonomic relation H^C that expresses (IS-A relation) becomes CM Generalization/ Specialization relationship.

$OWLsubClassOf$ element is transformed into Generalization/Specialization relationship in UML.

Example: According to Rule c3, the OWL hierarchical relationships:

```
<owl:class rdf:about="UrbanArea">
    <owl:subClassOf rdf:resource="#Destination"/>
</owl:class>
<owl:class rdf:about="RuralArea">
    <owl:subClassOf rdf:resource="#Destination"/>
</owl:class>
```

Are translated as shown in Fig. 3.

Rule *c4***.** no-taxonomic relation R is translated as follow:

- i) relation having concept as domain and range becomes a CM Association relationship. The association has the domain as the source CM class and the range as the target CM Class. The relation local name becomes the target class name.
- ii) relation that expresses *Part-whole* relation becomes CM Aggregation/ Composition relationship.

$OWLObjectProperty$ element is transformed into Relationship element in UML. An object property in OWL relates an individual to other individuals. An object property is defined as an instance of the built-in OWL class

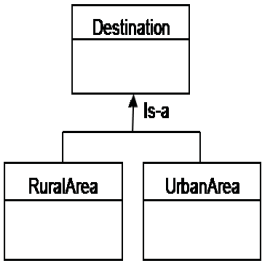


Fig. 3. Inheritance relationship.

owl : *ObjectProperty*. Relationships element in UML represents connections, links, or associations between two or more entities.
Example: According to Rule c4, the *OWLObjectProperty* element "HasAccommodation":

```
<owl:ObjectProperty rdf:about="#HasAccommodation">
    <rdfs:domain rdf:resource="#Destination"/>
    <rdfs:range  rdf:resource="#Accommodation"/>
</owl:ObjectProperty>
```

Is translated as shown in Fig. 4.

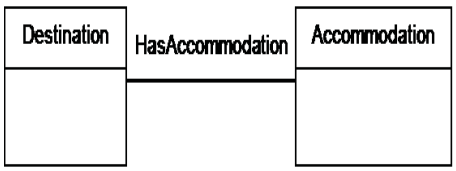


Fig. 4. Binary UML association.

Rule *c5.* Ontological axioms X are translated into CM constraints.

3 Overall architecture and implementation issues

The rules presented above are general and can be adapted to most conceptual modelling languages. The conceptual modelling language we have used is the UML, but we believe that our results could be applied to any similar language. On the other hand, the method is fully automatic. To evaluate our ontology-based Web application reverse engineering approach, a prototype has been implemented.

3.1 Architecture of the implemented tool

We have implemented the proposed approach in Java. The developed tool interacts with the Java WordNet and Jena APIs to parse HTML pages, compute semantic distance, and generate conceptual schema. It provides a set of features for personalizing the calculations performed during the reverse engineering process. Details on the obtained conceptual schema can also be visualized. The overall Framework is shown in Fig. 5.

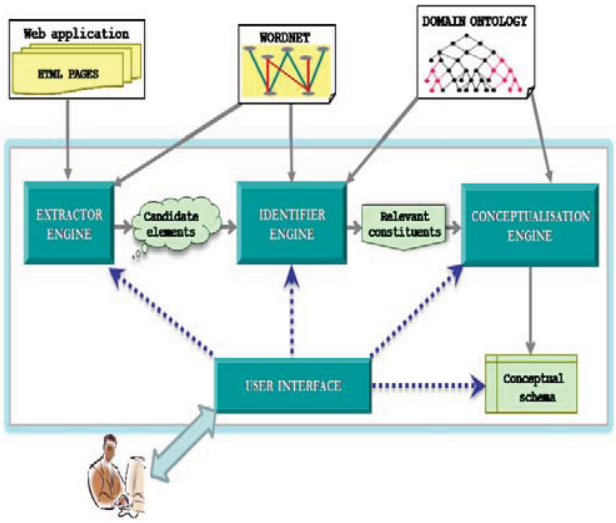


Fig. 5. The overall Framework.

- *User interface*: allows the acquisition of Web application's HTML pages, as well as domain ontology. It allows user to fix semantic distance threshold and to choose method and strategy for calculating this distance.
- *Extractor Engine*: represents an implementation of the extraction phase in the reverse engineering process. It is used to extract useful information from the acquired HTML pages.
- *Identifier Engine*: represents an implementation of the identification and enrichment phases in the reverse engineering process. It covers calculation of semantic distance, identification of a set of ontological concepts and relations, and enrichment of the identified set.
- *Conceptualization Engine*: is responsible for reverse engineering domain ontology to corresponding conceptual data model, it has been implemented according to the rules mentioned above, and executed using the identified ontological concepts and relations.

3.2 Illustrative Example

To illustrate our experiments, we have chosen the Web site of USA tourism Travel Guide[1], and the tutorial ontology for a Semantic Web of tourism[2] as running example. Using path measure as similarity measure between element names, multidimensional scaling strategy as semantic distance between element attributes, and a threshold as 0.7, we generate the conceptual schema presented in Fig. 6.

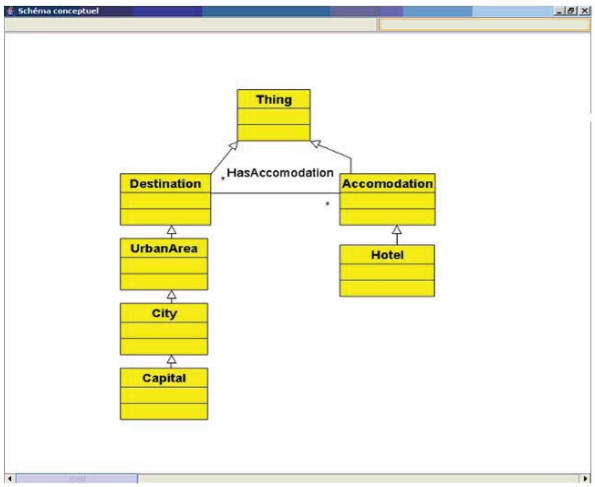


Fig. 6. Conceptual schema with threshold as 0.7.

While changing the threshold to 0.3, we obtain more complex conceptual schema (see Fig. 7).

3.3 Tool evaluation

To evaluate the quality of the match calculation, we compare the match result returned by the automatic matching process (P) with manually determined match result (R). We determine the true positives, i.e., correctly identified matches (I). Based on the cardinalities of these sets, the following quality measures are computed:

$Precision = |I|/|P|$, is the faction of the automatic discovered mapping which is correct. It estimates the reliability for the match prediction.

[1] http://www.hm-usa.com
[2] http://protege.stanford.edu/plugins/owl/owl-library/travel.owl

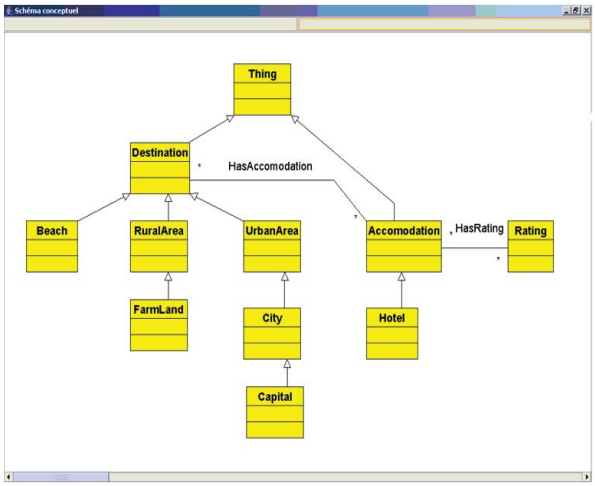


Fig. 7. Conceptual schema with threshold as 0.3.

$Recall = |I|/|R|$, is the fraction of the correct matches (the set R) which has been discovered by the mapping process. It specifies the share of real match that is found.

After several tests, we noted the following remarks:

- For the path measure, more the semantic distance threshold is small, more the conceptual schema becomes complex, less is the matching result and vice versa.
- The Multidimensional scaling strategy gives more efficient results than the other strategies.
- The recursive application of enrichment rules will provide more complete conceptual schema. Nevertheless, a significant number of iteration can generate superfluous concepts and relations. Thus, is to the designer to define the necessary and sufficient number of enrichment rules iteration.

Fixing a threshold for semantic distance means that an error rate was tolerated. Modelling a domain of interest is not deterministic, but it is heuristic.

4 Conclusion

In this paper we have proposed a reverse engineering approach of semi-structured and undocumented Web application. The proposed approach provides a reverse engineering rules to derive from domain ontology a conceptual schema modelling a Web application. The reverse engineering process consists in four phases: extracting useful information; identifying a set of ontological constructs representing the concepts of interest; enriching the identified set by additional constructs; and finally deriving a conceptual schema.

A prototype has been developed to implement the proposed approach. Some validation experiments have been carried out and they showed the usefulness of the proposed approach and highlighted possible areas for improvement of its effectiveness. We have formalized the method independently of the conceptual modelling language used. However, the method can be adapted to most languages. On the other hand, our method can be used with any domain ontology. The strong point of our approach is that it relies on a very rich semantic reference that is domain ontology. However, it is not possible to transform all elements from domain ontology into conceptual data model straight forward because ontology is semantically richer when data conceptual model.

The developed approach provides very satisfactory and encouraging results and supports the potential role that this approach can play in providing a suitable starting point for conceptual data model development. Nevertheless the derived conceptual schema should undergo a validation process that needs to be performed by domain specialist. Moreover, by using WordNet we can analyze only English Web applications. This problem can be solved in future work by using multilingual lexical knowledge.

References

1. P. Tramontana, Reverse engineering web applications, in: IEEE (Ed.), Proceedings 21st International Conference on Software Maintenance (ICSM05), 2005, pp. 705–708.
2. J. Lopez, P. Szekely, Web page adaptation for universal access, in: Proceedings of the 1st International Conference on Universal Access in Human-Computer Interaction, NewOrleans, 2001, pp. 690–694.
3. F. Ricca, P. Tonella, Using clustering to support the migration from static to dynamic web pages, in: Proceedings of the 11th International Workshop on Program Comprehension, Portland Oregon, USA, 2003, pp. 207–216.
4. D. W. Embley, Towards semantic understanding - an approach based on information extraction ontologies., in: Proceedings of the 25th Australasian Database Conference, New Zealand, 2004, pp. 3–12.
5. Y. A. Tijerino, D. W. Embley, D. W. Lonsdale, Y. Ding, G. Nagy, Towards ontology generation from tables., World Wide Web 8 (3) (2005) 261–285.
6. S. Benslimane, D. Benslimane, M. Malki, Z. Maamar, P. Thiran, Y. Amghar, M. Hacid, Ontology development for the semantic web: An html form-based reverse engineering approach, International Journal of Web Engineering, (JWE) 6 (2) (2007) 143–164.
7. DOM, Document object model. w3c recommendation (2004).
URL `http://www.w3.org/DOM/`
8. V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Cybernetics and Control Theory 10 (1966) 707–710.
9. OMG, Unified modelling language (uml), object management group (2003).
URL `http://www.omg.org/uml/`
10. R. Baeze-Yates, B. Ribeiro-Neto, Modern information retrieval, Addison-Wesley (Reading, Massachusetts).

Finding the Best Relaxations of Flexible Database Queries

Amine BRIKCI-NIGASSA[1], Allel HADJALI[2]

[1] Département d'Informatique, Faculté des Sciences de l'Ingénieur
Université Abou Bakr-Belkaid , Tlemcen, Algérie
nh2@LibreTlemcen.org

[2] IRISA/Enssat, Université Rennes 1
22305 Lannion Cedex, France
hadjali@enssat.fr

Abstract. This paper discusses an approach for searching for a best relaxation of a failing query (i.e., query that produces empty answers), semantically speaking. To do so, we propose a semantic proximity measure between flexible queries. This measure relies on particular distance between sets. By scanning the set of relaxed queries, the best relaxation that produces non-empty answers can be found thanks to the proximity measure defined.

Keywords: Databases, flexible queries, relaxation, semantic proximity.

1 Introduction

When retrieving and searching desired data over large databases, in particular those accessible via the web, users might be confronted with the common problem of *empty answers*, i.e., the queries submitted return an empty set of answers. In this case, the users' desires would be to find alternative answers that are related to the original failing queries. One cooperative approach that could enable for providing such answers is called *relaxation*. Query relaxation [7][8] aims at expanding the scope of a query by relaxing the constraints involved in the query.

In the context of flexible (or fuzzy) queries (i.e., queries that contain fuzzy conditions), the empty answer problem could still arise. Namely, there is no available data in the database that *somewhat satisfies* the user query. Only few works have been done for dealing with this problem (for an overview, see [1]). They mainly aim at relaxing the fuzzy requirements involved in the failing user query. This can be done by applying a transformation to some or all elementary conditions of that query. Recently in [1], a flexible query relaxation approach has been proposed. It is based on a particular *tolerance relation* modeled by a *relative proximity* parameterized by a tolerance indicator. This notion of proximity is intended for defining a set of predicates that are close, semantically speaking, to a given predicate *P*.

The above approach iteratively applies the relaxation mechanism defined on the failing query of interest and leads to a lattice of modified queries. This incremental relaxation process will stop when the answer to one of resulting queries is not empty or when some semantic condition is violated (in this case, the modified queries obtained are semantically far from the initial failing query). However, no guarantee is

provided that the non-failing modified query returned is the best relaxation of the failing initial query, i.e., it is the closest query to the initial one, semantically speaking. The main contribution of this paper is to attempt to overcome this shortcoming. To do this, we propose a semantic proximity measure between queries. This measure is based on particular distance between sets, called the Hausdorff distance. By scanning the lattice of modified queries, the best relaxation (in the above sense) which produces non-empty answers can be returned using that distance measure.

The paper is structured as follows. Section 2 gives a necessary background on some basic notions and introduces the Hausdorff distance measure. In section 3, we present the tolerance-based approach for flexible query relaxation. Section 4 discusses the method proposed to find the best relaxation of the query at hand. An illustrative example is proposed in section 5. Last, we briefly recall the main features of our proposal and conclude.

2 Background and the Hausdorff Distance

2.1 Flexible Queries

Flexible queries [11] are requests in which user's preferences can be expressed. Here, the fuzzy sets[1] framework is used as a tool for supporting the expression of preferences. The user does not specify crisp conditions, but fuzzy ones whose satisfaction may be regarded as a matter of *degree*. As a consequence, the result of a query is no longer a flat set of elements but is a set of discriminated elements according to their global satisfaction of the fuzzy constraints appearing in the query.

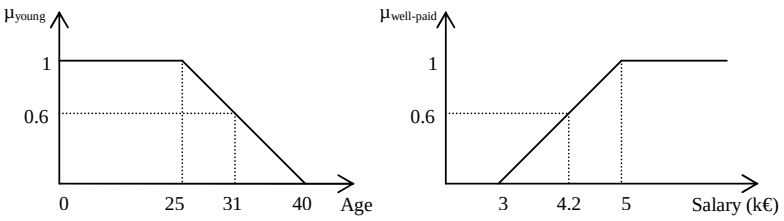


Fig. 1. Fuzzy predicates *young* and *well-paid* ($\mu_{young}(31) = 0.6$ and $\mu_{well\text{-}paid}(4.2) = 0.6$).

A typical example of a flexible query is: "retrieve the employees which are *young* and *well-paid"*, where *young* and *well-paid* are gradual predicates represented by means of fuzzy sets as illustrated in Figure 1. For instance, the predicate "*young*" is modeled by the trapezoidal membership function (*t.m.f.*) (A, B, a, b) = (0, 25, 0, 15) where [A, B] = [0, 25] (resp. [A–a, B+b] = [0, 40]) represents the core (resp. the support) of this predicate.

[1] A fuzzy set F in the referential U is characterized by a membership function $\mu_F: U \rightarrow [0, 1]$, where $\mu_F(u)$ represents the grade of membership of u in F. Two crisp sets are of particular interest when defining a fuzzy set F: the core (i.e., $\mathcal{C}(F) = \{u \in U / \mu_F(u) = 1\}$) and the support (i.e., $\mathcal{S}(F) = \{u \in U / \mu_F(u) > 0\}$).

2.2 Tolerance Relation

A *tolerance relation* (or a *proximity relation*) is a fuzzy relation R on a scalar domain U, such that for $u, v \in U$,

i) $\mu_R(u, u) = 1$ *(reflexivity)* *(ii)* $\mu_R(u, v) = \mu_R(v, u)$ *(symmetry)*

The quantity $\mu_R(u, v)$ evaluates the proximity between elements u and v. Here, we use the interpretation that evaluates to what extent the ratio u/v is *close* to *1* or not.
Relative Closeness [10]. The idea of relative closeness which expresses an approximate equality between two real numbers x and y, can be captured by the following relation:

$$\mu_{Cl}(x, y) = \mu_M(x/y), \quad (1)$$

where M, called a *tolerance parameter*, is a fuzzy number modeling "*close to 1*", such that: i) $\mu_M(1) = 1$: since x is close to x; ii) $\mu_M(t) = 0$ if $t \leq 0$: two numbers which are close should have the same sign; iii) $\mu_M(t) = \mu_M(1/t)$ (i.e., $M = 1/M$): this guarantees the property of symmetry of Cl. From property (iii), we can deduce that the support $\mathcal{S}(M)$ of the parameter M is symmetric and has the following form $\mathcal{S}(M) = [1 - \varepsilon, 1/(1 - \varepsilon)]$ with ε is a real number. Hence, in terms of *t.m.f.* M can be represented by $(1, 1, \varepsilon, \varepsilon/(1 - \varepsilon))$. It has been demonstrated in [10] that the fuzzy number M which parameterizes closeness (and also negligibility relation, Ne, defined as $\mu_{Ne[M]}(x, y) = \mu_{Cl[M]}(x+y, y)$) should be chosen so that its support $\mathcal{S}(M)$ lies in the validity interval $V = [(\sqrt{5} - 1)/2, (\sqrt{5} + 1)/2]$[2].

Interestingly enough, the validity interval V will play a key role in the proposed query relaxation process. As will be shown later, it constitutes the basis for defining a stopping criterion for controlling the relaxation.

2.3 The Hausdorff Distance

We recall here the principle of this kind of distance and we review an approach that can be followed to compute such a measure.

2.3.1 Crisp Sets. Consider two subsets A and B of a space U (equipped with a metric). The most popular scalar extension of distance between A and B is the *Hausdorff distance* defined as [5][12]:

$$d_H(A, B) = \max \{H(A, B), H(B, A)\}, \quad (2)$$

where $H(A, B)$ stands for the relative Hausdorff distance between A and B. We have $H(A, B) = \sup_{u \in A} d(u, B)$ and $d(u, B) = \inf_{v \in B} d(u, v)$. The expression $d(u, v)$ stands for a standard distance (such as Euclidean distance). Formula (2) can be written in the following compact form:

$$d_H(A, B) = \max \{\sup_{u \in A} \inf_{v \in B} d(u, v), \sup_{v \in B} \inf_{u \in A} d(u, v)\}. \quad (2')$$

[2] The following assumption holds: if x is close to y then neither is x negligible w.r.t. y, nor is y negligible w.r.t. x. Then, the interval V is the solution to the inequality $\mu_{Cl[M]}(x, y) \leq 1 - \max(\mu_{Ne[M]}(x, y), \mu_{Ne[M]}(y, x))$.

The idea that governs this distance is the following: for each element in *A* look for the closest element in *B*, then check for the element in *A* for which the distance to the closest element in *B* is maximal. The same is done exchanging *B* and *A* and the *longest* distance of the two component is kept.

As pointed out in [3] the Hausdorff distance can be seen as a measure of how much two non-empty compact (closed and bounded) sets *A* and *B* in a metric space resemble each other with respect to their positions. Note that d_H is a metric and the following statement holds: $d_H(A, B) = 0$ if and only if $A = B$. Usually, the following equalities are assumed to be true $d_H(A, \varnothing) = d_H(\varnothing, B) = +\infty$ and $d_H(\varnothing, \varnothing) = 0$.

Example 1. Let $A = [a_1, a_2]$ and $B = [b_1, b_2]$ be two regular intervals and let $d(u, v) = |u - v|$. Then, we have $d_H(A, B) = max(|a_1 - b_1|, |a_2 - b_2|)$. ♦

2.3.2 Fuzzy Sets. The Hausdorff distance between fuzzy sets can be either fuzzy or scalar. Hereafter, we only focus on the scalar version. For the fuzzy evaluation, more details are available in [3][5].

Scalar distances between fuzzy sets that retain good properties can be defined by merging the values $\{d_H(F_\alpha, G_\alpha), \alpha \in (0, 1]\}$[3]. For instance, Puri and Ralescu have proposed the following indices [12]:

$$d_H^{\infty}(F, G) = sup\{d_H(F_\alpha, G_\alpha), \alpha \in (0, 1]\}; \quad (3)$$

$$d_H^{1}(F, G) = \int_0^1 d_H(F_\alpha, G_\alpha) d\alpha \quad (4)$$

The only serious drawback of the above distance indices is the fact that they are limited to fuzzy sets that have equal *maximum* membership values. For our purpose this drawback is excluded. Indeed, all fuzzy sets that are considered are normalized (hence, with the maximum membership value equals to 1).

Example 2. Let *U* represents a numeric universe of discourse of the variable "*age*" of a person. Let also *F* = "*about thirty*" and *G* = "*between_26_and_28*" two fuzzy sets on U defined by the following two t.m.f.: $F = (30, 30, 3, 3)$; $G = (26, 28, 1, 1)$. Now, we evaluate the distance between *F* and *G* using formula (4). First, let us precise that F_α and G_α are intervals and can be expressed as follows: $F_\alpha = [3\alpha + 27, 33 - 3\alpha]$; $G_\alpha = [\alpha + 25, 29 - \alpha]$. Then, one can easily check that

$$d_H^{1}(F, G) = \int_0^1 max(|(\alpha + 25) - (3\alpha + 27)|, |(29 - \alpha) - (33 - 3\alpha)|) d\alpha = 7/2.$$

3 Flexible Query Relaxation

In this section, we first introduce a *tolerance–based approach* for relaxing a Single-Predicate (SP) Query. Then, we discuss the relaxation strategy in case of conjunctive queries. This section is mostly borrowed from [1].

[3] F_α stand for the α-cut of *F*, i.e., $\{u \in U / \mu_F(u) \geq \alpha\}$.

3.1 SP Queries: Principle of the Approach

Relaxing a failing flexible query consists in modifying the constraints involved in the query in order to obtain less restrictive variants. Such a modification can be achieved by applying a *basic transformation* $T^{\uparrow}$ to all or some predicates of the failing query. Some desirable properties are required for any transformation $T^{\uparrow}$ when applied to a predicate P, see [1] for more detail.

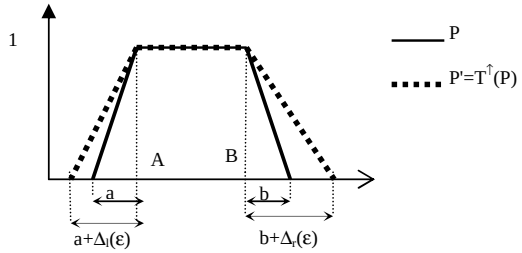


Fig. 2. Basic transformation using a proximity relation

Let $Q = P$ be a SP query and P a fuzzy predicate. To relax Q we replace P by an enlarged fuzzy predicate P' defined as follows:

$$\forall u \in U, \quad \mu_{P'}(u) = sup_{v \in U}\, min\,(\mu_P(v), \mu_{Cl[M]}(v, u)) = sup_{v \in U}\, min\,(\mu_P(v), \mu_M(v/u)).$$

Using the extension principle, it is easy to check that $P' = P \otimes M$ where $\otimes$ is the product operation extended to fuzzy numbers [6]. Formally, the transformation of interest $T^{\uparrow}$ writes:

$$P' = T^{\uparrow}(P) = P \otimes M. \tag{5}$$

Clearly, the modified predicate P' gathers the elements of P and the elements outside P which are *somewhat close* to an element in P. $T^{\uparrow}(P)$ results then in a predicate P' which is *less restrictive* than P, but still *semantically close* to P.

Input: Q := P: Initial Failing Query
ε: a tolerance value /* $\varepsilon \in [0, (3 - \sqrt{5})/2]$ */

1. $i := 0$; $Q_i := Q$; /* *i denotes the number of relaxation steps* */
2. compute Σ_{Q_i} ; /* Σ_{Q_i} *represents the set of answers to* Q_i */
3. **while** ($\Sigma_{Q_i} = \varnothing$ and $S(M^{i+1}) \subseteq V$) **do**
4. **begin**
5. $i := i+1$;
6. $Q_i := T^{\uparrow(i)}(P) = P \otimes M^i$;
7. compute Σ_{Q_i};
8. **end**

Output: Σ_{Q_i} (if $\neq \varnothing$): The set of non-empty answers to Q

Algorithm 1. Incremental relaxation of a single-predicate query

In terms of *t.m.f.*, if $P = (A, B, a, b)$ and $M = (1, 1, \varepsilon, \varepsilon/(1-\varepsilon))$ where ε stands for the *relative tolerance value* and lies in $[0, (3-\sqrt{5})/2]$ (this interval results from the inclusion $\mathcal{S}(M) \subseteq V$, see [10]), the predicate P' is such that $P' = (A, B, a + \Delta_l(\varepsilon), b + \Delta_r(\varepsilon))$ using (5) and where $\Delta_l(\varepsilon) = A \cdot \varepsilon$ and $\Delta_r(\varepsilon) = B \cdot \varepsilon/(1-\varepsilon)$. See Figure 2.
A *maximal relaxation*, denoted by $T^{\uparrow max}(P)$, of a predicate P can be reached using the tolerance value $\varepsilon_{max} = (3-\sqrt{5})/2$. Hence, $T^{\uparrow max}(P) = (A, B, a + \Delta_l(\varepsilon_{max}), b + \Delta_r(\varepsilon_{max}))$.

In practice, if $Q = P$ is an SP query and if the set of answers to Q is empty, then Q is relaxed by transforming it into $Q_1 = T^{\uparrow}(P) = P \otimes M$. This transformation can be repeated n times until the answer to the revised question $Q_n = T^{\uparrow(n)}(P) = P \otimes M^n$ is not empty. In order to ensure that the revised query Q_n remains semantically close enough to the original one, the support of M^n should be included in V. Then, the above iterative relaxation procedure will stop either when the *answer is non-empty* or when $S(M^n) \not\subset V$. This relaxation procedure can be formalized by Algorithm 1.

3.2 Flexible Conjunctive Queries

A conjunctive flexible query Q is of the form $P_1 \wedge \ldots \wedge P_N$, where the '$\wedge$' stands for the connector *'and'* and is interpreted by the '*min*' operator, and P_i is a fuzzy predicate.

3.2.1 Relaxing Strategy

The relaxation strategy we use is based on the *local query modification*. This kind of transformation affects only some predicates (or subqueries). Given a set of transformations $\{T_1^{\uparrow},\ldots,T_N^{\uparrow}\}$ and a query $Q = P_1 \wedge \ldots \wedge P_N$, the set of modified queries related to Q resulting from applying $\{T_1^{\uparrow},\ldots,T_N^{\uparrow}\}$ is $\{T_1^{\uparrow(i_1)}(P_1) \wedge \ldots \wedge T_N^{\uparrow(i_N)}(P_N)\}$, where $i_h \geq 0$ and $T_j^{\uparrow(i_h)}$ means that the transformation $T_j^{\uparrow}$ is applied i_h times to P_j.

As pointed out in [1], an ordering ($\prec$) over the set of revised queries can be defined. That ordering can be expressed on the basis of the number of applications of the transformation associated to each predicate. If Q' and Q'' are two relaxed queries of Q, we say that

$$Q' \prec Q'' \text{ if } \sum_{i=1}^{N} \text{count}(T_i^{\uparrow} \text{ in } Q') < \sum_{i=1}^{N} \text{count}(T_i^{\uparrow} \text{ in } Q'').$$

Now let us mention that the set of modified queries related to Q (i.e., $\{T_1^{\uparrow(i_1)}(P_1) \wedge \ldots \wedge T_N^{\uparrow(i_N)}(P_N)\}$ can be organized according to a lattice structure. For instance, the lattice associated with the query "$P_1 \wedge P_2$" is given in Figure 3.
The advantage of this approach is the fact that it leads to a *bounded lattice* of relaxed queries. Indeed, as mentioned in section 3.1, every predicate P_i can be maximally relaxed into $T_i^{\uparrow(max)}(P_i)$. Hence, the modified query given by

$$T^{\uparrow(max)}(Q) = T_1^{\uparrow(max)}(P_1) \wedge \ldots \wedge T_N^{\uparrow(max)}(P_N)$$

can be considered as the *maximal relaxation* of a query $Q = P_1 \wedge \ldots \wedge P_N$. This lower bound query is directly inherent to the semantic limits provided by the proximity-

based transformation. Now, assume that each predicate in Q can be relaxed at most ω times. Then, the number of relaxation steps authorized is $\omega \cdot N$.

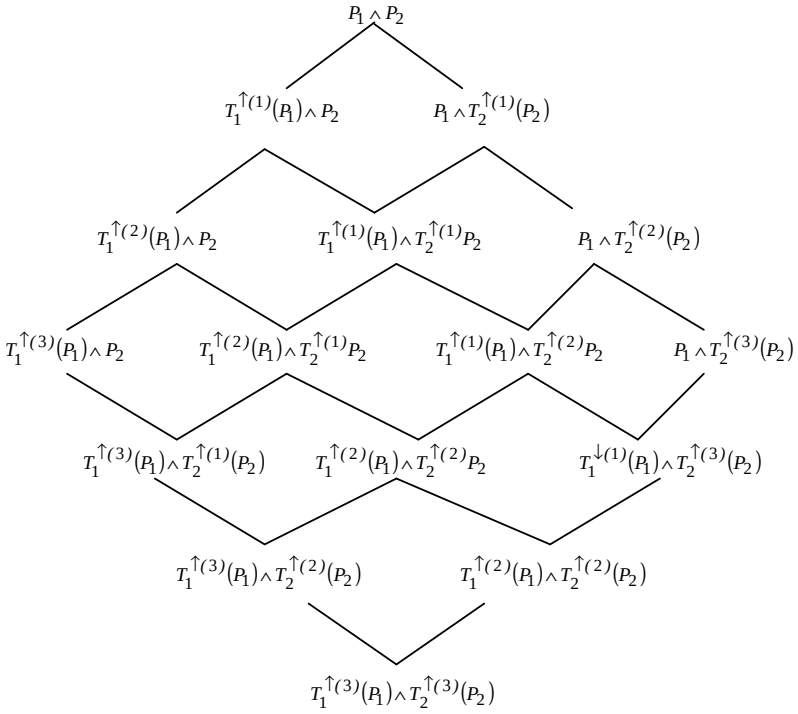


Fig. 3. Bounded lattice of relaxed queries (for ω = 3).

3.2.2 Lattice Traversal: An MFSs-based approach

In what follows, we show how to scan the lattice in an efficient way by exploiting the *Minimal Failing Subqueries* (MFSs) of the failing original query Q. Recall that MFSs stand for the smallest subqueries of Q that fail.

Let $Q = P_1 \wedge \ldots \wedge P_N$ be a failing query, $T^{\uparrow}(Q)$ a relaxed query of Q, and $SQ^{[j]}$ a subquery of Q obtained by deleting the predicate P_j from Q. Let also $mfs(Q) = \left\{ P_{s_1^1} \wedge \ldots \wedge P_{s_{m_1}^1}, \ldots, P_{s_1^k} \wedge \ldots \wedge P_{s_{m_k}^k} \right\}$ be the set of *MFSs* of Q with $\left\{ s_1^h, \ldots, s_{m_h}^h \right\} \subset \{1, \ldots, N\}$ for $1 \leq h \leq k$. To characterize the set of *MFSs* of $T^{\uparrow}(Q)$ with respect to that of Q, we introduce the following propositions.

Proposition 1. If $T^{\uparrow}(Q) = SQ^{[j]} \wedge T_j^{\uparrow}(P_j)$ with $j \in \bigcup_{h=1}^{k} \{s_1^h, \ldots, s_{m_h}^h\}$ then the *MFSs* of $T^{\uparrow}(Q)$ must be searched in *mfs(Q)* by substituting $T_j^{\uparrow}(P_j)$ to P_j in *mfs(Q)*.

Proposition 2. If $T^{\uparrow}(Q) = SQ^{[j]} \wedge T_j^{\uparrow}(P_j)$ with $j \notin \bigcup_{h=1}^{k} \{s_1^h, \ldots, s_{m_h}^h\}$ then the set of *MFSs* of $T^{\uparrow}(Q)$ is the set *mfs(Q)*.

Example 3. Assume that $Q = P_1 \wedge P_2 \wedge P_3 \wedge P_4$ and *mfs(Q)*= $\{P_1 \wedge P_3, P_1 \wedge P_4\}$.Then,

- If $T^{\uparrow}(Q) = T_1^{\uparrow}(P_1) \wedge SQ^{[1]} = T_1^{\uparrow}(P_1) \wedge P_2 \wedge P_3 \wedge P_4$, the *MFSs* of $T^{\uparrow}(Q)$ are searched in $\{T^{\uparrow}(P_1) \wedge P_3, T^{\uparrow}(P_1) \wedge P_4\}$.
- If $T^{\uparrow}(Q) = SQ^{[2]} \wedge T_2^{\uparrow}(P_2) = P_1 \wedge T_2^{\uparrow}(P_2) \wedge P_3 \wedge P_4$, Q and $T^{\uparrow}(Q)$ have the same *MFSs*.

From the above propositions, it results that the set of MFSs of a relaxed query $T^{\uparrow}(Q)$ can be obtained from the set of MFSs associated with Q. Thus, in practice, it suffices to compute the set of MFSs of Q for deducing the MFSs of any relaxed query $T^{\uparrow}(Q)$. Now, to search for a set of non-failing relaxed queries over the lattice, we make use of a two-step procedure as follows:

- **Step 1:** *identifying k MFSs of Q*
  To do this, we make use of the algorithm proposed in [9] which is designed for computing *k MFSs* in acceptable time (when *k* is not too large). By propositions 1 and 2, we run only once this algorithm since the MFSs of any relaxed query $T^{\uparrow}(Q)$ are deduced from those of Q.
- **Step 2:** *intelligent search technique*
  Information about *MFSs* allows for avoiding evaluating some nodes of the lattice. Indeed, a node in the lattice that preserves at least one *MFS* of its father-node (i.e., the node from which it is derived) does not have to be evaluated (since we are certain that it fails). This technique is sketched in Algorithm 2 where:

  - *Level(i)* stands for the set of relaxed queries at level *i* of the lattice;
  - *father(Q')* is the set of nodes from which Q' can be derived, i.e., Q' is an immediate relaxed variant of any query contained in *father(Q')*. In Figure 3, if $Q' = T^{(1)}(P_1) \wedge T^{(1)}(P_2)$, then *father(Q')* = $\{T^{(1)}(P_1) \wedge P_2, P_1 \wedge T^{(1)}(P_2)\}$.
  - *evaluate(Q')* is a function that evaluates Q' against the target database. It returns *true* if Q' produces non-empty answers, *false* otherwise.

As can be seen, Algorithm 2 returns a set (i.e., *List-Req*) of non-failing relaxed queries to the initial query Q. According to the order defined in section 3.2, the elements of *List-Req* are incomparable. Indeed, they result from the application of the same number of transformations: $\forall Q', Q'' \in$ *List-Req*, we have

Input: Q = $P_1 \wedge \ldots \wedge P_N$: a failing query

$$mfs(Q) = \left\{ P_{s_1^1} \wedge \ldots \wedge P_{s_{m_1}^1}, \ldots, P_{s_1^k} \wedge \ldots \wedge P_{s_{m_k}^k} \right\}$$: set of MFSs of Q

1. *List-Req* = ∅; *i := 1*;
2. **while** (*i* ≤ ω) **and** (*List-Req* = ∅) **do**
3. **begin**
4. Level(i) = { $Q_i^1, \cdots, Q_i^{n_i}$ };
5. **for** *req* **in** Level(i) **do**
6. **begin**
7. modif := true;
8. **for** a_mfs **in** mfs(father(req)) **do**
9. *modif := modif* ∧ *(a_mfs* ∉ *req)* ;
10. **if** modif **then**
11. **if** evaluate(*req*) **then**
12. *List-Req := List-Req* ∪*{req};*
13. **endif;**
14. **endif;**
17. **end**;
18. *i:=i+1*;
19. **end**;
20. **If** *List-Req* ≠ ∅ **then return** *List-Req*;
21. **endif;**

Output: *List-Req*: a set of non-failing relaxed query associated with *Q*

Algorithm 2. Search for a set of non-failing relaxed queries.

$$\sum_{i=1}^{N} \text{count}(T_i^{\uparrow} \text{ in } Q') = \sum_{i=1}^{N} \text{count}(T_i^{\uparrow} \text{ in } Q'').$$

Since semantics is one of our starting points (our ultimate aim is to find a relaxed query that produces non-empty answers and that is the closest one to the initial query Q, semantically speaking), the question of interest now is how to select the best relaxation to *Q* among the elements of *List-Req*?

To answer this question, we propose an ordering based on a semantic proximity measure induced by the Hausdorff distance between queries. The principle of that ordering is detailed in the following section.

4. Searching for the Best Relaxation

4.1 Semantic Proximity Query

Taking into account the strong intuitive connection between proximity and distance, and by using the Hausdorff distance measure, we may want to estimate to what extent two flexible queries are close, semantically speaking.

SP Queries. Let $Q = P$ and $Q' = P'$ be two SP queries (where P and P' are predicates pertaining to the same attribute, say A). To evaluate to which extend Q and Q' are close, semantically speaking, we make use of the Hausdorff distance index between the fuzzy predicates involved in those two queries. Then, we have

$$Dist(Q, Q') = d_H^1 (P, P'), \tag{6}$$

where $Dist(Q, Q')$ stands for a distance measure between Q and Q'. It is well known that the distance measure produces values that have the reverse ordering of proximity measures. The smaller the index $Dist(Q, Q')$, the closer Q and Q'. Let $Prox(Q, Q')$ denotes a proximity measure between Q and Q'. The index $Prox(Q, Q')$ can be defined using a conversion function on the distance measure. For instance, $Prox(Q, Q')$ can be defined by [4]:

$$Prox(Q,Q') = \left(1 + \left(\frac{Dist(Q,Q'))}{s}\right)^t\right)^{-1}. \tag{7}$$

The positive constants s and t adjust the size of the proximity measure. The simplest conversion function can be obtained by setting $s = 1$ and $t = 1$.

It is worth noticing that for our purpose, the amount to which Q is *semantically close to* Q' is *not too crucial* since we are only interested in the order induced by the measure $Prox(Q, Q')$. This order is obtained by reversing the order induced by the measure $Dist(Q, Q')$. To illustrate this matter, let us consider the following example:

Example 4. Let Q, $Q1$, $Q2$ and $Q3$ be four SP queries. Assume that we have obtained the following distance measures:

$$Dist(Q, Q1) = 6, \quad Dist(Q, Q2) = 17/3, \quad Dist(Q, Q3) = 8/3.$$

One can observe that $Dist(Q, Q1) > Dist(Q, Q2) > Dist(Q, Q3)$. By reversing this order, we obtain the following ordering based on proximity measure:

$$Prox(Q, Q1) < Prox(Q, Q2) < Prox(Q, Q3).$$

Hence, Q is closer to $Q3$ than $Q2$ (resp. $Q1$).

Compound Queries. Let $A_1, A_2, \ldots, A_n$ be n attributes with $D(A_i)$ being the domain of values of A_i. Let also Q (resp. Q') be a flexible compound query of the form $P_1 \wedge \ldots \wedge P_k$ (resp. $P'_1 \wedge \ldots \wedge P'_k$) where, for $i=1,k$, P_i (resp. P'_i) is a fuzzy predicate pertaining to the attribute A_i. To evaluate the distance between Q and Q' in the spirit of formula (6), one can use the following formula:

$$Dist(Q,Q') = \frac{1}{k}\sum_{i=1}^{k} Dist(P_i, P'_i) \tag{8}$$

One can easily recognize the above expression which stands for an arithmetic mean.

5.2 Principle of the Method

Assume that *Q* denotes the initial failing query and *List-Req* the set of relaxations of *Q* provided by Algorithm 2. In order to return the best relaxation of *Q* among the elements of *List-Req*, we proceed as follows:

- **Step 1**: *Distance measure Calculus*
  for each $relax_Q$ in *List-Req*
  Compute $Dist(Q, relax_Q)$
- **Step 2**: *Rank-order List-Req*
  Rank-order *List-Req* in increasing sense w.r.t. the distance measure
- **Step 3**: *Best relaxation*
  Return the first element of the ordered set *List-Req*

This above three-step procedure can be formalized in the following algorithm:

Input: *Q;*
List-Req;

1. **begin**
2. **for** $relax_Q$ **in** *List-Req*;
3. compute $Dist(Q, relax_Q)$ [4];
4. Sort(List-Req); /* sort in ascending way */
5. return First(List-Req);

Output: The best relaxation of *Q*;

Algorithm 3. Search for the best relaxation of Q.

5. An Illustrative Example

To illustrate our proposal, we have tailored an example inspired from [2]. It concerns a user who wants to find the employees in a department who satisfy the condition: *young* and *well-paid*. The relation describing the employees is given in Table 1.

Table 1. Relation of the employees

Name	Age	Salary (k€)	$\mu_{P_1}(u)$	$\mu_{P_2}(v)$	$\mu_Q(t) = \min(\mu_{P_1}(u), \mu_{P_2}(v))$
Dupont	46	3	0	0	0
Martin	42	2.5	0	0	0
Durant	28	1.5	0.8h	0	0
Dubois	30	1.8	0.67	0	0
Lorant	35	2	0.34	0	0

[4] One can consider $Dist(Q, relax_Q)$ as only equals to the relative distance index $H(relax_Q, Q)$. Indeed, the answers to $relax_Q$ which are given as alternative answers to *Q* and not the reverse.

Then, the query of interest writes Q = *"find employees who are young and well-paid"* where *young* and *well-paid* are labels of fuzzy sets represented respectively by the t.m.f. $P_1 = (0, 25, 0, 15)$ and $P_2 = (5, +\infty, 2, +\infty)$ as drawn in Figure 1. In the following, we will simply write $Q = P_1 \wedge P_2$.

As can be seen, each item of the database gets zero as satisfaction degree for the user's query Q with the content of Table 1. Now, in order to return alternative answers to the user, we try to cooperate with him by relaxing his/her question. Let us assume that $\varepsilon_1 = 0.09$, $\varepsilon_2 = 0.12$ and $\omega = 3$. Then, the lattice of relaxed queries is similar to the one depicted in Figure 3. It is easy to see that mfs(Q) = $\{P_2\}$.

Table 2.

Name	**Age**	**Salary** (k€)	$\mu_{T_1}{}^{\uparrow(P_1)}(u)$	$\mu_{T_2}{}^{\uparrow(P_2)}(v)$	$\mu_{Q_{22}}(t) = \min(\mu_{T_1}{}^{\uparrow(P_1)}(u), \mu_{T_2}{}^{\uparrow(P_2)}(v)$
Dupont	46	3	0	0.23	0
Martin	42	2.5	0.028	0.03	0.028
Durant	28	1.5	0.81	0	0
Dubois	30	1.8	0.69	0	0
Lorant	35	2	0.39	0	0

Table 3.

Name	**Age**	**Salary** (k€)	$\mu_{P_1}(u)$	$\mu_{T_2}{}^{\uparrow(2)(P_2)}(v)$	$\mu_{Q_{23}}(t) = \min(\mu_{P_1}(u), \mu_{T_2}{}^{\uparrow(2)(P_2)}(v)$
Dupont	46	3	0	0.37	0
Martin	42	2.5	0	0.21	0
Durant	28	1.5	0.8	0	0
Dubois	30	1.8	0.67	0	0
Lorant	35	2	0.34	0.06	0.06

By applying Algorithm 2, we obtain *List-Req* = $\{Q_{22}, Q_{23}\}$ where Q_{ij} is the relaxation number j (from the left to the right) of Q of the level i (in the lattice). We have $Q_{22} = T_1{}^{\uparrow}(P_1) \wedge T_2{}^{\uparrow}(P_2)$ and $Q_{23} = P_1 \wedge T_2{}^{\uparrow(2)}(P_2)$ with $T_1{}^{\uparrow}(P_1) = (0, 25, 0, 17.5)$, $T_2{}^{\uparrow}(P_2) = (5, 100, 2.6, 0)$[5] and $T_2{}^{\uparrow(2)}(P_2) = (5, 100, 3.2, 0)$. See Tables 2 and 3.

To select which element of *List-Req* is the best relaxation for Q, we first estimate the distance measures $Dist(Q, Q_{22})$ and $Dist(Q, Q_{23})$. Using formula (8), it is easy to check that

$Dist(Q, Q_{22}) = 0.55$ (where $Dist(P_1, T_1{}^{\uparrow}(P_1)) = 0.25$ and $Dist(P_1, T_2{}^{\uparrow}(P_2)) = 0.3$)
$Dist(Q, Q_{23}) = 0.6$ (where $Dist(P_2, T_2{}^{\uparrow(2)}(P_2)) = 0.6$)

[5] To make the calculus simple, we have fixed the upper bound of the support of P_2, i.e., $P_2 = (5, 100, 2, 0)$. Due to the fact that this bound is never reached by the actual salaries, the relaxation transformation will affect only the left side of *P2*.

We deduce that $Dist(Q, Q_{22}) < Dist(Q, Q_{23})$. This implies that Q_{22} is the best relaxation of Q. Then, the employee Martin is returned to the user query.

6. Conclusion

In this paper, we have proposed an approach aiming at finding the best relaxation of failing queries in a flexible setting. The key concept of this approach is the semantic query proximity defined using the Hausdorff distance measure. The approach proposed can apply both for point and range queries as well. In this work, only attributes with domains endowed with a metric have been considered. It would be extremely interesting to extend the approach to attributes with non metricized domains (as *color* attribute). We acknowledge also that some experimental studies are needed to demonstrate the efficiency and effectiveness of the approach.

References

1. P. Bosc, A. Hadjali, O. Pivert, "Incremental controlled relaxation of failing flexible queries", *Journal of Intelligent Information Systems*, in press, 2008.
2. P. Bosc, A. Hadjali, O. Pivert, Weakening of fuzzy relational queries: An absolute proximity relation-based approach, *Journal of Mathware & Soft Computing*, Vol. 14(1), pp. 35-55, 2007.
3. B. Chaudhuri and A. Rosenfeld, A modified Hausdorff distance between fuzzy sets, *Information Sciences*, Vol. 118, pp. 159-171, 1999
4. V. Cross and T. Sudkamp, Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications, Studies in Fuzziness and Soft Computing, No93, Physica-Verlag, 2002.
5. D. Dubois and H. Prade, On distances between fuzzy points and their use for plausible reasoning, In Proc. Int. Conf. on Systems, Man and Cybernetics, 1983, pp. 300-303.
6. D. Dubois, Prade H., Possibility Theory, Plenum Press, 1988.
7. T. Gaasterland, Cooperative answering through controlled query relaxation, *IEEE Expert*, 12(5), pp. 48-59, Sep/Oct 1997.
8. T., Gaasterland, P. Godfrey, and J. Minker, Relaxation as a platform for cooperative answering. *Journal of Intelligent Information Systems*, 1(3-4), pp. 293-321, 1992.
9. P. Godfrey, Minimization in cooperative response to failing database queries, *Int. Journal of Cooperative Information Systems*, 6(2), pp. 95-149, 1997.
10. A. Hadjali, D. Dubois and H. Prade, Qualitative reasoning based on fuzzy relative orders of magnitude. *IEEE Transactions on Fuzzy Systems*, Vol. 11, No 1, pp. 9-23, 2003.
11. H. Larsen, J. Kacpryk, S. Zadrozny, T. Andreasen, and H. Christiansen (Eds.), Flexible Query Answering Systems, Recent Advances, Physica Verlag, 2001.
12. M.L. Puri and D.A. Ralescu, Differentials of fuzzy functions, Journal of Mathematical Analysis and Applications, Vol. 91, pp. 552-558, 1983.

SNNHLM : SNN hierarchical clustering

Guillem Lefait[1,2], Gilles Goncalves[2], Michael Whelan[1],
Tahar Kechadi[1], Tiente Hsu[2]

[1] University College Dublin, School of Computer Science & Informatics, PCRG
[2] Université d'Artois, Faculté des Sciences Appliquées, LGI2A

Abstract. Shared Nearest Neighbour (SNN) similarity measure defines the relationship between two objects based on how similar their shared neighbourhood is. The SNN similarity is more robust than the usual measures, such as Euclidean distance for two reasons: firstly, the similarity function has to be symmetrical as to facilitate the identification of outliers. Secondly, because the SNN similarity is based on more than two objects, structural information is more easily discovered even with high-dimensional data. We present a hierarchical clustering that relies on the SNN similarity to produce high-quality clusters and we show that our method is efficient when applied to high-dimensional data sets.
Key words:data mining, clustering, hierarchical clustering, similarity measure, high-dimensional data

1 Introduction

The aim of clustering is to group unlabeled datasets into a small number of sets of natural, hidden data structures [1]. A widely accepted way of classifying clustering techniques is to consider the processes of how clusters are generated: hierarchical or partitional [2].
Partitional clustering techniques attempt to identify the partitions that optimise (usually locally) a given clustering criterion [3]. Partitioning algorithms typically start with randomly created partitions and iteratively optimise the partitions.
Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters. They are of two categories: agglomerative and divisive algorithms. Agglomerative algorithms start by considering each data as a cluster (singleton) and merge iteratively clusters altogether according to a given similarity measure. Divisive approaches start by considering all the dataset as one cluster and then try to divide the data in such a way as to increase the similarity between the elements of the resulting groups.
Hierarchical clustering algorithms have some advantages compared to partitional clustering. Firstly, they do not need to specify the number of the clusters in advance because they output a hierarchy. The user will decide a posteriori how many clusters to keep. Secondly, the hierarchy can be a very efficient way to represent relationships between data objects or items. If the user aim is to create taxonomy, then hierarchical algorithms may be a good choice as they reveal how the data can be structured or organised.

However hierarchical algorithms have some disadvantages. Their inherent complexity is still a challenge and mainly for real datasets and thus they often rely on very efficient pre-processing sampling methods. Moreover, the size of the hierarchy produced may be too large to be adequately investigated. Hierarchical clustering algorithms are greedy and will never try to improve previous decisions, e.g. an inappropriate merge or split will be propagated throughout the whole process without any possibility of being corrected. The hierarchy generated is highly dependent on the relationship function used. Therefore, the users have to choose carefully the metrics between both patterns and clusters.
In addition, in high dimensional data sets, distances or similarities between objects become more uniform, making clustering more difficult to discover the relationship between patterns.
In this paper, we propose an algorithm to deal with these issues, and mainly, reduce the complexity of clustering to an acceptable threshold, use a similarity measure that is able to isolate outliers and be scalable (be efficient on high-dimensional datasets).

2 Related work

2.1 Notation

Before starting the problem description and some related works, we introduce some notations. Given a dataset P composed of n patterns, $P = (p_1, ..., p_n)$, we want to obtain a set H, the hierarchy, composed of a set of clusters $C_h = (c_{h_1}, ..., c_{h_K})$, such that every pattern belong to one and only one cluster (hard clustering) and such that $|H_{C_h}| < |H_{C_{h+1}}|$ (the number of clusters decreases through the hierarchy).
The ordered list of the k nearest neighbours of a pattern i given a distance measure d is defined as $N_k(i)$ with $N_k(i)[j]$ the id of the $j-th$ neighbour of i and where $d(p_i, p_{N_k(i)[a]}) \leq d(p_i, p_{N_k(i)[b]})$ if $a < b$ (the neighbour list is ordered by increasing distances).
Figure 1 shows an example of a dataset and the first four neighbours of a pattern, E. The **N**eighbour matrix N is computed by retrieving all the four nearest neighbours of each pattern.

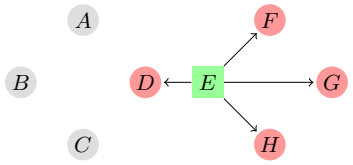


Fig. 1. Identification of the four Nearest Neighbour of pattern E

The seminal work on shared near neighbours similarity is described in [4]. The authors proposed a clustering algorithm, called SNNO, which

consists of two main phases : 1) the similarity computation and 2) the cluster construction.
The similarity S_UW (**U**n**W**eighted) is defined as the number of neighbours that two patterns i and j have in common:

$$S_{UW}(i,j,k) = |N_k(i) \cap N_k(j)| \tag{1}$$

This similarity can be improved by taking into account the rank of the shared neighbours. Therefore the closer the shared neighbours are, the greater the similarity will be. In the next equation, the ranks are **S**ummed (S_S):

$$S_S(i,j,k) = \sum_{z=0}^{k} \sum_{z'=0}^{k} \big((k+1-z) + (k+1-z')\big) \tag{2}$$

where $N_k(i)[z] = N_k(j)[z']$. S_P is defined in the same way, but it involves the **P**roduct of the ranks.
Finally, these patterns must appear in the other pattern's neighbourhood, otherwise the similarity is set to 0 :

$$S_*(i,j,k) = \begin{cases} S_*(i,j,k) & \text{if } i \in N_k(j) \text{ and } j \in N_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The star $*$ stands for UW, P, S and SP, PD (see below).
Equation 3 identifies the noisy patterns and the bounds of clusters of different densities. The choice of the similarity function will be discussed in the section 4.1. Given the matrix of nearest neighbours N (see Table 1) obtained from the data shown in Figure 1, similarity is calculated for S_{UW} (Table 2) and for S_S (Table 3).

id	N_1	N_2	N_3	N_4
A	B	D	C	E
B	A	C	D	E
C	B	D	E	A
D	E	C	A	B
E	D	F	H	G

Table 1. 4-NN

	A	**B**	**C**	**D**	**E**
A		5	5	5	0
B	5		5	5	0
C	5	5		5	0
D	5	5	5		2
E	0	0	0	2	

Table 2. S_{UW} similarities

	A	**B**	**C**	**D**	**E**
A		30	30	30	0
B	30		30	30	0
C	30	30		30	0
D	30	30	30		18
E	0	0	0	18	

Table 3. S_S similarities

Once the similarity values between patterns have been performed, the clusters are created by grouping the patterns given a certain threshold. A hierarchical version, SNNOH, may be derived by reducing gradually this threshold.
Finally the authors discussed two approaches for refining the agglomeration process. Firstly, the cluster agglomeration is based on a successive merging operations of the two closest patterns given the SNN measure.

Secondly, a large number of clusters may be initially formed by SNNO with a high threshold and then the similarity measure is adapted to reflect a group-to-group similarity.
A revised version of SNNO, SNNOR, was presented in [5]. The computational complexity for the nearest neighbour search is reduced to $NlogN$ by using a KD Tree structure. Then an inverted neighbourhood table is constructed, the similarity matrix is derived and the clusters are produced in the same way as in the SNNO algorithm.
However, this solution presents some issues. Firstly, KD Trees are not efficient when the number of dimensions is large. Secondly, the way the similarities are calculated does not respect Equation 3, which weakens the algorithm against noise.
SNNDB [6] is a combination of SNN and DBSCAN [7]. Once the SNN similarities are defined between patterns, the density for each pattern is set to be the number of patterns with a similarity larger than the radius parameter Eps. Patterns with a density greater than the threshold $MinPts$ are defined as *core points*, the remaining patterns farther than Eps from any *core point* are labeled as noise and discarded. Finally *core points* with similarities above Eps are grouped in the same cluster and the remaining data are assigned to the cluster of their closest *core point*.
As KD Trees are not suitable for high dimensional data sets, the authors proposed to use canopies [8] to retrieve the neighbourhood. The use of canopies involves creating small and overlapping groups of patterns with a simple metric and performing the neighbourhood query only in the groups in which the patterns appear. A very similar solution is proposed in [9].
SNNDB2 [10] aimed to extend SNNDB both for outlier detection and *core point* selection. Outliers are identified and removed based on nested loops in conjunction with randomisation and a pruning rule [11]. Then very close *core points* are pruned to limit the number of *core points* per cluster.
In [12], the authors addressed the problem of low-cardinality in high-dimensional data. They presented a top-down hierarchical clustering algorithm based on the shared farthest neighbours, SFN. They argued that using the farthest patterns is more robust than using the nearest patterns, as greater distances are less sensitive to perturbations and produce more stable partitions. As the algorithm has $O(n^2)$ memory requirement, it is not suitable for large data sets.
A multi-cluster combiner, SNNC, is proposed in [13] and uses shared neighbour similarity to create a graph where edges exist between vertices if vertices appear frequently together and if vertices share a sufficient number of neighbours. Finally the graph is partitioned into K clusters such that the min-cut is minimised. Experimental results show that this method improves previous consensus-based clustering, especially for unbalanced data.

$$S_{PD}(i,j,k) = \sum_{z=0}^{k} \sum_{z'=0}^{k} \left((k+1-zd(i, N_k(i)[z])) * (k+1-z'd(i, N_k(j)[z'])) \right) \tag{4}$$

In [14] the authors applied the SNNH algorithm for image clustering. They modified the similarity function (Eq. 2) to take into account the underlying similarity measure. S_{PD} (**P**roduct and **D**istance) is defined in Eq. 4 and S_{SD} (**S**um and **D**istance) may be constructed similarly. They noted that this extended version performs slightly better than the original one to the cost of an additional access to the similarity measure.

3 Toward an Efficient Hierarchical SNN

We present an agglomerative clustering, SNNHLM that differs from SNNOH in three aspects.
Our algorithm extends the first approach discussed in [4] : given similarities between patterns and an agglomeration method, we merge successively the two nearest clusters (See Algo 1).

Algorithm 1 $SNNHLM(Patterns\ P, Int\ K, Distance\ d)$

$N \leftarrow$ nearestNeighborSearch(P, K, d) #
$S \leftarrow$ ComputeSimilarities(N) # Individual similarities are sorted
$Top \leftarrow$ sort(maxValueOfEachRow(S)) # Keep record best value per row
$H = \emptyset$ # Hierarchy
while $Top \neq$ empty **do**
  $(a, b) \leftarrow closestCluster(Top)$ # Pop element from the stack
  $S_a \leftarrow mergeSimilarity(a, b)$ # Merge S_b in S_a and discard S_b
  if $|S_a| > k$ **then**
    $S_a \leftarrow prune(S_a, k)$ # Constant-size constraint
  end if
  for all $e \in N_a$ **do**
    $S_{a,e} = S_{e,a} \leftarrow updateSim(a, b, e)$ # Update similarity value
    $Top \leftarrow updateBest(S_e)$ # update pos of e in the best list
  end for
  $Top \leftarrow updateBest(S_a)$ #update pos of a in the best list
  $H \leftarrow$ insert(a, b) # update the hierarchy
end while
return H

In the original paper, authors stated that both the memory and computational requirements are of $O(n^2)$ and $O(n^3)$ complexity respectively, which make this method *prohibitive*. However, with the use of proper data structures and Eq. 3, the complexity is reduced to $O(nk)$ and $O(nklog(n))$ respectively (without considering the knn queries).
We used NB-Tree [15] to perform the k-nearest neighbours searches. NB-Tree maps data to a 1 dimensional line and then uses B^+-Tree to store the data. The KNN search is performed in three steps (see Figure 2) : 1) the data norm is extracted, 2) the bucket that contains the norm is returned and 3) while not enough points have been explored, the search is performed in both directions with the use of circular linked lists.

In the worst case, the KNN search in NB-Tree is performed in $O(d + ndlogk)$, being of similar complexity of a naive search. Experiments show that its complexity is very data dependent. However, recent study [16] showed that distributed computations of the KNN queries may be achieved with nearly optimum efficiency. Therefore, KNN computations should not be considered as an issue.
Initially, the maximum size of the similarity matrix S is nk : each pattern has k neighbours and a similarity measure can only be defined to these k neighbours, then at least $n - k$ measures are equal to zero and may be discarded by using a sparse matrix. The size of this matrix is guaranteed to decrease through the agglomeration process. Consider the merge of two clusters c_i and c_j, with S_i and S_j respectively their similarity list, the resulting cluster $c_{i \cup j}$ will be related to at most $|S_i| + |S_j| - 2$ other clusters and as S_i and S_j are not used anymore, the memory requirement after the merging will be lower, and thus it is bound to nk. That initial modification is called *SNN**H***, as we focus on the **H**ierarchical operations.

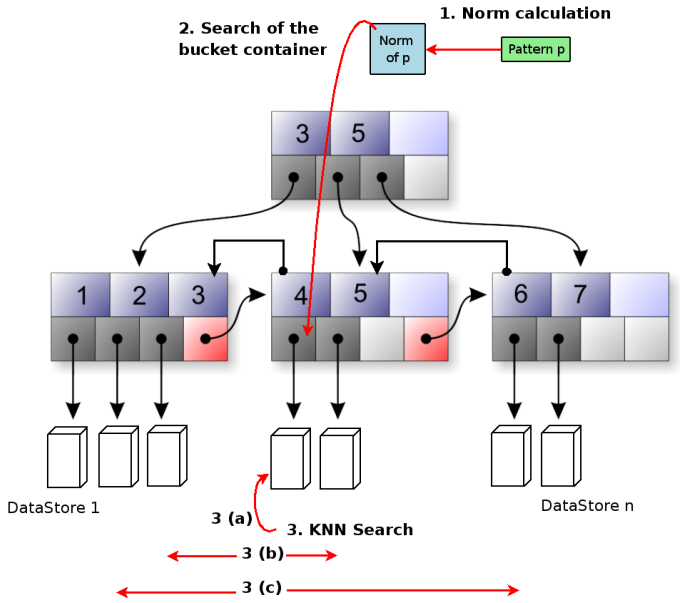
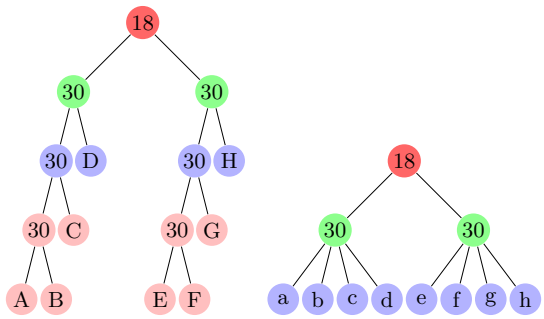


Fig. 2. Knn query on NB-Tree

For each iteration, two main tasks are required : the search and the agglomeration of the two closest clusters.
The closest clusters search complexity may be reduced to $log(n)$ by the use of a stack with construction cost of $nklog(k) + nlog(n)$.

The agglomeration process requires the merge of two neighborhood lists and two similarity lists. The internal similarity list is then updated to reflect the agglomeration objective. In this paper, only the single link is used, but contrarily to SNNOH, other linkage metrics such as the group average or the farthest link may be selected instead.
Without Eq. 3, we will potentially make n external updates, although with, we only need to update the related neighbours of the merged clusters. However we have no guarantee that the successive merges will not have a neighbuorhood larger than k. Consequently, in order to maintain constant the neighborhood size, we added a post-merging constraint : if the resulting cluster has a neighbourhood larger than k, we prune it to keep only the k most similar clusters.
We chose to keep the similarity matrix consistent by removing the symetrical relation that may have been deleted. We called this version that **L**imits the number of neighbours through the process *SNNH**L***.



(a) SNNHL (b) SNNHLM

Fig. 3. Hierarchy size reduction

Finally, we proposed another variation : SNNHL**M** that is able to merge **M**ultiple sub-clusters at each iteration. While the $i-th$ similarity obtained from the stack is identical to the first similiarity returned, we include it in the merging operation. This technique is able to reduce the hierarchy size by removing uninteresting merges, *i.e.* the merges that cannot be distinguished. Figure 3 shows an example with the similarities of Table 3. The number in the node is the similarity. Note that there is no reason of doing binary merges as the similarity is constant for various levels.
The multiple merges are therefore not desctructive. Moreover, this modification highlights how the similarity measure may impact on the hierarchy size. Scattered similarity values will induce nearly binary merges while similarities defined on a smallest domain may produce a more condensed structure.

4 Experiments

We compare the three versions of our algorithm with AGNES, the standard agglomerative clustering, and the original SNNOH algorithm. We perform tests on two simulated and six realworld datasets obtained from the UCI-ML repository [17] (see Table 4).

name	Number of patterns	Number of dimensions	Number of classes
simulated0	322	2	7
simulated3	8000	2	6
mammographic	961	5	2
pendigits	7494	16	10
letter recognition	20000	16	26
landsat	4435	36	6
optdigits	3823	64	10
isolet	6238	617	26

Table 4. Datasets used in experiments

We used FMeasure and Purity to assess the quality of the clusters produced. Purity measures the proportion of patterns of the majority class in a cluster. F-Measure is the harmonic mean of precision and recall. It describes, to what extend a cluster contains objects of a particular class (precision) and all objects of that class (recall). In both measures, a higher value means the result is of better quality.

4.1 Similarity function comparison

We first present a comparison between five different measures between individual patterns : unweighted (U), sum (S), product (P), sum combined with distance (SD) and product combined with distance (PD). These measures have an impact both on quality and on the hierarchy size.

From a quality point of view, we found that the measures S_{UW}, S_S and S_P have very similar behaviours on the datasets tested. When used on different versions of our algorithm, they produce nearly identical results. However, S_{SD} and S_{PD} differ significantly from the previous similarity measures. This is because S_{SD} and S_{PD} give more importance to the distance measure used. Unlike to [14], we did not found that the use of the distance impacts highly on our algorithm. These differences are highlighted in the Figure 4. In this figure we can see that S_{UW}, S_S and S_P have the same performance, while S_{SD} and S_{PD} have very distinct behaviours.

The impact on hierarchy size is induced by the diversity in the measures produced and occured only with methods that may agglomerate multiple clusters at a time (SNNOH and SNNHLM). S_{UW} can only produce k

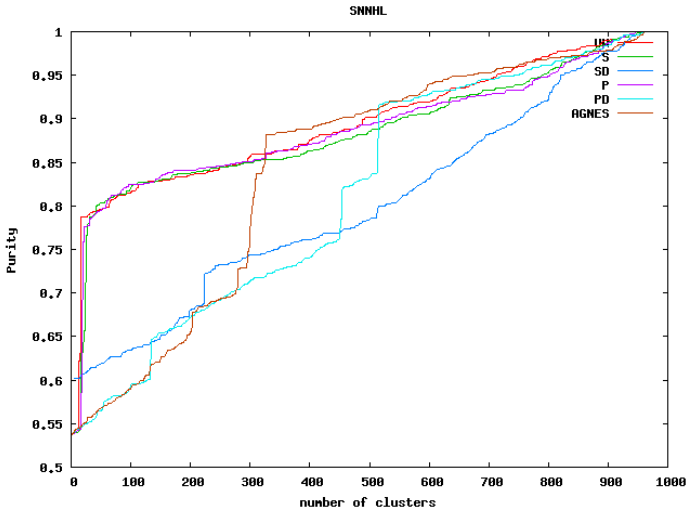
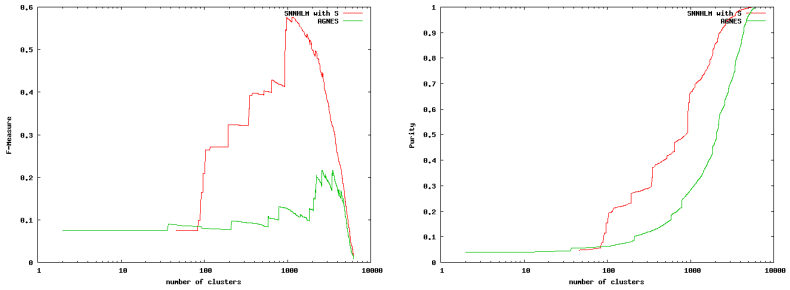


Fig. 4. Purity for mammographic dataset with SNNHL

different measures (k is the number of neighbours) and consequently, the hierarchy size may be reduced to k levels. However, S_{PW} involves product of ranks and distances, and then may potentially produces $\frac{nk}{2}$ different similarity values (recall that the measure is symmetric).
The choice of the similarity measure on individual patterns has an impact though all the process and has to be chosen carefully. When no specific knowledge is available, we argue that simple measures such as UW, S or P should be used, because they do not involve overhead and enable the hierarchy size to be based on the number of neighbours only. When a meaningful distance is available, it should be used instead.

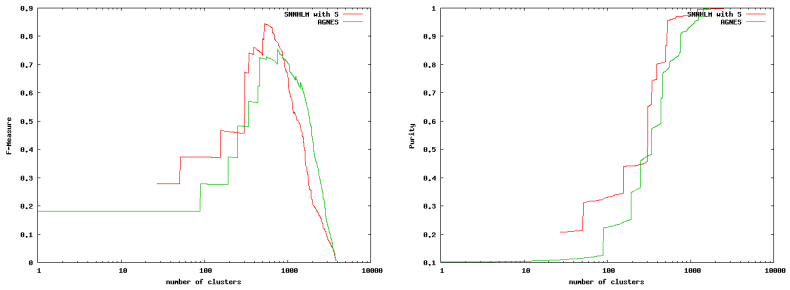
4.2 Quality results

Figure 5 shows the F-Measure for the ISOLET dataset. This data set is very challenging as it contains more than six hundred dimensions. The results show that our algorithm performs better than AGNES both in terms of purity and F-Measure. However, both algorithms do not perform well when the number of clusters becomes small. This is due to the agglomeration strategy used in these experiments. We rely only on the single-link, e.g. the best similarity induced by two individual points from two different clusters. With a more complex agglomeration scheme, like border agglomeration where the similarity between two clusters is not based only on one point but on the border points of the cluster frontier, results should improve.



(a) FMeasure (b) Purity

Fig. 5. Results on ISOLET dataset



(a) FMeasure (b) Purity

Fig. 6. Results on Optidigits dataset

Figure 6 shows the results obtained from the optidigits dataset. Similarly, SNNHLM outperform AGNES both for F-Measure and Purity.
The differences between SNNOH and the revised versions are of two different types. Firstly, the hierarchy is complete for both SNNH and SNNHL and may be reduced for SNNOH and SNNHLM. A complete hierarchy enables experts to navigate and select any level they need to inspect. However, the number of levels may be too large to be efficiently used or explored. It is up to the end-user to select the type of hierarchy. However, unlike SNNOH where a threshold is required to build the hierarchy, we do not need it for SNNHLM. Moreover the multiple simultaneous merges are not destructive because all of them have identical coherence. Secondly, our versions can be combined with any agglomeration scheme as we are using linkage metrics to group clusters instead of relying only on the underlying similarity measure. The line 12 in the SNNHL algorithm (Algo 1) shows how the similarity may be

updated. In single link, the best similarity toward a neigbour is kept, but other schemes allow to update the similarity to reflect both the similarity between clusters given the cluster sizes. Finally, one major issue for SNNOH is that it needs some parameters to merge successive clusters with similarity values above the threshold. Setting the parameters may not be obvious if the similarity function is combined with the distance. If SNNOH is transformed to keep the record of all different similarity levels between the clusters, then it will behave nearly (without updates) as SNNH. SNNHL limits the number of neighbours a cluster may have, from the beginning to the end of the process. Consequently, some relationships are lost when the post-merging constraint is applied. Finally SNNHLM makes the merging of multiple clusters possible.

5 Conclusions

We presented a revised version of SNN, a hierarchical clustering based on the shared nearest neighbour similarity. Our version uses NB^+Tree to efficiently perform the searches for KNNs, it is able to merge multiple clusters in one pass, it limits the neighbourhood of agglomerated clusters, and it uses efficient structures to perform the agglomeration. We have shown that this approach is able to process high-dimensional data sets and produce good results compared to methods that rely only on distance measure only.
We plan to extend our method with a hybrid agglomeration scheme (single-border link) to improve its efficiency when the number of clusters become very small. We also plan to propose a distributed version of this algorithm to be able to process very large data sets in terms of both dimensions and number of instances.

References

1. Rui Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.
2. L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
3. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
4. R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Trans. Comp.*, C(22):1025–1034, 1973.
5. I. Hofman and R. Jarvis. Robust and efficient cluster analysis using a shared near neighbours approach. *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, 1:243–247 vol.1, Aug 1998.
6. Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *In Proceedings of Third SIAM International Conference on Data Mining*, 2003.

7. Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.
8. Andrew Mccallum, Kamal Nigam, and Lyle Ungar. L.h.: Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*, pages 169–178, 2000.
9. Hong-Bin Wang, Yi-Qing Yu, Dong-Ru Zhou, and Bo Meng. Fuzzy nearest neighbor clustering of high-dimensional data. *Machine Learning and Cybernetics, 2003 International Conference on*, 4:2569–2572 Vol.4, Nov. 2003.
10. Jian Yin, Xianli Fan, Yiqun Chen, and Jiangtao Ren. High-dimensional shared nearest neighbor clustering algorithm. In *FSKD (2)*, pages 494–502, 2005.
11. Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38, New York, NY, USA, 2003. ACM.
12. Stefano Rovetta and Francesco Masulli. Shared farthest neighbor approach to clustering of high dimensionality, low cardinality data. *Pattern Recogn.*, 39(12):2415–2425, 2006.
13. Hanan Ayad and Mohamed S. Kamel. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In Terry Windeatt and Fabio Roli, editors, *Multiple Classifier Systems*, volume 2709 of *Lecture Notes in Computer Science*, pages 166–175. Springer, 2003.
14. Pierre-Alain Moéllic, Jean-Emmanuel Haugeard, and Guillaume Pitel. Image clustering based on a shared nearest neighbors approach for tagged collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 269–278, New York, NY, USA, 2008. ACM.
15. M.J. Fonseca and J.A. Jorge. Indexing high-dimensional data for content-based retrieval in large databases. *Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. Eighth International Conference on*, pages 267–274, March 2003.
16. Erion Plaku and Lydia E. Kavraki. Distributed computation of the knn graph for large high-dimensional point sets. *J. Parallel Distrib. Comput.*, 67(3):346–359, 2007.
17. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

Introduction Du Data Mining Pour L'Amélioration De La Recherche D'images Par Le Contenu

Djerroud Souhila [1,1], Zaoui Lynda [1],

Université des sciences et de la technologie - Mohamed Boudiaf (U.S.T.O. M.B.)
B.P. 1505 El-Mnaouer, Oran, 31036, Algerie.
[1,1]dsii_umd@yhoo.fr, [1].Zaoui_Lynda@yahoo.fr

Résumé : Les méthodes de recherche d'images par le contenu procèdent dans un premier temps à une indexation dans une base d'images. Il s'agit d'extraire des descripteurs de bas niveaux (couleur, texture, forme.) Afin de pouvoir établir un critère de similarité entre elles, ensuite, pour chaque requête de l'utilisateur, une distance est calculée en fonction de ces descripteurs et les images les plus proches de la requête sont retournées. Le principal objectif de cet article est de présenter une méthode de segmentation issue du domaine du Data Mining permettant de regrouper les images d'une base généraliste en groupe d'images similaires qui seront stockées par une structure arborescente particulière : « Arbre Quaternaire générique » permettant la minimisation de l'espace de stockage par partage d'informations entre arbres quaternaires représentants les images. Cette méthode est une adaptation de la méthode de segmentation hiérarchique appelée CHAMELEON et appliqué aux images, elle nécessite la définition d'une mesure de dissimilarité (ou similarité) entre images représentées en arbre quaternaire.

Mots Clé : Base d'images ; arbre quaternaire générique ; méthode de Segmentation ; segmentation hiérarchique; CHAMELEON.

1 Introduction

La nature des documents numériques a profondément évolué au cours des dernières décennies. Quelque soit leurs type (texte, image, son,…) les moyens de création, de duplication et de transmission se sont rapidement développés, conduisant leurs nombres à s'accroître considérablement. En conséquence, la gestion, la recherche et l'exploration de ces documents nécessitent des moyens plus performants afin de décrire ces derniers en fonction de leur contenu multimédia notamment visuel.
La navigation a pour but de répondre au problème de la recherche d'un document précis ou d'un type de document en catégorisant et structurant la base à laquelle il appartient. Les méthodes de recherche d'images par le contenu procèdent, dans un premier temps, a une indexation de la collection, autrement dit, à l'extraction des descripteurs de bas niveaux des images (couleur, texture, forme,...) afin de pouvoir établir un critère de similarité entre elles. Pour chaque requête de l'utilisateur, une distance de similarité est calculée en fonction de ces descripteurs et les images à priori inconnues. Deux contraintes principales sont imposées par le problème de l'indexation

et la recherche d'images par le contenu: la première concerne le temps de calcul, du fait que le calcul de similarité se fait en parcourant toute la base d'images. La deuxième concerne la taille de stockage de la base d'images.
Plusieurs méthodes de classification de base d'images existent, et apportent une aide précieuse dans la résolution de ce type de problème en découpant la base en groupes d'images similaires. Une fois que la base d'images est partitionnée, les systèmes de visualisation choisissent de définir une image représentative pour chaque groupe, plutôt que de représenter la totalité de la base. Ainsi, l'utilisateur peut naviguer rapidement et efficacement dans la base d'images. De manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer une image à une classe. Ces problèmes se déclinent essentiellement en deux variantes : la classification dite « supervisée » et la classification dite « non supervisée ».Dans l'approche de la classification supervisée, les classes existent à priori. Par opposition, dans l'approche de classification « non supervisée », on dispose au départ d'un ensemble d'objets non étiquetés. A partir de ces données, l'idée est de parvenir à détecter des objets similaires afin de les regrouper. Un clustering (segmentation) sera jugé satisfaisant si on obtient en sortie de la méthode des groupes d'images similaires [1], [2], [3].

Le terme de Data Mining signifie littéralement: forage de données .Comme dans tout forage, son but est de pouvoir extraire des connaissances à partir des données qui peuvent être stockées dans des entrepôts, dans des bases de données distribuées ou sur Internet (Web Mining). Ces concepts s'appuient sur le constat qui existe au sein de chaque entreprise des informations cachées dans le gisement de données ; ils permettent grâce à un certain nombre de techniques spécifiques, de faire apparaître des connaissances [4], [5], [6]. Le Data Mining offre des moyens pour aborder les corpus en langage naturel (Text Mining), les images (Images Mining), le son (Sound Mining) ou la vidéo et dans ce cas on parle plus généralement de Multimédia Mining.

Dans cet article, nous nous intéressons à l'une des méthodes de segmentation du data Mining [7] appelée CHAMELEON. Cette dernière est réalisée sur des images stockées en arbre quaternaire Générique. Cette structure minimise le stockage des images similaires organisées en arbre quaternaires par partage de partie communes entre elles.

2 Méthodes de segmentation

La segmentation est la division des données dans des groupes d'objets similaires. Chaque groupe, appelé segment (cluster), est composé d'objets similaires entre eux et dissimilaires des autres groupes d'objets. Elle consiste à former des groupes homogènes à partir d'un ensemble de données hétérogène ainsi, il n'y a pas de classe à expliquer ou de valeur à prédire définis à priori, Il appartient ensuite à un expert du domaine de déterminer l'intérêt et la signification des groupes ainsi constitués

1.1 Principe de la segmentation

Le principe général de tout système de clustering est de maximiser la similarité intra-classe (à l'intérieur d'un cluster) et de minimiser la similarité inter-classe (entre cluster). Traditionnellement, la segmentation est généralement divisée en hiérarchique et partitionnement. Les algorithmes de partitionnement démarrent avec un nombre fixe de segments. Leur principe est de découvrir des segments par transfert d'itération de points entre groupes, ou d'identifier les segments qui ont une haute densité des données. Pour les algorithmes de la première classe on cite (PAM, CLARA et CLARANS) [8]. Par opposition aux algorithmes de partitionnement, Les algorithmes hiérarchiques construisent les segments graduellement. La segmentation hiérarchique est aussi subdivisée en agglomération et division. BIRCH, CURE et CHAMELEON sont des algorithmes bien adaptés au data mining.

1.2 La Segmentation Hiérarchique

En principe, il existe deux classes de méthodes de segmentation hiérarchique: *la segmentation descendante:* Qui démarre avec un cluster réunissant tous les objets, puis va diviser les clusters jusqu'à ce qu'un critère d'arrêt soit satisfait, *la segmentation ascendante:* Dont le but est de former une hiérarchie de clusters, telle que plus on descend dans la hiérarchie, plus les clusters sont spécifiques à un certain nombre d'objets considérés comme similaires; Afin d'apporter des améliorations pour le clustering hiérarchique ascendant plusieurs algorithmes ont été définis dans le domaine du data mining. Tels que: BIRCH (Balanced Iterative Reduced and Clustering using Herarchises) [9], CURE (Clustering Using Representatives) [10] et CHAMELEON (A Hierarchical Clustering Algorithm Using Dynamic Modelling) [7].

L'algorithme de BIRCH n'est pas complexe mais il ne permet pas de détecter les clusters de formes ou de tailles différentes. D'un autre coté CURE ne considère pas l'inter-connectivité des deux clusters, et par conséquent il peut y avoir une pénalisation de deux groupes homogènes qui ont des points non condensés. Pour remédier à ces inconvénients on doit utiliser une distance Inter-connectivité et proximité relative pour prendre en compte le rapport de dispersion des points à l'intérieur d'un groupe. Pour cela, la méthode CHAMELEON proposée consiste en la maximisation de ses deux mesures de similarité entre deux clusters.

Dans ce travail, nous intégrons la méthode CHAMELEON dans un système d'indexation et de recherche d'image par le contenu qui a été développé en 2005,

appelé REQUIT [11]. Ce dernier permet de représenter chaque image par un arbre quaternaire [12] dont le critère de découpage est l'homogénéité de la couleur, ce qui permet de stocker l'ensemble des images en arbre quaternaire générique. Cette structure [13] minimise l'espace de stockage, par partage de parties communes entre images. Différentes opérations sur l'image où entre images sont proposées par ce système (la recherche globale, par région, par niveaux).

2 L'Arbre Quaternaire

C'est une structure hiérarchique, dite aussi quad-tree, construite par division récursive de l'espace en quatre quadrants disjoints de même taille [12], en fonction d'un critère de découpage (exemple : homogénéité de la couleur) de telle sorte que chaque nœud de l'arbre quaternaire représente un quadrant dans l'image. L'image entière est le nœud racine, si une image n'est pas homogène par rapport au critère de découpage, le nœud racine de l'arbre quaternaire représentant l'image a quatre fils représentant les quatre premiers quadrants de l'image, un nœud d'arbre quaternaire est dit feuille, si le critère d'homogénéité est vérifié sinon le nœud est interne. La similarité entre images est calculée en fonction des trois distances T, Q, V [11].

Exemple

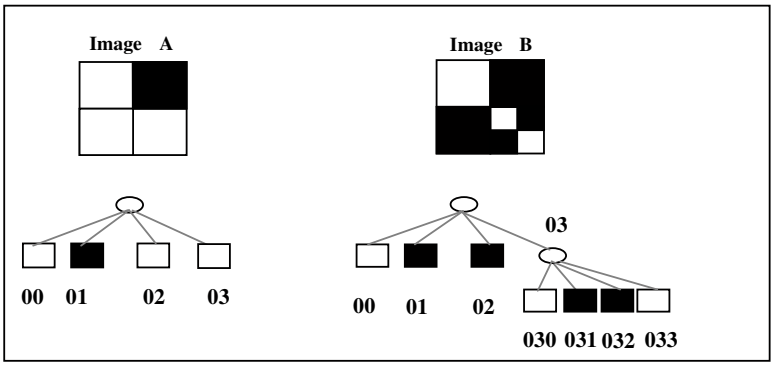


Fig. 1. Exemples de représentations des images en arbre Quaternaire.

2.1 Définition De La Distance Entre Images

La distance Δ est une distance entre images représentées par des arbres quaternaires. La distance Δ entre deux images i et j est définie par une somme de distance δ_k (i,j) entre les nœuds des arbres quaternaires représentant les images i et j, pondérées par des coefficients $C_k > 0$:

$$\Delta (i, j) = \Sigma C_{\mathbf{k}} \delta_{\mathbf{k}} (i, j) / \Sigma C_{\mathbf{k}} \qquad (1)$$

-δk(i,j) est une distance normalisée entre les nœuds homologues **k** des arbres quaternaire **i** et **j** définie précédemment.

- **k** est l'identificateur d'un nœud pris parmi l'union des identificateurs de nœuds apparaissant dans les arbres quaternaires des images **i** et **j**.

- $\mathbf{C_k}$ est un coefficient positif représentant le poids du nœud k dans le calcul de la distance $\mathbf{C_k}$. Chaque poids $\mathbf{C_k}$ est choisit selon l'importance qu'on souhaite donner à certains quadrants d'image par rapport à d'autres dans le calcul de la distance $\mathbf{\Delta}$ (on peut même donner à l'utilisateur la possibilité de choisir lui-même les poids de certains quadrants). Par exemple, si certains quadrants ne doivent pas apparaître dans le calcul de $\mathbf{\Delta}$ (le cas de la distance par région ou par niveau), alors ils peuvent être associés à un poids nul. Si la surface des quadrants doit entrer en jeu dans le calcul de $\mathbf{\Delta}$, alors chaque coefficient $\mathbf{C_k}$ doit être proportionnel à la surface représentée par le quadrant par rapport à l'image entière.

2.1.1 Cas particuliers de la distance Δ

En fonction des différents poids Ck associés aux nœuds et de la distance choisie entre les nœuds, plusieurs familles de distances peuvent être définies à partir de la distance Δ. Nous définissons deux familles de distances basées sur la structure des arbres quaternaires, appelées T-distance (T pour Tree) et Q-distance (Q pour quadrant), et une famille de distances de similarité visuelle entre images, appelée V-distance (V pour visuel). Les deux premières familles de distances comparent les arbres quaternaires représentants les images. La dernière famille compare visuellement les images en utilisant leur représentation en arbre quaternaire [13].

3 Méthode CHAMELEON

C'est un algorithme de classification hiérarchique agglomérative qui utilise une modélisation dynamique pour l'agrégation de classes. A travers cet aspect dynamique, le but est de pouvoir retrouver des classes avec des formes irrégulières et des tailles différentes, Il s'agit de créer un ensemble initial important de clusters, puis les fusionner selon une nouvelle mesure. L'algorithme possède deux étapes (phases).

Phase 01. Une matrice de similarité est calculée. Celle-ci regroupe toutes les similarités entre les images prises deux à deux. A partir de cette matrice, un graphe des **k** plus proches voisins (k-ppv) sera, dans un premier temps, construit. Il existe un lien entre l'image **i** et l'image **j** si et si est seulement si l'image **i** fait partie des k-ppv de l'image j et l'image j fait partie des k-ppv de l'image **i**. La valeur de **k** étant un paramètre de l'algorithme. Le graphe des k-ppv permet de diminuer le temps de calcul pour la suite de l'algorithme, car seule une partie des données initiales est utilisée. Une fois le graphe des k-ppv crée, les auteurs appliquent dans un second temps un algorithme de partitionnement de graphe appelé Hmétis pour créer un ensemble de classes initiales.

Phase 02. Consiste en une combinaison itérative de l'ensemble de classes initiales en utilisant un nouvel algorithme de classification hiérarchique. Cet algorithme repose sur deux mesures pour regrouper deux classes: l'inter connectivité relative (Relative Interconnectivity ou **RI**) et la proximité relative (Relative Closeness ou **RC**).

Inter-connectivité Relative RI : L'inter-connectivité relative entre deux classes est la valeur absolue de l'inter-connectivité entre ces deux classes normalisées ($EC_{\{ci, cj\}}$) par la moyenne arithmétique de l'inter-connectivité de chaque classe (EC_{ci}). Ainsi l'inter-connectivité entre deux classes C_i et C_j est définie de la façon suivante :

$$RI\ (C_i, C_j) = \frac{\left|EC_{\{C_i, C_j\}}\right|}{\frac{1}{2}\left(\left|EC_{C_i}\right| + \left|EC_{C_j}\right|\right)} \tag{2}$$

EC $\{c_i, c_j\}$: Somme des arcs du graphe des k-ppv regroupant les classes.

EC_{ci}: Somme minimum des arcs pour diviser en deux la classe C_i.

Proximité relative RC : Les concepts évoqués pour la proximité relative sont analogues à ceux définis pour l'inter-connectivité relative. La proximité absolue entre deux classes C_i et C_j est la moyenne pondérée des arcs (alors que l'inter-connectivité absolue est la somme des arcs) qui connectent un élément de C_i a un élément de C_j.

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{Ci,Cj\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{Ci}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{Cj}}} \tag{3}$$

$|C_i|$: Nombre d'éléments de la classe C_i.

$S_{EC\ \{Ci,Cj\}}$: **Moyenne** des poids des arcs connectant les sommets de C_i à ceux de C_j;

S_{ECci} : Moyenne des poids des arcs reliant les deux sous classes du cluster C_i.

Pour regrouper deux classes en utilisant ces deux mesures, plusieurs approches sont possibles, pour notre part, nous utilisons une approche qui maximise une fonction combinant les deux mesures, donnée par

$$RI\ (Ci, Cj) \times RC\ (Ci, Cj)^{\alpha} \tag{4}$$

Le but de cette fonction consistant à regrouper deux classes en fonction des deux mesures. La valeur de α va accroître l'importance d'une mesure par rapport à l'autre. Si **α=1**, alors les deux mesures ont la même importance. Par contre, si on veut donner plus d'importance à la proximité relative, alors en prend **α > 1**; pour **α < 1** alors on donne plus d'importance à l'inter-connectivité relative.

1.4 Application de CHAMELEON

L'exemple suivant va faire l'objet de démonstration de l'adaptation de la méthode CHAMELION dans notre prototype. L'exemple ne contient que 11 images satellites (Fig.2), par la suite nous donnons des résultats pour des bases d'images de tailles plus importantes.

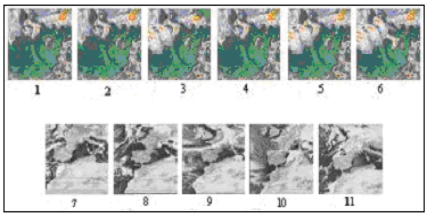


Fig. 2. Exemple d'une base d'images généraliste.

Dans notre prototype, on peut dérouler la méthode avec l'une des trois distances T, Q ou V pour calculer la distance entre une paire d'images i et j, représentée sous forme d'une matrice et de réaliser des opérations telles que la construction du graphe k-ppv, à partir duquel le graphe de partitionnement est conçu et représente en réalité les classes initiales.

Lorsque la matrice des similarités est calculée, on peut tracer le graphe k-ppv. Dans notre application le graphe est représenté sous forme d'un tableau (Tableau.1) où chaque ligne représente une classe (une image avec ses k plus proches images). La matrice des distances nous permet de construire le graphe des k-ppv illustré dans la Fig.3 avec k=2.

Tableau.1. Les Classes du graphe 2 – ppv.

Classes	Images	1ier voisin	2iéme voisin
C1	1	2	4
C2	2	4	1
C3	3	5	2
C4	4	2	1
C5	5	3	6
C6	6	5	3
C7	7	10	9
C8	8	9	10
C9	9	8	10
C10	10	9	11
C11	11	10	8

Comme le montre la Fig.3 il existera un lien entre deux images s'il fait partie de ces deux plus proches images à titre d'exemple : pour l'image 1 les deux plus proches images sont l'image 2 et 4. (arc en bleu).

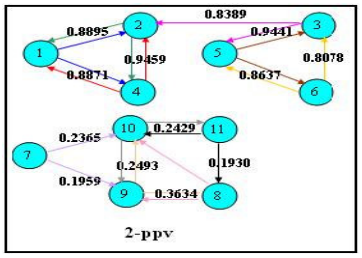


Fig. 3. Graphe 2-ppvs du tableau 3.2.

Chaque couleur d'arc dans la Fig.3 représente une classe dans le graphe 2-ppvs.

On passe maintenant à la construction des classes initiales (voir la Fig.4), qui illustre le principe de partitionnement entre quatre classes du graphe 2-ppvs de la Fig.3.

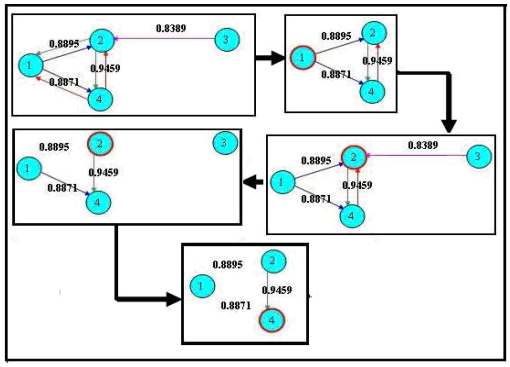


Fig.4. Application d'algorithme de partitionnement sur le graphe de la Fig. 3.

La Fig.5 Présente quatre classes inter connecté entre elle (arcs vert, bleu, rouge, et rose). L'algorithme de partitionnement parcours les classes du 2-ppvs pour chaque classe on vérifie la présence de ses éléments dans les autres classes, élément par élément.

Commençant par la classe 1 **(a)** ayant les images 1, 2 et 4 (un lien entre 1 et 2, 1 et 4) ; on sélectionne le premier élément de la classe en cours ; on constate que l'élément (1) appartient à 3 classe (arcs bleu (1, 2), arcs vert (2, 1) , rouge(4 ,1)) ; on a trouver deux liens plus fort ;le lien (1, 2) de la classe en cours dont il est propriétaire et le lien (2, 1) de la deuxième classe (arc en vert) qui sont identique(0,8895) ; dans ce cas il sera éliminé de la $2^{iéme}$ classe est maintenu dans ça propre classe **(b)**. Concernant le deuxième élément (2) il appartient à quatre classes différentes **(c)** (arcs vert, bleu, rouge, rose) deux liens plus fort trouvé, le lien (2, 4) arc en vert et le lien (4, 2) arc en rouge qui ont la même valeur (0,9454). On garde le lien (2, 4) arc en vert de la classe dont l'élément (2) est propriétaire. Le troisième élément (4) appartient à

deux classe **(d)** (arc bleu et vert), le lien le plus fort est (2, 4) de la deuxième classe **(e)** qui sera maintenu avec la valeur 0,9454.

On réitère le même procédé pour les autres classes du graphe 2.-ppv. Les classes initiales résultantes. Sont présentées dans Fig.5.

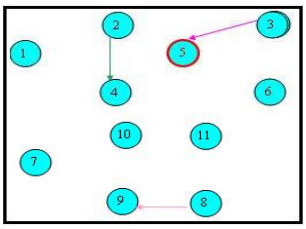


Fig. 5. Les classes initiales sous forme de graphe.

Afin de fusionnée les paire de clases les plus proches, on calcule les deux mesures de similarité l'inter-connectivité relative et la proximité relative.

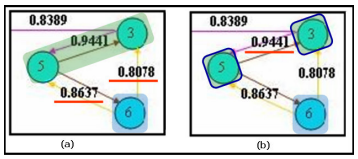


Fig. 6. Deux classes initiales.

Pour calculer l'inter-connectivité relative on doit d'abord calculer l'inter-connectivité absolue et l'inter connectivité interne :

*Inter - connectivité absolue***. :** Est calculer entre toutes les paire de classe ; pour la paire de classe C_3, C_4 ; C'est la somme des poids des arcs [(5, 6) et (6, 5)] et [(6. 3)] qui les relies de la fig.5 (a) : EC $\{c_3, c_4\}$ = (0.8637 * 2) + (0.8078) = 2.535.

Inter-connectivité intern. : Chaque classe étant coupée en deux sous-classes de taille identique, le calcule de l'inter-connectivité interne consiste a additionnées les poids des arcs reliant les deux sous-classes ; A titre d'exemple, la classe 3 contient deux élément ces deux sous-classes seront alors C_{31} avec l'élément (5) et la C_{32} avec l'élément (3) qui sont relier a travers deux arcs [(5, 3) et (3, 5)] de poids identique : ECc_3 = 0.9441 * 2 = 1.88. Par conséquence, si une classe ne comporte qu'un élément dans ça classe alors son inter-connectivité interne est égale à zéro, a titre d'exemple la classe (4) : $ECc_4 = 0$.

Pour calculer la proximité relative on doit d'abord calculer la proximité absolue et la proximité interne :

Proximité absolue : Par déduction la proximité absolue est la moyenne de l'inter-connectivité relative. $S_{EC\ \{C3,\ C4\}}$ = EC $_{\{C3,\ C4\}\,/}$ nombre d'arcs qui les relie =2.535 / 3 = 0.845.

Proximité interne : Même principe la proximité interne est la moyenne des poids des

arcs reliant les deux sous-classes. On peut aussi la calculer on fonction de l'inter-connectivité interne sur le nombre d'arcs qui relie les deux sous classes. La proximité interne de la classe (3) : S_{ECc3} = ECc_3 / nombre d'arcs qui relie les deux sous-classes =1.88 / 2 = 0.944.

Le calcule de la proximité relative se fait entre chaque paire de classes ; La proximité relative des deux classes C_3 et C_4 :

$$RC(C_3, C_4) = \frac{0.8389}{\frac{|3|}{|1|+|2|} * 0.944 + \frac{|1|}{|1|+|2|} * 0} = 1.343$$

Idem pour les paires de classes restante, pour décider de la fusion d'une paire de classes C_i, C_j on a utilisé l'approche qui maximise la fonction :

$RI\ (C_i, C_j) \times RC\ (C_i, C_j)^{\alpha}$ en fonction des deux mesures calculée précédemment. Pour notre exemple : $RI*RC_{\{C3, C4\}}^{\alpha} = 2.865 \times 1.343 = 3.606$ avec $\alpha = 1$.

Tableau. 2. Similarité a maximisé (matrice combinatoire).

$RI*RC^{\alpha}_{\{Ci,Cj\}}$	CI_1	CI_2	CI_3	CI_4	CI_5	CI_6	CI_7	CI_8
CI_1	- / -	5.292	0	0	0	0	0	0
CI_2	5.292	- / -	0.394	0	0	0	0	0
CI_3	0	0.394	- / -	3. 606	0	0	0	0
CI_4	0	0	3.606	- / -	0	0	0	0
CI_5	0	0	0	0	- / -	0.4314	0	0
CI_6	0	0	0	0	0.4314	- / -	1.956	0.4266
CI_7	0	0	0	0	0	1.956	- / -	0
CI_8	0	0	0	0	0	0.4266	0	- / -

Deux classes peuvent être fusionnée s'ils ont la même valeur maximale ; exemple des paires de classes ayant la même valeur maximale du Tableau.2, sont :($CI'_1 = \{CI_1, CI_2\}$, $CI'_2 = \{CI_3, CI_4\}$) ; par contre la valeur maximale de la classe CI_5 est de 0,4314 avec la classe CI_6 mais la valeur maximale de la classe CI_6 est avec la classe CI_7 de 1,956 et la classe CI_7 est aussi la même valeur maximale avec cette dernière donc la classe C_6 sera fusionnée avec la classe CI_7. Les classes résultantes de la première itération sont :

($\mathbf{CI'_1} = \{CI_1, CI_2\}$, $\mathbf{CI'_2} = \{CI_3, CI_4\}$, $\mathbf{CI'_3} = CI_5$, $\mathbf{CI'_4} = \{CI_6, CI_7\}$, $\mathbf{CI'_5} = CI_8$).

On arrête le processus de regroupement des classes intermédiaires une fois que la matrice combinatoire ne contient que des zéro ; pour cet exemple les clases finale sont obtenue au bout de la troisième itération Fig.7.

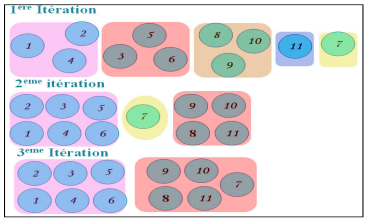


Fig. 7. Les classes résultantes de chaque itération.

4 Experiences, résultats et critiques

La méthode CHAMELEON est validée sur une base d'images représentées sous forme d'arbres quaternaires [11] la similarité entre deux images est définie par la distance (T, Q ou V) entre les arbres les représentants. Une fois que les groupes d'images sont conçus, chaque groupe d'images va faire l'objet d'un arbre quaternaire générique. La Fig. 8 montre un exemple d'application de notre prototype de base d'images. Les classes finales résultantes pour cet exemple sont présentées dans la Fig.9 (les classes (c) et (g)) sont représentées dans deux classes séparément, cela revient au choix de la valeur de k ; si on veut que les classes soient fusionnées, on augmente la valeur de k (Fig.10) ; la même chose pour les classes (d) et (h) de la Fig.9 qui sont aussi proches.

Après plusieurs tests, on a constaté que le graphe des k-ppv influe considérablement sur la construction des classes initiales. La construction du graphe k-ppv repose sur la matrice des similarités. Lors de sa construction l'algorithme prend pour chaque image les k premières images considérées comme proches sans prendre en compte que la distance de similarité soit faible ou forte. On remarque aussi que si le nombre k dépasse le nombre d'images similaires à une image donnée cela va influencer négativement sur la construction des classes initiales, et par conséquent, on peut trouver des images divergentes dans une même classe (voir la classe (g) de la Fig.10). Pour éviter cette faille, et afin d'améliorer cette méthode on a ajouté un seuil appelé seuilkppvs ayant pour objectif de limiter la plage des plus proches voisins. Ce dernier est vérifié lors de la construction du graphe k-ppv.

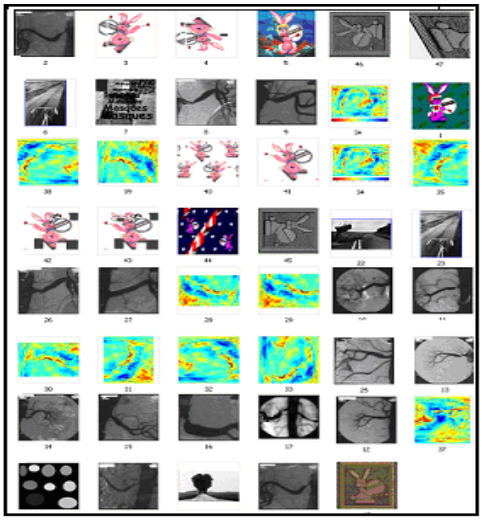


Fig. 8. Les classes résultantes de chaque itération

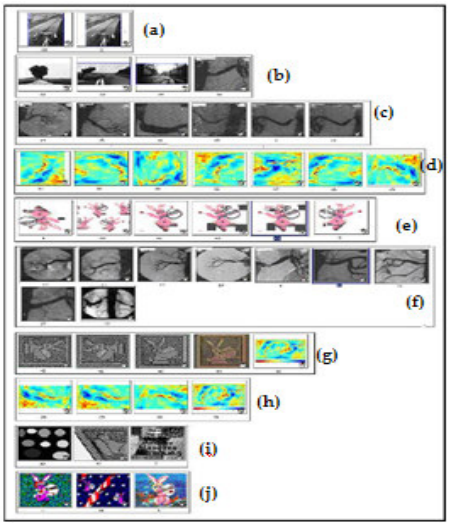


Fig. 9. Les classes finales avec k=2.

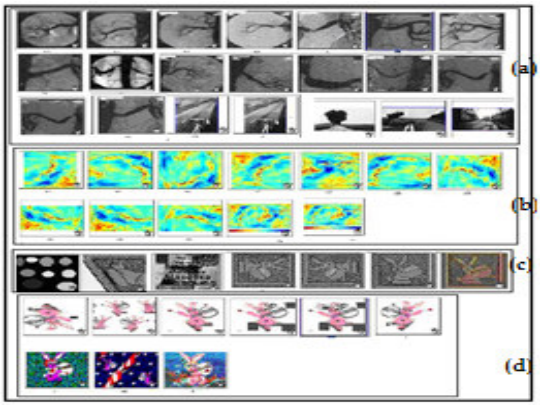


Fig. 10. Les classes finales pour k= 5 et seuilkppvs =40.

Plusieurs essais ont été appliqués sur notre prototype. La Fig.10 donne des résultats pour une base d'images. Des valeurs de k sont choisies de zéro à dix et, pour chaque valeur de k, elle est testée avec **10** valeurs différentes du seuil ***seuilkppvs.*** Le temps d'exécution de ces expériences prend seulement six minutes pour chaque expérience, sur un PC Pentium II, 400 MHz.

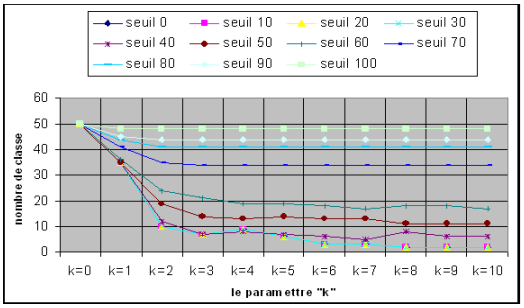


Fig. 11. L'influence du seuilkppvs et paramètre k sur les classes finales.

Le découpage de l'image en quadrant suivant l'organisation en arbre quaternaire se fait selon un critère particulier (ex : homogénéité de la couleur des pixels). Lorsque l'image est en niveau de gris, et/ou contient des nuances de couleurs, le découpage de l'image par ce critère peut aller jusqu'au niveau des pixels ; dans ce cas, la taille de l'images sera plus grande que la taille au format BMP (Fig.12, Fig.13). Pour remédier à cela, il est possible de fixer un seuil d'homogénéité pour limiter ce découpage.

Dans la Fig.13, un seuil d'homogénéité choisi a partir de 5, donne des tailles de fichiers au format QT plus petites que ceux enregistrés au format BMP(par rapport a l'image1). par contre, pour l'image 2,la Fig.13 indiques des tailles de fichier au

format QT plus petites que ceux enregistrés au format BMP pour un seuil d'homogénéité choisi à partir de 50.

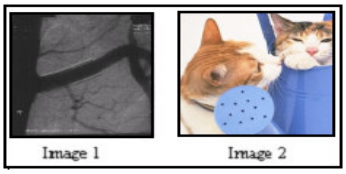


Fig. 12. Exemple d'images.

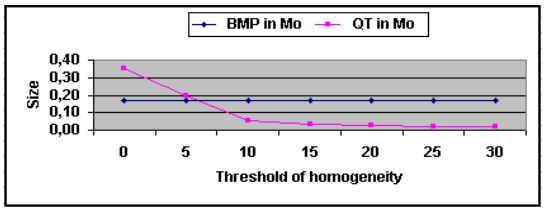


Fig.13. l'influence du seuil d'homogénéité sur la taille de l'image1 (Fig. 12).

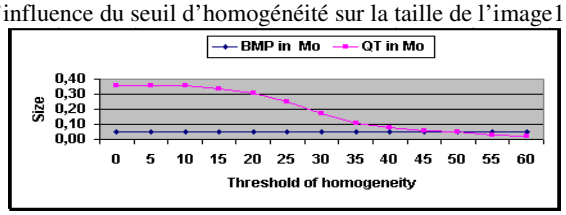


Fig. 14. L'influence du seuil d'homogénéité sur la taille de l'image2 (Fig. 12).

La Fig.15 comporte trois images format QT (a),(b) et (c) avec différentes valeurs du critère d'homogénéité de la couleur ((a) : 10%, (b) : 25% et (c) : 30%), et la Fig.16 montre l'amélioration de la taille de l'image1 selon le critère d'homogénéité de la couleur. On remarque que le meilleur résultats en gain d'espace est de 89,64% pour l'image(c) (Fig.16), mais on perd en qualité d'image, l'image (a) donne un meilleur résultat avec 53,92% en gain d'espaces.

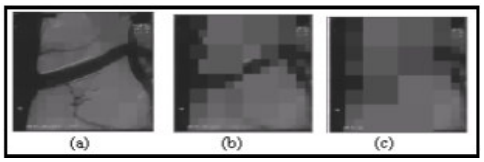


Fig. 15. Les résultats obtenus avec différentes valeurs du seuil d'homogénéité pour l'image1**.**

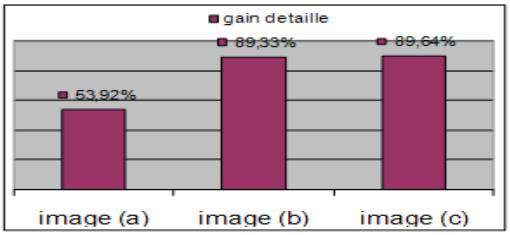


Fig. 16. Gain d'espace selon le critère d'homogénéité pour l'image1.

Les paramètres de la Fig.15 sont employées avec l'image2 (Fig.12), et la Fig.17 donne les images résultantes avec différentes valeurs du critère d'homogénéité, et la Fig.18 montre l'amélioration des tailles. Nous avons noté que le meilleur résultat en gains d'espace de 91,49% pour l'image (c) (Fig.16), mais avec perte en qualité d'image, pour l'image (a) le meilleur résultat en gain d'espace est de 71,66%.

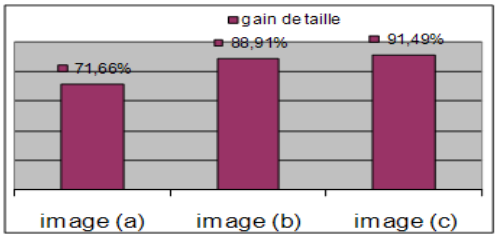


Fig. 17. Les résultats obtenus avec différentes valeurs du seuil d'homogénéité pour l'image2.

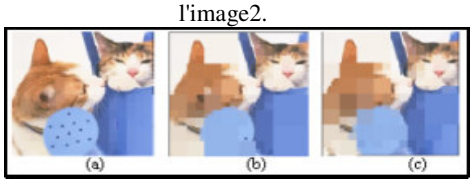


Fig. 18. Gain d'espace selon le critère d'homogénéité pour l'image2.

La Fig.19 présente l'amélioration des tailles entre le système REQUIT [11] et notre nouveau système ARICHA ***(Accélération de la Recherche d'Image par le Contenu Méthode CHAMELON)*** appliqué a quatre différentes bases d'images. Les meilleurs résultats en taille du format Qt sont donnés par ARICHA (Amélioration De La Recherche D'images Par Le Contenu Intégration De La Méthode CHAMELEON), cela est du aux modifications apportées aux structures de données utilisées dans ce format.

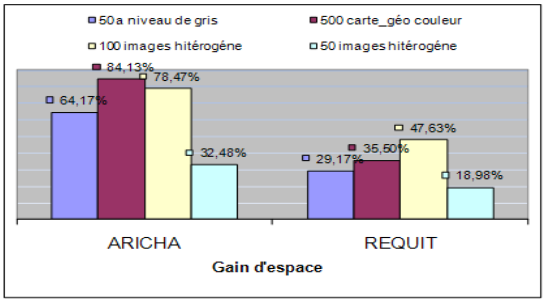


Fig. 19. Amélioration des tailles entre le système REQUIT et notre nouveau système ARICHA appliqué a quatre différentes bases d'images.

Dans notre prototype, la recherche d'image similaire procède dans un premier temps a la recherche de la classe la plus similaire, puis on cherche les images similaire a l'image requête dans la classe sélectionnée. Le temps de la recherche d'images similaires dans une base généraliste de mille images avant la segmentation est de 2132,35s le temps de recherche a été réduit de 216,13s après application de notre prototype (22,23s pour la recherche de la classes et 193,9s pour la recherche des images dans la classe) soit un rapport de 10.

5 Conclusion

Après l'application de la méthode CHAMELEON, la recherche de similarité d'une image requête par rapport aux images stockées dans la base passe par deux étapes: La première étape consiste à calculer la similarité de l'image requête avec les images représentant chacune leurs classes, afin de repérer la classe la plus proche à l'image requête, en utilisant l'une des distances de similarité T, Q ou V (parcours de l'ensemble d'images résumées). La seconde étape consiste en la recherche d'images similaires dans le groupe de la classe dont elle est la plus proche.

Le nombre d'itération pour les regroupements des paires de classe dépend du paramètre k, en augmentant k, le nombre d'itération diminue. Le choix du critère de découpage dépend du domaine d'application, de l'expert et de la nature des images traitées, ainsi le choix du paramètre k de l'algorithme est aussi expérimental si le nombre k est faible on aura plus de classe plus de précision, est vice versa d'un coté; de l'autre l'apparence du paramètre ***seuilkppvs*** qui limite la plage des plus proches voisins, nous a évité de tombé sur des images divergentes en terme de distance dans une même classe au moment de construction du graphe des k-ppv sur lequel se base le graphe Hmétis pour la construction des classes initiales.

Références

1. Nackache, D. « Data warehouse et Data Mining », Conservatoire National des arts et Métiers de Lille, juin 1998.
2. Delorme, J.M. « L'apport de la fouille de données dans l'analyse de texte », Conservatoire National des Arts et Métiers, Centre régional de Montpellier, avril 2002.
3. Gilleron, R. Tommasi, M. « Découverte de connaissances à partir de données », Université de lille3, 2000.
4. Brahmi, A. Gueddache, A. «Data Mining et SMA»; exposé RFIA USTO, 2003.
5. Nakache, D. " probatoire en ingénierie des systèmes décisionnels : Data Mining sur Internet" ; conservatoire National des arts et métiers de Lille; 15-12-1998.
6. Abdelkaderzired, D. Rakotomalala, R." Extraction des connaissances à partir des donnés (ECD), Data Mining " ; 2002.
7. Karypis, G. Han, E. Kuner, V. "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling" 1999.
8. Ng Raymond, T. Han, J. Member, « CLARANS: A Method for Clustering Objects for Spatial Data Mining» IEEE Computer Society, 1994.
9. Zhang, T., Ramakrishan, R.: BRICH. An Efficient Data Clustering Method for Very Large Data Bases".Mirou 1998.
1. 10. Guha, S., Rastagi, R., Kshim ,K. : CURE: An Efficient Clustering Algorithm for Large Data Bases , 1998.
10. Belal, A. Djelil, M. "Similarité entre images basées sur les arbres quaternaires", PFE, Informatique Université Usto ,2005.
11. Van, P. Oostrom. "Reactive Data Structures for Geographic Information systems", Oxford University Press, ISBN: 0-19-823320-5, 1993.
12. Manouvrier, M. : Objets de grande taille dans les bases de données ", Thèse de doctorat informatique, université de Paris, Jan 2000.
13. Manouvrier, M., Rukoz, M., Jomier, G. "Quadtree representations for storage and manipulation of clusters of images", Image and Vision Computing, vol. 20, n° 7, 2002.
14. Djerroud, S., Zaoui, L. : Accélération de la recherche d'images stockées en arbre quaternaire dans les bases d'images généralistes par CHAMELEON », mémoire de magistère de l'université d'USTO Oran, soutenu le 02 juillet 2007.
15. Djerroud, S., Zaoui, L. : Improprement of Content Based Image Retrieval: intégration of CHAMELEON method. a la : 3emes journées internationales sur l'informatique graphique JIG'2007, université de Constantine, 28-29 octobre 2007.
16. Djerroud, S., Zaoui, L. : Accélération de la recherche d'images stockées en arbre quaternaire dans les bases d'images généralistes par CHAMELEON », colloque internationale MOAD'07 METHODE ET OUTILS D'AIDE A LA DESCISION, université de Béjaia 18-20 novembre 2007.
17. Djerroud, S., Zaoui, L. : Amélioration de la recherche d'images par le contenu en utilisons une méthodes issue du domaine du data mining ., 10éme Conférence Maghrébine de le technologie d'information (Conférence on Software Engineering and Artificial Intelligence) MCSEAI'08 , université d'USTO Oran, 28-30 avril 2008.
18. Djerroud, S., Zaoui, L. : Optimisation Du Temps De La Recherche D'images Par Le Contenu., Journées d'Etudes Algero-Françaises en Imagerie Médicale, JETIM'08, université saad dahlab blida, 22-25 novembre 2008.

Le routage dans les systèmes Pair-à-Pair

Nassima Adjissi

Département Informatique, UFA, Sétif, Algérie
nadjissi@hotmail.com

Résumé. La technologie client/serveur domine actuellement le calcul distribué. Le plus grand inconvénient de cette technologie est la concentration de charge sur les serveurs. La technologie Pair-à-Pair (Peer-to-Peer: P2P) vise à résoudre cet inconvénient, elle permet la recherche de données dans des environnements dynamiques à large échelle en distribuant totalement la charge de traitement. Elle ne nécessite aucun contrôle centralisé ou hiérarchique et intègre des mécanismes permettant la cohérence du système alors que les noeuds du réseau physique joignent et quittent le réseau de façon anarchique. Dans ce papier, nous présentons les systèmes Pair-à-Pair, le routage dans les différentes architectures ainsi qu'un tableau récapitulatif résumant le principe de fonctionnement et les avantages et inconvénients de chaque type de routage.

Mots clés : Système P2P, Recherche d'information, Routage Pair-à-Pair, Réseau de recouvrement, DHT

1 Introduction

En 1999, un étudiant de l'université de Boston, Shawn Fanning, alors âgé de 19 ans vient bouleverser le monde bien établi du client-serveur. Il écrit, pour lui et ses amis un logiciel gratuit qui permet d'échanger facilement de la musique au format mp3 via internet. La raison d'être de ce logiciel repose sur le constat suivant: rechercher des mp3 sur les moteurs de recherche habituels conduisait à une perte de temps énorme tant les réponses étaient inappropriées. Fanning, dont le pseudonyme sur les forums était Napster donnera ce nom à son application Napster [1].

L'originalité de Napster réside dans le fait que les fichiers transférés ne le sont pas au travers d'un serveur mais directement d'un utilisateur vers un autre, définissant ainsi le principe d'échange du peer-to-peer qui facilite le partage de l'information.

Du coté du monde académique, en 1999 toujours, Ian Clark écrivait son mémoire : "*A distributed decentralized information storage and retrieval system*" qui est l'article de base qui permettra de construire le logiciel Freenet [2]. Il s'agit d'un système permettant la diffusion d'information, et ce quels que soient les moyens mis en place pour le contrer. Il fournit entre autre un système d'anonymat, de duplication, et de cryptographie. Ce système est l'un des rares systèmes académiques a être implémenté et utilisé à l'heure actuelle.

D'un point de vue purement technique, un système Pair-à-Pair est un ensemble de machines coopérant pour exécuter une tâche (calcul, partage de fichiers, service). Ces machines exécutent un programme au but identique. Il n'y a alors plus une entité indépendante qui gère le service. De nombreuses techniques ont été élaborées afin d'optimiser la recherche de données sur un réseau P2P, des logiciels comme CAN [3], Chord [4], Pastry [5], et Tapestry [6], permettent d'optimiser les recherches grâce à des tables de hachage distribuées.

Le reste du papier est structuré comme suit : Dans la section 2 nous étudions en détail les différentes architectures et les différents types de routage Pair à Pair. Une discussion est présentée dans la section 3. Dans la section 4 nous présentons un tableau récapitulatif des avantages et des inconvénients des différents systèmes P2P. et nous concluons notre travail dans la section 5.

2 Routage dans les systèmes Pair-à-Pair

Nous allons maintenant étudier plus en détail les différentes architectures utilisées pour les systèmes Pair-à-Pair. En effet, malgré la multiplicité des projets, il existe relativement peu de type d'architectures différentes. En effet lorsqu'on s'abstrait de caractéristiques tels que l'anonymat, ou la façon de gérer la découverte des ressources, nous obtenons ce qui différencient les systèmes Pair-à-Pair des autres approches.

2.1 Système d'indexation centralisé

Dans ce modèle, les participants doivent se connecter à un serveur central avant de se connecter au réseau. Le serveur ne contient aucune ressource, il permet juste de référencer l'ensemble des participants et les ressources associées, ainsi éventuellement que de méta-données (par exemple des informations sur les possesseurs de tel ou tel type de données).

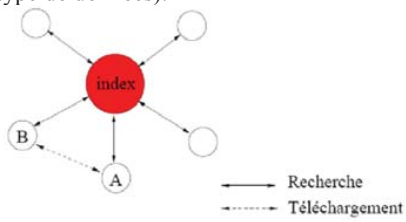


Fig. 1. Système Peer-to-peer centralisé

Dans ce type de système, lorsqu'un internaute souhaite partager un fichier, il le déclare au serveur central. Celui-ci répertorie alors son adresse IP. Tout autre internaute qui souhaite obtenir ce fichier interroge l'index central, comme il interrogerait un moteur de recherche. L'index central lui communique une adresse IP. Ensuite, le logiciel de partage lui permet de se connecter directement sur l'ordinateur qui propose le fichier. Le fichier en lui-même ne transite pas par le serveur central, qui fonctionne simplement comme un annuaire. Une fois l'adresse IP trouvée, les ordinateurs peuvent se connecter directement entre eux.

L'intérêt de ce modèle réside dans une localisation rapide et performante des ressources grâce à l'utilisation d'un index central (le serveur). D'autre part cet index a ici l'avantage d'être mis à jour régulièrement et facilement. Cela permet entre autre de décider facilement si une ressource est disponible ou pas.

Le point faible d'un tel système est le risque que le serveur ne supporte pas la charge lorsque le nombre de fichiers et/ou le nombre de requêtes augmentent ou plus simplement qu'il tombe en panne. De plus, avec une approche centralisée, il est difficile de garder une vue cohérente d'un réseau fortement dynamique.

Le système le plus médiatique qui rentre dans la catégorie des systèmes Pair-à-Pair est sans conteste Napster [1]. Napster a pour but de permettre l'échange de fichiers musicaux de type mp3. Un autre exemple de ce type de système est SETI@Home [7]. Ce système utilise la puissance inutilisée des ordinateurs connectés à l'Internet pour la recherche d'intelligence extra-terrestre. Les données captées par le télescope Arecibo sont divisées en petites unités, téléchargées et traitées par un logiciel fonctionnant comme un économiseur d'écran sur de nombreux ordinateurs. À la fin d'un traitement, le logiciel envoie le résultat au serveur de SETI@Home et continue avec de nouvelles données.

2.2 Système décentralisé

Dans ce modèle, le serveur central n'est plus nécessaire à la connexion. Chaque internaute indexe lui-même ses propres fichiers. Ceux qui sont à la recherche d'un fichier interrogent, de proche en proche, tous les ordinateurs du réseau. L'exemple le plus connu est le réseau Gnutella [8].

Gnutella repose sur un algorithme d'inondation, le premier problème à résoudre est de trouver, lors de la première connexion, d'autres internautes qui participent au partage. Dès que l'on rencontre un ordinateur connecté au réseau, celui-ci peut communiquer les adresses IP d'autres participants. Pour trouver ce premier internaute, on utilise les « Gwebcache ». Ce sont des ordinateurs du réseau Gnutella qui fonctionnent également comme des serveurs web. Ainsi, ils sont répertoriés par les moteurs de recherche. Mais chaque internaute ne peut se connecter qu'à un nombre limité d'autres participants. Ainsi, de proche en proche, se constitue ce qu'on appelle un «réseau d'overlay» qui se greffe sur le réseau global. En pratique, quand on recherche un fichier, c'est sur ce réseau logique qu'on diffuse sa requête. Celle-ci est d'abord adressée aux ordinateurs voisins, puis peu à peu propagée jusqu'à trouver un ordinateur possédant le fichier en question. La réponse, constituée par l'adresse IP de cet ordinateur, suit le même chemin en sens inverse.

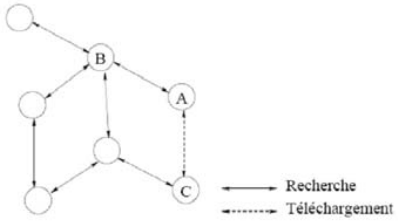


Fig. 2. Système Peer-to-peer décentralisé

Ce modèle, du à son architecture, est beaucoup plus robuste et plus autonome qu'un modèle centralisé. De plus, il gère facilement la dynamique des nœuds car si l'un des nœuds se déconnecte du réseau, la requête pourra être poursuivie vers les autres ordinateurs connectés. En revanche, en raison du type de communication utilisé pour le transfert des requêtes (Broadcast), la bande passante nécessaire pour chaque requête croit exponentiellement avec le nombre de pairs: ce qui peut entraîner une inondation et même la chute du réseau. Ce système garantit une certaine forme d'anonymat : celui qui détient le fichier ne connaît pas l'identité de celui qui le lui demande mais la sécurisation d'un tel système reste très difficile car il n'existe aucune trace des échanges.

2.3 Système hybride

Le principe des architectures hybrides est d'utiliser les avantages des deux premiers modèles. La plupart du temps ce sont des modèles dis "super-nœud".

Ce modèle opère une distinction entre deux niveaux de pairs : ceux qui ont une connexion haut débit et ceux qui ont une connexion par modem. Les ordinateurs disposant d'une connexion par modem se relient à un ordinateur ayant une connexion haut débit. Ce dernier devient dès lors un « superpair ». Chaque superpair indexe alors les fichiers des pairs bas débit qui lui sont rattachés, comme le faisait autrefois le serveur central des réseaux avec un index central. Entre deux superpairs en revanche, le système continue à fonctionner comme les réseaux décentralisés. Mais la propagation des données est plus rapide, puisqu'elle n'utilise plus que les connexions haut débit. Une fois l'adresse IP communiquée à l'ordinateur d'origine, une connexion directe s'établit entre les deux pairs, quel que soit leur niveau. Il s'agit donc d'une solution hybride entre les réseaux avec index central et les réseaux décentralisés.

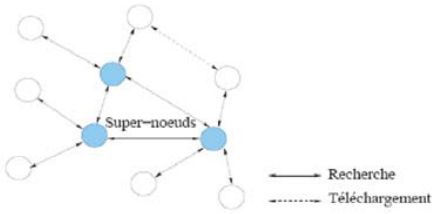


Fig. 3. Système Peer-to-peer hybride

Les avantages de ce type de structure sont que les communications réseau ne sont plus polluées par les trames de Broadcast, et que la sécurité est gérable grâce au réseau des superpairs. En contrepartie, l'anonymat n'est plus assuré. Le réseau FastTrack, associé au logiciel de partage Kazaa, constitue un bon exemple de réseau Pair-à-Pair hybride.

2.4 Système non structuré

Un système P2P non structuré est un système qui n'impose pas de règle de connexion entre les pairs. Ils reposent sur le fait que l'utilisation en elle même du système va le structurer. Les réseaux P2P non structurés reposent sur la génération de graphes aléatoires entre les noeuds. Chaque nouveau noeud se connectant doit connaître un noeud appartenant au réseau, qui lui sert de bootstrap pour s'insérer dans le réseau. Les requêtes se passent ensuite sous la forme d'inondation (demande à tous les voisins qui demandent à tous leurs voisins. . .) ou de marche aléatoire (demande à un voisin qui demande à un voisin. . .).

Les premiers paramètres importants pour ce type de réseau sont les degrés entrant et sortant (nombre de connexions) de chaque noeud. Les autres paramètres importants sont les paramètres de requêtes. Dans le cas d'une inondation, il s'agit du TTL et du nombre de voisins à inonder (dans le cas d'une inondation optimisée) : une valeur trop faible ne retournera pas toutes les réponses attendues alors qu'une valeur trop forte génèrera un fort trafic inutile. Dans le cas d'une marche aléatoire, il s'agit de l'algorithme de choix du voisin qui recevra la requête : si le voisin est mal choisi, alors la recherche ne rendra pas de résultats intéressants. Nous présentons dans ce qui suit le système Freenet qui est un exemple des systèmes non structurés.

Freenet

Freenet [2, 9, 10] est un système de partage de fichier qui protège la libre publication, le téléchargement et l'anonymat des usagers. Chaque pair de Freenet maintient un tableau de routage contenant l'adresse de quelques autres pairs (voisins) ainsi que les clés des fichiers qu'il croit que ces voisins stockent. La localisation d'un fichier applique la méthode de recherche en profondeur. La figure 4 illustre ce mécanisme.

Un message de requête passe d'un pair à l'autre. S'il atteint le pair (D) stockant le fichier recherché, une réponse spécifiant le lieu de stockage est retournée au pair demandeur (A) (en suivant le même chemin que la requête). Sinon, le pair courant achemine le message au voisin qui contient la clé la plus proche de celle cherchée. Si le message atteint un pair déjà visité (B) ou un cul-de-sac (E) (c.-à-d., échec de recherche), il retourne au pair précédent et essaie un autre chemin. Un message a un champ TTL (time-to-live) qui limite la longueur du chemin de recherche.

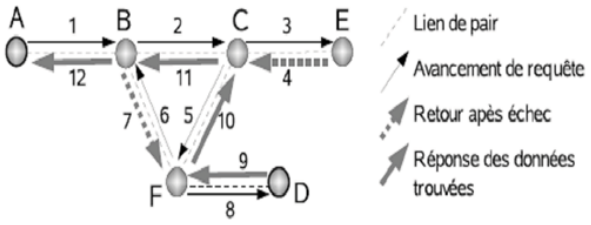


Fig. 4. Chemin de requête dans Freenet (adaptée de Clarke et al. [10])

L'insertion d'un fichier suit le même chemin que la recherche de la clé du fichier. Le message d'insertion a un champ TTL qui détermine le nombre de copies du fichier à stocker. Si un fichier avec la même clé existe déjà sur le chemin du message, l'insertion échoue. Sinon (TTL expire sans collision), le dernier pair répond « all clear » au pair d'origine du fichier pour accueillir le fichier au stockage des pairs sur le chemin.

Ce mécanisme de routage a un certain nombre d'effets. On peut supposer que le routage va en s'améliorant au cours du temps pour deux raisons :

- Les noeuds devraient finir par se spécialiser dans la recherche de clés proches. Si un noeud est associé à une clé dans une table de routage, il va avoir tendance à traiter principalement des clés semblables à celles qu'il possède.
- Les noeuds devraient de même se spécialiser dans le stockage d'objets de clés semblables. Comme transférer une requête revient à mettre dans son cache en retour l'objet associé, le fait de router des requêtes pour des clés semblables entraîne la possession d'objets semblables.

Ces deux effets doivent entraîner une boucle d'amélioration de la réponse du système. Ce système agit comme un système de cache au sens traditionnel du terme. Les objets très populaires seront plus souvent copiés et donc se diffuseront dans l'ensemble du réseau. Leur accès sera accéléré car les données seront plus proches des utilisateurs.

Les systèmes Pair-à-Pair non structurés supportent des opérations simples et efficaces d'arrivée et de départ des pairs parce que les pairs n'ont pas à maintenir une topologie. Cependant, la localisation d'objet n'est pas réalisée exhaustivement. Le champ TTL peut empêcher de trouver le fichier cherché même s'il existe. De plus, l'inondation de message dans Gnutella cause un grand coût de trafic du réseau. Les systèmes Pair-à-Pair structurés résolvent ce problème.

2.5 Système structuré

Un système Pair-à-Pair structuré doit maintenir une topologie. Il définit un espace de clés et projette les objets sur cet espace. La plupart des systèmes utilisent une fonction de hachage pour effectuer cette projection et donc sont appelés tableaux de hachage distribués (Distributed Hash Tables ou DHT).

Une table de hachage est un ensemble d'objets décrits sous la forme de couples (clef, valeur) représentant par exemple un nom de fichier et la machine qui le fournit : (file.mp3, 140.55.123.4). Ces couples sont stockés physiquement dans m emplacements mémoire par application de la fonction de hachage h : X → {0, 1, ..., m-1} sur la clef.

Dans le cadre des systèmes distribués, cette fonction vise à répartir ces clefs sur les nœuds du réseau. La fonction de hachage doit être aléatoire uniforme à valeur dans un ensemble d'identifiants logiques avec une probabilité négligeable de collision (deux clefs distinctes donnant une même valeur). Les DHT forment un réseau de recouvrement logique permettant de structurer le réseau physique. L'emplacement d'un noeud dans le réseau logique est défini par l'application de la fonction de hachage sur son adresse IP. Les couples (clef, valeur) sont répartis de façon uniforme sur les différents noeuds logiques par application de h sur la clef. Il est important de remarquer que seule une référence de l'objet déclaré, et non l'objet lui-même est stocké dans la DHT. La recherche d'un objet par sa clef (Lookup(clef)) permet de retrouver l'ensemble des valeurs associées à une clef, en routant la requête en fonction de la clef recherchée.

Un système Pair-à-Pair structuré distribue la responsabilité des clés aux pairs. Le système définit une structure de connexion entre les pairs en fonction des clés dont chacun s'occupe. La localisation d'un objet se fait en calculant la clé de l'objet et routant une requête au pair voisin le rapprochant le plus de la cible. Malgré l'absence de connaissance globale, chaque pair peut acheminer correctement la requête à l'aide de la structure de connexion connue.

En théorie, un système Pair-à-Pair structuré garantit de trouver n'importe quel objet s'il existe. Il supporte habituellement un routage très efficace dont le coût est souvent O(log n) où n est le nombre de pairs. Cependant, l'arrivée et le départ d'un pair ne sont plus aussi simples car le système doit réorganiser les pairs pour maintenir la structure de connexion. Différentes topologies entraînent différentes efficacités de routage et de maintenance du système. Trois groupes de topologies des systèmes Pair-à-Pair structurés existants : topologies basées sur un tableau de routage à plusieurs niveaux, topologies basées sur un espace de clés multidimensionnel et topologies inspirées d'un graphe.

Topologies basées sur un tableau de routage à plusieurs niveaux

Le tableau de routage d'un pair contient les pointeurs vers certains pairs (voisins). Le principe de ce groupe de topologies est de construire le tableau de routage des pairs en utilisant plusieurs niveaux. Rappelons que chaque pair s'occupe d'un ensemble donné de clés (ex., les clés les plus proches numériquement de l'id du pair ou les clés partageant un préfixe identique à l'id du pair). On peut donc projeter les pairs sur l'espace de clés et définir la distance entre les pairs dans cet espace. À chacun des niveaux, le pair maintient un certain nombre de pointeurs vers d'autres pairs. Les différents niveaux de pointeurs permettent au routage de passer de pair à pair en précisant progressivement la position de la cible. Des exemples de ce type de topologie sont Tapestry [11], Pastry [12] et Chord [4] dont nous présentons ici.

Chord

Chord [4] définit un système Pair-à-Pair sur un espace circulaire de 2^m clés. Les ids des pairs sont choisis dans l'espace de clés. Un id est donc une chaîne de m bits. Les pairs immédiatement avant et après un pair a dans le cercle des clés sont appelés, respectivement, le prédécesseur et le successeur de a. Le pair a est responsable des clés qui précèdent a (incluant a), mais qui succèdent le prédécesseur de a. On écrit $a = successor(k)$ si a est responsable de la clé k. Le tableau de routage de a possède m niveaux.

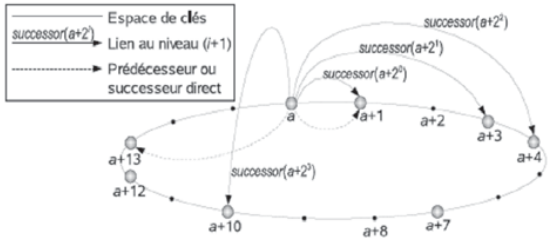


Fig. 5. Exemple des liens d'un pair a dans Chord

Chaque $i^{ième}$ niveau ne contient qu'un lien qui pointe au pair successor $(a+2^{i-1})$. Le pair a a aussi deux liens vers son prédécesseur et son successeur, respectivement. La figure 5 illustre les liens d'un pair a dans un réseau Chord ayant 8 pairs et $m = 4$. Cette structure permet un routage efficace. Le routage d'une clé k à partir du pair a cherche dans le tableau de routage le $j^{ième}$ niveau qui donne $(a + 2^{j-1})$ le plus proche mais avant k dans le cercle des clés. Puis le routage suit le lien indiqué. Ce routage coûte $O(logN)$ sauts dans un système de N pairs. Le degré des pairs est $O(logN)$.

Topologies basées sur un espace de clés multidimensionnel

Un système Pair-à-Pair de ce type définit l'espace de clés comme un espace Cartésien à plusieurs dimensions. L'espace multidimensionnel permet au routage de s'approcher de la cible (étant un point dans l'espace) en « marchant » sur ces différentes dimensions. Cela raccourcit donc la longueur de routage par rapport à une transmission naïve dans un espace unidimensionnel.

CAN

Un exemple de tel système est le Content-Addressable Network (CAN) [3]. Le routage dans CAN utilise un algorithme glouton qui traverse les frontières des zones dans les différentes dimensions pour atteindre la cible. La figure 6 présente un exemple de routage dans un CAN ayant 7 pairs et d = 2. La requête est initiée au pair 1, et va rejoindre le pair 6 en passant par le pair 7. L'espace complet est un carré dont l'abscisse et l'ordonnée varient entre 0 et 1. Initialement, un pair est responsable de tout l'espace. Dans ce système, un pair est toujours responsable d'un sous rectangle du carré initial. Lorsqu'un noeud veut se rajouter, il tire aléatoirement une position, puis contacte le pair responsable de cette position. L'espace dont est responsable ce pair est alors coupé en deux, le nouveau pair prenant la responsabilité d'une partie de l'espace.

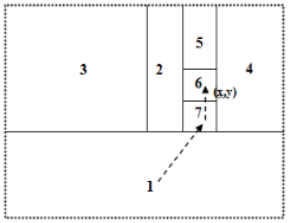


Fig. 6. Exemple de routage du pair 1 au pair 6 dans CAN

Dans ce système, tout pair connaît l'adresse des pairs responsables des zones qui lui sont contiguës. C'est grâce à cette connaissance que le routage est possible. Pour router une requête dans le système, un pair cherche dans ses voisins le pair le plus proche du résultat, puis lui transmet la requête. Dans [3] une étude de la taille des chemins et du coût de maintien des voisinages. L'un des intérêts de ce système est la possibilité de faire varier la dimension suivant les impératifs du système. En effet, le coût de mise à jour des voisinages des pairs est en O(d), alors que la longueur moyenne des chemins est en O(dn1/d), avec n le nombre de pairs dans le système. Ainsi suivant la dynamicité escomptée on peut régler finement l'efficacité de CAN.

L'un des plus gros défauts actuels de CAN réside dans le fait qu'il n'existe pas, à l'heure actuelle, de système basé sur cette architecture. Ainsi il n'est pas possible d'avoir une évaluation complète des capacités de ce système prometteur.

Topologies inspirées d'un graphe

Les topologies de ce groupe s'inspirent de graphes qui ont de bonnes propriétés pour le routage. Des exemples de ce type de topologie sont: Viceroy, CAN D2B, Koorde et DH DHT. Tandis que Viceroy s'inspire des graphes butterfly, les autres appliquent les graphes de De Bruijn.

Topologies inspirées des graphes de De Bruijn

Un graphe de De Bruijn [13], dénoté B(b,m), se compose de b^m noeuds dont l'id, dénoté $x_1 \cdot \cdot \cdot x_m$, appartient à l'ensemble $\{0, \cdot \cdot \cdot, b-1\}^m$. Les arcs connectent chaque nœud $x_1 \cdot \cdot \cdot x_m$ aux noeuds $x_2 \cdot \cdot \cdot x_m y$ où $y \in \{0, \cdot \cdot \cdot, b-1\}$. Cette connexion dote le graphe de De Bruijn d'un algorithme de routage très efficace.

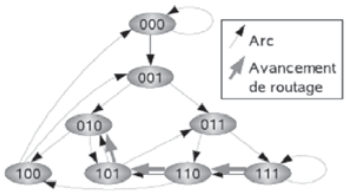


Fig. 7. Graphe de De Bruijn B(2, 3) et le chemin de routage de 111 à 010

Pour router un message d'un nœud $X = x_1 \cdot \cdot \cdot x_m$ à un noeud $Y = y_1 \cdot \cdot \cdot y_m$, on détermine la chaîne s la plus longue étant un suffixe de X et un préfixe de Y (c.-à-d., $Y = sy_{|s|+1} \cdot \cdot \cdot y_m$). Puis, on progresse en passant au noeud voisin $x_2 \cdot \cdot \cdot x_m y_{|s|+1}$. L'algorithme rallonge s à chaque étape jusqu'à Y. La figure 7 illustre le graphe B(2, 3) et le routage de 111 à 010. Le chemin de routage passe par les noeuds 111, 110, 101 et 010 qui amènent l'évolution de la chaîne s : vide, 0, 01 et 010.

Un graphe de De Bruijn possède certaines des propriétés d'un arbre complet : degré constant des noeuds et coût logarithmique de routage. Tandis qu'un arbre a seulement une racine (c.-à-d., le noeud au bout d'un chemin vers n'importe quel autre noeud), dans un graphe de De Bruijn, tous les noeuds sont une telle « racine ». Le graphe de De Bruijn présente des avantages dans la construction des réseaux d'interconnexion : la longueur du chemin de routage et le degré (sortant) des noeuds sont limités par m et b, respectivement.

3 Discussion

Les systèmes Pair-à-Pair apportent une bonne solution au problème du calcul à grande échelle. Ils distribuent la charge de travail sur les participants du système afin d'assurer la performance globale vis-à-vis des tâches intensives. Dans ce papier, nous avons présenté les systèmes Pair-à-Pair, parmi lesquels nous avons distingué cinq grandes familles : les systèmes centralisés, les systèmes décentralisés, les systèmes hybrides, les systèmes non-structurés et les systèmes structurés.

Chacun de ces systèmes possède des caractéristiques et des avantages bien différents. Les premiers systèmes Pair-à-Pair se sont fondés soit sur un service centralisé de recherche tels Napster, construisant des systèmes fragiles et peu enclins au passage à l'échelle, soit sur des techniques d'inondation des noeuds du réseau tels Gnutella, surchargeant inutilement le réseau pour des recherches non-exhaustives Les systèmes non-structurés permettent l'utilisation de requêtes complexes, n'impose pas de structure sur la connexion des pairs. Il en découle une construction simple et une maintenance efficace mais ne garantissant pas le déterminisme de recherche. La localisation d'un objet peut ne pas atteindre la cible, même si elle existe, alors que les systèmes structurés résous ce problème en appliquant une structure de connexion entre les pairs. Cette structure assure le routage efficace et la localisation déterministe des objets. Cependant, la maintenance du système devient compliquée parce qu'elle doit conserver la structure imposée.

4 Tableau récapitulatif

Dans ce tableau récapitulatif nous présentons une comparaison entre les différents types des systèmes Pair-à-Pair, Avantages et Inconvénients de chaque type.

Table 1. Comparaison entre les différents types des systèmes Pair-à-Pair

Type de système Pair-à-Pair	Avantages	Inconvénients
Système centralisé	• localisation rapide et performante des ressources • Mise à jour régulière et facile de l'index. • Décider facilement si une ressource est disponible ou pas.	• Goulot d'étranglement sur le serveur central. • Défaillance du système si le serveur central tombe en panne. • Difficulté de garder une vue cohérente d'un réseau fortement dynamique. • Aucun anonymat n'est possible, puisque chaque utilisateur est identifié sur le serveur.
Système décentralisé	• Robustesse et autonomie. • Réseau très tolérant aux fautes. • Gère facilement la dynamique des nœuds. • Garantit l'anonymat	• Grande consommation de la bande passante nécessaire (Risque d'inondation et même la chute du réseau). • La sécurisation très difficile. • Pas de garantie de succès.
Système hybride	• La propagation des données est plus rapide (pas de Broadcast). • La sécurité est gérable grâce au réseau des superpairs	• Sensibilité aux défaillances (si le super-peer n'est plus joignable, tous ses clients sont coupés du réseau). • L'anonymat n'est pas assuré.
Système non structuré	• Traitements simples et efficaces des opérations d'arrivée et de départ des pairs.	• La localisation d'objet n'est pas réalisée exhaustivement • Le champ TTL peut empêcher de trouver le fichier cherché même s'il existe. • L'inondation de message cause un grand coût de trafic du réseau.
Système structuré	• Routage efficace d'un point à un autre. • Garantit de trouver n'importe quel objet s'il existe.	• Traitements difficiles des opérations d'arrivée et de départ des pairs. • Réorganisation des pairs pour maintenir la structure de connexion.

5 Conclusion

Nous avons présenté dans ce papier une étude des systèmes Pair-à-Pair en nous intéressant particulièrement à la problématique de la recherche et plus généralement au routage dans ce type de système. Nous pouvons distinguer trois étapes primordiales de routage dans un tel réseau : premièrement, l'adhésion du pair au réseau; deuxièmement, la recherche d'un contenu par le pair ; troisièmement, le transfert à proprement parler du contenu.

On peut classifier ces systèmes selon leur modèle de recherche sous-jacent qui peut être soit non structuré (propagation aléatoire des requêtes dans le graphe des pairs), soit structuré (propagation des requêtes selon une structure d'organisation des pairs basée en général sur du hachage). L'approche non structurée bien que moins efficace sur le plan du routage des requêtes offre l'avantage de respecter au mieux l'autonomie des pairs et de pouvoir supporter des langages de requêtes plus expressifs. Nous avons aussi présenté un tableau récapitulatif des avantages et inconvénients des divers mécanismes de routage P2P ainsi une discussion générale des différentes approches.

Référence :

1. Napster. Http://www.napster.com
2. The Freenet project. Http://freenet.sourceforge.net.
3. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In ACM SIGCOMM'01, pages 160–172, San Diego, California, USA, August 2001.
4. Ion Stoica, Robert Moris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In ACM SIGCOMM'01, pages 149–160, San Diego, California, USA, August 2001.
5. Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001.
6. Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerance wide-area location and routing. Technical Report UCB/CSD-01-1141, University of California Berkeley, April 2001.
7. SETI@Home. Http://setiathome.free.fr/index.html.
8. Gnutella. Http://www.gnu.org/philosophy/gnutella.fr.html
9. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet : A distributed anonymous information storage and retrieval system. In Designing Privacy Enhancing Technologies : Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, Springer Berlin / Heidelberg, page 46, 2001.
10. Ian Clarke, TheodoreW. Hong, Scott G. Miller, Oskar Sandberg, and Brandon Wiley. Protecting free expression online with Freenet. IEEE Internet Computing, 6(1):40–49, 2002.
11. Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerance wide-area location and routing. Technical Report UCB/CSD-01-1141, University of California Berkeley, April 2001.
12. Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001.
13. N. G. de Bruijn. A combinatorial problem. In *Koninklijke Nederlands Akademie van Wetenschappen*, volume 49, 1946.

Méthode numérique de calcul de contrôle optimal des systèmes compartimentaux

N. Messaoudi, S. Manseur
Département de mathématiques.
Université de Blida.B.P. 270, Blida, Algérie

Résumé

Le but de ce travail est l'utilisation da la méthode décomposionnelle d'Adomian pour le calcul d'un contrôle optimal d'un système non linéaire. Le problème est ramené à la minimisation d'une fonction à une variable sur des intervalles réduits. Une application de cette méthode a été réalisée sur un modèle bicompartimental avec des échanges non linéaires de type Mechaelis-Menten.

Mots clés: Méthode décompositionnelle d'Adomian, système différentiel non linéaire, modèle bicompatimental, contrôle optimal .

1 Introduction

De manière générale, un problème de contrôle optimal consiste à trouver l'enchaînement des opérations permettant à un système dynamique de passer d'un état initial à un état final prédéterminé tout en optimisant un certain critère.

La modélisation des systèmes biologiques nous conduit souvent à des équations différentielles non linéaires, contenant un vecteur de paramètres qu'on appelle contrôle ou commande. Le but est de déterminer l'évolution de ce paramètre dans le temps de manière à amener la solution à satisfaire certains critères.

La théorie de contrôle optimal est applicable à des cas concrets comme par exemple :

- contrôler la croissance de certaines populations (cellules cancéreuses,bactéries,virus,..., etc) en utilisant des traitements chimiques. Dans le cas particulier de la chimiothérapie du cancer, l'objectif consiste à minimiser le nombre de cellules cancéreuses.

- maintenir la concentration de médicament dans la tumeur autour d'une valeur souhaitée.

- minimiser le nombre de cellules tumorales avec un minimum de substance active et en un temps minimum.

Généralement, ce problème il est résolu par le principe de pontriaguine qui se ramène à la résolution d'un système Hamiltonien en général non linéaire avec conditions aux limites dont la résolution exige l'application de méthode numérique [7].

L'objectif de ce travail est l'utilisation de la méthode décompositionnelle d'Adomian au problème de contrôle optimal des systèmes non linéaires. Cette méthode permet la résolution analytique d'un système différentiel linéaire et non linéaire sous forme de série convergente qui dépend explicitement des paramètres du système et sur la décomposition en série de l'opérateur non linéaire, en utilisant des polynômes appelés « polynômes d'Adomian ». Ces polynômes sont calculés par des formules récurrentes ([1],[4]). La résolution se fait sur de petits intervalles de temps où le contrôle est supposé constant et borné.

La solution du système est alors introduite dans la fonction objective, et le problème devient alors un problème de minimisation d'une fonction à une seule variable, le contrôle.

Cet article est organisé comme suit, dans la deuxième section on donne une formulation du problème, la section 3 présente une méthode de calcul numérique de contrôle optimal dans le cas de systèmes différentiels non linéaires. Une application de cette méthode au modèle bicompartimental est présentée dans la section 4 et nous terminons par une conclusion.

2 Formulation du problème

Considérons le système différentiel suivant :

$$\begin{cases} \dot{x} = \frac{dx_i}{dt} = f\left(x, u, t\right), \\ x(0) = \beta_i \qquad i = 1, .., p;\ n \leq p \end{cases} \tag{1}$$

et le vecteur d'observation est sous la forme :

$$y(t) = h\left(x, u, t\right)$$

où $x(t) \in \mathbb{R}^p$ est le vecteur d'état, $u(t) \in \mathbb{R}^n$ est le vecteur de contrôle, et $y(t) \in \mathbb{R}^m$ est le vecteur d'observation. f est le vecteur de fonctions non linéaires supposées entièrement connues et les conditions initiales β_i (généralement égales à 0) sont données, $i = 1, .., p$.

On suppose que chaque composante de $u(t)$ est mesurable et bornée, c'est-à-dire admissible.

Le problème de contrôle optimal consiste à chercher le contrôle $u(t)$ admissible qui minimise le critère :

$$J = \int_0^T g\left(x, u\right) dt \tag{2}$$

où x et u doivent satisfaire le système différentiel (1) et g est une fonction continue par rapport à ses variables positive connue et $[0, T]$ est l'intervalle d'observation avec $T \geqslant 0$ qui peut être fixé ou indéterminé.

Des méthodes classiques permettent la résolution du problème comme celles de la programmation dynamique et ses variantes, le principe du maximum de Pontriaguine (voir [4],[6],[7]).

Dans les exemples concrets, il est souvent possible d'obtenir des relations du type :

$$x = \Psi(u) \text{ ou bien } \ u = \Lambda\,(x) \tag{3}$$

où Ψ et Λ sont des fonctions vectorielles qui peuvent être déterminer par un raisonnement mathématique ou par un calcul formel. Lorsque les fonctions f_i sont non linéaires par rapport à x et u la détermination du contrôle optimal exige l'application de méthode numérique [7]. Ici on propose une méthode numérique, basée sur l'utilisation de la méthode décmpositionnelle d'Adomian et permettant de simplifier considérablement le problème posé.

3 Méthode décopositionnelle d'Adomain

La méthode décompositionnelle d'Adomain est utilisée pour résoudre des équations fonctionnelle linéaires et non linéaires de différents type : différentielles, aux dérivées partielles, intégrales, algébriques,...etc.

Rappelons quelques définitions et les propriétés de la méthode décompositionnelle [4].

Considérons l'équation fonctionnelle suivante :

$$x - N(x) = F \tag{4}$$

où N représente un opérateur non linéaire (différentiel, différentiel partiel, intégral,...), F une fonction connue et x est la solution de l'équation (4). L'équation (4) est dite forme canonique d'Adomain.

La méthode décompositionnelle consiste à chercher la solution x (si elle existe) sous la forme d'une série :

$$x = \sum_{n=0}^{\infty} x_n \tag{5}$$

Et à décomposer l'opérateur non linéaire $N(x)$ en série :

$$N(x) = \sum_{n=0}^{\infty} A_n(x_0, x_1, ..., x_n) \tag{6}$$

où les A_n sont les polynômes d'Adomian qui dépendent de $x_0, x_1, ..., x_n$ ([1], [4],[5]). On suppose que ces deux séries sont convergentes. Ils sont donnés par la relation suivante :

$$n!A_n = \frac{d^n}{d\lambda^n}\left[N\left(\sum_{i=0}^{n} \lambda^i x_i]\right)\right]_{\lambda=0}, \ \ n = 0.1, 2, ...$$

où λ est un paramètre introduit par commodité.

Reportons les expressions (5) et (6) dans (4), on peut écrire :

$$\sum_{n=0}^{\infty} x_n - \sum_{n=0}^{\infty} A_n = F$$

Adomian proposé le schéma récursive suivant :

$$\begin{cases} x_0 = F \\ x_1 = A_0(x_0) \\ x_2 = A_1(x_0, x_1) \\ \vdots \\ x_{n+1} = A_n(x_0, ..., x_n), ... \end{cases}$$

On peut facilement calculer les termes de la série x_n de notre solution, il suffit de connaître les polynômes d'Adomian. Y.Cherruault et K.Abbaoui ont prouvés que la série $\sum x_n$ converge si l'opérateur non linéaire N satisfait certaine conditions. Les polynômes d'Adomian existent et la série $\sum A_n$ converge, des formules pratiques pour calculer ces polynômes sont proposées ([1],[4]).

En pratique, il est difficile d'obtenir tous les termes de la série solution, aussi utilise-t on une approximation de la solution sous la forme de série tronquée d'ordre s :

$$\phi_s = \sum_{i=0}^{s-1} x_i$$

4 Méthode numérique de calcul du contrôle optimal

Considérons le système différentiel non linéaire :

$$\begin{cases} \dot{x}_i = \frac{dx_i}{dt} = f_i\left(x_1, \ldots, x_p, u_1, \ldots, u_n\right) \\ x_i\left(0\right) = \beta_i, \qquad\qquad i = 1, \ldots, p \qquad n \leq p \end{cases} \tag{7}$$

dans lequel les fonctions f_i sont non linéaire.

On suppose que les f_i sont de classe C^1 et que le système (7) admet une solution unique lorsque les u_j sont fixés. Nous allons utiliser une technique de linéarisation et pour cela nous allons effectuer un développement de Taylor à l'ordre 1 des f_i, sur des petits intervalles de longueur Δt, $t \in [t_k, t_{k+1}]$ où $t_k = k.\Delta t$ et $\Delta t = \frac{T}{\eta}, k = 0, .., \eta$, où η est le nombre de subdivision de l'intervalle $[0, T]$ avec $k, \eta \in \mathbb{N}$. $t_0 = 0$.

Sur chaque intervalle $[t_k, t_{k+1}]$ le système (7) devient linéaire par rapport aux x_i et aux u_j :

$$f(x,u,t) = \frac{\partial f}{\partial x\,(t=t_k)}x(t) + \frac{\partial f}{\partial u\,(t=t_k)}u(t), \quad t \in [t_k, t_{k+1}]$$

Et on peut donc écrire :

$$\dot{x}_i = A_k\ x(t) + B_k\ u(t), \quad t \in [t_k, t_{k+1}] \tag{8}$$

où $A_k = \frac{\partial f}{\partial x(t=t_k)}$ et $B_k = \frac{\partial f}{\partial u(t=t_k)}$.

Pour lequel on appliquera la méthode décompositionnelle d'Adomian ([2],[3],[4]). On exprimera les $x_i(t)$ sous la forme de série tronquée d'ordre (s) :

$$x_i = \sum_{l=0}^{s-1} v_l^i\,(u_1, \ldots, u_n)\,,\ i = 1, \ldots, p\ \ ,\text{sur}\ \ [t_k, t_{k+1}] \tag{9}$$

Posons $J = \int_0^T g\,(x,u)\,dt$. En décomposant $[0,T]$ en sous intervalle, on a l'égalité :

$$J = \sum_{k=0}^{\eta-1} \int_{t_k}^{t_{k+1}} g\,(x,u)\,dt \tag{10}$$

où $\eta\Delta t = T$.

On reporte les expressions (9) dans la fonctionnelle (10) et l'on obtient une fonctionnelle J où seuls les u_j interviennent de façon explicite. Cette fonctionnelle est à son tour minimisée par des méthodes d'optimisation.

Des conditions de raccordement seront nécessaires aux points $t_k, t_{k+1}, ...$ pour assurer la continuité de u . Si la fonction g est positive pour tout couple (x,u) on peut procéder comme suit :

- On résout d'abord le problème :

$$\underset{\substack{u \\ t\in[0,t_1]}}{Min} \int_0^{t_1} g\,(x,u)\,dt \tag{11}$$

Autrement dit, on cherche le minimum du critère (11) sur l'intervalle $[0,t_1]$ .C'est un problème de minimisation classique qui nous donnera en particulier les conditions initiales en $t = t_1$ pour l'étape suivante.

On résout ensuite :

$$\underset{\substack{u \\ t\in[t_1,t_2]}}{Min} \int_{t_1}^{t_2} g\,(x,u)\,dt \tag{12}$$

et ainsi de suite. On déterminera ainsi de proche en proche $u(t)$ sur tout l'intervalle $[0,T]$.

4.1 Algorithme de calcul de contrôle optimal : cas d'un seul contrôle constant

L'algorithme suivant permet le calcul de contrôle optimal :

Début
(0) Fixer Δt, T, β_i, $i = 1, \ldots, p$. Initialiser $k = 0$, $t_k = 0$,
$\eta = k.\Delta t$,
Tant que $\eta < T$ faire
(1) Poser $u = u_k$,
le système différentiel devient :

$$\begin{cases} \dot{x}_1 = \frac{dx_1}{dt} = f_1(x_1, \ldots, x_p, u_k) \\ \vdots \\ \dot{x}_p = \frac{dx_p}{dt} = f_p(x_1, \ldots, x_p) \\ x_i(t = t_k) = \beta_i, \qquad i = 1, \ldots, p \end{cases}, \ f_i \text{ non linéaire}$$

(2) Linéariser f_i, on obtient le système linéarisé :

$$\begin{cases} \dot{x}_i = A_k\, x(t) + B_k\, u_k, \quad t \in [t_k, t_{k+1}] \\ x_i(t = t_k) = \beta_i \end{cases}, i = 1, \ldots, p$$

où $A_k = \frac{\partial f}{\partial x(t=t_k)}$; $B_k = \frac{\partial f}{\partial u_k(t=t_k)}$, avec $B_k = [1, 0, ..., 0]^t$

(3) Résoudre le système linéarisé par la méthode décompositionnelle d'Adomian, on obtient :

$$x_i = \sum_{l=0}^{s-1} v_l^i(u_k, t), \quad i = 1, \ldots, p \ , \qquad \text{sur } [t_k, t_{k+1}]$$

(4) Chercher u_k sur l'intervalle $[t_k, t_{k+1}]$, en résolvant le problème :

$$\underset{\substack{u_k \\ t \in [t_k, t_{k+1}]}}{Min} \int_{t_k}^{t_{k+1}} g(x, u_k)\, dt$$

(5) Représenter le graphe des u_k et les courbes x_i sur $[t_k, t_{k+1}]$.

(6) $k := k + 1$;
(7) Modification des conditions initiales du système différentiel linéarisé :
En utilisant une solution tronquée x_i de l'étape précédente, on peut changer les conditions initiales en posant $t = t_k$:

$$x_i(t = t_k) := \sum_{l=0}^{s-1} v_l^i(u_{k-1}, t_k), \ i = 1, \ldots, p \ , \text{sur } [t_k, t_{k+1}]$$

$\eta := k.\Delta t$,
Fin de tant que ;
Fin.

Algorithme de calcul numérique de contrôle optimal (un seul contrôle constant)

Sur chaque intervalle $[t_k, t_{k+1}]$, on a un problème de minimisation d'une fonction à une

seule variable u_k. Si on cherche u_k sur un ensemble fermé et borné de $\mathbb{R}$ et si la fonction objective est continue sur cet ensemble alors nous aurons l'existence d'un contrôle optimal. Quant à l'unicité de la solution elle pourra résulter des propriétés particulières de la fonction.

5 Application au modèle bicomparimental

Considérons le modèle à deux compartiments traduit par le système différentiel suivant [4] :

$$\begin{cases} \dot{x}_1 = -\left(K_e + \frac{V_1}{K_1+x_1}\right) x_1 + \frac{V_2}{K_2+x_2} x_2 + u \\ \dot{x}_2 = \frac{V_1}{K_1+x_1} x_1 - \frac{V_2}{K_2+x_2} x_2 \\ x_1(0) = 0,\ x_2(0) = 0 \end{cases} \tag{13}$$

Ce modèle décrivant l'évolution d'une substance chimique (médicament) dans un organisme humain.

Les variables x_1 , x_2 représentent respectivement la concentration de substance dans le premier et le deuxième compartiment.

K_e constant d'échange linéaire du compartiment 1 avec l'extérieur.

On suppose que le type d'échange du compartiment 1 vers le compartiment 2 et ou du compartiment 2 vers le compartiment 1 est non linéaire de type Michaelis-Menten.

$u(t)$ est une fonction de contrôle inconnue.

Une fois le modèle (11) est identifié. On peut agir à son contrôle de façon à optimiser un certain critère.

On veut maintenir la concentration de la substance chimique dans le deuxième compartiment autour d'une constante fixée "a" pendant $[0, T]$ et la quantité totale de substance active pendant une période T fixée. δ est un facteur de pondération qui sera fixé en fonction des objectifs de la thérapie.

L'objectif est de minimiser le critère :

$$J = \int_0^T (x_2(t) - a)^2 + \delta \int_0^T u^2(t) dt \tag{14}$$

avec $0 \leq u(t) \leq 1$

On va résoudre ce problème par la méthode numérique de calcul de contrôle optimal. En linéarisant le système différentiel (13).

Le but est de chercher la quantité optimal $u^*(t)$ tel que :

$$u^*(t) = Min(1, u(t))$$

On considère ici le contrôle $u(t)$ comme étant une fonction constante.

6 Résultas numériques

- On suppose que les paramètres et les conditions initiales du système sont fixés :

$$\begin{array}{lll} x_1(0)=0 & V_1=0.32 & K_1=0.52 \\ x_2(0)=0 & V_2=0.12 & K_2=0.13 \\ & K_e=0.0004 & \end{array}$$

Puis appliquer la méthode numérique de calcul de contrôle optimal où le contrôle $u=u_k$ est constant sur $[t_k,t_{k+1}]$.

On pose :

$f_1=-\left(K_e+\frac{V_1}{K_1+x_1}\right)x_1+\frac{V_2}{K_2+x_2}x_2+u$

$f_2=\frac{V_1}{K_1+x_1}x_1-\frac{V_2}{K_2+x_2}x_2$

On linéarise le système différentiel (13), on obtient le système linéarisé :

$$\left[\begin{array}{c}\dot{x}_1\\ \dot{x}_2\end{array}\right]=\left[\begin{array}{cc}-\left(K_e+\frac{V_1}{(K_1+x_1(t=t_k))^2}\right) & \frac{V_2}{(K_2+x_2(t=t_k))^2}\\ \frac{V_1}{(K_1+x_1(t=t_k))^2} & -\frac{V_2}{(K_2+x_2(t=t_k))^2}\end{array}\right]\left[\begin{array}{c}x_1\\ x_2\end{array}\right]+\left[\begin{array}{c}1\\ 0\end{array}\right]u$$

Posons $u=u_k$:

$$\left\{\begin{array}{l}\dot{x}_1=-\left(K_{1e}+\frac{V_1}{(K_1+x_1(t=t_k))^2}\right)x_1+\frac{V_2}{(K_2+x_2(t=t_k))^2}x_2+u_k\\ \dot{x}_2=\frac{V_1}{(K_1+x_1(t=t_k))^2}x_1-\frac{V_2}{(K_2+x_2(t=t_k))^2}x_2\\ x_1(0)=0,\ x_2(0)=0\end{array}\right. \tag{15}$$

Résolvant le système linéarisé (15) par la méthode décompositionnelle d'Adomian, on obtient :

$$\begin{array}{l} x_{10}=x_1(t_k)+u_k,\quad x_{20}=x_2(t_k)\\ x_{1j+1}=-\left(K_{1e}+\frac{V_1}{(K_1+x_1(t=t_k))^2}\right)L^{-1}x_{1j}+\frac{V_2}{(K_2+x_2(t=t_k))^2}L^{-1}x_{2j}\\ x_{2j+1}=\frac{V_1}{(K_1+x_1(t=t_k))^2}L^{-1}x_{1j}-\frac{V_2}{(K_2+x_2(t=t_k))^2}L^{-1}x_{2j}\ ,\ \ j=0,1,...\end{array}$$

où $L^{-1}(.) = \int_{t_k}^{t_{k+1}}(.)ds$ et $t_0=0$.

Une solution approchée à l'ordre s pour les $x_i(t)$, $i=1,2$:

$$\phi_{1s}=\sum_{j=0}^{s-1}x_{1j},\qquad \phi_{2s}=\sum_{j=0}^{s-1}x_{2j}$$

On fixe $a=0.035,\ T=2.5$ et δ =0.00001, on obtient les courbes des solutions représentant la concentration de la substance dans le premier compartiment (voir figure 1) et la

concentration de la concentration de la substance dans le deuxième compartiment comme montre la figure 2 :

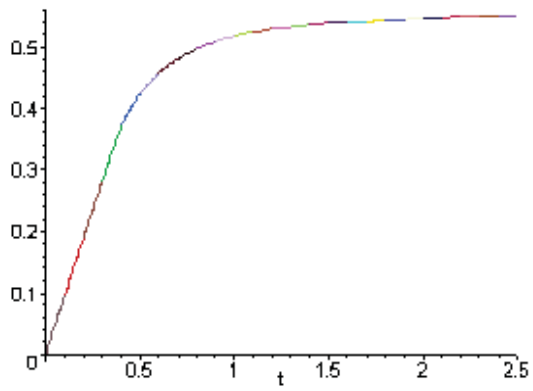


Fig1: Concentration de la substance x_1

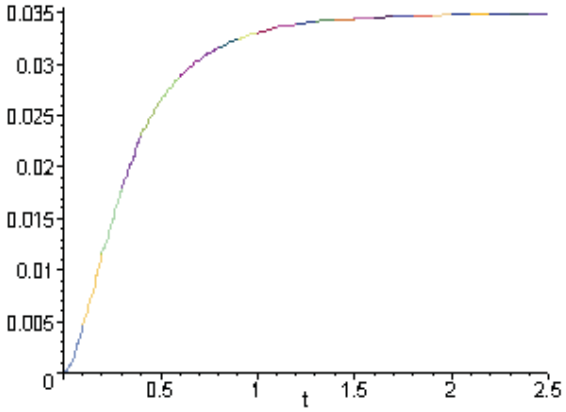
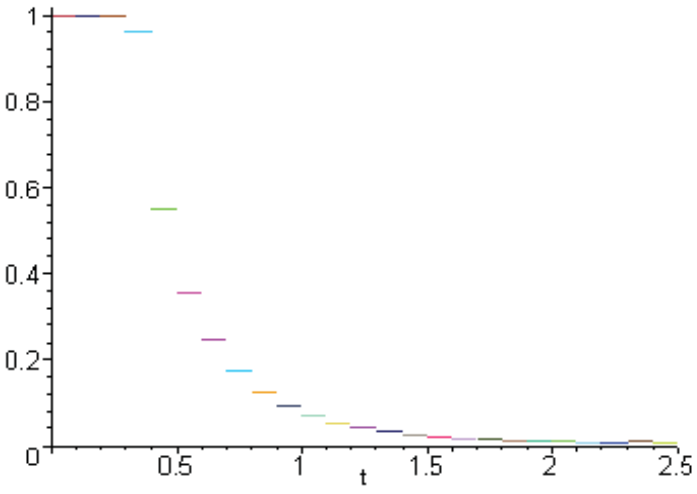


Fig 2 : Concentration de la substance x_2

La courbe de contrôle est illustrée par la figure suivante :



Conclusion

Nous avons considéré le problème de contrôle optimal dans la cas où le système a contrôlé est non linéaire. Pour résoudre ce problème, on linéarise d'abord le système non linéaire, puis on propose l'utilisation de la méthode décompositionnelle d'Adomian pour le résoudre. Cette méthode permet d'exprimer les solutions explicitement des contrôles. Une application de cette méthode au modèle bicompartimental a été réalisée et les résultats obtenus sont satisfaisants, dans le sens où la concentration est proche de la valeur désirée.

References

[1] Abbaoui, K. (1995), *Les fondements mathématiques de la méthode décompositionnelle d'Adomian et application à la résolution de problèmes issus de la biologie et de la médecine*. Thèse de l'Université Paris VI. Laboratoire MEDIMAT.

[2] Abbaoui, K., Cherruault,Y., and N'Dour, M.N. (1995), *The decomposition method applied to differentiel systems,* Kybernetes, Vol. 24, No.8, pp. 32-40.

[3] Abbaoui, K., Cherruault, Y. (1994), *Convergence of Adomian's method applied to differentiel equations,* Computers Math applic, Vol. 28, No. 5, pp 103-109.

[4] Cherruault, Y. (1998), *Modèles et méthodes mathématiques pour les sciences du vivant*, Presses Universitaires de France (P.U.F), Paris.

[5] Cherruault, Y. (1999) : *Optimisation, méthodes locales et globales*, Presses Universitaires de France (P.U.F), Paris, (1999).

[6] Manseur, S., Attalah, K., Cherruault, Y.(2005): *Optimal control of chemotherapy of HIV model using the combined Adomian/Alienor methods*, Kybernetes, V.34, n0 7/8, 1200-1210.

[7] Trélat, E (2005), *Contrôle optimal : Théorie et applications* . Edition Vuibert.

Secure In-Network Data Aggregation in Wireless Sensor Networks: Issues and Solutions

Nabila Labraoui [1], Mourad Guerroui [2], Tanveer Zia [3]

[1] University of Tlemcen, Algeria
[2] PRISM University of Versailles, France
[3] Charles Sturt University, Australia

Abstract. Data aggregation in wireless sensor networks (WSN) is a rapidly emerging research area. It can greatly help conserve the scarce energy resources by eliminating redundant data thus achieving a longer network lifetime. However, securing data aggregation in WSN is made even more challenging by the fact that the sensor nodes and aggregators deployed in hostile environments are exposed to various security threats. In this paper we survey the current research related to security in data aggregation in wireless sensor networks. We have classified the security schemes studied in two main categories: cryptographic based scheme and trust based scheme. We provide an overview and a comparative study of these schemes and highlight the future research directions to address the flaws in existing schemes.

Key words: Data Aggregation, Security, Sensor Networks .

1 Introduction

Advancements in micro electro mechanical systems (MEMS) [9] and wireless networks have made possible the advent of tiny sensor nodes called “smart dust” which are low cost small tiny devices with limited coverage, low power, smaller memory sizes and low bandwidth [2]. A wireless sensor network (WSN) [5] is an ad-hoc network consisting of a large number of sensor nodes deployed to sense their surrounding environment. Sensor nodes are usually used to collect and report application-specific data to the monitoring node, known as a “sink node” [1]. WSNs are expected to be solutions to many applications [12,13], such as providing health care for the elderly, surveillance, emergency, and battlefield intelligence data gathering.

Along with the attractive features and increasingly important roles, sensor networks however have their inherent limitations: resource constraints, which is determined by the design goal of small-size and low-cost; security vulnerability, due to the open nature of wireless communication channels and the lack of physical protection of individual sensor nodes which makes it easy for the adversaries to eavesdrop the communication and compromise sensor nodes [3]. Security solutions require high computation, memory, storage and energy resources which creates an additional challenge when working with tiny sensor nodes [2]. Extensive research has been conducted to address these limitations by developing schemes that can improve resource efficiency and enhance security.

In the resource constrained WSN environment, forwarding of large amounts of data becomes the major focus of energy and bandwidth optimization efforts. Data aggregation has thus been put forward as an essential technique to achieve power and bandwidth efficiency in WSN. Based on the principle that the sink does not necessarily need all raw pieces of data collected by each sensor but only a summary or aggregated data thereof, data aggregation is done by aggregators

which are comparatively powerful sensor nodes having the ability to aggregate and process data forwarded by source nodes at each intermediate node enroute to the sink. The Data communication constitutes an important share of the total energy consumption of the sensor network; sending one bit requires almost the same amount of energy as executing 50 to 150 instructions [38]. Thus, data aggregation can greatly help conserve the scarce energy resources by eliminating redundant data [6], and achieving a longer network lifetime. Typical aggregation functions include SUM, AVERAGE, MAX/MIN, and so on [4, 11]. However, data aggregation in sensor networks is even more challenging by the fact that the sensor nodes and aggregators deployed in hostile environments are exposed to various threats such as node compromise, injection of bogus aggregators, disclosure of sensed data and aggregate values to intruders or tampering with nodes and data transmitted over wireless links. Therefore, the processing and aggregation mechanisms need to be resilient against attacks where the aggregator and a fraction of the sensor nodes may be compromised.

The rest of the paper is organized as follows. Section II introduces the problem statement of secure data aggregation in WSN. While Section III presents the classification of secure aggregation schemes with description of existing works and Section IV discusses the performance comparison of the cited schemes. Concluding remarks are given in Section V.

2 Problem statement

2.1 Security Requirement in WSN aggregation

In-network data processing, such as data fusion and aggregation [40], has emerged in the recent years as an active research area in WSN. One of the important issues related to data aggregation is to find a realistic balance between computational overhead, delay, data resolution and trustworthiness [7]. This section describes the required security primitives to strengthen the security in aggregation schemes.

Data Integrity: This property ensures that the content of a message has not been altered, during transmission process. An adversary near the aggregator point will be able to change the aggregated result sent to the sink by adding some fragments or manipulating the packet's content without detection.

Data confidentiality: It is also essential to prevent leakage of sensitive data. If, for example, a network was responsible for monitoring a military target for the purpose of planning a surprise attack, then it would be necessary to ensure that the privacy of the information is preserved so that the target does not become aware of the ensuing plans. For this reason, a sensor network that uses data aggregation is also required to protect the confidentiality of the aggregated data.

Data Authentication: guarantees that the reported data is the same as the original one and it has come from reliable source "identification". In secure data aggregation, both identification and authentication are important to ensure the legitimate data transfer between sensors.

Data Freshness and Availability: Given that sensor networks are used to monitor time-sensitive events, it is important to ensure that the data provided by the network is current and available at all times. This means that an adversary can not replay old messages in the future.

2.2 Attacks against WSN aggregation

WSNs are vulnerable to different types of attacks [16] due to the nature of the transmission medium (broadcast), remote and hostile deployment location, and the lack of physical security in each node. However, the damage caused by these attacks varies in applications depending on the assumed threat model. Therefore, data aggregation must be done securely so as to prevent a deceptive reading of the state of the environment being monitored. In this section, we discuss about the attacks that might affect the aggregation in the WSN.

Node compromise: Current sensor hardware does not provide any resistance to physical tampering. If an adversary captures a node, he can easily extract the cryptographic primitives as well as exploit the shortcomings of the software implementation. This allows the adversary to launch attacks from within the system as an insider, bypassing encryption and password security systems. Considering the data aggregation scenario, the compromised nodes can successfully authenticate bogus reports to their neighbours, which have no way to distinguish bogus data from legitimate ones [8].

Denial of Service Attack: is a standard attack on the WSN by transmitting radio signals that interfere with the radio frequencies used by the WSN and is sometimes called jamming. As the adversary capability increases, it can affect larger portions of the network. In the aggregation context, an example of the DoS can be an aggregator that refuses to aggregate and prevents data from travelling into the higher levels.

Sybil attacks: refers to the scenario when a malicious node pretends to have multiple identities. For example, the malicious node can claim false identities, or impersonate other legitimate nodes in the network [16]. It affects aggregation schemes in different ways [15]. Firstly, an adversary may create multiple identities to generate additional votes in the aggregator election phase and select a malicious node to be the aggregator. Secondly, the aggregated result may be affected if the adversary is able to generate multiple entries with different readings. Thirdly, some schemes use witnesses to validate the aggregated data and data is only valid if n out of m witnesses agreed on the aggregation results. However, an adversary can launch a Sybil attack and generate n or more witness identities to make the base station accept the aggregation results.

Selective Forwarding Attack: In selective forwarding, a malicious node acts like a black hole and refuses to forward every packet. Adversary uses the compromised node to forward the selected messages. In the aggregation context, any compromised intermediate nodes have the ability to launch the selective forwarding attack and this subsequently affects the aggregation results.

Replay Attack: In this case an attacker records some traffic from the network without even understanding its content and replays them later on to mislead the aggregator and consequently the aggregation results will be affected.

Stealthy Attack: the adversary's goal is to make the user accept false aggregation results, which are significantly different from the true results determined by the measured values, while not being detected by the user.

2.3 Security Design Challenges for WSN Aggregation

Consequently, it is believed that a secure data aggregation is very challenging issue, and requires more attention during design process. According to the properties required by the application and according to the type of attack and the type of adversary, a secure data aggregation scheme should have as well as possible following properties:

Low communication overhead: the purpose of conducting aggregation is to reduce communication overhead. Thus a secure scheme should maintain this purpose.
Scalability: Secure aggregation techniques should provide high-security features for small networks, but also maintain these characteristics when applied to larger ones.
Flexibility: secure aggregation techniques should be able to function well in any kind of environments and support dynamic deployment of nodes.
Effectiveness: it is important to ensure the accuracy of the final aggregation result.
Generality: the secure aggregation scheme should apply to various aggregation function, such as MAX/MIN, MEAN, SUM, COUNT, and so forth.

3 Classification of secure aggregation schemes

Many innovative and intuitive secure aggregation schemes for WSNs have been proposed for solving the problem of security in sensor networks. In this section we survey these schemes and classify them into two classes: cryptographic-based secure data aggregation and trust-based secure data aggregation schemes. See figure 1.

3.1 Cryptographic-based secure aggregation

The security issues, such as data confidentiality and integrity in data aggregation become vital when the sensor network is deployed in a hostile environment. Most current research in securing data aggregation in WSNs, have been achieved through cryptographic scheme. We distingue two techniques: techniques based on concealed data (End-to-End privacy) and techniques based on revealed data (Hop-By-Hop privacy).

A. Techniques based on Concealed Data

Concealed data aggregation (CDA) is an improved version of the in-network aggregation, which in contrast to the classic Hop-by-Hop ensures the End-to-End privacy, i.e. encrypted values do not need to be decrypted for the aggregation. Instead, the aggregation is performed with encrypted values and only the sink can decrypt the result. The fundamental basis for CDA are cryptographic methods that provide the privacy homomorphism (PH) property. An encryption algorithm E() is homomorphic, if for given E(x) and E(y) one can obtain E(x*y) without decrypting x,y for some operation *. The concept was introduced by Rivest et al. [17] in 1978. The two most common variations of PHs are the additive PH and the multiplicative PH. The latter provides the property $E(x \times y)=E(x) \otimes E(y)$.

Girao et al. [18] propose a CDA scheme (CDAM) that is based on the PH proposed in [20]. They claimed that, for the WSN data aggregation scenario, the security level is still adequate and the proposed PH method can be employed for encryption.

Castelluccia et al. proposed a simple and provable secure additively homomorphic stream cipher (HSC) that allows for the efficient aggregation of encrypted data [19]. The new cipher replace the xor (Exclusive-OR) operation with modular addition and is therefore very well suited for CPU-constrained devices such as those in WSNs. One limitation of this proposal is the important overhead and scalability problem that generates if the network is unreliable.

Recently, Önen et al. [21] and Castellucia [22] propose a new scheme that combines a PH and multiple encryptions, (PHM1 and PHM2). Theses two works are quite similar but were

developed in parallel and independently. The homomorphic of the underlying encryption technique allows sensors to aggregate their cleartext measurements with the encrypted aggregate values whereas the multiple encryption scheme assure that aggregates values and individual measurements results remain oblivious to all intermediate nodes enroute to the sink. The proposed scheme assures the end-to-end confidentiality and scales efficiency. It improves the bandwidth performance of secure aggregation scheme described in [19] and allows resisting against n compromised nodes.

B. Techniques based on revealed data

Many protocols based on revealed data (hop-by-hop privacy) provide more efficient aggregation operation and highly consider data integrity.

Hu and Evans proposed the first secure data aggregation (SDA) protocol for WSNs that is resilient to both intruder devices and single device key compromises [23]. However, the protocol may be vulnerable if a parent and a child node in the hierarchy are compromised.

Przydatek et al. proposed a secure information aggregation (SIA) framework for sensor networks [24]. This framework provides resistance against stealthy attacks. It consists of three node categories: a home server, a base station, and sensor nodes. A base station is a resource enhanced node which is used as intermediary between the home server and the sensor nodes, and is also the candidate to perform the aggregation task. SIA assumes that each sensor has a unique identifier and shares a separate secret cryptographic key with both the home server and the aggregator. The keys enable message authentication and encryption if data confidentiality is required. Moreover, it assumes that the home server and base station can use a mechanism, such as μTESLA, to broadcast authentic messages. SIA consist of three parts: collecting data from sensors and locally computing the aggregation result, committing to the collected data, and reporting the aggregation result while providing the correctness of the result.

Çam et al. proposed an energy-efficient secure pattern based data aggregation (ESPDA) protocol for wireless sensor networks in [25,26]. ESPDA is applicable for hierarchy-based sensor networks. In ESPDA, a cluster head first requests sensor nodes to send the corresponding pattern code for the sensed data. If multiple sensor nodes send the same pattern code to the cluster head, only one of them is permitted to send the data to the cluster head. ESPDA is secure because it does not require encrypted data to be decrypted by cluster heads in order to perform data aggregation.

Du et al. proposed a witness-based data aggregation (WDA) scheme for WSNs to assure the validation of the data sent from data fusion nodes to the base station [27]. In order to prove the validity of the fusion result, the fusion node has to provide proofs from several witnesses. A witness is one who also conducts data fusion like a data fusion node, but does not forward its result to the base station. Instead, each witness computes the message authentication code (MAC) of the result and then provides it to the data fusion node who must forward the proofs to the base station.

Discussion

The field of cryptography within in-network data processing is a very promising research field, and introduces many interesting challenges [7]. However, selecting the appropriate cryptography method for sensor nodes is fundamental to providing security services in WSNs. Symmetric key cryptography is commonly used in WSN, and is superior to public key cryptography in terms of speed and low energy cost.

In spite of the diversity and the proved efficiency of these solutions (cryptography methods),

many of them assume that the sensor nodes are trustworthy and reporting data truthfully. In practice though, sensors are usually deployed in open unattended environments, and hence are susceptible to physical tampering. When a node is compromised, the adversary can inject bogus data into the network. However, we argue that the conventional view of security based on cryptography alone is not sufficient for the unique characteristics and novel misbehaviours encountered in open networks. Even though cryptography can provide integrity, confidentiality, and authentication, it fails in the face of insider attacks. This necessitates a system that can cope with such internal attacks.

3.2 Trust-based secure data aggregation

As we cite above, WSNs are often deployed in unattended territories that can often be hostile, they are subject to physical capture by adversaries. A simple tamper-proofing is not a viable solution [10]. Hence, sensors can be modified to misbehave and disrupt the entire network. This allows the adversary to access the cryptographic material held by the captured node and allow the adversary to launch attacks from within the system as an insider, bypassing encryption and password security systems. The compromised nodes can successfully authenticate bogus reports to their neighbors, which have no way to distinguish false data from legitimate ones [8]. Trust and reputation systems have been proposed as an attractive complement to cryptography in securing WSNs. They provide the ability to detect and isolate both faulty and malicious nodes that are behaving inappropriately in the context of the specific WSN.

Recently, attention has been given to the concept of trust to increase security and reliability in Ad Hoc [32,33] and sensor networks [35, 34]. The notion of trust to be used throughout this paper is briefly defined as: trust is the degree of belief about the future behavior of other entities, which is based on the ones the past experience with and observation of the others actions. Reputation is another complex notion that spans across multiple disciplines. It is quite different from but easily confused with trust.

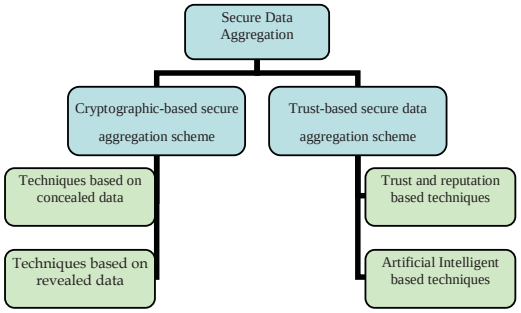


Fig. 1. Classification of secure aggregation schemes.

The mathematical foundation for reputation management is rooted in statistics and probability [39]. Furthermore, reputation is based on collection of evidence of good and bad behaviour undertaken by other entities. It is based on past experience with a given entity, whereas trust is not restricted to this. In this section, we study the proposed schemes based on

trust and reputation for secure data aggregation in networks. We classify them into two categories: trust and reputation-based framework, and artificial intelligent-based framework.

A. Trust and reputation-based framework

In [28], the authors propose a trust based framework for secure data aggregation in WSN based on Bayesian model and beta distribution probability (TKL). They first evaluate trust in individual sensor nodes based on Kullback-Leibler (KL) distance or relative entropy. The idea is to calculate the distance between an ideal node behaviour and the actual node behaviour. The authors assign a confidence value to aggregated sensor data. Based on sensor data confidence, the framework computes an opinion which encompasses belief and uncertainty on the aggregation of sensor data by means of the consensus operator [37]. Nevertheless, this approach is still time-consuming for establishing a stable reputation on sensor nodes. The reputation on a node, based the inverse square of its KL distance, suffers from severe oscillation for the first reputation evaluation.

Hur et al. propose in [29] a trust-based aggregation scheme for WSN based on local trust evaluation (LTE). This local trust evaluation mechanism is suitable for resource-limited sensor networks. The trustworthiness of a node is computed based upon several trust evaluation factors, such as battery lifetime, sensing communication ratio, sensing result and consistency level. Each sensor node computes only their neighbour node's trust accumulatively. Prior to a data aggregation, sensor nodes elect an aggregator node in their own grid, which has the highest trust value among all the nodes in an identical grid by the majority of vote. A trust agreement process is necessary, because the trust value of a node is evaluated by its neighbour nodes, that is any node never know it's own trust value. Sensing data from multiple nodes are aggregated in consideration of agreed trust values of member nodes per each grid. Deceitful data from malicious or compromised nodes whose trust value are lower than those of the other legal nodes can be excluded. One drawback of this scheme is that it considers that the trust evaluation is computed by only trustworthiness nodes.

Rabinovich et al. [30] propose a mechanism to detect and mitigate stealthy attacks against sensor networks by using distributed localized constraint validation and randomized routing (RTM). More specifically, they consider those applications of sensor networks where measurements are spatially correlated. This correlation is expressed in the form of measurement constraint. Sensors observe their neighbors during transmission and file "complaints" if their neighbours' data violate the constraint when compared with their own. Protection against compromised aggregators is achieved via construction of randomized delivery trees. Every time the sensor network sends data to the server it includes both current and small number of recent measurements. This allows the server to detect an attack when a fairly small fraction of sensors have been compromised. Based on complaints, the server uses a beta reputation system [36] to identify compromised sensors. In order to test the effectiveness of the scheme, authors developed a custom simulator to analyse the time to detect a compromised sensors, the fraction of compromised sensors successfully detected by the system, and the communication overhead introduced by the security protocol.

		CRYPTOGRAPHIC BASED SCHEMES								TRUST BASED SCHEMES			
		Concealed Data Techniques				Revealed Data Techniques				Trust and Reputation			IA
		CDAM [18]	HCS [19]	PHM1 [21]	PHM2 [22]	SDA [23]	SIA [24]	ESPDA [25,26]	WDA [27]	TKL [28]	LTE [29]	RTM [30]	IAF [31]
Security services	Confidentiality	*	*	*	*		*	*					
	Integrity					*	*		*				
	Authentication					*	*						*
	Freshness					*	*			*	*	*	*
	Availability			*						*	*	*	*
Attacks existence	Node compromise	*	*	*	*	*	*	*	*				
	DoS	*	*	*	*	*	*	*	*	*	*	*	*
	Selective forwarding			*	*	*	*		*				
	Sybil							*	*	*	*	*	*
	Stealthy					*			*				
Design Goal	Scalability			*	*			*			*		
	Overhead	high	high	high	high	low	low	low	high	low	low	high	low
	Flexibility	*		*	*								
	Effectiveness		*				*		*	*	*	*	
	Generality		*	*	*	*		*	*	*	*	*	*

Table 1. a summary of comparison between secure data aggregation schemes.

B. Artificial intelligent-based framework

Interest on applying Artificial Intelligence techniques on securing sensor network environments is rising. [11] Use offline neural network based learning technique to model spatial patterns in sensed data. [13] applies reinforcement learning techniques for intrusion detection. [14] and [12] base their detection systems on multi-agent systems. The critical issue of aggregation, however, is not taken into consideration by any of these researches.

In [31] the authors propose the first mechanism (IAF) that combines statistics and artificial intelligence techniques for robust detection of malicious nodes in a sensor network environment without unnecessarily eliminating honest nodes, e.g., the descendants of a malicious node. In particular, hypothesis testing mechanisms are used by a child node to estimate the probability of error reporting by a child node over one data reporting epoch, while reinforcement learning schemes are used to update such reputation over successive epochs. To create a more robust system, reputations must be accumulated over successive epochs, and only if consistent deviations are observed should a parent node be labelled to be malicious. One drawback of this scheme is that it cannot detect unbiased errors introduced by the aggregator node. A very high learning rate can increase detection but also introduce unacceptable false positives.

Discussion

Building a robust trust and reputation system presents several important challenges on its own [10]. The most pressing is the possibility that a malicious node that participates in the reputation system can prevent it from functioning by lying. A compromised node can falsely accuse well-behaving nodes of malicious actions or falsely praise bad-behaving nodes (pollution reputation). To maintain its integrity the reputation system must be able to prevent these kinds of attacks. Another important issue when building reputation is determining when a node has performed a malicious action and being able to distinguish it from natural failures. Due to the uncertain nature of WSN environment, such as collisions on the wireless channel, it is not always possible to distinguish these two kinds of erroneous behaviors.

4 Performance Comparison

This section, attempts to compare the secure data aggregation schemes that were reviewed in the last section. Comparison of security schemes can be difficult since the designers solve secure aggregation from different angles. Therefore, these schemes are compared in a number of different ways: security services provided (confidentiality, integrity, authentication, freshness and availability), design goal (scalability, overhead, flexibility, effectiveness and generality), and resilience against attacks described in section 2.2. We summarise this comparison in table 1.

Since the assumed adversary varies from one scheme to other, each proposed scheme has different requirements. In cryptographic based scheme, the data confidentiality represents the minimum security requirements. As we see in table, the techniques based on revealed data provide more efficient aggregation operation in term of overhead and highly consider data integrity. However, they represent weaker model of data confidentiality perspective than techniques based on concealed data. However, in trust based scheme, the availability and the network lifetime are the principle concern and should be provided as well as possible.

We notice that all the proposed schemes based on cryptography are vulnerable to DoS and physical attacks, while all the trust based schemes are vulnerable to DoS and Sybil attacks.

Metrics used to evaluate the design goals are: Scalability, Overhead, Flexibility, and Effectiveness.

Scalability: as we have cited in section 2.3, a scheme is considered scalable if it can provide high-security feature as well for small networks that for a larger ones. Alzaid et al. [15] classify the secure data aggregation into two models: the single aggregator and the multiple aggregators. In the single aggregator model all individual data in the WSN travels to only one aggregator point in the network before reaching the base station (the querier). This model is useful when the network is small, however large networks are not suitable places to implement this model especially when data redundancy at the lower levels is high [15]. This is the case of SIA. In the multiple aggregators model, the collected data in the WSN are aggregated more than one time before reaching the querier [15]. We could think that this model is well suitable in the large WSNs because it achieves greater reduction in the number of bits transmitted. However, this affirmation is credible only if the scheme does not generate a high overhead in communication and bandwidth. We cite the example

of PHM1 and PHM2 which in spite of the high overhead (caused by the multiple encryption that consumes energy), scale efficiency because it improves the bandwidth gain, while HCS is not scalable especially if the network is unreliable because it generates an important traffic in bandwidth when transmitting the ID of nodes which participate (or not participate) in the aggregation function.

Overhead: the scheme introduces overhead that consumes bandwidth and energy. This overhead is measured by the size of transmitted message and the energy consumption. In concealed data techniques, data encryption is very expensive, and generates important energy consumption especially if the data is encrypted more than one time (multiple encryptions). It is the case in CDAM, HCS, PHM1 and PHM2. In the case of WDA, to prove the validity of aggregation result, the aggregator has to provide proofs from several witnesses. It must then forward the proofs to the base station by piggybacking them with the aggregation result. This introduces high bandwidth consumption. RTM [30] which is a trust based scheme makes the network more reliable but introduces a high overhead.

Effectiveness: the scheme which ensures the accuracy of the final aggregation result can contain a verification phase to enhance the querier ability in distinguishing between the valid and the invalid aggregated readings, as the case of HCS, SIA and WDA in cryptographic based schemes. The scheme can also ensure the effectiveness of aggregation by monitoring the aggregator when sending data aggregation result to the querier. If monitors detect that the aggregator send an altered data, they send an alarm to the querier. This is the case of TKL, LTE and RTM in the trust based schemes. However, IAF cannot ensure completely the accuracy of aggregation because this scheme cannot detect unbiased errors introduced by the aggregator node.

Flexibility: all the proposed schemes are designed for static sensor networks. They don't consider the mobility of the node. However, the topology of WSNs can vary some times. The scheme must take into account this characteristic and consider addition of new nodes or removing them in the network while supporting the same security level. It is true that this property is very difficult to ensure. Only CDAM, PHM1 and PHM2 are flexible.

Finally, there is no scheme which is perfect. Each proposal has its advantages and its limitations. A tradeoff between security level and performance must be carefully balanced.

5 Conclusion

Sensor networks are vulnerable to insider and outsider attacks much more than other wireless networks for the reasons discussed in section 2. When designing a security protocol it is important to understand the dangerous and damaging effects these attacks can have so that the protocol can guard against them. Secure data aggregation in wireless sensor networks is a critical issue that has been addressed through many proposed schemes. These proposals are classified into two categories: cryptographic based scheme and trust based scheme.

Some promising results have been recently achieved in secure data aggregation. They are based on advanced cryptographic concepts, such as privacy homomorphism. Despite of the diversity and the proved efficiency of these solutions, many of them assume that the sensor nodes are trustworthy and reporting data truthfully. Even though cryptography can provide integrity, confidentiality, and authentication, it fails in the face of insider attacks. Another important issue is related to assessment of trustworthiness and reliability of the data provided by WSNs. However, this issue is still in its infancy, and there are not clear trust evaluation models

which can be applied to sensor networks properly.

This paper provides an overview of theses techniques, each of which offers different advantages and disadvantages. A balance between the requirements and resources of a WSN determines which technique should be employed. We notice that no secure aggregation technique is ideal to all the scenarios where sensor networks are used; therefore the techniques employed must depend upon the requirements of target application and resources of each individual sensor network.

Despite of potentially great importance and very interesting theoretical and practical challenges, the topic of secure data aggregation in wireless sensor networks have received until recently much less attention than, e.g., secure routing or key management. Therefore, despite of many interesting initial results, the security questions related to data aggregation in WSN remain largely open, and in our opinion constitute an interesting area for further research.

Références

[1] O. Moussaoui, A. Ksentin, M. Naimi and M. Gueroui, " a novel clustering algorithm for efficient energy saving in wireless sensor networks" in the 7th IEEE International Symposium on Computer networks (ISCN'06), Istanbul, Turkey, June 2006.

[2] T.A Zia and A.Y Zomaya, "A security framework for wireless sensor networks". In Proceedings of the IEEE Sensor Applications Symposium (SAS) , Houston, Texas, February 7-9, 2006.

[3] H. Wenbo, L. Xue, N. Hoang, K. Nahrstedt And T. Abdelzaher, "PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks", In 26th IEEE International Conference on Computer Communications, (INFOCOM), Anchorage, AK, May 2007, pp. 2045-2053

[4] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring,", In Proceedings of the ACM Conference on Embedded Networked Sensor Systems, Baltimore, MD, November 2004.

[5] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, "SPINS: Security protocols for sensor network", Wireless Networks, 2002, vol. 8, no. 5, pp. 521-534.

[6] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey", In Communications Surveys & Tutorials, IEEE, Fourth Quarter 2006, Vol. 8, Issue 4, pp 48-63.

[7] A.Sorniotti, L.Gomez, K.Wrona and L.Odorico "Secure and trusted in-network data processing in wireless sensor networks: a survey", JIAS, Journal of Information Assurance and Security, September 2007, Vol. 2, Issue 3.

[8] A. Perrig, J. Stankovic, D.Wagner, "Security in Wireless Sensor Networks", Communication of the ACM, June 2004.

[9] B. Warneke, K.S.J. Pister, "MEMS for Distributed Wireless Sensor Networks," 9th Int'l Conf on Electronics, Circuits and Systems, Dubrovnik, Croatia, September 2002.

[10] S. Ganeriwal and M. Srivastava, "Reputation-based framework for high integrity sensor networks", In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN), October 2004 , pp. 66-77.

[11] P.Mukherjee and S.Sen, "Detecting malicious sensor nodes from learned data patterns", In Proceedings of the Workshop on Agent Technology for Networks, 2007, pp. 11-17.

[12] R. M. Ruairi and M. T. Keane, "An energy-efficient, multi-agent sensor network for detecting diffuse events", In IJCAI, 2007, pp. 1390–1395.

[13] A. L. Servin and D. Kudenko, "Multi-agent reinforcement learning for intrusion detection", In Adaptive Learning Agents and Multi Agent Systems, 2007, pp. 158–170.

[14] J. Wu, C. jun Wang, J. Wang, and S. fu Chen, "Dynamic hierarchical distributed intrusion detection system based on multi-agent system", In Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, Washington, DC, USA, 2006, pp. 89–93.

[15] H. Alzaid, E. Foo And J.M Gonzalez. "secure data aggregation in wireless sensor networks: a survey" In L. Brankovic and M.Miller, editors, Sixth Australasian Information Security Conference (AISC) Wollongong, NSW, Australia, 2008, vol. 81 of CRPIT, pp. 93-105.

[16] T. Roosta, S. Shieh, And S. Sastry, "Taxonomy of security attacks in sensor networks", In 'The First IEEE International Conference on System Integration and Reliability Improvements', IEEE International, Washington, DC, USA, 2006.

[17] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On Data Banks and Privacy Homomorphisms,", In Foundations of Secure Computation, New York: Academic, 1978, pp. 169–79.

[18] J. Girao, D. Westhoff, and M. Schneider, "CDA: Concealed Data Aggregation for Reverse Multicast Traffic wireless Sensor Networks," In Proceeding IEEE Int'l. Conf. Commun. , Seoul, Korea, May 2005.

[19] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient Aggregation of Encrypted Data Wireless Sensor Network," In Proceeding .ACM/IEEE Mobiquitous, San Diego, CA, July 2005.

[20] J. Domingo-Ferrer, "A Provably Secure Additive and Multiplicative Privacy Homomorphism," Lecture Notes Comp. Sci., vol. 2433, 2002, pp. 471–83.

[21] M. Onen and R. Molva, "Secure data aggregation with multiple encryption" 4th European conference on Wireless Sensor Networks, Delft, The Netherlands, January 29-31, 2007, Also published as LNCS Vol. 4373 , pp 117-132.

[22] C.Castellucia, "Securing very dynamic groups and data aggregation in wireless sensor networks", IEEE International Conference on Mobile Adhoc and Sensor Systems, (MASS), Pisa, Italy, October 2007, pp. 1-9.

[23] L. Hu and D. Evans, "Secure aggregation for wireless networks", In Proceeding Of Workshop on Security and Assurance in Ad hoc Networks, Orlando, FL, January 2003.

[24] B. Przydatek, D. Song, and A. Perrig, "SIA : Secure information aggregation in sensor networks", In Proceeding Of SenSys'03, Los Angeles, CA, Nov 5-7, 2003.

[25] H. Çam, D. Muthuavinashiappan, and P. Nair, "ESPDA: Energy Efficient and Secure Pattern-Based Data Aggregation for Wireless Sensor Networks," In Proceding IEEE Sensors, Toronto, Canada, Oct. 2003, pp. 732–36.

[26] H. Çam, D. Muthuavinashiappan, and P. Nair, "Energy-Efficient Security Protocol for Wireless Sensor Networks,", In Proc.IEEE VTC Conf., Orlando, FL, Oct. 2003, pp. 2981–84.

[27] Du, W., Deng, J., Han, Y. S. & Varshney, P. A witness-based approach for data fusion assurance in wireless sensor networks, In 'IEEE Global Communications Conference (GLOBECOM), 2003, Vol. 3, pp. 1435– 1439.

[28] W. Zhang, S. Das and Y. Liu, A trust based framework for secure data aggregation on wireless sensor networks. In Proceedings of th 3rd Annual IEEE Communications Society and Networks (SECON), 2006, pp. 60-69.

[29] J. H. Junbeom , L. Yoonho, H. Seongmin, Y. Hyunsoo, "Trust-based aggregation in wireless sensor networks", International Conference on Computing, Communications and Control Technologies, July 2005.

[30] P. Rabinovich, R. Simon, "Secure aggregation in sensor networks using neighbourhood watch" , In IEEE International Conference, Glasgow, June 2007, pp. 1484-1491.

[31] A. Gursel, O. Mistry, S. Sandip, " Robust Trust Mechanisms for Monitoring Aggregator Nodes in Sensor Networks", Int. Workshop on Agent Technology for Sensor Networks (ATSN-08), Estoril,

Portugal, May 2008.
[32] S. Buchegger and J. L. Boudec, "Performance analysis of the CONFIDANT protocol," In Proceeding in 3rd ACM int. symp. Mobile ad hoc networking & computing, 2002.
[33] P. Michiardi and R. Molva, "CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks," 2001.
[34] A. Srinivasan, J. Teitelbaum, and J. Wu, "DRBTS: Distributed Reputation-based Beacon Trust System," In 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06), 2006.
[35] S. Ganeriwal and M. Srivastava. Reputation-based framework for high integrity sensor networks. In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04), October 2004, pp. 66-77.
[36] A. Jsang and R. Ismail, "The Beta Reputation System", In Proceedings of the 15th Bled Electronic CommerceConference, June 2002.
[37] Audun Joang, A logic for uncertain probabilities. Int. J. Uncertain. Fusiness Knowl-based system, 9(3):279-311, 2001.
[38] S. Peter, K.Piotrowski, and P. Langndoerfer. "On concealed data aggregation for aggregation for wireless sensor networks", In Proceedings of the IEEE Consumer Communications and Networking Conference, January 2007.
[39] P. Robinson and M. Beigl, "Trust context spaces: An infrastructure for pervasive security", In Proceedings of the first International Conference on Security in Pervasive Computing, 2003.
[40] R.Rajagopalan and P.K Varshney "data aggregation techniques in sensor networks: a survey", communications surveys and tutorials IEEE, 2006.

Vers une solution d'appariement ontologique

A.Boubekeur [1], M. Malki [2], A.Chouarfia[3]

[1] Département d'Informatique Université de Tiaret, Algérie Boubakeur@univ-tiaret.dz
[2] Laboratoire EEDIS, Département d'Informatique, Université de Sidi Bel-Abbes, Algérie
Malki_m@univ-sba.dz
[3] Département d'Informatique Université USTO-Oran, Algérie Chouarfia@univ-usto.dz

Résumé : Les ontologies sont la solution pour parvenir à l'interopérabilité sémantique sur le Web. Mais leur grande diversité sur un même domaine est en soi une source d'hétérogénéité. Cette hétérogénéité peut être d'ordre terminologique, structurelle et/ou sémantique. Une solution à cette difficulté se situe dans le développement de techniques de coordination et d'appariement d'ontologies. Dans ce cadre, l'appariement des ontologies, permettant d'établir des liens sémantiques (équivalence, spécialisation,…) entre ontologies du même domaine, est une autre problématique centrale afin de pouvoir manipuler de façon conjointe plusieurs ontologies au sein d'une même application, ou de permettre à deux applications utilisant des ontologies différentes de communiquer. Notre focalisation est sur la composition d'une variété de mesures de similarité entre les entités des ontologies OWL et sur un processus d'appariement de calcul itératif.

Mots clefs: Ontologies, intéroperabilité sémantique, alignement, mesure de similarité, Word Net, OWL, XML

1 Introduction

L'interopérabilité et l'intégration d'ontologies provenant de sources hétérogènes deviennent incontournables vu le développement croissant du Web et la diversité des domaines. L'appariement d'ontologies permet les échanges de manière sémantique. Cet appariement peut se situer au niveau sémantique et conceptuel ou au niveau des instances et des données [17]. Il est au coeur de la fusion, l'alignement, l'extension,… [2]. Les travaux existant pour établir les relations de mise en correspondance se focalisent à la fois sur les aspects d'automatisation et de découverte de ces relations et aussi sur l'aspect formalisation pour le raisonnement. Ces approches se basant sur des heuristiques permettent d'identifier des similarités syntaxiques et structurelles. L'absence de structure commune, de similarités syntaxiques et d'instances communes rend le problème de mise en correspondance difficile et souvent basé sur de fortes hypothèses pas toujours très réalistes. Le problème principal auquel est confronté le processus de d'appariement est lié à l'hétérogénéité sémantique introduite quand différents modèles sont construits avec

différentes terminologies pour représenter la même information dans les différentes sources.
Les ontologies sont la solution pour parvenir à l'interopérabilité sémantique sur le Web. Et pourtant, de plus en plus, nous nous rendons compte que nous n'arriverons pas toujours à un consensus. Au contraire, un grand nombre d'ontologies verra le jour sur un même sujet, mais elles différent, non seulement au niveau des choix terminologiques ou de la langue choisie, mais également au niveau sémantique et conceptuel : il sera nécessaire d'apparier les termes (concepts, relations, propriétés, …) sous-jacents parce que les points de vue seront différents.
La découverte de similarités repose sur des techniques variées: terminologiques, structurelles et/ou sémantiques [16]. Les techniques terminologiques permettent d'exploiter toute la richesse des noms des termes, en particulier dans les domaines où les homonymes sont rares. Les techniques structurelles permettent de trouver des mises en correspondance potentielles du fait de la position dans la taxonomie des termes candidats lorsque l'exploitation de leur syntaxe ne suffit pas. Enfin, lorsque terminologiquement, ou structurellement, il n'est pas possible de trouver de correspondances, les techniques sémantiques montrent tout leur intérêt [16].
Notre travail se veut d'aborder ces techniques et de fournir un procédé d'appariement d'ontologies OWL. Le but est, et en utilisant WordNet [10], de trouver pour chaque terme (concept, relation, attribut,…) de l'ontologie source de quels termes de l'ontologie cible il peut être similaire ou sémantiquement rapproché (ceux avec lesquels ils partagent des généralisants) puis d'identifier le terme dont il est le plus proche. Notre démarche consiste donc à définir un seuil et procède par la suite à des calculs de similarité entre les différents termes. A partir de ce seuil, on déterminera les entités à considérer comme similaires. S'il s'agit de distance entre entités, les valeurs inférieures au seuil indiqueront une quasi-équivalence des entités, tandis pour les mesures de similarité, les valeurs supérieures au seuil indiquent que les entités sont similaires.
La section 2 de ce papier présente un état de l'art sur les travaux relatifs. Les sections 3 et 4 décrivent respectivement les caractéristiques ontologiques liées à notre processus d'appariement des ontologies. En section 5, nous montrons les résultats expérimentaux et les évaluations de notre stratégie. La dernière section fait des suggestions de conclusion pour les travaux futurs.

2 Etat de l'art

L'appariement d'ontologies est généralement défini comme un processus qui prend deux ontologies en entrée et renvoie un lien de correspondance ou « mapping » qui identifie des termes correspondants dans les deux ontologies, en tenant compte de leurs descriptions et des contraintes en termes de propriétés et de relations sémantiques. Comme résultat d'appariement d'ontologies, des mappings sont établis entre les entités des différentes ontologies. Un mapping est une correspondance entre une entité de la première ontologie et une ou plusieurs entités de la deuxième ontologie. L'appariement est utilisé lorsque l'on veut rendre des sources compatibles et cohérentes entre elles tout en les maintenant séparément [8].

Le problème de l'appariement d'ontologies, et bien avant le problème de l'appariement de schémas, a été largement étudié dans la littérature et un certain nombre d'approches et d'outils ont été proposés dans le domaine de la gestion de données et de connaissances [6].

L'appariement d'ontologies est généralement décrit comme l'application du prétendu opérateur de l'égalité [17]. L'entrée de l'opérateur est un certain nombre d'ontologies et la sortie est une spécification de correspondances entre les ontologies. Il y a beaucoup d'algorithmes différents qui implémentent l'égalité. Ces algorithmes peuvent être généralement classés le long de deux dimensions. D'une part il y a la distinction entre l'assortiment qui se base sur le schéma de l'ontologie et celui qui se base sur les instances de l'ontologie [17]. Un assortiment basé sur le schéma prend les différents aspects de concepts et de relations des ontologies et emploie une certaine mesure de similarité pour déterminer la correspondance [13]. Un assortiment basé sur les instances prend les instances qui appartiennent aux concepts dans les différentes ontologies et compare ces dernières pour découvrir la similarité entre les concepts [14]. D'autre part, il y a la distinction entre l'assortiment au niveau élément et l'assortiment au niveau structure. Un appariement au niveau élément compare les propriétés particulières d'un concept ou d'une relation, telle le nom, et emploie ces dernières pour trouver des similarités [11]. Un appariement niveau structure compare les structures (par exemple, la hiérarchie de concept) des ontologies pour trouver des similarités [11], [7]. Pour une classification plus détaillée des techniques d'alignement nous nous référons [17]. Ces techniques peuvent également être combinées [7]. Par exemple, Ancre-Prompt [11], rapprochent des ontologies vues comme des graphes au sein desquels les noeuds sont des classes et les liens sont des propriétés. Le système prend en entrée un ensemble d'ancres qui sont des couples d'éléments liés. Un algorithme analyse les chemins dans le sous graphe délimité par les ancres et détermine quelles classes apparaissent fréquemment dans des positions similaires sur des chemins similaires. Ces classes correspondent ainsi vraisemblablement à des concepts similaires [11]. Maedche et Staab, dans [9], proposent d'exploiter les relations entre les concepts. Des travaux de Ehrig et Staab sont focalisés sur l'efficacité des algorithmes de génération de mappings [5]. Des heuristiques pour restreindre le nombre de mappings candidats sont utilisées et une ontologie permet de classer les mappings candidats selon qu'ils sont plus ou moins prometteurs. Dans le travail de Maedche et Staab, une mesure globale de similarité entre deux hiérarchies est calculée, consistant à comparer les éléments parents et fils de tous les éléments communs [9]. Cette méthode n'est adaptée que si les hiérarchies sont bien structurées et contiennent de nombreux éléments en commun. Enfin Euzenat et Valtchev dans [6] proposent une mesure de similarité dédiée aux ontologies décrites en OWL-Lite, permettant d'agréger différentes techniques de comparaison exploitant les constructeurs de OWL-Lite dans une mesure commune.

D'autres travaux exploitent les données associées aux ontologies et appliquent des techniques d'apprentissage automatique sur ces données [4]. Enfin, des travaux réalisés à l'université de Trento en Italie mettent l'accent sur les aspects sémantiques et proposent d'utiliser WordNet pour aider à mettre en correspondance sémantiquement des éléments d'une hiérarchie de classification [3]. Notre approche se différencie des travaux susmentionnés par le mécanisme itératif de la mise en

correspondance entre termes (concepts et relations) présents dans les ontologies d'entrée et par la composition de multiples mesures de similarité. Cette composition aide à raffiner les résultats obtenus au préalable. Notre étude traite d'une part, l'héterogenieté syntaxique en définissant des moyens pour représenter ces ontologies dans un seul langage, langage pivot, le standard OWL, et d'autre part, l'héterogenieté sémantique qui se base sur l'interrogation d'une ressource linguistique, telle le WordNet, tout en l'accompagnant d'un degré de similarité appelé le degré de tolérance qui indique la plausibilité de cet appariement. Notre étude définit donc techniquement le processus d'appariement de plusieurs ontologies et de son résultat : alignement.

3 Les caractéristiques liées à notre stratégie d'appariement

L'appariement d'ontologies pose un nouvel ensemble de problèmes pour traiter et exiger également de nouvelles méthodes et de techniques qui n'ont pas été fournies par les outils d'appariement de schémas [17]. Les caractéristiques particulières à adresser par notre appariement d'ontologies sont:

La *complexité sémantique* : le nombre de constructeurs, de types d'éléments, de contraintes, et de liens sémantiques à être appariés est plus élevé dans l'appariement d'ontologies que dans l'appariement de schémas. Ceci signifie que la complexité de notre algorithme adopté dans l'appariement d'ontologies est également plus élevée. En fait, les différents types de constructeurs présentent également un certain nombre de nouveaux types d'hétérogénéités pour les traiter entre les sources de données.

Le *rôle de la sémantique*: les ontologies testées sont caractérisées par le fait que la sémantique de leurs constructeurs est explicitement formalisée en termes d'interprétation formelle des termes d'ontologies. Notre processus d'appariement doit donc traiter les contraintes et les informations fournies par cette interprétation.

Le *rôle des instance*s: parfois dans les ontologies, on ne détermine pas clairement la différence entre le schéma et l'instance et entre les méta-données et les données. La raison de ça est double : d'un côté, quelques modèles d'ontologies permettent au concepteur d'employer une classe comme une instance dans la description d'ontologies, comme dans RDFS ou dans le OWL Full. De l'autre côté, les différents concepteurs dans différentes ontologies peuvent choisir de décrire le même objet réel au moyen d'une classe ou au moyen d'une instance de la classe. D'un point de vue formel, les classes et les instances sont différentes (comme dans les logiques de description [18]) même si elles se réfèrent au même objet réel. Cependant, il est important pour nous de capturer la similarité entre deux descriptions différentes du même objet réel afin de fournir un résultat plus précis à la fin du processus d'appariement.

4 Notre processus d'appariement

Notre processus d'appariement d'ontologies est récapitulé en trois étapes principales:

La *première étape* est l'acquisition des ontologies à apparier. Le problème ici est de traiter la représentation et le modèle d'ontologies. Puis, les ontologies sont représentées au moyen d'un modèle interne pour les buts d'appariements.
La *deuxième étape* est indiquée par l'analyse des ontologies et par l'exécution des procédures d'alignements. Cette étape diffère des travaux précités par la composition en mesures de similarité [15]. Plusieurs techniques d'appariements (terminologique, linguistique et sémantique) sont adoptées par nos procédures d'appariements. Pour cette raison, cette étape est souvent réitérée plusieurs fois afin de raffiner les résultats obtenus en exécutions précédentes.
Dans la *troisième étape*, les mappings parmi des éléments d'ontologies sont déterminés. Ici nous pouvons avoir de différentes tâches selon le type de processus d'appariement qui a été effectué. Dans ce cas, les différents résultats de similarité sont analysés dans une valeur compréhensive de similarité pour les éléments d'ontologies et les résultats qui ne sont pas considérés comme appropriés sont écartés. Et ça pour le contrôle de la consistance (l'uniformité) des mappings. En conclusion, des ensembles de mappings sont déterminés entre les éléments d'ontologies en entrée.
L'appariement automatique, réalisé dans un temps fini d'exécution, est considéré comme une bonne réponse, en respectant les limites prédéfinies de confidentialité. Dans ce cas, les résultats d'alignements réalisés ne demandent pas à être stockés, parce que des alignements sont réalisés à chaque fois que c'est nécessaire. Leurs résultats peuvent être rejetés et, donc, non persistants.

5 Expérimentations

Bien que notre objectif soit d'apparier des ontologies de domaines d'application spécifiques, notre étude de cas a été réalisée avec l'intention d'évaluer son comportement devant un traitement d'ontologies avec un grand nombre de termes et de structures taxonomiques bien différentes. Le choix des ontologies utilisées n'implique pas un intérêt spécifique. La priorité a été donnée à la disposition des ontologies exprimées en OWL, d'institutions et de contexte connu dans des domaines d'applications spécifiques, *Ontologies Web*. Le Choix d'une ontologie d'un domaine d'application spécifique, demande encore le choix d'une autre ontologie, de domaine complémentaire à ce de la première, avec des classes « concepts/Relations » équivalentes à être appariées entre elles. Les ontologies choisies ne devraient pas être aussi créées par les mêmes experts d'ontologies (multi-points de vues).

Ces ontologies sont de la compagne I3CON[1]. I3CON, le premier effort dans la communauté de la recherche de l'intégration et de l'interprétation de l'ontologie et du schéma, est pour fournir des outils et un framework d'évaluation systématique de l'intégration des ontologies et des schémas. I3CON fournit 10 paires d'ontologies accompagnées de leur fichier de référence qui contient les correspondances déterminées manuellement entre deux ontologies.

1 Information Interpretation and Integration Conference; http://www.atl.lmco.com/projects/ontology/i3con.html.

5.1 La similarité des noms

En OWL, des ressources sont identifiées par des URIs. Par exemple, le concept « Article » est référencée en OWL par l'URI http://www.inria.fr/acacia/exemple#Article *<rdfs:Class rdf:about=" http://www.inria.fr/acacia/ exemple#Article " />*. L'URI se compose de deux parties: l'espace de noms (namespace) et le nom local. Dans l'exemple ci-dessus, l'espace de noms de l'URI est « *http://www.inria.fr/acacia/exemple* », et le nom local est « *Article* ». Si deux ressources ont une même URI, elles sont exactement la même ressource. Alors, si deux classes de deux ontologies sont référencées par une même URI, elles sont parfaitement le même concept et donc elles sont similaires (la valeur de similarité entre elles est de 1). Cependant, l'appariement d'ontologies traite normalement différentes ontologies ayant différents espaces de noms, il est intéressant donc de ne comparer que les noms locaux des classes, appelé désormais pour simplifier les noms de classe.

5.2 Des mesures pour l'évaluation des résultats

Pour évaluer la précision, l'efficacité et la performance de notre algorithme d'appariement, nous employons les mesures de *précision*, de *rappel* et de *f-mesure*. Ce sont des mesures largement utilisées dans le domaine de recherche d'information qui ont été appliquées par la suite dans le domaine d'alignement d'ontologies pour permettre une analyse fine des performances de système.

Le *rappel* est la proportion de correspondances correctes renvoyées par l'algorithme parmi toutes celles qui sont correctes (en incluant aussi des correspondances correctes que l'algorithme n'a pas détectées). Le rappel mesure l'efficacité d'un algorithme. Plus la valeur de rappel est élevée, plus le résultat de l'algorithme couvre toutes les correspondances correctes.

La *précision* est la proportion des correspondances correctes parmi l'ensemble de celles renvoyées par l'algorithme (ce sont les correspondances dans la liste des quadruplet). Cette mesure reflète la précision d'un algorithme. Plus la valeur de précision est élevée, plus le bruit dans le résultat de l'algorithme est réduit, et donc plus la qualité de résultat est imposante.

Enfin, la *f-mesure* est un compromis entre le rappel et la précision. Elle permet de comparer les performances des algorithmes par une seule mesure. La f-mesure est définie par

$$\text{f-mesure} = (2*\text{rappel}*\text{précision}) / (\text{rappel}+\text{precision}) \qquad (1)$$

Nous utilisons aussi la mesure globale (overall measure) définie dans [19]. Cette mesure, appelée l'exactitude, correspond à l'effort exigé pour corriger le résultat renvoyé par l'algorithme afin d'obtenir le résultat correct. La mesure globale est toujours inférieure à la f-mesure. Elle n'a un sens que dans le cas où la précision

n'est pas inférieure à 0,5, c'est-à-dire si au moins la moitié des correspondances renvoyées par l'algorithme sont correctes. En effet, si plus de la moitié des correspondances sont erronées, il faudrait à l'utilisateur plus d'effort pour enlever les correspondances incorrectes et d'ajouter les correspondances correctes mais absentes, que pour mettre en correspondance manuellement les deux ontologies à partir de zéro.

$$\text{Overall} = \text{rappel}*(2 - 1/\text{précision}) \quad (2)$$

Nous notons :

- *F* – le nombre des correspondances renvoyées par l'algorithme
- *T* – le nombre des correspondances correctes déterminées manuellement par des experts (celles dans le fichier de référence)
- *C* – le nombre des correspondances correctes trouvées (appelé aussi vrais positifs)
- Ainsi, *précision* = *C / F* ; *rappel* = *C / T ;*

5.3 Méthodologie d'évaluation

Nous avons testé notre stratégie proposée avec les paires d'ontologies de test de la campagne: I3CON. Pour chaque paire d'ontologie, notre outil d'appariement appelé OntAlign est exécuté 6 fois avec 6 seuils différents de similarité. Le seuil est utilisé dans la stratégie pour déterminer des correspondances considérées comme correctes. La valeur de similarité totale de deux entités appartenant aux ontologies, calculée par des techniques terminologique et sémantique, est comparée avec le seuil. Deux entités (concepts ou relations) sont considérées comme similaires si leur valeur de similarité dépasse ce seuil. Le résultat final de l'algorithme (la liste des correspondances entre deux ontologies) dépend donc du seuil choisi. Plus le seuil de similarité est élevé, plus la qualité du résultat est élevée (la *précision* augmente, le nombre des correspondances incorrectes trouvées diminue) mais plus des correspondances correctes sont considérées comme incorrectes. En continuant à augmenter le seuil de similarité, les valeurs de *f-mesure* et d'*overall* diminuent car le nombre des correspondances correctes mais pas trouvées devient très important. La raison est qu'il y a des entités qui représentent des concepts similaires mais leurs définitions dans des ontologies sont un peu différentes, donc leur valeur de similarité, calculée en se basant sur leurs définitions, n'atteint pas une valeur élevé (le maximum est 1.0), et donc ne dépasse pas le seuil.

Les correspondances trouvées par notre algorithme sont comparées avec les correspondances correctes dans le fichier de référence (fourni aussi par la campagne de test I3CON), pour calculer le nombre de correspondances correctes et incorrectes trouvées par l'algorithme. Les valeurs des mesures d'évaluation (précision, rappel, f-mesure, overall) sont calculées pour chaque cas et affichées dans un tableau.

♦ ***Evaluation de paires d'ontologies dans la campagne I3CON***

Les Tableaux tab.1 et tab.3 résument des informations concernant les paires d'ontologies de test, avec le nombre de concepts, de relations, de propriétés, d'instances dans les deux ontologies, la ressource externe interrogée et le nombre de correspondances correctes T dans le fichier de référence fourni.

♦ *La paire « Animals » (*O_A *:animalsA.owl -* O_B *:animalsB.owl)*

Tabl. 1. Informations concernant la paire d'ontologies « Animals »

O_A – O_B	**C**oncepts	**R**elations	**P**ropriétés	**I**nstances	**T**	**R**ess. **E**xterne
Manuel.	13 – 13	15 – 14	-	11 - 0	24	Expert
OntAlign	9 - 9	12 – 11	3 - 3	11 - 0	-	WordNet

Les deux ontologies de cette paire sont très proches aussi bien au niveau structurel qu'au niveau linguistique. La première a 11 instances tandis que la deuxième n'en a aucune, mais cela n'influence pas la performance de notre algorithme OntAlign. C'est ces éléments qui donnent à l'ontologie toute sa puissance en matière d'expressivité et permettent de réaliser des inférences.

Les deux ontologies, il est clair, modélisent le même domaine ; elles présentent des similarités évidentes. Cependant, elles présentent aussi des différences.

A titre d'exemples, nous remarquons que dans la première ontologie O_A le concept d'*hermaphrodite* n'a pas été évoqué, ce qui a été fait au niveau de la deuxième O_B. *hasMother*, *hasMom*, *hasFemalParent* sont utilisés comme équivalents dans O_A, alors que dans O_B, on s'est contenté du terme *hasMother*. Dans la première ontologie, on précise que *HumanBeing* veut dire la même chose que *Person*, tandis que dans la deuxième on ne le souligne pas. Pour décrire la même chose, les deux ontologies utilisent des termes différents : *TwoLeggedThing*, *TwoLeggedPerson* dans la première et *BipedalThing*, *BipedalPerson* dans la seconde. Ces différences suffisent pour empêcher une exploitation effectivement efficace des deux ontologies de manière conjointe. Il faudra donc les apparier.

La différence entre ces deux ontologies est la définition des concepts et des propriétés équivalentes. Par exemple, la première ontologie définit *hasFemaleParent* comme la propriété équivalente de la propriété *hasMother*, et *HumanBeing* comme la classe équivalente de la classe *Person*. L'application OntAlign exécute un post-traitement permettant de détecter des entités équivalentes. Cela permet de trouver toutes les correspondances de ce type.

Avec le seuil de similarité de 0,82 pour OntAlign, le *f-mesure=1.000* et toutes les correspondances correctes sont bien trouvées (1.000, tabl. 2). Si nous augmentons ces seuils, le nombre des correspondances correctes mais non-détectées est aussi augmenté, et donc, la valeur de *f-mesure* diminue.

Tabl. 2. Résultat de OntAlign sur la paire d'ontologies « Animals »

Seuil	F	T	C	P=C/F	Rl=C/T	f-mesure	Overall
0.70	32	24	32	1.000	1.333	1.142	1.333
0.80	26	24	26	1.000	1.083	1.039	1.083
0.82	24	24	24	1.000	1.000	1.000	1.000
0.85	18	24	18	1.000	0.750	0.545	0.750
0.90	18	24	18	1.000	0.750	0.545	0.750
1.00	6	24	6	1.000	0.250	0.400	0.250

♦ *La paire « Computer Networks »*

Comme la paire d'ontologies « *Animals* », les deux ontologies suivantes ont une structure similaire et les termes nommant les entités le sont aussi. Par contre, la structure ou la taxonomie d'ontologie est assez complexe, avec la profondeur maximale de 4.

Tabl. 3. Informations concernant la paire d'ontologies « Computer Networks »

$O_A - O_B$	**C**oncepts	**R**elations	**P**ropriétés	**I**nstances	**T**	**R**ess. **E**xterne
Manuel.	27 – 27	5 - 6	-	0 - 0	30	Expert
OntAlign	27 - 27	5 - 6	0 -0	0 - 2	-	WordNet

L'algorithme OntAlign se base principalement sur la linguistique, il détecte incorrectement donc que la classe *FTPServer* de la première ontologie, qui est sous-classe de *ServerSoftware*, correspond à la classe *Server* de la deuxième ontologie, qui est sous-classe de *Computer*, car la valeur de similarité totale calculée est de 0,80 par *Wu & Palmer[12]* et de 0,90 par la mesure terminologique (basé string). Tandis que la valeur de similarité entre *FTPServer* et sa vraie classe correspondante dans la deuxième ontologie, *FTP*, n'est que 0,50 par *Wu & Palmer[12]* et 0,825 par la mesure terminologique *StringDistance*. L'algorithme OntAlign prend en compte la structure d'ontologie et trouve donc correctement la classe correspondante de la classe *FTPServer*, qui est la classe *FTP* (sous-classe de *ServerSoftware) à 0,94 par StringDistance*.

La valeur de f-mesure est maximale entre (1.016, 0.775 et 0.666) avec les seuils de similarité de 0.80, 0.825, 0.925 et 0.9 dans OntAlign, respectivement (tabl. 4). La valeur de *f-mesure* pour OntAlign est de $1.016 \approx 1.00$, elle est la meilleure valeur pour ce test. Si nous diminuons les seuils, le nombre de correspondances incorrectes augmente et diminue dans le cas contraire.

Pour les deux paires d'ontologies, les résultats renvoyés par f-mesure reflètent la précision de l'algorithme. Les résultats sont donc bons.

Suite à ces résultats, et à d'autres expérimentations et évaluations antérieures sur l'ensemble de paires d'ontologies de test fournies dans le cadre de la campagne de tests I3CON, tous les résultats obtenus par OntAlign sont encourageants.

Cependant, les mesures de performance de notre stratégie via les mesures telles que la précision, le *rappel*, la *f-mesure*, l'*over*all montrent que les résultats renvoyés par notre OntAlign proposé sont de qualité.

Tabl. 4. Resultat de OntAlign sur la paire d'ontologies « Computer Networks »

Seuil	F	T	C	P=C/F	R=C/T	F mesure	Overall
0.70	76	30	76	1.000	2.533	1.433	2.533
0.80	31	30	31	1.000	1.033	1.016	1.033
0.852	19	30	19	1.000	0.633	0.775	0.633
0.90	15	30	15	1.000	0.500	0.666	0.500
0.925	15	30	15	1.000	0.500	0.666	0.500
0.951	11	30	11	1.000	0.366	0.535	0.366

6 Conclusion

Notre approche est donc une des approches attaquant le problème d'appariement d'ontologies et spécialement pour les ontologies représentées en OWL. Elle emploie les mesures de similarité basant sur la comparaison des chaînes de caractères inspirée du domaine de la recherche d'information et l'intégration de WordNet pour produire la similarité linguistique lexicale entre deux entités. Elle exploite aussi des heuristiques pour le calcul de la similarité de deux entités. Ainsi, notre algorithme est une application et une intégration des mesures de similarité de base [1], [12], [15] en exploitant des caractéristiques du formalisme OWL et diffère des autres approches dans le domaine de l'appariement des schémas ou des ontologies.

OntAlign procède une étape d'une comparaison uniforme des entités et seulement les termes de la même catégorie sont comparés. Ensuite il les compare en employant des mesures basées sur la similarité terminologique (similarité des chaînes de caractères) ou sémantique (en interrogeant WordNet). Mais OntAlign applique les mesures de préfixe, de suffixe et StringDistance pour l'axe terminologique et une composition d'autres mesures telles que Wu & Palmer [12], Resnik, Rada et Leacock [15] pour l'axe sémantique. OntAlign retourne un résultat meilleur dans la plupart des cas et dans un temps de calcul plus réduit. OntAlign n'exploite pas la similarité des instances pour déduire la similarité des classes ou des relations correspondantes à ces instances comme les approches GLUE ou NOM/QOM [17] le font, bien que la similarité de deux instances est calculée analogiquement comme dans le calcul de similarité de deux classes ou deux relations (grâce à leurs nom, leurs étiquettes et leurs commentaires). Au lieu de cela, OntAlign déduit la similarité entre deux entités grâce aux valeurs de similarités agrégées qui dépassent un seuil prédéfini.

Cependant des perspectives de notre travail s'orientent donc vers :

- Etendre le processus d'appariement aux restrictions, fonctions et axiomes,
- Explorer les mesures de similarité basées sur le contenu informationnel,

- Tester l'approche sur d'autres domaines d'application, tel l'environnement dynamique peer-to-peer,
- Elargir l'approche pour des ontologies décrites dans de multiples langages tels que DAML, OIL, DAML+OIL, RDFs…,
- Traiter le processus d'appariement en intégrant la dynamicité des systèmes multi-agents (SMA),
- Interroger une autre ressource linguistique comme le Euro-Wordnet.

Réferences

1. J. Madhavan, P.A. Bernstein, E. Rahm, "Generic Schema Matching with Cupid", VLDB 2001.
2. Deborah L.McGuinness (Standford University) Frank Van Harmelen (Vrije Universiteit , Amsterdam) « OWL Web Ontology Language Overview » W3C « http://www.w3.org/TR/2004/REC-owl-features-20040210/ », 2004.
3. Deliverable : Stefano Spaccapietra coordinator (Ecole Polytechnique Fédérale de Lausanne) "D2.1.3.1 Report on Modularization of Ontologies", 2005
4. A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, A. Halevy, "Learning to match ontologies on the Semantic Web", The VLDB Journal, 12:303-319, 2003.
5. M. Ehrig, S. Staab, "QOM – Quick Ontology mapping", in proc. of the 3rd International Semantic Web Conference (ISWC2004), November 7 to 11, Hiroshima, Japan, 2004.
6. J. Euzenat, P. Valtchev, "An integrative proximity measure for ontology alignment", in Proc. ISWC 2003.
7. F. Giunchiglia, P. Shvaiko, M. Yatskevich, "S-Match: an algorithm and and implementation of semantic matching", In proceedings of the European Semantic Web Symposium, LNCS 3053, pp. 61-75, 2004.
8. Y. Kalfoglou, M. Schorlemmer, "Ontology mapping: the state of the art", in Knowledge Engineering Review, Vol. 18, pp. 1-31, 2003.
9. A. Maedche, S. Staab, "Measuring similarity between Ontologies", in proc. of the European Conference on Knowledge Acquisition and management – EKAW-2002, Madrid, Spain, October 1-4, LNCS/LNAI 2473, Springer, 2002, pp. 251-263, 2002.
10. G.A. Miller, "Word Net: A lexical Database for English", Communications of the ACM, Vol. 38, N°11, p. 39-45, November, 1995.
11. N. Noy, M. Musen, "Anchor-PROMPT: Using Non-Local Context for Semantic Matching", IJCAI 2001.
12. Wu Z. & Palmer M. Verb Semantics and Lexical Selection, Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics, pages 133-138, 1994.
13. Noy, N. F. & Musen, M. A., PROMPT: Algorithm and tool for au-tomated ontology merging and alignment, in 'Proc. 17th Natl. Conf. On Artificial Intelligence (AAAI2000)', Austin, Texas, USA 2000b.
14. Doan, A., Madhaven, J., Domingos, P. & Halevy, A., Ontology matching:A machine learning approach, in S. Staab & R. Studer, eds, 'Handbook on Ontologies in Information Systems', Springer-Verlag, pp. 397–416,2004.
15. Zargayouna H. , Contexte et sémantique pour une indexation de documents semi-structurés. ACM COnférence en Recherche Information et Applications, CORIA'2004.
16. Kefi, H.,B. Safar, C. Reynaud, Alignement de taxonomies pour l'interrogation desources d'information hétérogènes. RFIA, Tours, 2006.
17. Jerome Euzenat and Pavel Shvaiko. Ontology matching. Springer-Verlag, Heidelberg (DE), 2007.

18. M. Sabou, M. d'Aquin, E. Motta. Using the semantic web as background knowledge for ontology mapping. International Workshop on Ontology Matching, collocated with the 5th International Semantic Web Conference ISWC-2006, November 5, 2006, Athens, Georgia, USA.
19. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: a versatile graph matching algorithm. In Proceedings of the International Conference on Data Engineering (ICDE), pages 117– 128, 2002.

Détection des cas de débordement flottant avec une recherche locale

Mohamed SAYAH[1], Yahia LEBBAH[1]

{sayahxfr,ylebbah}@yahoo.fr
[1] Université d'Oran, Dép. Informatique, BP-1524, El-M'Naouer, Oran, Algeria

Résumé. Dans ce papier, nous proposons une recherche locale pour la résolution de contraintes définies sur les nombres flottants. Ces contraintes sont issues de la modélisation de la problématique de la détection des cas de débordement flottant dans le calcul scientifique. L'ensemble des nombres flottants est un ensemble discret avec une grande cardinalité. Nous détaillons comment modéliser la problématique sous forme d'un système de contraintes dont la résolution exacte nécessite un calcul coûteux qui motive amplement le recours à la recherche locale. La fonction de voisinage et la fonction d'évaluation au sein de la recherche locale sont proposées en exploitant la structure particulière des contraintes et des flottants.

1 Introduction

L'assertion "*L'ordinateur calcule faux et déborde*" reflète une réalité que beaucoup de scientifiques essayent de considérer avec beaucoup de difficultés. La raison évidente à cette réalité est le fait suivant : les algorithmes numériques sont conçus et prouvés sur les nombres réels, alors que leur mise en oeuvre est faite sur les nombres flottants qui violent presque toutes les propriétés mathématiques des nombres réels. En pratique, l'ordinateur ne fait avec l'arithmétique à virgule flottante qu'une approximation de l'arithmétique exacte. Cette approximation se traduit par des erreurs soudaines de débordement au niveau des calculs.

Par ailleurs et de nos jours, les applications de calcul scientifique utilisent les nombres à virgule flottante, qui sont la représentation en machine des nombres réels. La première apparition du concept de nombre flottant dans les processeurs a conduit à une multitude de formats de représentation flottante suivant le type du processeur utilisé [4, 7, 8]. Par la suite, la création de la norme IEEE-754 [1] a permis d'uniformiser le calcul flottant et de donner la possibilité aux utilisateurs de développer des programmes dont les résultats sont les mêmes quelque soit la machine utilisée.

Toutefois, l'utilisation fréquente des programmes informatiques au profit de l'industrie nécessite une analyse approfondie de ces derniers (programmes) afin de les sécuriser de tout aléa possible. En fait, l'exploitation d'un programme informatique en milieu industriel, est conditionnée à priori par le "verdict positif" d'une pile de tests. Dans ce contexte, notre approche propose la génération automatique des cas de test qui donnent lieu à des débordements flottants.

Notre approche procède en trois étapes : (1) Transformation du programme en un système de contraintes mathématiques en exploitant la forme SSA [5]; (2) Transformation du critère de débordement en contraintes ajoutées aux contraintes générées dans l'étape précédente; (3) Résolution du système de contraintes produit avec une recherche locale dédiée, en exploitant la bibliothèque MPFR. Le point clé dans cette démarche réside dans les fonctions de voisinage et d'évaluation de la recherche locale qui tirent profit des structures particulières des contraintes et des flottants. L'utilisation de la bibliothèque MPFR nous permet de faire des calculs avec une grande précision dont le résultat contiendrait les cas de débordement.

Le plan du papier se présente comme suit. La section 2 introduit un exemple illustratif qui montre le déroulement de notre démarche. La transformation d'un programme informatique en un système de contraintes est détaillée dans la section 3. La structure des nombres flottants est explicitée dans la section 4. Dans la section 5, nous introduisons notre algorithme par recherche locale pour résoudre les contraintes sur les flottants. Nous terminons ce papier avec une section expérimentale et une conclusion.

2 Exemple de motivation

Soit le programme *SQUARE* qui calcule la racine carrée R de deux nombres réels positifs X et Y. Dans le cas où X est égal à Y, R prend la valeur de X.

double SQUARE (**double** x, **double** y){
if $(x == y)$ **then**
  $r = x$;
else
  $r = sqrt(x * y)$;
end if
return r;
}

Les nombres réels X, Y et R sont transcrits respectivement vers les variables informatiques x, y et r. Nous notons aussi que plusieurs cas de débordements peuvent surgir lors de l'exécution de ce programme. Pour détecter ces cas, notre démarche consiste, dans un premier temps, à transformer le programme en un ensemble de contraintes comme explicité dans la section 3 ci dessous. La transformation du programme *SQUARE* se fait dans une forme intermédiaire dite *SSA : Static Single Assignement*. Cette forme préserve la sémantique du programme informatique en entrée et facilite sa réécriture sous forme d'un ensemble de contraintes sur les flottants. Le résultat de la transformation en forme *SSA* du programme *SQUARE* est donné comme suit :

Etapes	Instructions de SQUARE	Forme SSA
1.	Double x, y, r;	Double x, y, r;
2.	$if(x == y)$	$if(x_0 == y_0)$
3.	$r := x$;	$r_0 := x_0$;
4.	$else$	$else$
5.	$r := sqrt(x * y)$;	$r_1 := sqrt(x_0 * y_0)$;
6.	Fonction ϕ	$r_2 := \phi(r_0, r_1)$;

Par la suite, cette forme SSA est traduite en un systèmes de contraintes, donné comme suit :

Forme SSA de SQUARE	Contraintes associées
Double x, y, r;	$x_0, y_0, r_0 \in [-1.8 * 10^{308}, 1.8 * 10^{308}]$
$if(x_0 == y_0)$	$x_0, y_0, r_0 \notin [-4,94 * 10^{-324}, 4,94 * 10^{-324}]$
$r_0 := x_0$;	$(x_0 = y_0) \wedge (r_0 = x_0)$
$else$	ou
	$x_0, y_0, r_1 \in [-1.8 * 10^{308}, 1.8 * 10^{308}]$
$r_1 := sqrt(x_0 * y_0)$;	$x_0, y_0, r_1 \notin [-4,94 * 10^{-324}, 4,94 * 10^{-324}]$
$r_2 := \phi(r_0, r_1)$;	$\neg(x_0 = y_0) \wedge (r_1 = sqrt(x_0 * y_0))$

Finalement et en dernière étape, les contraintes exprimant le débordement sont ajoutées au système de contraintes courant. Chaque situation de débordement (voir section 4.1) est modélisée par une ou plusieurs contraintes. Par exemple, un débordement de l'expression $(x*y)$ dans le cas où la garde de la conditionnelle est fausse est exprimable avec la contrainte $(|(x * y)| > (1.8 * 10^{308}))$.

Le système final de contraintes S associé à cette situation de débordement est le suivant :

$$S^1 = \begin{cases} (x_0 \neq y_0) \\ (r_1 = sqrt(x_0 * y_0)) \\ (x_0 * y_0) > (1.8 * 10^{308}) \\ x_0, y_0, r_1 \in [-1.8 * 10^{308}, 1.8 * 10^{308}] \\ x_0, y_0, r_1 \notin [-4,94 * 10^{-324}, 4,94 * 10^{-324}] \end{cases}$$

Dans ce cas de figure, la configuration $D_1 = (\underbrace{1.0 * 10^{220}}_{x_0}, \underbrace{4.0 * 10^{180}}_{y_0}, \underbrace{NAN}_{r_1})$ est un cas de débordement[1].

3 Transformation d'un programme en contraintes

Cette transformation [5, 3] consiste, premièrement, à traduire le programme en entrée en une forme intermédiaire dite *SSA* ou *Static Single Assignement*. Puis, cette forme SSA du programme est réécrite en un système de contraintes.

3.1 Forme SSA d'un programme informatique

Considérons un if_while langage comportant en plus des types de base *double* et *float*, les structures de contrôle suivantes : L'affectation "*variable := expression*", la conditionnelle "*if (condition) then* $I_1, \ldots, I_n$" , et la boucle "*while (condition) do* $I_1, \ldots, I_n$ endo".

Notons que dans le cas de l'affectation, toute variable affectée x provoque la création d'une nouvelle variable x_i avec $i = 1, \ldots, n$, comme suit :

[1] Toute solution du système S est un cas de débordement de la variable x dans le programme SQUARE.

Affectation	Forme SSA équivalente
$x := x + y;$	$x_1 := x_0 + y_0;$
$y := x - y;$	$y_1 := x_1 - y_0;$
$x := x - y;$	$x_2 := x_1 - y_1;$

Dans le cas de la conditionnelle, des fonctions spéciales ϕ sont ajoutées dans la forme SSA afin de regrouper les définitions de ces variables.

La conditionnelle [2]	Forme SSA
$if(x == y)$ then $r := x;$	$if(x_0 == y_0)$ then $r_0 := x_0;$
$else$	$else$
$r := sqrt(x * y);$	$r_1 := sqrt(x_0 * y_0);$
	$r_2 := \phi(r_0, r_1);$

Quant à l'instruction de *boucle*, des fonctions ϕ sont introduites avant la boucle et regroupent les variables définies à l'intérieur de la boucle avec celles qui précédent directement l'instruction de la boucle. Cette transformation se présente donc comme suit :

L'instruction de boucle	Forme SSA[3]
	$x_3 := \phi(x_1, x_2);\ r_3 := \phi(r_1, r_2)$
$while(x < y)$ do	$while(x_1 < y_0)$ do
$r := r + x;$	$r_2 := r_1 + x_0$
$x := x + 1;$	$x_2 := x_1 + 1;$
enddo	enddo

3.2 De la forme SSA aux contraintes

Une fois la forme SSA du programme C est générée, une autre phase de réécriture est engagée afin de déduire les contraintes modélisant la sémantique du programme en question. Pour ceci, toute déclaration d'une variable x d'un type T est traduite en une contrainte d'appartenance à un intervalle $x \in [T_{min}, T_{max}]$. L'instruction conditionnelle de la forme "**if** (condition) **then** I_1; ... ; I_n **else** $\acute{I}_1$; ...; $\acute{I}_n$ **fsi**" est transformée en un opérateur "**ite**" défini comme suit : "**ite**(c,C_1 $\wedge$ ... $\wedge$ C_n,$\acute{C}_1$ $\wedge$...$\wedge$ $\acute{C}_n$)". Cet opérateur est équivalent sémantiquement à "(c $\wedge C_1 \wedge ... \wedge C_1$) ou ($\neg$ c $\wedge$ $\acute{C}_1$ $\wedge$ ...$\wedge$ $\acute{C}_1$)". c est la contrainte primitive associée à la *condition* et C_i et $\acute{C}_i$ correspondent respectivement aux contraintes primitives de I_i et $\acute{I}_i$. Les instructions de boucle sont réécrites en opérateur [4]**w** suivant : "**w**(c,$\boldsymbol{v}_0$, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$,$C_1 \wedge ... \wedge C_n$)", où $C_1 \wedge ... \wedge C_n$ correspond à la forme SSA du corps de la boucle, $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2$ étant respectivement le vecteur des variables qui atteignent l'instruction *while*, le vecteur des variables définies dans le corps de la boucle et le vecteur des variables utilisées dans la boucle.

A la fin et une fois que le programme est transformé en un système de contraintes, des contraintes spécifiques aux débordements sont rajoutées à ce dernier système.

[2] Voir la conditionnelle de l'exemple illustratif de la section 2.

[3] On signale le caractère combinatoire dans la forme SSA de la boucle. Il est donc intéressant de procéder à une détection de débordement en fonction d'une profondeur donnée de la boucle.

[4] Le dépliage de la boucle while et le renommage des variables a pour conséquence la réduction de la complexité du graphe de flot de contrôle.

4 Structure des nombres flottants

La norme IEEE-754 [1] définit quatre formats de représentation des nombres à virgule flottante : simple précision, simple précision étendue, double précision et double précision étendue. Tout flottant x peut s'écrire $\mathbf{x} = \varepsilon.\mathbf{b^e}.\mathbf{m}$ avec $b \in \mathbb{N}$, $\epsilon \in \{-1, +1\}$, $e \in \mathbb{Z}$ et $m \in [0, b]$. b est la base , ε est le signe, e est l'exposant et m est la mantisse. Coder un réel sur ordinateur consiste à trouver le triplet $\{\epsilon, e, m\}$ qui soit le plus proche possible.

La norme **IEEE-754** utilise aussi la base 2, ainsi :

$$e = \sum_{i=0}^{p} b_i.2^i \quad et \quad m = \sum_{i=0}^{\infty} a_i.2^{-i} \quad avec\,(a_i, b_i) \in \{0, 1\}$$

Le codage de ϵ tient sur un bit qui est le bit du signe, et vaut 0 si $x > 0$ et 1 si $x < 0$. Le format définit la taille de la mantisse (pM), de l'exposant (pE), le valeur minimale et la valeur maximale de l'exposant (Emin et Emax) et la valeur du biais. Le biais consiste à ajouter une quantité fixe, qui dépend du format, et de la valeur de l'exposant. Ainsi, il est possible de travailler avec des exposants toujours positifs. Les valeurs de ces paramètres, en fonction du format, sont données par le tableau ci dessous :

Format	Taille	p_M	p_E	E_{max}	E_{min}	Biais
Simple	32	24	8	$+127$	-126	$+127$
Simple étendu	≥ 43	≥ 32	≥ 11	$\geq +1023$	≤ -1022	/
Double	64	53	11	$+1023$	-1022	$+1023$
Double étendu	≥ 79	≥ 64	≥ 15	$\geq +16383$	≤ -16382	/

Table 1. Représentations et formats des nombres à virgule flottante

4.1 Cas de débordements (Définitions)

Tout programme de calcul consiste en une suite fini d'évaluations des expressions. Dans le cas d'un calcul flottant qui déborde, une opération arithmétique peut conduire soit à une valeur trop petite (*UNDERFLOW*) ou une valeur trop grande (*OVERFLOW*) [9, 6]. Un résultat indéfini est stocké, dans la norme IEEE 754, avec une valeur particulière dite NAN (*Not A Number*).

Dans la norme IEEE 754, les cas d'erreurs sont traités comme des exceptions dans le calcul flottant. Quatre types d'exception sont à considérer : *Underflow*, *Overflow*, *Opération Invalide ou NAN*, et *Division par Zéro*. Nous avons donc respectivement :

– Un Underflow est généré lorsqu'une opération arithmétique produit un résultat trop petit pour être représenté comme un nombre flottant normalisé

(le premier chiffre de la mantisse est différent de zéro). La valeur renvoyée peut être zéro ou la valeur la plus petite représentable selon le type de la variable.
- Un Overflow est généré lorsqu'une opération arithmétique produit un résultat trop grand pour être représenté comme un nombre flottant normalisé (le premier chiffre de la mantisse est différent de zéro). La valeur renvoyée est alors l'infini (Inf), signée.
- Une opération invalide se produit lorsqu'au moins l'un ou plusieurs opérandes ne sont pas valides pour l'opération implémentée. La valeur renvoyée est une valeur particulière appelée Not A Number (NaN).
- Une division par zéro se produit lorsqu'un nombre non nul est divisé par un nombre nul. La valeur renvoyée est alors l'infini (1), signée.

4.2 Arithmétique flottante et débordements

Les propriétés de l'arithmétique machine ne sont pas celles des réels et toute évaluation d'une expression manipulant des flottants peut mener à des exceptions [2]. Ces dernières sont classées en cas de débordements généraux ou particuliers comme indiqués dans les tables (02) et (03) ci-dessous :

Opération arithmétique	Type d'opération	Résultat par défaut	Nature de l'exécution
$\beta * (-\beta)$	Multiplication	$-\infty$	Overflow
$\alpha_1 * \alpha_2$	Multiplication	0.0 (non normalisé)	Underflow
$\alpha_1 - \alpha_2$	Soustraction	∞	Overflow
$\beta_1 - \beta_2$	Soustraction	0.0 (non normalisé)	Underflow
α/β	Division	0.0 (non normalisé)	Underflow

Table 2. Cas de débordements généraux

Opération arithmétique	Type d'opération	Résultat par défaut	Nature de l'exécution
$\infty * (0.0)$	Multiplication	NAN	Invalide
∞/∞	Division	NAN	Invalide
$f/0.0$	Division	$+\infty$	Overflow
0.0/0.0	Division	∞	Overflow
$\infty/0.0$	Division	∞	Overflow
f/NAN	Division	NAN ($\forall f$)	Invalide
$\infty - \infty$	Soustraction	NAN	Invalide

Table 3. Cas de débordements particuliers

Remarque : Les variables α, β, f, et ∞ sont respectivement un flottant très petit, un flottant très grand, un flottant quelconque et le flottant représentant l'infini.

5 Résolution des équations par une recherche locale

Le principe d'une recherche locale consiste à démarrer d'une configuration initiale P_0 et d'essayer ensuite de l'améliorer, en cherchant une meilleure configuration dans le voisinage de celle qui est courante.

Local Search(IN P_0, OUT P)
% P_0 : la configuration initiale
% P : la configuration finale
$P := P_0$;
while $(P \neq Pi)$ **do**
  $Pi := P$;
  $P := meilleurVoisin(Pi)$;
end while

Le voisinage d'une configuration P_c correspond à des éléments adjacents à P_c dont chacun peut être atteint par un changement dans la configuration de P_c. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration ne pourrait être faite. Cette possibilité de comparer les configurations dans le voisinage de P_c est garantie par une fonction de coût $eval(P)$ introduite dans la section 5.1.

Dans notre approche, on rappelle que les solutions à chercher sont sur l'ensemble des nombres flottants $I\!F$. Cet ensemble qui est discret et grand[5], nécessite une fonction de voisinage qui doit couvrir la totalité de l'espace $I\!F$. En plus, notre fonction de coût *eval*, qui consiste à évaluer [6] le nombre de contraintes violées, prend en compte la nature et la forme des contraintes générées, à partir de la forme SSA, afin d'augmenter la convergence vers un point solution. L'algorithme 1 ci-dessous montre notre implémentation de la recherche du meilleur voisin, le voisin ayant le minimum de contraintes violées. Quand le nombre de contraintes violées est nul, le point Pm représente le cas de débordement du programme analysé :

Algorithme 1:

meilleurVoisin(IN P_0, IN profondeur, IN niveau, IN delta, IN/OUT Pm)
% P_0 : la configuration initiale
% Pm : la meilleure configuration
% profondeur : épaisseur du voisinage
% delta : saut au niveau des voisinages
% niveau : paramètre d'exploitation ou d'intensification
% N est le rang de la configuration P
if profondeur == 0 **then**
  return;
end if
for (i = 1 .. N) **do**
  $P_1 \leftarrow P_0$;
  $P_2 \leftarrow P_0$;

[5] La taille est de l'ordre de $2 * (1.8 * 10^{308})$ pour une variable de type *double*.
[6] Une équation ou contrainte non violée a un coût égal à zéro (0).

$P_1(i) \leftarrow NextVoisin(P_1(i), Niveau, delta, -\infty)$; % voisins gauches et descendants
if $(eval(P_1) < eval(Pm))$ **then**
$Pm \leftarrow P_1$;
end if
meilleurVoisin(P_1, profondeur-1, niveau, delta, Pm) ;
$P_2(i) \leftarrow NextVoisin(P_2(i), Niveau, delta, +\infty)$; % voisins droits et descendants
if $(eval(P_2) < eval(Pm))$ **then**
$Pm \leftarrow P_2$;
end if
meilleurVoisin(P_2, profondeur-1, niveau, delta, Pm) ;
end for

avec NextVoisin ($P(i)$, $Niveau$, $delta$, Dir) détermine le flottant voisin se trouvant à une distance $delta$ de $P(i)$ avec un déplacement de $Niveau$ flottants dans la direction Dir (voir *Algorithme* 2 et figure 1).

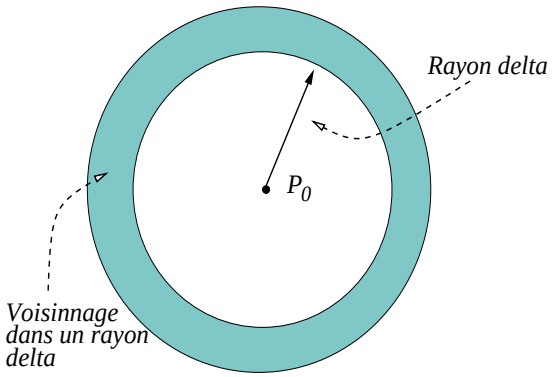


Fig. 1. Description du voisinage d'un point courant P_0.

Par exemple, dans le langage C, il suffit de faire, en premier temps, un déplacement de $delta$ par rapport à $P(i)$ ($P(i) = P(i) + delta$) et d'appliquer ensuite $Niveau$ fois, dans la direction Dir, la fonction système NEXT_AFTER.

Algorithme 2:

NextVoisin (IN/OUT $P(i)$, IN $niveau$, IN $delta$, IN Dir)
% P est configutation quelconque
% $P(i)$ est la i^ieme de la configuration P
% niveau : paramètre d'exploitation ou d'intensification
% Dir paramètre spécifiant la direction de l'exploration (et/ou) l'exploitation
if $Dir ==$ '+' **then**

$P(i)$ = $P(i)$ + delta ;
end if
if Dir == '-' **then**
$P(i)$ = $P(i)$ - delta ;
end if
for (i = 1 à $niveau$) **do**
if Dir == '+' **then**
$P(i)$ = le premier flottant voisin de $P(i)$ dans la direction $+\infty$;
else
$P(i)$ = le premier flottant voisin de $P(i)$ dans la direction $-\infty$;
end if
end for

Cet algorithme fait un parcours en profondeur sur les deux sens de chacune des variables, en faisant des sauts de longueur *delta*. Le paramètre *delta* permet donc de fixer l'éloignement de la configuration courante, alors que le paramètre $profondeur$ influence le nombre de voisins visités qui est de l'ordre de $(2n)^{profondeur}$.

5.1 Fonction d'évaluation

Nous avons adopté une fonction d'évaluation *eval* qui estime le coût des contraintes violées par un point donné P. Ce coût est calculé suivant la nature et la forme des contraintes générées à partir de la forme SSA, comme suit :

$$eval(P) = \begin{cases} 0 & si \ \forall \ \ C_{i,(i=1,\ldots,m)} \ \ PCviol[i] = 0. \\ \sum_{i=1}^{m} PCviol[i]. & sinon \end{cases}$$

On note aussi que $PCviol[i]$, le coût de violation d'une contrainte C par un point P, est calculé en fonction du type de la contrainte C. L'algorithme 2 suivant présente comment est évalué le coût global d'une violation du système d'équations par une configuration P :

Algorithme 3:

eval(IN P, OUT $Pviol$)
% P : configuration initiale
% $Pviol$: coût de violation des contraintes
% M étant le nombre de contraintes
if P est une solution du système d'équations **then**
  $Pviol \leftarrow 0$;
else
  $Pviol \leftarrow 0$; % $Pviol$ est le Coût global de violation
  for (i= 0 .. M) **do**
    Calculer le coût de violation de P pour la contrainte C_i noté $PCviol[i]$;
    $Pviol \leftarrow PCviol[i] + Pviol$;
  end for
end if

Le coût de violation $Pviol$ (voir Algorithme 3) du système de contraintes $C_{i,(i=1,\ldots,m)}$ par une configuration ou point P, est la somme des violations $PCviol$ calculées pour chaque type de contrainte C_i. Ces contraintes sont classées en trois types, à savoir: $C^{=}$: $(e_1 = e_2)$ ou contrainte d'égalité, $C^{<}$: $(e_1 < e_2)$ ou contrainte d'inégalité strictement inférieure et $C^{>}$: $(e_1 > e_2)$ ou contrainte d'inégalité strictement supérieure.

Nous supposons la disponibilité de la fonction $val(P, e)$ qui calcule la valeur d'une expression donnée [7]e à une configuration donnée $P = (v_1, \ldots, v_n)$. La fonction $h(P, C^t)$ permet d'évaluer le coût de violation $PCviol$ d'un type de contrainte donnée C^t par une configuration P. La fonction h est spécifiée comme suit :

$$h(P, C^t) = \begin{cases} |val(P, e_2) - val(P, e_1)| & si \quad (C^t = C^{=}). \\ |val(P, e_2) - val(P, e_1)| & si \quad (C^t = C^{<}) \wedge (val(P, e_2) - val(P, e_1)) < 0. \\ |val(P, e_2) - val(P, e_1)| & si \quad (C^t = C^{>}) \wedge (val(P, e_2) - val(P, e_1)) > 0. \\ 0 & sinon \end{cases}$$

6 Résultats expérimentaux préliminaires

Pour pouvoir gérer, manipuler et détecter numériquement les cas de débordement, nous avons adopté une bibliothèque fournissant une précision réglable dans laquelle il ne peut y avoir de débordement avec les valeurs flottantes de la machine. Cette bibliothèque est $MPFR$. Considérons le programme cité dans [6] où il est question de calculer la racine cubique d'un nombre. Dans notre

[7] Les expressions sont exprimées en fonction des variables déduites de la forme SSA et leurs évaluations, pour une configuration P, exploitent les valeurs flottantes $vi, (i = 1 \ldots n)$.

cas d'expérimentation, nous nous intéresons particulièrement aux cas de débordement de la variable $DELTA$ avant d'atteindre un point de contrôle donné dans ce programme. Le point de contrôle choisi a été modélisé par le système d'équations S suivant :

$$S = \begin{cases} (3.0 * B - (A * A)/3.0) = Q \\ (((2.0 * A * A * A - 9.0 * A * B) - (27.0 * C))/27.0) = R \\ ((Q * Q * Q)/27.0 + (R * R)/4.0) = DELTA \\ A, B, Q, R, DELTA \in [-1, +1] \\ A, B, Q, R, DELTA \notin [-4,94 * 10^{-324}, 4,94 * 10^{-324}]^{9} \\ ^{8}\mathbf{ABS(DELTA)} \leq \mathbf{4,94 * 10^{-324}} \end{cases}$$

Nous notons que les variables $A, B, C, Q, R, et\ \ DELTA$ sont représentées, sur MPFR, en précision arbitraire. Notre recherche locale a trouvé 4 cas de débordement donnés ci-dessous :

A	B	C	Q	R	DELTA
2.3438153e-320	2.34381521e-320	-7.4210235e-321	2.3438152e-320	7.424639e-321	**Underflow**
1.7950801e-321	1.7950801e-321	-3.03393301e-322	1.79508e-321	2.9907e-322	**Underflow**
1.4505767362e-322	1.4505767362e-322	-2.411040352e-323	1.4505767362e-322	2.8063e-323	**Underflow**
7.7074241e-323	7.7074241e-323	-1.97626259e-324	7.7074241e-323	-1.9762626e-324	**Underflow**

Table 4. Quelques cas de débordement dans le système S

7 Conclusion

Nous avons introduit dans ce papier un algorithme de résolution par recherche locale pour appréhender des systèmes de contraintes définis sur les flottants. Ces systèmes sont issus de la génération automatique de cas de débordement dans un calcul scientifique. Nous avons mis en oeuvre ces algorithmes avec la bibliothèque MPFR qui nous a permis de détecter d'une façon numérique les cas de débordement au niveau du *solveur*, et de gérer toute configuration de flottants.

Cette recherche locale a deux paramètres essentiels : le paramètre de profondeur et celui de l'éloignement de la configuration courante. Celui de l'éloignement permet de forcer la recherche locale à s'éloigner d'une région de minimums locaux. Celui de la profondeur a l'effet inverse, c'est-à-dire visiter les voisions proches. Une première expérimentation a montré un premier intérêt de cette approche. Notre perspective immédiate est de pousser loin nos expérimentations sur des programmes informatiques plus significatifs, notamment pour améliorer notre recherche locale.

[8] Contrainte de débordement de la variabe DELTA

[9] Intervalle de débordement flottant de type $Underflow$

References

1. ANSI/IEEE, New York. *IEEE Standard for Binary Floating Point Arithmetic*, Std 754-1985 edition, 1985.
2. Bernard Botella, Arnaud Gotlieb, and Claude Michel. Symbolic execution of floating-point computations. *Softw. Test., Verif. Reliab.*, 16(2):97–121, 2006.
3. Bernard Botella, Arnaud Gotlieb, Claude Michel, Michel Rueher, and Patrick Taillibert. Génération automatique de cas de test structurels avec les techniques de programmation par contraintes. *TSI (Hermes).*, pages 21:1163–1187, 2002.
4. Digital Equipment Corporation. *DECV AX Architecture Handbook.* MIT, 1981.
5. A. Gotlieb. *Automatic Test Data Generation using Constraint Logic Programming.* PhD thesis, Université de Nice — Sophia Antipolis, France, 2000.
6. Claude Michel, Michel Rueher, and Yahia Lebbah. Solving constraints over floating point numbers. *Lecture Notes in Computer Science (LNCS)*, pages 2239:524–538, 2001.
7. Richard M. Russell. The cray-1 computer system. *Commun. ACM*, 21(1):63–72, 1978.
8. J.E. Thornton. *Design of a Computer: The Control Data 6600.* Glenview, Illinois, 1970.
9. Bill Venners. Floating point arithmetic: A look at the floating point support of the java virtual machine. *Java World*, 1996.

Importance du Site dans le Calcul de la Probabilité A Priori de Pertinence d'une Page Web

Arezki Hammache[1], Mohand Boughanem[2], Rachid Ahmed-Ouamer[1]

[1] Laboratoire LARI, Université Mouloud Mammeri
15000 Tizi-Ouzou, Algérie
{arezki20002002, ahm_r}@yahoo.fr
[2] Laboratoire IRIT, Université Paul Sabatier
118 route de Narbonne 31062 Toulouse Cedex 09, France
bougha@irit.fr

Abstract. Les moteurs de recherche d'information sur le web sont les outils les plus utilisés pour l'accès à l'information. Cependant la plupart de ces moteurs négligent un facteur important qui est la qualité des documents restitués. Ceci est du en partie à la non prise en compte de toutes les caractéristiques de documents web dans les processus d'indexation et de recherche, entre autres la structure du web. Les modèles de langage offrent un cadre probabilistique qui permet de modéliser le processus de la recherche d'information et la possibilité d'incorporer des informations sur la pertinence a priori du document, en le conditionnant avec des caractéristiques de documents. Plusieurs caractéristiques on été utilisées pour estimer la probabilité a priori d'un document comme : la longueur du document, la structure des liens, le facteur temps, le rapport Information/Bruit. Cependant, ces caractéristiques dépendent seulement du document. Or, une page web fait partie en général d'un site web lequel fait partie du web. L'idée que nous explorons ici est l'utilisation des caractéristiques du site contenant la page concernée pour conditionner la probabilité de pertinence a priori de la page.

Mots-clés. Recherche d'information sur le web – Modèles de langage – Probabilité a priori de pertinence

1 Introduction

La plupart des moteurs de recherche d'information (RI) sur le web privilégient la minimisation du temps de réponse par rapport à la qualité des documents retournés à l'utilisateur. En effet, ces derniers délivrent des résultats massifs en réponse aux requêtes des utilisateurs, qui génèrent ainsi, une surcharge informationnelle dans laquelle il est difficile de distinguer l'information pertinente de l'information secondaire ou même du bruit.

L'une des raisons qui a engendré ceci est la non prise en compte de toutes les caractéristiques d'un document web dans les processus d'indexation et de recherche.

En effet, les moteurs de recherche implémentent les techniques traditionnelles de la RI, qui considèrent un document web comme un ensemble de termes (sac de mots). Cependant, les documents web, généralement sous format Html, sont des documents structurés via des balises et interconnectés par des liens hypertextes. De plus un document web traite un ou plusieurs thèmes exprimés implicitement par les liens entre les termes du document.

Dans cet article est traité l'exploitation de la structure des hyperliens pour estimer la probabilité a priori de pertinence d'un document web. En se basant sur le postulat qu'une page web plus populaire est plus probable à être pertinente qu'une page moins populaire. La popularité de la page est conditionnée elle même par la popularité du site qui la contient.

Les modèles de langage offrent un cadre probabilistique pour la description du processus de la RI. Les résultats obtenus ont montré des performances équivalentes voire supérieures a celles des modèles classiques (vectoriel, probabiliste).

Le modèle de langage convient bien à la recherche d'information sur le web, et s'appuie sur des bases mathématiques qui permettent d'analyser et de modéliser d'énormes masses de documents (le web). De plus il permet de combiner différentes représentations d'un document comme l'intégration des connaissances a priori de la pertinence d'un document web.

Plusieurs caractéristiques des documents web ont été explorées pour estimer la probabilité a priori de pertinence d'un document, comme : la longueur de document, la structures des liens, etc. Nous proposons dans cet article l'exploitation de la structure du web pour estimer cette probabilité a priori, en considérant le web comme un ensemble de sites composés d'un ensemble de pages web.

Nous organisons ce papier comme suit : dans la section 2 sont abordés la modélisation de langage et l'intégration dans les modèles de langage des informations sur la pertinence a priori de document. La section 3 est consacrée à la présentation de l'approche que nous proposons pour intégrer les informations a priori de pertinence d'un document web dans le modèle de langage. Un exemple d'illustration de l'approche proposée est donné dans la section 4. La dernière section fait la synthèse de ce travail.

2 Modélisation de Langue et Pertinence A Priori de Document

2.1 La Modélisation de Langue en Recherche d'Information

Les approches traditionnelles de la RI incluant les modèles : booléen, vectoriel, et probabiliste comportent un modèle d'indexation, qui décrit comment sont représentés les documents et les requêtes et un modèle de pertinence qui définit la fonction de correspondance entre deux représentations (de documents et de requêtes). Ces modèles utilisent d'une manière heuristique différentes statistiques telles que (la fréquence d'un terme dans un document, la fréquence en document d'un terme, etc.).

L'approche de modélisation de langue part d'un principe différent des approches traditionnelles ; on ne tente pas de modéliser directement la notion de pertinence (à l'exception de [7]) ; mais on considère que la pertinence d'un document face à une

requête est en rapport avec la probabilité que la requête puisse être générée par un modèle de langue d'un document. Ainsi un modèle de langue « Md » est construit pour chaque document, et le score d'un document est déterminé par la probabilité de génération de la requête sachant le modèle de ce document, notée ainsi :

$$P\ (q \mid Md).$$

Cette approche permet de combiner les deux composantes (indexation et Ranking) dans un seul modèle unifié. La plupart des modèles de langue développés pour la RI utilisent le principe de génération de la requête par un modèle de document.

Se basant sur la représentation des documents et la fonction de Ranking les approches de modélisation de langage pour la RI peuvent être classées en trois catégories :

– Génération de la requête par le modèle de document (Query Likelihood Models) : dans cette approche un modèle de langage est associé à chaque document. Les documents sont classés selon leurs probabilités de génération de la requête [4], [9], [10]. Le score d'un document vis-à-vis d'une requête (la probabilité de générer une requête « Q » sachant le modèle de document « Md ») est donné par la formule suivante :

$$Score\ (D,Q) = P(D) \prod_{i=1}^{n} P(t_i / M_d)\ . \qquad (1)$$

Cette catégorie de modèles de langage permet en particulier d'incorporer des informations sur la pertinence a priori du document, en utilisant le facteur P (D) et en le conditionnant avec les caractéristiques de document.

– Génération de document à partir du modèle de la requête : cette approche procède dans le sens inverse. Ainsi, un modèle de langage de la requête est construit, ensuite les documents sont classés selon leurs probabilités que leur contenu soit généré par le modèle de la requête. Le travail de Lavranko et Croft [7] s'insert dans cette catégorie d'approche.

– Similarité entre modèle de document et modèle de la requête : dans cette approche un modèle de langage est construit pour chaque document et un autre pour la requête. Les documents sont alors ordonnés selon la similarité de leurs modèles avec le modèle de la requête. C'est ce qui est proposé dans Lafferty et Zhai [6] en rapport avec la minimisation de risque basée sur la théorie de décision bayésienne.

Plusieurs travaux de recherche se sont penchés sur le problème lié à l'utilisation des modèles de langage en RI et qui concerne la clairsemence de données (Data Sparseness) : dans les modèles de langage développés pour la RI, l'idée est de classer les documents selon la probabilité P(Q|D) pour une requête « Q » donnée, généralement calculée par la multiplication des probabilités individuelles des termes de la requête P(ti|D). Cependant, si un document « D » contient tous les termes de la requête sauf un alors on lui attribue la probabilité nulle même si le document est pertinent. Pour y remédier la technique de lissage qui permet d'attribuer une probabilité non nulle aux termes non observés dans le document est utilisée. Par exemple «le lissage de Laplace, le lissage de Good-Turing, le lissage de backoff et le lissage par interpolation».

Zhai et Lafferty [11] ont expérimenté plusieurs techniques de lissage en utilisant le modèle de langage uni-gramme et ils ont rapporté que la méthode de lissage par interpolation donne de meilleurs résultats.
La méthode de lissage par interpolation consiste à lisser le modèle de document « Md » par le modèle de la collection « Mc ». La formule (1) devient alors :

$$Score\ (D, Q) = P\ (D) \prod_{i=1}^{n} \lambda P(t_i / M_d) + (1-\lambda) P(t_i / M_c). \qquad (2)$$

2.2 La Pertinence A Priori d'un Document

Le score d'un document vis-à-vis d'une requête est donné par la formule (2), et cela selon l'approche de génération de la requête par le modèle de document (Query Likelihood Models).

Selon l'approche adoptée les propriétés (taille de document, nombre de liens entrants, etc.) indépendantes des requêtes peuvent être utilisées pour conditionner la probabilité a priori de pertinence d'un document. Si la probabilité a priori de pertinence P(D) n'est pas conditionnée par l'une de ces propriétés alors cette probabilité représente la probabilité de prélever un document de la collection, par conséquent tous les documents sont équiprobables dans la collection, et la probabilité a priori de pertinence de document peut être ignorée lors du classement des documents.

Par contre, si la probabilité a priori est conditionnée par l'une de ces caractéristiques alors les documents de la collection n'ont pas la même probabilité a priori, et les documents ne sont pas équiprobables. Par exemple, si la caractéristique utilisée est le score de popularité de document alors un document populaire est plus probable d'être pertinent qu'un document moins populaire.

Plusieurs caractéristiques ont été utilisées pour estimer la probabilité a priori d'un document comme : la longueur du document [9], la structures des liens [3], [5], le facteur temps [2], [7], le rapport Information/Bruit [12]. Elles expriment qu'un document est plus probable parce que : il est plus long, il est plus populaire, il est plus récent, ou contient plus d'informations que de bruit.

Les caractéristiques les plus souvent utilisées sont :

– La taille du document : la probabilité a priori est proportionnelle à la taille du document, telle que $P(D) = \frac{|D|}{|C|}$, |D| est la taille du document et |C| la taille de la collection.

Un document plus long tend à contenir plus d'informations et par conséquent il est plus probable d'être pertinent. Les résultats obtenus avec l'utilisation de cette caractéristique ont été mixtes selon la collection utilisée [5], [9].

– La structure des liens : les documents populaires ou les plus cités tendent à être plus pertinents. La méthode utilise généralement le nombre de liens entrants, i.e. :

$$P(D)= \frac{n(l,D)}{\sum_{D'} n(l,D')}.$$

Où *n(l, D)* est le nombre de liens entrants.

D'autres facteurs peuvent être utilisés comme le Page Rank [1].

– Rapport Information/Bruit : il est défini comme le rapport entre la taille de document après prétraitement (élimination des mots vides et des balises Html) et la taille de document sans prétraitement [12] :

$$P(D)= \frac{L_{token}}{L_{document}}.$$

Tel que : L_{token} est la taille de document après le prétraitement et $L_{document}$ est la taille de document avant le prétraitement. Ainsi, un document avec moins de mots vides et peu de balises Html produit un haut rapport Information/Bruit, ce qui signifie que le document est de « bonne » qualité.

– Type d'URL du document : Kraaij et al [5] ont utilisé le type d'URL pour estimer la probabilité qu'une page soit une page d'entrée.

$$P(D) = P(PE|URLtype(D) = ti) = \frac{c(PE,t_i)}{c(t_i)}.$$

Où *URLtype* est le type d'URL de document D, *c(PE, ti)* est le nombre de documents de type d'URL « ti » qui sont des pages d'entrées « PE » pour un site web obtenu à partir des évaluations de pertinence, c(ti) est le nombre de documents de type d'URL « ti ».

Quatre types d'URL ont été définis :

– Racine : elle contient le nom de domaine seulement, par exemple : www.sigir.org

– Sous-racine : elle contient le nom de domaine suivi d'un seul répertoire, par exemple : www.sigir.org/sigirlist/

– Chemin (répertoire) : il contient le nom de domaine suivi d'un ou de plusieurs répertoires, par exemple : www.sigir.org/sigirlist/issues/

– Fichier : tout URL se terminant avec un nom de fichier autre qu'index.html

Sur la base de ces quatre types d'URL ont mené des expérimentations sur la collection WT10g (collection utilisée dans TREC 2001) pour estimer la probabilité qu'une page soit une page d'entrée sachant son type d'URL. Il a été constaté que cette information est un bon indicateur pour prévoir la pertinence d'une page d'être une page d'entrée.

3 Intégration de la Pertinence A Priori d'un Document dans le Modèle de Langage

Plusieurs caractéristiques ont été utilisées pour estimer la probabilité a priori P(D) d'un document. Cependant, les caractéristiques utilisées jusqu'ici dépendent du document web (page) seulement. Or, une page web fait partie en général d'un site lequel fait partie du web. L'idée explorée ici est l'utilisation des caractéristiques du site web qui contient la page concernée pour conditionner la probabilité de pertinence a priori de la page. Sous l'hypothèse que dans la plupart des cas les auteurs des pages web référencent la page principale « site » au lieu de référencer la page exacte « la page concernée », l'utilisation du nombre de liens entrants ou de facteurs (comme le Page Rank) ne reflète pas l'importance de la page web dans l'espace web. Autrement dit on doit assigner plus de confiance aux pages provenant de sites importants « intéressants » que celles provenant de sites moins importants ou même des sites spams.

Pour cela nous introduisons le facteur importance de site web « page principale » dans l'évaluation de l'importance d'une page.

Avec la notation suivante où :

Nb1 : nombre de liens pointant le site « page principale »

Nb2 : nombre de liens pointant la page concernée (p)

N : le nombre de pages dans le site contenant la page (p), (ce nombre peut être considéré comme constant).

La formule (3) suivante proposée, exprime la probabilité a priori de pertinence d'une page, qui intègre l'importance de site qui la contient :

$$P(D)= C[\alpha((Nb1)/N) + (1- \alpha) Nb2]. \quad (3)$$

Tel que C est une constante et α est compris entre 0 et 1.

Dans notre étude nous utilisons le modèle de Ranking exprimé par la formule (2) afin de classer les documents. Nous présentons ci-dessous un exemple d'illustration de l'approche proposée.

4 Exemple d'Illustration

La table 1 ci-dessous donne un exemple de caractéristiques de 10 pages web. L'importance de la page est fixée à une même valeur (0.6) pour toutes les pages (p1-p10). Le nombre de pages dans le site est aussi fixé à 5.

L'importance de site contenant la page prend les valeurs 0.1, 0.2,…,0.9 pour les 9 premières pages et la valeur 0.99 pour la $10^{ème}$ page.

On fait ensuite varier l'apport de l'importance de site dans la nouvelle importance de la page et cela en prenant deux valeurs pour le facteur « α » : 0.7 et 0.5.

Table 1. Caractéristiques de 10 pages web

pages	Facteurs	α=0,7	α=0,5	pages	Facteurs	α=0,7	α=0,5
	importance_page	0,6	0,6		importance _page	0,6	0,6
	importance_site	0,1	0,1		importance_site	0,4	0,4
	nombre_page_site	5	5		nombre_page_site	5	5
	nouvelle_importance	0,426	0,31		nouvelle_importance	0,444	0,34
	score_page	0,25	0,25		score_page	0,6	0,6
	classement_page	8	8		classement_page	2	2
	nouveau_score_page	0,177	0,129		nouveau_score_page	0,444	0,34
	rapport entre scores	0	0		rapport entre scores	0,03	0,05
p1	nouveau_classement	9	9	p4	nouveau_classement	2	3
	importance _page	0,6	0,6		importance _page	0,6	0,6
	importance_site	0,2	0,2		importance_site	0,5	0,5
	nombre_page_site	5	5		nombre_page_site	5	5
	nouvelle_importance	0,432	0,32		nouvelle_importance	0,45	0,35
	score_page	0,35	0,35		score_page	0,45	0,45
	classement_page	5	5		classement_page	4	4
	nouveau_score_page	0,252	0,187		nouveau_score_page	0,337	0,262
	rapport entre scores	0,01	0,017		rapport entre scores	0,04	0,067
p2	nouveau_classement	5	6	p5	nouveau_classement	4	4
	importance _page	0,6	0,6		importance _page	0,6	0,6
	importance_site	0,3	0,3		importance_site	0,6	0,6
	nombre_page_site	5	5		nombre_page_site	5	5
	nouvelle_importance	0,438	0,33		nouvelle_importance	0,456	0,36
	score_page	0,28	0,28		score_page	0,18	0,18
	classement_page	7	7		classement_page	10	10
	nouveau_score_page	0,204	0,154		nouveau_score_page	0,137	0,108
	rapport entre scores	0,02	0,033		rapport entre scores	0,05	0,083
p3	nouveau_classement	7	8	p6	nouveau_classement	10	10
	importance _page	0,6	0,6		importance _page	0,6	0,6
	importance_site	0,7	0,7		importance_site	0,9	0,9
	nombre_page_site	5	5		nombre_page_site	5	5
	nouvelle_importance	0,462	0,37		nouvelle_importance	0,474	0,39
	score_page	0,31	0,31		score_page	0,88	0,88
	classement_page	6	6		classement_page	1	1
	nouveau_score_page	0,239	0,191		nouveau_score_page	0,695	0,572
	rapport entre scores	0,06	0,1		rapport entre scores	0,08	0,133
p7	nouveau_classement	6	5	p9	nouveau_classement	1	1
	importance _page	0,6	0,6		importance _page	0,6	0,6

p8	importance_site	0,8	0,8	p10	importance_site	0,99	0,99
	nombre_page_site	5	5		nombre_page_site	5	5
	nouvelle_importance	0,468	0,38		nouvelle_importance	0,479	0,399
	score_page	0,55	0,55		score_page	0,24	0,24
	classement_page	3	3		classement_page	9	9
	nouveau_score_page	0,429	0,348		nouveau_score_page	0,192	0,160
	rapport entre scores	0,07	0,117		rapport entre scores	0,089	0,1483
	nouveau_classement	3	2		nouveau_classement	8	7

L'augmentation des scores (nouveaux scores) des pages (nsc (p) dont le calcul est précisé en table 2) est proportionnelle à la valeur du facteur « α » et de l'importance du site « imp(s) », et elle est inversement proportionnelle au nombre de pages dans le site « N» et de l'importance initiale de la page « imp (p) ».

Table 2. Formules de la nouvelle importance et du nouveau score

Formule de la nouvelle importance (imp'(p))	Formule de nouveau score (nsc (p))
imp'(p)= α*imp(p)+(1-α)*(imp(s)/N)	nsc(p)=[imp'(p)/imp(p)]*sc(p)

Le rapport entre l'ancien score et le nouveau score est donné par la formule suivante :

$$nsc(p)/sc(p) = (sc(p))/([imp'(p)/imp(p)]*sc(p))$$
$$= imp'(p)/imp(p)$$
$$= (\alpha*imp(p)+(1-\alpha)*(imp(s)/N))/ imp(p)$$
$$= \alpha+(1-\alpha)*imp(s)/(N* imp(p))$$

– Ainsi avec une valeur α = 0.7 la modification des scores des pages est faible (cf. figure 1), et le classement des pages n'est modifié que pour les deux pages p1 et p10 (cf. figure 2).
– Avec une valeur de α = 0.5 la modification des scores est un peu plus importante (cf. figure 1) et le classement des pages est modifié pour la plupart des pages p1- p4, p7, p8, et p10 (cf. figure 2).

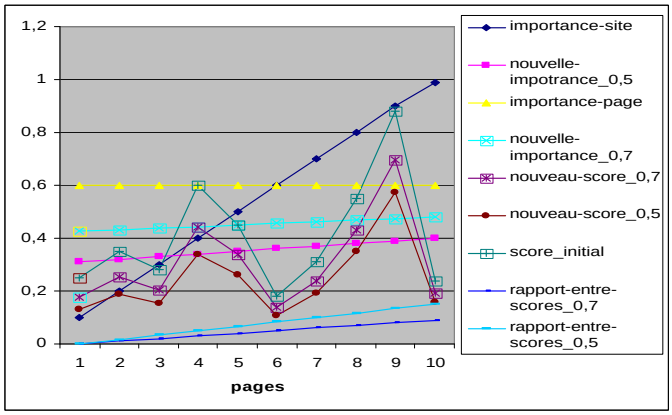


Fig. 1. Modification des scores selon les valeurs de α (0,5 et 0,7)

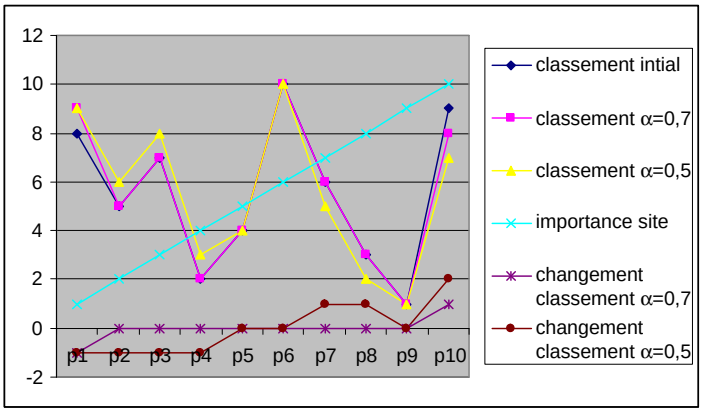


Fig. 2. Classement des pages selon les valeurs de α (0,7 et 0,5)

La table 3 suivante donne les résultats obtenus en fonction de l'importance du site :

Table 3. Résultats en fonction de l'importance du site

Page	classement initial	classement avec α=0,7	classement avec α=0,5	importance site	changement classement avec α=0,7	changement classement avec α=0,5
p1	8	9	9	1	-1	-1
p2	5	5	6	2	0	-1
p3	7	7	8	3	0	-1
p4	2	2	3	4	0	-1
p5	4	4	4	5	0	0
p6	10	10	10	6	0	0
p7	6	6	5	7	0	1
p8	3	3	2	8	0	1
p9	1	1	1	9	0	0
p10	9	8	7	10	1	2

5 Conclusion

Nous avons proposé dans cet article une approche qui permet d'intégrer des informations pour conditionner la probabilité a priori de pertinence d'un document web. Ces informations sont l'importance de la page et l'importance du site contenant la page. Et cela en se basant sur l'hypothèse suivante : « on doit assigner plus de confiance aux pages provenant de sites importants « intéressants » que celles provenant de sites moins importants. Nous avons réalisé cette intégration dans le cadre du modèle de langage. L'exemple d'illustration que nous avons donné, montre que la prise en compte de l'importance du site contenant la page modifie le classement des pages.

Bien que la prise en compte de caractéristiques complémentaires pour juger de la pertinence d'une page web soit justifiable, l'approche proposée tend à favoriser les pages provenant de sites « attestés » et renforcera le phénomène déjà observé de l'omniprésence de pointeurs vers Wikipédia, Amazon, eBay, etc. Ceci peut être au détriment de nombreux blogs créés récemment et potentiellement intéressants. Cette limite est déjà observable chez Google notamment où la prise en compte de l'ancienneté d'une page et la réputation du nom de domaine sont des critères de classement.

Cependant la prise en compte du site web dans sa globalité constitue une approche novatrice et intéressante. Les méthodes de classement actuelles n'en font pas usage. La prochaine étape consiste à mettre en œuvre cette approche proposée tout en estimant les différents paramètres utilisés et à évaluer ses performances en la comparant avec d'autres approches et solutions existantes.

Références

1. Brin S. et Page L. : The anatomy of a large-scale hypertextual web search engine. In: Proc. of www7, Brisbane, Australia, (1998). http://www7.scu.edu.au/programme/fullpapers/1921/com1921.html
2. Diaz F., Jones R. : Using temporal profiles of queries for precision prediction. The 27th annual international conference on Research and development in information retrieval, ACM Press, (2004) 18–24
3. Hauff C., Azzopardi L. : Age dependent document priors in link structure analysis. The 27th European Conference in Information Retreival ». Springer, (2005) 552–554
4. Hiemstra D. : A linguistically motivated probabilistic model of information retrieval. Second European Conference on Research and Advanced Technology for Digital Libraries, ECDL'98, LNCS number 1513, Nicolaou C., Stephanides C. (Eds.), Springer Verlag, (1998)
5. Kraaij W., Westerveld T., Hiemstra D. : The importance of prior probabilities for entry page search. ACM SIGIR Conference on Research and Bibliography and Development in Information Retrieval. Tampere, Finland, (2002) 27–34
6. Lafferty J., Zhai C. : Document language models, query models, and risk minimization for information retrieval. The 24th annual international ACM-SIGIR conference on research and development in information retrieval, New Orleans, Louisiana, Croft W.B., Harper D.J., Kraft D.H., Zobel J. (Eds.), New York: ACM, (2001) 111-119
7. Lavrenko V., Croft, W. B. : Relevance-based language models. The 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, Croft W.B., Harper D.J., Kraft D.H., Zobel J. (Eds.), New York: ACM, (2001) 120-127
8. Li X., Croft W. B. : Time-based language models. The twelfth international conference on Information and knowledge management CIKM '03, ACM Press, (2003) 469–475
9. Miller D. R. H., Leek T., Schwartz R. M. : A hidden markov model information retrieval system. Hearst *et al*. (Eds.), (1999) 214–221
10. Ponte J.M., Croft W. B. : A language modeling approach to information retrieval. Croft *et al*. (Eds.), (1998) 275–281
11. Zhai C., Lafferty J. : A study of smoothing methods for language models applied to ad hoc information retrieval. The 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, Croft W.B., Harper D.J., Kraft D.H., Zobel J. (Eds.), New York: ACM, (2001) 334-342
12. Zhu X. L., Gauch S. : Incorporating quality metrics in centralized / distributed information retrieval on the World Wide Web. The 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, (2000) 288–295

Hybridation STM SVM pour classifier des trajectoires multidimensionnelles phonétiques

Mourtada Benazzouz, Mohammed Amine Chikh
Génie Biomédical (GBM), Dépt. Informatique, Université Abou Bakr Belkaid Tlemcen, Algérie
m_benazzouz@mail.univ-tlemcen.dz, mea_chikh@mail.univ-tlemcen

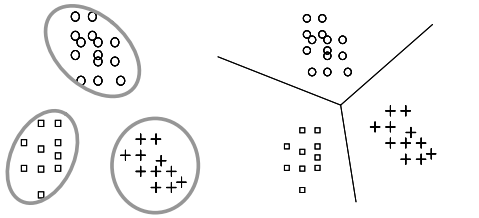
Résumé. Lors de la conception d'un système de reconnaissance de formes, l'objectif principal est de minimiser les erreurs de classification. La variabilité de la longueur des différentes représentations des formes, constitue un problème majeur pour l'intégralité des algorithmes de classification, d'où l'utilisation des outils Outer-Product & Inner-Product pour la normalisation de ces représentations. Face à des problèmes complexes comme celui de la Reconnaissance Automatique de la Parole (RAP), les systèmes à classificateurs multiples ont bénéficié d'un intérêt croissant durant ces dernières années. Fondés sur des principes de complémentarité, ils visent à accroître les performances d'un système de reconnaissance en limitant l'erreur liée à l'utilisation d'un classificateur unique. Dans cet article, on présente un classificateur hybride qui s'appuie sur une combinaison hiérarchique de deux approches de classification différentes. Une technique probabiliste STM « modèle stochastique de trajectoire » et une autre discriminante SVM « Séparateurs à Vastes Marges ».Le premier type d'approche cherche à déterminer un modèle le plus fidèle possible de chacune des classes, alors que l'objectif du second type est d'optimiser des frontières de décision de manière à séparer au mieux les classes. En outre, cette combinaison présente l'avantage de réduire la complexité de calcul associée à la prise de décision des SVM.

Mots-clés: RAP, Outer-Product, Inner-Product, modèle stochastique de trajectoire (STM), Support Vector Machines (SVM).

1 Introduction

Si l'homme a la faculté de comprendre un message vocal provenant d'un locuteur quelconque, dans des environnements souvent perturbés par le bruit, quelques soient son mode d'élocution, la syntaxe et le vocabulaire utilisés, la machine est-elle capable d'en faire autant ? Une solution peut-elle répondre en globalité à ces difficultés ? Le problème de la reconnaissance vocale est un sujet d'actualité et pour l'instant, seules les solutions partielles sont aptes à répondre aux différentes tâches que la machine doit effectuer. Compte tenu de l'ampleur du domaine, ce travail décrit deux approches adaptées aux phénomènes transitoires propres à la parole. La modélisation stochastique est une méthode souple pour tenir compte de la grande variabilité de la parole. Contrairement à la programmation dynamique qui utilise des méthodes

heuristiques pour construire des formes de référence robustes, les modèles stochastiques permettent un apprentissage rigoureux reposant sur la théorie des probabilités [5]. Support Vector Machines ou les Machines à Vecteurs de support est une méthode de classification qui fut introduite par Vapnik [10]. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent.



(a) par modélisation (b) par discrimination

Fig. 1. Deux catégories d'approches de classification

2 Modèles stochastiques de trajectoires

Dans un espace de paramètres spécifiques à la parole, le signal de parole est un point qui se déplace lorsque l'articulation évolue. En se déplaçant, ce point décrit une certaine trajectoire dont l'expression analytique est inconnue. Cette trajectoire peut être considérée comme une réalisation d'une fonction aléatoire, ou plus simplement comme une réalisation interpolée d'une séquence de vecteurs aléatoires. Cette séquence sous-jacente de vecteurs aléatoires constitue le modèle de l'ensemble des trajectoires associées à une unité de parole donnée. Bien évidemment, la notion de trajectoire peut caractériser différentes unités de paroles : phonèmes, syllabes, mots, etc., le signal de parole étant constitué d'une concaténation d'unités élémentaires. La reconnaissance de la parole consiste alors à rechercher la succession de modèles expliquant au mieux, selon un certain critère, le signal de parole observé. Dans la suite de notre travail, nous supposons que l'unité de parole associée à la trajectoire est le phonème.

2.1. Trajectoires de parole

On définit P comme un ensemble de H symboles représentant les phonèmes : $\mathrm{P} = \{s_1, s_2, ...s_H\}$, et on souhaite construire un modèle caractérisant les trajectoires associes à un symbole s issu de P. Soit Ns un ensemble de trajectoires spécifique au symbole s, chaque trajectoire Y_i est constituée d'une succession de d_i vecteurs dans un espace de dimension *D* spécifique à la parole (exemple espace cepstral). On note $Y_0, ..Y_{d_i-1}$ les d_i vecteurs constituant la trajectoire Y_i, où d_i est appelé durée de la trajectoire Y_i [2].

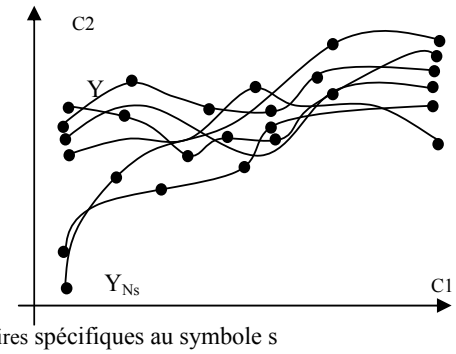


Fig. 2. N_s trajectoires spécifiques au symbole s

Ces Ns trajectoires doivent être considérées comme autant de réalisations d'une trajectoire aléatoire sous-jacente qui constitue le modèle.

Définir un modèle pour ces trajectoires pose deux problèmes majeurs :

- La prise en compte des variations de durée des trajectoires (d'où nécessité d'échantillonnage ou normalisation).
- La modélisation d'une distribution complexe des trajectoires dans l'espace des trajectoires.

2.2. Normalisation de la base

Il est plus simple de modéliser sous un cadre stochastique des trajectoires ayant toutes la même durée, plutôt que des trajectoires de durées différentes. Ainsi, avant l'étape de modélisation, on doit d'abord échantillonner chaque trajectoire en un nombre fixe de points. Une trajectoire devient alors une séquence de Q vecteurs. Pour cela, on a utilisé deux approches, échantillonnage linéaire et Outer-Product de matrice des trajectoires.

2.2.1 Échantillonnage linéaire de la base

Dans le modèle STM, Gong [4] choisit d'effectuer l'échantillonnage linéairement, de la façon suivante :

Soit Y une trajectoire observée de durée *d* trames, la trajectoire échantillonnée X s'écrit :

$$Y = (y_0, \ldots y_{d-1}) \rightarrow X = (x_0, \ldots x_{Q-1}) \text{ avec } x_i = y_{i \times \frac{d-1}{Q-1}},\ 0 \le i \le Q \quad (\mathbf{1})$$

Notons que la division utilisée pour calculer l'indice de y dans l'équation est une division entière.

Si d < Q, correspond à un sur-échantillonnage de Y, ce qui signifie que certains vecteurs de Y peuvent être répétés plusieurs fois dans X.

Dans le cas contraire où d > Q, la trajectoire Y va être sous-échantillonnée, ce qui signifie que certains vecteurs constituant Y n'apparaîtront pas dans X.

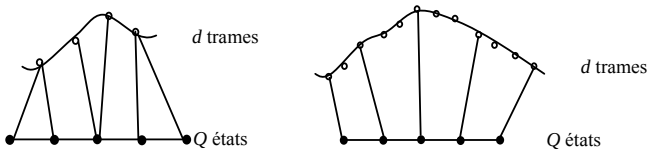


Fig. 3. Sur-échantillonnage & Sous-échantillonnage des trajectoires

En pratique, l'échantillonnage est effectué en 5 points (Q = 5), l'intervalle de temps entre 2 trames consécutives y_i et y_{i+1} étant de 10 ms.

2.2.2 Outer-Product de matrice de trajectoire

L'analyse d'un segment de la parole donne une séquence de vecteurs paramétriques de dimension *l*. Un tel ordre est considéré comme trajectoire dans l'espace de dimension *l*, et *d* étant la durée pour un segment donné. La matrice de trajectoire pour le segment s'écrit : X(*l*,*d*), $X = [x_1 x_2 ... x_d]$.

La matrice d'Outer-Product, Z, d'une matrice de trajectoire X est donnée par : $Z = XX^T$ [1], de dimension *l* par *l*. La dimension de la matrice d'Outer-Product est indépendante du nombre de colonnes *d* dans la trajectoire. La matrice d'Outer-Product sera par la suite vectorisée pour obtenir un modèle de dimension fixe qui peut être employé comme entrée pour soit un modèle stochastique de trajectoires STM ou bien une machine à vecteur de support SVM.

Le calcul de la matrice Outer-Product d'une trajectoire peut être considéré comme opération de prétraitement sur la trajectoire pour la tracer un modèle de dimension fixe.

2.3 Modélisation des trajectoires échantillonnées

Le résultat de la normalisation des vecteurs de différentes durées obtenue par l'une des deux méthodes citées ci-dessus, est une trajectoire $X=(x_1,...,x_Q)$
Pour un symbole s de l'ensemble des phonèmes S, sera caractérisé par probabilité donnée par la loi de Bayes [8], [11].

$$p(s/X,d) = \frac{p(X,d,s)}{\sum_{s \in P} p(X,d,s)} \quad \textbf{(2)}$$

La fonction de densité de probabilité conjointe $p(X,d,s)$ d'une trajectoire échantillonnée X où *d* représente la durée de la trajectoire non échantillonnée Y.

$$p(X,d,s) = p(X/d,s) \,.\, p(d/s)^{\lambda} \,.\, \Pr(\tilde{s}=s)^{\gamma} \quad \textbf{(3)}$$

avec $\Pr(\tilde{s}=s)=\frac{N_s}{N}\ \forall s\in P$: la probabilité a priori du symbole *s*.

$p(d/s)$: la *pdf* de la durée d sachant le symbole *s*, modélisée par une loi Gamma Γ.

$p(X/d,s)$: la *pdf* conditionnelle d'une trajectoire connaissant la durée *d* et le symbole *s*.

Les facteurs de pondération λ, γ sont introduits afin de modifier l'influence de la probabilité de la durée et celle du symbole [8].

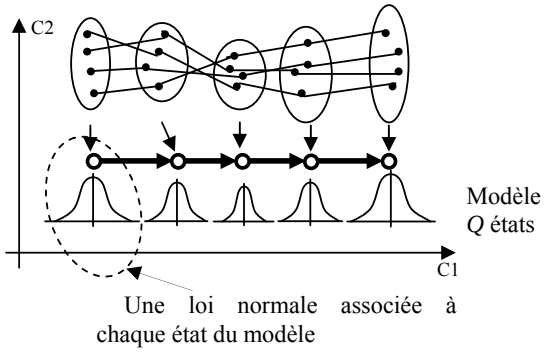


Fig. 4. Ensemble des trajectoires échantillonnées spécifique au symbole s

Pour procéder à la reconnaissance, il est nécessaire d'estimer les différents termes de l'équation (3), à savoir la distribution a priori des symboles $P_r(\tilde{s})$, les paramètres de la pdf de la durée connaissant le symbole $p_{\tilde{d}|\tilde{s}}(d|s)$ et enfin les paramètres de la pdf de la trajectoire échantillonnée connaissant le symbole et la durée $p_{\tilde{X}|\tilde{d},\tilde{s}}(X_n|d,s)$.

Les paramètres de ces différentes pdfs et distributions sont estimés selon le critère du maximum de vraisemblance (MLE) à partir d'un corpus d'apprentissage étiqueté au niveau phonétique sur la base TIMIT [2].

3 Support Vector Machines

Pour la classification des trajectoires multidimensionnelles à l'aide des SVM, il est nécessaire de définir l'Inner-Product de deux trajectoires dans le noyau de l'espace caractéristique.

L'inner-Product de deux vecteurs X et Y est donné par :

$$\langle x,y\rangle = x^T y = \frac{1}{2}\left(\|x\|^2+\|y\|^2-d^2(x,y)\right) \quad (4)$$

Ici $\|x\|$ est la norme du vecteur X et $d(x,y)$ est la distance euclidienne entre les vecteurs donnés.

L'Inner-Product entre deux trajectoires $X=\{x(t)\}$ et $Y=\{y(t)\}$ de la même longueur peut être obtenu à partir de la définition de l'Inner-Product de deux fonctions, et est donné comme suit :

$$\langle X,Y\rangle=\int_{t1}^{t2} x^T(t)y(t)dt \quad (5)$$

Cependant, cette définition d'Inner-Product ne peut pas être prolongée pour la trajectoire de deux longueurs différentes. On considère deux trajectoires discrètes dans le temps $X=\{x_1,x_2,...,x_m\}$ et $Y=\{y_1,y_2,...,y_n\}$ de longueur m et de n respectivement.

Le Inner-Product de la trajectoire X et Y peut être défini comme :

$$\langle x,y\rangle_P=\frac{1}{2}\left(\|x\|_P^2+\|y\|_P^2-D^2(X,Y)\right) \quad (6)$$

Ici $\|X\|_P$ est la norme ou la longueur de la trajectoire X et D(X,Y) est la distance entre les trajectoires. La norme de la trajectoire X peut être définie comme suit :

$$\|X\|_P=\|X_1\|^2+\sum_{i=2}^{m}\|x_i-x_{i-1}\|^2 \quad (7)$$

La distance entre les trajectoires, D(X,Y), peut être calculée en utilisant la méthode (DTW) « Dynamic Time Warping ».

Pour la classification des trajectoires multidimensionnelles à l'aide des SVM, il est nécessaire de définir le Inner-Product de deux trajectoires dans le noyau de l'espace caractéristique. $\phi(x_i)$ Étant le vecteur du noyau caractéristique de l'espace correspondant au vecteur paramétrique x_i. Alors la trajectoire $\phi(X)$ dans l'espace caractéristique correspondant à la trajectoire X dans l'espace paramétrique peut être définie comme suit :

$$\phi(X)=\{\phi(x_1),\phi(x_2),...,\phi(x_m)\} \quad (8)$$

L'Inner-Product des trajectoires dans l'espace caractéristique peut alors être défini:

$$\langle\phi(X),\phi(Y)\rangle_P=\frac{1}{2}\left(\|\phi(X)\|_P^2+\|\phi(Y)\|_P^2-D_\phi^2(X,Y)\right) \quad (9)$$

Où $D_\phi(X,Y)$ est la distance entre $\phi(X)$ et $\phi(Y)$.

La norme d'une trajectoire dans l'espace caractéristique est définie comme :

$$\|\phi(X)\|_P=\|\phi(x_1)\|^2+\sum_{i=2}^{m}\|\phi(x_i)-\phi(x_{i-1})\|^2 \quad (10)$$

La norme d'un vecteur dans l'espace caractéristique peut être calculée en utilisant le noyau d'Inner-Product :

$$\|\phi(x_i)\|=\sqrt{K(x_i,x_i)}=\sqrt{K_{ii}} \quad (11)$$

La distance euclidienne entre deux vecteurs dans l'espace caractéristique peut être calculée comme suit :

$$\|\phi(x_i)-\phi(x_j)\|=\sqrt{K_{ii}+K_{jj}-2*K_{ij}} \quad (12)$$

La définition précédente de la distance euclidienne est également employée en calculant la distance entre deux trajectoires dans l'espace caractéristique en utilisant la méthode de DTW.

La méthode Inner-Product emploie les données complètes disponibles dans le segment de la parole des unités de base. Il s'est avéré théoriquement que l'Inner-Product basé sur le noyau employé dans la construction des machines à vecteur de support est valable aussi pour noyau de Mercer [7].

4 Système à deux niveaux de décision

Il est possible de distinguer deux types de données pouvant causer des problèmes à un classificateur : les données ambiguës et les données aberrantes. Or, les algorithmes de classification peuvent être séparés en deux grandes catégories. Les approches agissant par séparation ont pour objectif de minimiser le premier type d'erreur, mais ne permettent pas de rejeter efficacement le deuxième type de données. Par contre, les approches agissant par modélisation sont adaptées à ce type de rejet, mais s'avèrent généralement peu discriminantes [6].

L'idée consiste alors à utiliser dans un premier niveau de décision une approche par modélisation pour rejeter les outliers, classer les données ne présentant aucune ambiguïté et isoler les classes en conflits. Le second niveau de décision, utilisera ensuite les SVM appropriés pour permettre une meilleure classification. De plus, cette combinaison présente l'avantage de réduire le principal fardeau des SVM : la complexité de calcul nécessaire à la prise de décision.

4.1 Premier Niveau STM

Déterminer la liste LC des classes en conflit à partir des probabilités $P_i(X,d,s)$ données du premier niveau

$$\sum_{i=1}^{P} P_i(X,d,s) \geq S_C \qquad (\mathbf{13})$$

Le seuil SC contrôle la tolérance du premier niveau, en l'augmentant, le nombre de classes en conflit aura tendance à croître.

Trois cas de figure sont alors envisageables :

- Toutes les distances sont très grandes. Il s'agit alors vraisemblablement d'un outlier qui pourra être rejeté.
- Une seule distance s'avère faible. Il s'agit d'une donnée facile à classer. La décision peut donc être prise directement.
- Plusieurs distances sont faibles. Il s'agit d'une donnée ambiguë. Le conflit sera alors réglé dans un second temps par le ou les SVM appropriés.

4.2 Deuxième Niveau SVM

L'objectif du second niveau de décision est de retraiter les données ambiguës à l'aide de classificateurs discriminants de manière à prendre la décision parmi les p classes en conflits. Ainsi on entraîne les SVM à toutes les paires de classes. Mais lors

de la classification seuls les *p*(p-1)/2* classificateurs défini par la liste LC seront utilisés, et la décision sera alors prise en effectuant un vote majoritaire et en cas d'égalité nous choisirons la classe ayant la plus petite distance *di*.

5 Expériences et Résultats

Les expériences de la classification des phonèmes ont été réalisées sur un noyau de la base de données TIMIT constitué de 6 voyelles, 6 fricatives et 6 plosives (table 1) et sur un PC PIV de 2.4Ghz et 512Mo de RAM. Les signaux ont été échantillonnés à 16 KHz avec une analyse cepstrale sous l'échelle Mel, prise toutes les 20ms dans des fenêtres de Hamming de 25ms donnant chacune 12 coefficients MFCC et l'énergie résiduelle correspondante.

Table 1. Le sous-corpus de TIMIT utilisé

Classe	Phonème	Train N_S	Test	$\Pr(s=s)$
Voyelles	/ah/	2200	879	0.0698
	/aw/	700	216	0.0222
	/ax/	3352	1323	0.1064
	/ax-h/	281	95	0.0089
	/uh/	502	221	0.0159
	/uw/	536	170	0.0170
Fricatives	/dh/	2058	822	0.0653
	/f/	2093	911	0.0664
	/sh/	2144	796	0.0680
	/v/	1872	707	0.0594
	/z/	3574	1273	0.1134
	/zh/	146	74	0.0046
Plosives	/b/	399	182	0.0127
	/d/	1371	526	0.0435
	/g/	1337	546	0.0424
	/p/	2056	779	0.0652
	/q/	3307	1191	0.1138
	/t/	3586	1344	0.0698

Avec Ns : nombre d'occurrences par phonème au niveau de l'apprentissage (train), $\Pr(\tilde{s}=s)$ probabilité a priori du symbole s.

Le taux de reconnaissance pour un phonème est obtenu à partir du nombre de phonème reconnu sur le nombre total de celui-ci dans la base test, en d'autre terme, la valeur de la diagonale divisé sur la somme de la ligne dans la matrice de confusion.

Pour le taux de reconnaissance globale est estimé par la somme de la diagonale divisé sur 12055 qui n'est autre que le nombre total de phonème de la base test. Dans

ce qui suit, on expose les résultats des expériences réalisées avec des taux sur les performances obtenues.

Tableau 2. Taux de reconnaissance des Expériences

Phonèmes	1	2	3	4	5	6	7	8
/ah/	63,48	51,30	30,94	51,42	63,25	64,73	43,00	65,42
/aw/	81,02	81,01	23,61	41,67	80,56	79,17	0,00	53,70
/ax/	48,37	52,98	46,03	47,24	50,79	50,26	40,06	7,63
/ax-h/	29,47	11,57	0,00	10,53	25,26	26,32	0,00	0,00
/uh/	50,23	27,60	4,98	8,60	48,42	49,32	24,43	28,05
/uw/	69,41	46,63	14,71	32,94	67,65	67,65	41,76	35,88
/dh/	47,57	37,59	28,71	37,23	48,30	48,05	39,05	7,79
/f/	89,57	83,97	50,16	52,80	89,24	88,80	49,51	47,09
/sh/	89,45	92,58	75,00	68,59	89,82	89,95	63,57	79,52
/v/	60,54	40,16	29,99	34,79	59,97	60,82	31,82	34,94
/z/	80,44	84,21	73,61	65,59	81,15	81,30	52,95	89,95
/zh/	50,00	24,32	9,46	9,46	48,65	50,00	0,00	0,00
/b/	30,77	68,68	0,55	2,20	26,37	28,57	36,26	0,00
/d/	33,08	51,71	3,99	23,95	30,04	31,37	32,70	8,94
/g/	67,03	59,15	32,97	47,25	66,67	67,03	37,36	44,51
/p/	67,01	39,28	29,14	35,82	68,29	67,65	46,34	27,09
/q/	73,30	59,11	43,24	58,44	73,97	74,31	44,25	18,72
/t/	59,60	42,33	40,63	33,63	60,49	60,57	50,37	46,06
Taux Global	**60,57**	**53,01**	**40,68**	**45,52**	**65,22**	**65,41**	**43,28**	**39,62**

Dans la première expérience E1, on a utilisé le modèle stochastique de trajectoires STM muni d'un échantillonnage linéaire, à base d'états modélisé par une simple

gaussienne pour la classification des phonèmes avec les facteurs de pondération λ =1,γ =1. Le nombre d'états d'un STM a été fixé à Q=5 dans toutes les expériences. On remarque qu'elle donne un taux de reconnaissance assez satisfaisant (60.57%).

Alors que pour l'expérience E2, c'est un modèle STM non pas avec un échantillonnage linéaire mais avec une normalisation Outer product des matrices de trajectoires. Notons un net recul du taux (53,01%), d'où on peut tirer une déduction que l'échantillonnage linéaire est mieux adapté que l'échantillonnage outer-product dans le cas des données phonétique multidimensionnelles.

L'approche discriminante SVM est entrée dans notre travail dans la troisième expérience E3, non pas avec une fonction noyau (Polynomial, Linéaire, Gaussien ou Laplacien) mais avec un noyau Inner Product présenté dans section 3, avec un échantillonnage linéaire de la base. Ici, Par contre les SVM ont été pénalisés par des phonèmes à caractère dominant dans la base d'une part, et d'autre part par la complexité qui augmente avec le nombre de classe, car on sait bien que le SVM est un bon classifieur binaire mais qui trouve des difficultés dans le cas multi classe.

L'expérience E4 est similaire à l'expérience E3 sauf qu'elle a été appliquée sur le corpus de la base TIMIT sans passer par la normalisation.

La cinquième expérience Ex5 est une combinaison de deux expériences E1 et E3. Donc c'est un système à deux niveaux de décision. La première décision se fait au niveau du modèle STM appliqué sur la base TIMIT avec un échantillonnage linéaire et le deuxième niveau est réalisé grâce au classificateur SVM avec noyau Inner product appliqué sur une base ayant subi un échantillonnage linéaire.

Pour l'expérience E6, c'est encore une hybridation STM échantillonnage linéaire et SVM noyau Inner mais sur base non normalisée. Pour E5 et E6, les résultats du système hiérarchique à deux niveaux de décision montrent qu'on a bel et bien profité des avantages des deux approches (65.22% & 65,41%), et cela se manifeste par un gain important en matière de temps d'exécution dans la passe SVM, puisque les STM lui ont permis d'éliminer à chaque fois la majorité des phonèmes non concernés.

Et en dernier lieu l'expérience E7, tout simplement un classificateur SVM doté d'un noyau gaussien mais avec une normalisation Outer-Product des matrices de trajectoires. Cette dernière expérience est faite pour tester l'impact de l'outil Outer-Product sur d'autres méthodes de classification et ainsi voir les performances de cette nouvelle transformation des données.

Une comparaison avec d'autres techniques comme le perceptron multicouche (mlp) des réseaux de neurones[9], dans la dernière colonne du tableau 2 d'expériences avec un taux de 39,62%, on peut dire que nos résultats sont assez satisfaisants et permettent une nette amélioration sur deux plans majeurs, le taux de reconnaissance étant le premier et le deuxième se présente par un gain de temps surtout à l'étape d'apprentissage.

6 Conclusion et perspectives

Dans l'utilisation des différentes approches permettant d'améliorer la robustesse des systèmes de reconnaissance automatique de la parole, il est difficile de se prononcer

d'une manière ou d'une autre sur la supériorité de telle ou telle méthode. Cependant il est impossible de réaliser une intégration simultanée de toutes améliorations.

Les SVM sont des techniques très récentes [10]. Plusieurs travaux utilisant ces techniques dans différentes applications ont montré qu'elles sont très efficaces, très intéressantes et surtout très prometteuses.

L'objectif de ce travail est d'introduire les techniques d'apprentissage statistique SVM « Support Vector Machines » dans le domaine de la RAP.

La reconnaissance automatique de la parole reste une tâche difficile, un grand nombre de voies de recherche restent ouvertes pour traiter tous les aspects de la communication Homme Machine notamment dans le domaine médical et assistance des handicapés [2].

Les résultats que nous avons obtenus par nos systèmes sont très intéressants. Cependant, tous les problèmes liés aux approches que nous avons proposés sont loin d'être résolus. De nombreuses études peuvent être à envisager afin de valider ces approches et beaucoup de travail reste à faire pour améliorer les systèmes proposés. Une question importante s'impose à ce niveau là. Est-ce que ces résultats sont dus à l'utilisation des SVM ou aux nouvelles représentations des données que nous avons proposées ?

Pour répondre à cette question, on peut envisager une étude qui consiste à utiliser ces mêmes représentations de données avec d'autres techniques discriminantes.

References

1. Anita, R., Srikrishna, D., Sekhar, C.: Outerproduct of trajectory matrix for acoustic modeling using support vector machines. International Workshop on Machine Learning for Signal Processing, Brazil, pp.355-363, Sep. 2004
2. Benazzouz, M. : Classification des Trajectoires Phonétiques par les Modèles Stochastiques de Trajectoires et les Machines à Vecteurs de Supports, Mémoire de magister en reconnaissance des formes et intelligence artificielle, département informatique, USTMB Oran, Avril 2007.
3. Cole, R.A., Mariani, J., Zuenen, A.: Survey of state of the art in human language technology. Center for Spoken Language Understanding, pp.35-42, Novenber 1995. Oregon graduate institute.
4. Gong, Y.: Stochastic Trajectory Modeling and sentence Searching fot Continious Speech Recognition. IEEE Transactions on Speech and Audio Processing, 5:33-44, January 1997.
5. Gong, Y., Haton, J.P.: Stochastic trajectory modeling for speech recognition. Proc.IEEE int. conf. on Acoustic, Speech and Signal Processing, ICASSP, 1 :57-60, 1994. Adelaide, Australia.
6. Miligram, J, Sabourin, R., Cheriet, M.: Système de classification à deux niveaux de décision combinant approche par modélisation et machines à vecteurs de support. Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle, École de Technologie Supérieure de Montréal.
7. Sekhar, C., Palaniswami, M.: Classification of multidimensional trajectories for acoustic modeling using support vector machines, 2004. Dept. of Comput. Sci. & Eng., Indian Inst. of Technol., Chennai, India;
8. Siohan, O.: Reconnaissance automatique de la parole continue en environnement bruitée:application à des modèles stochastiques de trajectoires. Thèse , Université H.Poincarré-Nancy I,1995

9. Tlemçani, R., Neggaz, N. :″reconnaissance de la parole par les réseau de neurones″ mémoire de pfe, département informatique, USTMB Oran, 2003
10. Vapnik, V.: The Nature of Statistical Learning Theory, Springer-Verlag, 1995.
11. Verhasselt, J.: The importance of segmentation probability in segment based speech recognizers. Proc. ICASSP-97, Apr 21-24, Munich, 1997

Reconnaissance de l'écriture manuscrite arabe basée sur une approche hybride de type MMC / PMC

Brahim Farou[1] , Samir Hallaci[1], Hamid Seridi[1,2]

[1]Département d'informatique, Université 08 mai 45 Guelma, B. P. 401, Algérie
[2]CResTIC, EA 3804, Université de Reims, B.P 1035, 51687, Reims, Cedex, France
LAIG, Université 08 mai 45 Guelma, B. P. 401, Guelma 24000, Algérie
{fbrahim24, s.hallaci,seridi}@yahoo.fr

Résumé. Dans ce papier, nous présentons un système de reconnaissance de l'écriture manuscrite arabe utilisant les modèles de Markov cachés (MMC) et les réseaux neuronaux (RN) dans une architecture probabiliste en tirant avantage des deux outils. Le classifieur MMC va générer une liste des N meilleures hypothèses de mots ainsi que leurs segmentations en caractères. Le classifieur RN utilise la segmentation du classifieur MMC afin de retourner à l'image du mot et d'extraire les caractéristiques convenables à la reconnaissance de caractères isolés. Le classifieur RN réévalue chacune des N meilleures hypothèses des mots et les scores générés sont combinés avec ceux du classifieur MMC. Finalement, une nouvelle liste des N meilleures hypothèses est réordonnée selon les nouveaux scores faisant ainsi ressortir la meilleure hypothèse. Pour tester les performances de notre système, nous avons utilisé deux bases. La première contient 14400 mots manuscrits constituant un lexique de 48 mots des wilayas algériennes. La deuxième contient 2800 caractères segmentés manuellement à partir des mots de la première base. La deuxième base a servi seulement pour l'apprentissage du classifieur RN. Le système proposé a atteint un taux de réussite de 91,77 % que nous estimons encourageant.

Mots clés : Reconnaissance de l'écriture manuscrite, MMC, RN, Viterbi, Baum welch.

1 Introduction

Des progrès considérables ont été réalisés dans le domaine de la reconnaissance de l'écriture manuscrite pendant la dernière décennie. Ces progrès est du d'une part aux nombreux travaux effectués dans ce domaine et d'autre part a la disponibilité de bases de données internationales standards relatives à l'écriture manuscrite qui permettait aux chercheurs de rapporter de façon crédible les performances de leurs approches dans ce domaine, avec la possibilité de les comparer avec d'autres approches vu qu'ils utilisent les mêmes bases. L'industrie des téléphones portables intelligents et des assistants numériques personnels a bénéficié largement de ces progrès ce qui a ramené les chercheurs à travailler d'avantage dans le domaine et d'essayer d'améliorer la qualité de la reconnaissance.

Contrairement au latin, la langue arabe n'a pas eu cette chance, elle reste encore au niveau de la recherche et de l'expérimentation [1], c'est-à-dire que le problème reste encore un pari ouvert pour les chercheurs. L'écriture arabe étant par nature cursive, elle pose de nombreux problèmes aux systèmes de reconnaissance automatique. Le problème le plus difficile lors de la conception d'un système de reconnaissance de l'écriture manuscrite est la segmentation des mots manuscrits en vue de leur reconnaissance, qui n'est pas toujours triviale et demande beaucoup de temps et de calcul. D'autre part les systèmes se basant sur une analyse globale négligeant les informations locales ce qui peut diminuer considérablement la performance du système [2]. Pour remédier à ces problèmes, des approches hybrides ont été proposées pour la reconnaissance des mots arabes manuscrits dans un vocabulaire limité. Dans un tel système, il doit nécessairement intégrer la prise en compte d'un nombre important de variabilités.

Le travail présenté dans ce papier, consiste à la conception d'un système de reconnaissance d'écriture manuscrite arabe dans un vocabulaire limité, nous proposons une approche hybride basée sur les réseaux de neurones et les modèles de Markov cachés comme outil de classification.

2 Prétraitement

Afin d'éliminer le bruit dans l'image et de simplifier les traitements ultérieurs, nous utilisé la binarisation, le lissage, la normalisation, le cadrage, la squelettisation, la correction de la déformation des caractères et l'estimation de la ligne de base.

Le but de la binarisation est de faire surgir l'information utile par rapport à l'arrière-plan, malheureusement à cause de la mauvaise qualité de l'image reçue en entrée (Niveau du gris de l'arrière-plan très élevé) nous étions obligés de tester plusieurs seuils afin de trouver un compromis entre les différentes images utilisées.

L'opération de lissage est appliquée afin d'éliminer les bruits introduits dans l'image à cause des systèmes d'acquisition, les effets de temps ou tout simplement à cause de la qualité du papier et du stylo utilisé, en vue de la décrire par une séquence de vecteurs de caractéristiques plus au moins stables.

La normalisation est une tache nécessaire lorsque l'acquisition n'est pas réalisée avec un scanner relié au système (image existante). En effet si l'entrée du système est une image externe par rapport au système (Notre Système par exemple) il faux impérativement ramener les caractères a la même taille, car à cause de la variation des fontes ou des opérations d'agrandissement ou de réductions de la taille des images, les caractères peuvent subir une légère déformation dans la taille se qui complique les tâches de segmentation et influence sur la stabilité des paramètres.

L'opération de cadrage consiste à chercher la première et la dernière ligne / colonne signifiante (pixel <> de l'arrière-plan), ensuite créer une nouvelle image cadrée à partir de l'image mère.

La squelettisation est l'une des techniques les plus utilisées dans la reconnaissance des formes. Elle permet de diminuer l'information utile en ne gardant que le squelette de la forme. Le principe est de ramener l'image du mot à une écriture linéaire d'une épaisseur égale à un pixel, en préservant la forme, la connexité et la topologie du tracé.

Nous désirons de cette étape la correction de la déformation des caractères. En effet après la squelettisation du corps des caractères, plusieurs déformations de type concave et convexe ont apparu au niveau des lignes continues (supposées sans concave ni convexe). La fig.1 montre quelques déformations détectées après l'étape de squelettisation.

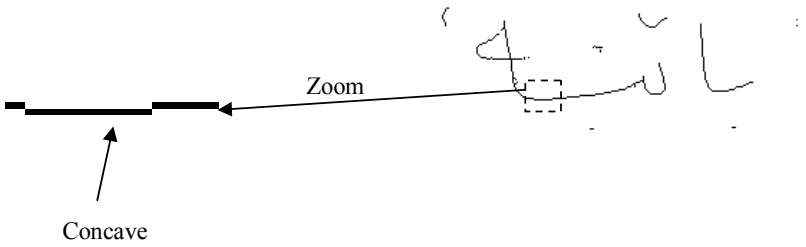


Fig. 1. Déformation du mot **" باتنة "** après la squelettisation
(Création de concave et convexe non désirée)

Pour remédier à ce problème, nous avons proposé un algorithme qui permet de redresser les lignes. Cet algorithme se base sur le principe de continuité c.-à-d. si deux segments horizontaux ou verticaux qui se trouvent sur la même ligne respectivement colonne et s'il existe un autre segment horizontal respectivement vertical et ce dernier contient des pixels voisins avec les deux segments alors c'est une déformation horizontale respectivement verticale. La correction s'effectue par le déplacement du segment (voir Fig. 2). L'algorithme peut se résumer dans les instructions suivantes :

Algorithme
Répéter
Pour chaque ligne / colonne faire
1) Détecter les bords du segment **A**
2) Détecter les bords du segment **B**
3) S'il existe un troisième segment **C** dont les pixels des bords sont voisins avec le premier et le deuxième segment alors :
   a. Créer un nouveau segment entre les segments **A** et **B**
   b. Supprimer le segment **C**

Fsi
Fpour
Jusqu'à « aucune modification n'est possible sur l'image »

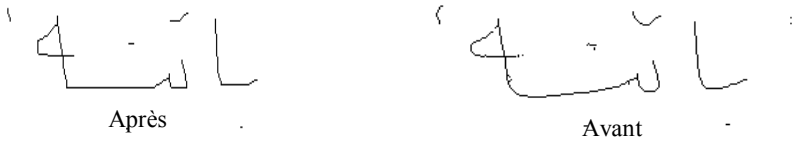


Fig.2. Résultats de l'opération de correction de la déformation sur le mot « **باتنة** »

L'extraction de certaines caractéristiques (c.-à-d. points diacritiques) demande l'estimation de la ligne de base d'écriture du mot. La méthode utilisée donne une bonne estimation de la ligne de base [6]. Elle est basée sur l'analyse de l'histogramme de projection horizontale.

3 Segmentation

Nous avons utilisé dans notre système une segmentation non uniforme basée sur l'analyse de l'histogramme de projection verticale. Le but de cette méthode est de diminuer la dimension de l'information contenue dans l'image. Cette technique se base sur le principe que la liaison entre deux caractères est la partie la plus mince du tracé manuscrit.

4 Extraction des caractéristiques

L'identification directe du mot à partir de son image (matrice de pixels) est presque impossible à cause de la grande variabilité inhérente au style d'écriture utilisé et au bruit entachant l'image. D'où la nécessité d'extraire, à partir de la représentation en pixels du mot, un ensemble de caractéristiques permettant d'identifier facilement ce dernier. Ces primitives doivent être discriminatives et invariantes vis-à-vis les différentes transformations que peut subir l'image telle que la rotation, variation de la taille…etc.

Nous avons utilisé dans notre système un mélange entre les caractéristiques statistiques et structurelles qui peut d'après la littérature donner des meilleurs résultats. Le choix des caractéristiques n'est pas une tâche simple, malheureusement il n'y a pas de théorie qui permet de choisir tel ou tel caractéristique. Après plusieurs teste-nous avons choisi les caractéristiques suivantes :

- Les sept moments invariants.
- Le nombre de boucles.
- Le nombre et le type de chaque concave (vers la Gauche, vers la Droite, vers le Haut et vers le Bas).
- Les hampes et les Jambes.
- Le nombre, le type et la position des points diacritiques (Hamza, Madda et chapeau inclus).
- Nombre de points extrêmes.

- Nombre de points de branchement.
- Nombre de points de croisement.
- Le nombre de composantes connexes

Le résultat de cette étape est un vecteur de caractéristiques comportant 26 caractéristiques statistiques et structurelles.

5 Classifieur MMC

Un modèle de Markov caché discret est un automate probabiliste à nombre d'états finis constitué de N états. C'est un processus aléatoire qui se déplace d'état en état à chaque instant, et on note q_t le numéro de l'état atteint par le processus à l'instant t.

L'état réel q_t du processus n'est pas directement observable, il est caché, mais peut être observé par un autre processus aléatoire qui émet après chaque changement d'état un symbole o_t. Dans le cas d'un processus markovien d'ordre 1, la probabilité de passer de l'état i à l'état j à l'instant t et d'émettre o_t ne dépend ni du temps, ni des états aux instants précédents [3].

La modélisation du classifieur nécessite la définition des observations émises par les états du modèle et l'architecture des modèles de mots [4].

Pour définir l'architecture du modèle, nous devons tenir compte de la topologie et le nombre d'états du modèle [5]. La topologie adoptée dans notre système est de type droite gauche conformément à l'écriture arabe avec saut inter états et intraétat. Ce type de modèles a l'avantage de conserver la notion du temps dans la modélisation, s'approchant ainsi de la nature de l'écriture. En outre, c'est le type le moins gourmand en temps de calcul et en nombre de paramètres à estimer lors de l'apprentissage. Dans notre modélisation chaque état correspond à un caractère c.-à-d. le nombre d'états est différent d'un modèle a un autre, ce qui nécessite la création de 48 modèles représentant les 48 wilayas algériennes.

Nous avons considéré les caractéristiques statistiques et structurelles extraites à partir de l'image de mot comme des observations pour notre modèle. La quantification vectorielle garantit la transformation du vecteur de caractéristiques en une séquence discrète d'observation.

Quantification vectorielle

Les MMCs utilisés pour la modélisation des mots sont de nature discrète, leurs densités de probabilités d'observations sont discrètes, ce qui nécessite l'utilisation d'un quantifieur vectoriel pour faire correspondre chaque vecteur continu à un indice discret d'un dictionnaire de référence (CodeBook). Une fois le dictionnaire de référence obtenu, cette correspondance entre les vecteurs caractéristiques des trames et les indices du dictionnaire deviennent un simple calcul de type plus proche voisin.

Apprentissage et reconnaissance

L'apprentissage des paramètres des modèles MMCs (A, B, Π) correspondant aux classes de mots est réalisé par l'algorithme de Baum-Welch [5] en appliquant les formules de Russel [7] et Levinson [8] pour estimer les paramètres des différentes

distributions de probabilité d'état. Cet algorithme permet d'aligner les observations sur les états, et d'une façon générale, converge vers un minimum local du fait de manque de données [9]. La reconnaissance est effectuée par la recherche du modèle discriminant, elle peut se faire simplement par le calcul des probabilités d'émission de la forme par les modèles que l'on suppose a priori équiprobables. La forme à reconnaître est affectée à la classe dont le modèle fournit la probabilité la plus importante.

$$\lambda^* = \text{argmax}_{\lambda \in A} \; P(O/\lambda) \quad \textbf{(1)}$$

Où A désigne l'ensemble des modèles. L'évaluation de la probabilité de chaque modèle est réalisée grâce à une méthode à base de programmation dynamique qui est l'algorithme de Viterbi [10].

6 Classifieur RN

Le classifieur que nous avons utilisé est un PMC à rétro-propagation du gradient d'erreur à une couche cachée [11]- [12]. Les 26 caractéristiques statistiques et structurelles extraites à partir des segments de caractère isolés générés par le module de segmentation sont les entrées du réseau. La couche cachée est composée de 25 neurones. Les classes à discriminer sont les 28 caractères de l'alphabet arabe, d'où le choix de 28 neurones pour la couche de sortie. La fonction d'activation des neurones est la fonction sigmoïde unipolaire. Ce choix de classifieur est basé sur les critères suivant : sa rapidité, sa capacité à traiter des données hétérogènes et le plus important, s'il est bien entraîné, un PMC estime des probabilités bayésiennes a posteriori [14] [15]. Ce dernier point est très important dans notre modélisation vu que nous avons utilisé les modèles de Markov cachés et que ce dernier génère aussi des probabilités a posteriori. Donc la mise en œuvre de la combinaison des résultats va être simplifiée.

7 Combinaison des résultats

Nous voulons de cette étape la correction des inconvenances générées par les modèles de Markov cachés. En effet, malgré le progrès réalisé au niveau de la reconnaissance de la parole et de l'écriture manuscrite par les MMCs, nous les reprochons de négliger un peu les informations locales. De plus, la condition d'indépendance imposée par le modèle de Markov (chaque observation doit être indépendante des observations voisines) rend les MMC incapable de tirer avantage de la corrélation qui existe réellement parmi les observations d'un même caractère, mais le faible pouvoir discriminatif reste l'inconvénient majeur des MMCs.

Pour effectuer la combinaison des résultats, nous avons calculé un nouveau score des mots à partir des probabilités a posteriori de chaque classifieur. En effet étant donné que les deux classificateurs estiment des probabilités a posteriori en sortie,

nous pouvons calculer un score composé P^* par combinaison des sorties des classificateurs.

$$P^* = \log(P_{MMC}) + \log(P_{PMC}) \quad \textbf{(2)}$$

La sortie du système (complet) une liste des N meilleures hypothèses. Le choix du mot candidat est effectué par rapport à la probabilité la plus élevée.

8 Tests et résultats

Pour construire notre système et évaluer ces performances, nous avons utilisé deux bases de données. La première contient 14400 échantillons écrits sur du papier par 100 scripteurs différents et 3 occurrences pour chaque mot. Les mots de cette base constituent un lexique de 48 mots des wilayas algériennes. Les échantillons de la base de données utilisée ont été collectés au sein du laboratoire de recherche en informatique d'Annaba. La deuxième contient 2800 caractères segmentés manuellement à partir des mots de la première base. Le choix des mots à segmenter est pris au hasard parmi les 14400 mots contenus dans la première base. Nous avons pris de chaque caractère 100 exemplaires qui n'appartiennent pas forcement au même scripteur. Nous avons construit la deuxième base pour éviter l'apprentissage collectif.

La base des caractères a permis simplement de réaliser l'apprentissage du perceptron multicouche, tandis que la base des mots a servi pour l'apprentissage des modèles de Markov cachés et pour le test des performances du système. Nous avons divisé la base de mots en deux sous base, la première contient 75 % (10800 échantillons) des mots pour l'opération d'apprentissage et la deuxième contient 25 % (3600 échantillons) des mots pour les tests [16].

Nous allons dans ce qui suit montrer seulement les résultats finaux de la reconnaissance des mots manuscrits arabes de notre système c.-à-d. avec toutes les corrections effectuées au niveau des modules du système (version finale du système). La meilleure configuration de notre système a donné les résultats illustrés dans le Tableau 1.

Tableau 1 Taux de reconnaissance sur les ensembles d'apprentissage et test.

	Taux de reconnaissance (%)	Taux de rejet (%)
Base d'apprentissage	100 %	0 %
Base de Test	91.77 %	8.23 %

9 Conclusion

Nous avons présenté dans ce chapitre notre modélisation pour un système de reconnaissance de l'écriture manuscrite arabe à vocabulaire limité. Notre approche est basée sur l'hybridation des deux classifieurs les plus utilisés dans le domaine de la reconnaissance des formes et en particulier la reconnaissance de l'écriture et de la parole.

Malgré un taux de réussite de 91,77 % que nous estimons encourageant, notre système est loin d'atteindre la perfection. Les erreurs de classification sont dues d'une part à l'utilisation des MMCs discrets. En effet l'utilisation de la quantification vectorielle permet de représenter chaque classe par un vecteur unique (centroïde) ce qui influence sur la qualité de l'information véhiculée dans chaque vecteur. D'autre part, la qualité des images de la base est très médiocre ; la numérisation des images est une étape extrêmement importante dont il faut prendre le soin de le faire.

Références

1. Essoukhri Ben Amara, N. : Problématique et orientations en reconnaissance de l'écriture arabe, CIFED'2002, Colloque International Francophone sur l'Ecrit et le Document, pp.1-10, Hammamet, Tunisie, Octobre 2002.
2. Essoukhri Ben Amara, N., Belaïd, A., Ellouze, N.: Utilisation des modèles markoviens en reconnaissance de l'écriture arabe: Etat de l'art, CIFED'2000, Colloque International Francophone sur l'Ecrit et le Document, pp.181-191, Lyon, France, 2000.
3. Bourlard, H., Morgan, N.: Continuous speech recognition by connectionist statistical methods. IEEE Trans. on Neural Networks, 1993.
4. Benzenache, A.: Reconnaissance hors-ligne des mots arabes manuscrits par les modèles de Markov cachés, Mémoire de magister, Département Génie Electrique, Université 08 mai 45, Guelma, Algérie, 2007.
5. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE Vol. 77, No. 4, pp. 336-349, August 1989.
6. Pechwitz, M., Maergner, V.: HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database, ICDAR (Proceedings of the Seventh International Conference on Document Analysis and Recognition), pp: 890-894, 2003.
7. Russel, M. J., Moore, R. K.: Explict Modeling of State Occupancy in Hidden Markov Models for Speech Recognition, Proceeding of ICASSP (International Conference on Acoustic, Speech and Signal Processing), pp: 5-8, 1985.
8. Levinson, S.E: Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition. Computer, Speech & Language, Vol 1, N° 1, pp: 29-45, 1986.
9. Saon, G. : Modèles Markoviens uni-bidimensionnels pour la reconnaissance de l'écriture manuscrite hors-ligne, Thèse de doctorat, Université Henri Poincaré Nancy 1, 1998.
10. Forney, G.D. : The Viterbi Algorithm, Proc IEEE, Vol. 61, No. 3, pp. 268- 278, March 1973.
11. Koerich, A.: Large vocabulary off-line handwritten word rocognition, Thèse de Docteur Ecole de Technologie Supérieur, 2002.
12. Cheriet, M., Suen, C.Y.: Un système neuro-flou pour la reconnaissance de montants numériques de cheques arabes, Pattern Recognition letters 14(1993), pp, 1009-1017.
13. Davalo, E., Naim, P. : Des réseaux de neurones, Eyrolles.1993

14. Richard, M.D., lippmann, R.P.: Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities, Neural Computation, vol. 3, 1991, p. 461–483.
15. Farah, N., Souici, L., Sellami, M.: Arabic Word Recognition by Classifiers and Context, JCST, Journal of Computer Science and Technology, Vol. 20, No. 3, pp. 402-410, May 2005.
16. Pechwitz, M., Maergner, V.: HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database, ICDAR (Proceedings of the Seventh International Conference on Document Analysis and Recognition), pp : 890-894, 2003.

Optimisation à base de flot de graphe pour la mise en correspondance d'images stéréoscopiques

Sid Ahmed Fezza[1], Nacera Benamrane[1], Habib Zaidi[2]

[1] Groupe Vision et Imagerie Médicale, Laboratoire SIMPA, Département d'Informatique, Faculté des Sciences, Université des Sciences et de la Technologie d'Oran "Mohamed Boudiaf", B.P 1505 El ' Mnaouer 31000, Oran, Algérie
[2] Division of Nuclear Medicine Geneva University Hospital CH-1211 Geneva 4, Switzerland
sidahmed.fezza@gmail.com, nabenamrane@yahoo.com, habib.zaidi@hcuge.ch

Résumé. Dans ce papier, on propose une approche pour résoudre le problème de la mise en correspondance des images stéréoscopiques. Parmi les algorithmes récents d'appariement proposés dans la littérature, on trouve ceux qui sont basés sur la coupure de graphe, ils transforment le problème de la mise en correspondance stéréoscopique en un problème de minimisation d'une fonction d'énergie globale. L'idée est de montrer comment on peut appliquer la technique de « flot de graphe » pour résoudre ce problème de minimisation. Plusieurs méthodes ont été proposées pour construire ce graphe, mais toutes considèrent en chaque pixel, toutes les hypothèses possibles pour la disparité. Notre contribution consiste à construire un graphe réduit en gardant seulement quelques valeurs potentielles de la disparité pour chaque pixel. Ces valeurs potentielles sont trouvées par une approche locale d'appariement. Ce graphe réduit permet d'accélérer considérablement le temps de calcul de cette méthode globale, en même temps d'améliorer sensiblement la qualité des images de disparité en préservant les discontinuités sur les contours présents dans les images.

Mots-clés: Stéréovision, mise en correspondance, coupure de graphe, minimisation d'énergie, flot de graphe.

1 Introduction

La vision stéréoscopique binoculaire est une méthode importante pour la reconstruction 3D d'une scène. Cette méthode nécessite trois étapes fondamentales pour trouver le relief de la scène: en premier *le calibrage* qui consiste à trouver les paramètres des capteurs, ensuite vient l'étape de *la mise en correspondance* ou *appariement* dont le but est de trouver les points homologues entre les deux images et produit comme résultat une carte de disparité qui représente l'ensemble des correspondances entre les deux images. Chaque pixel d'une carte de disparité représente l'amplitude de la disparité, c'est-à-dire, la distance entre la position du pixel de l'image gauche et celle de son correspondant dans l'image droite. La dernière étape

est *la reconstruction 3D* qui à partir des paramètres des capteurs et des correspondances de points reconstruit un modèle 3D de la scène.

De manière générale, la mise en correspondance stéréoscopique consiste à retrouver dans les images gauche et droite, les primitives homologues, c'est-à-dire, les primitives qui sont la projection de la même entité de la scène. Ces primitives peuvent être les *pixels* de l'image ou des points d'intérêt (comme les points de Harris), dans ce cas on parle de *stéréo dense* (*pixel-based stereo*). Elles peuvent être aussi des primitives *structurées* tels que des segments, des contours…, dans ce cas on parle de *stéréo éparse* (*feature-based matching*). Dans le cadre de notre travail, les primitives que nous considérons sont les pixels des images.

La mise en correspondance s'avère être une tâche délicate dont la qualité du résultat détermine directement la qualité de la reconstruction 3D. Elle rencontre de nombreuses difficultés, notamment en présence de changements de luminosité entre les deux images, d'objets dont la surface est uniforme, d'occultations et de bruit dans les images. En raison de ces difficultés, les méthodes de mise en correspondance sont amenées à exploiter toutes les informations disponibles afin de faciliter la recherche et la détermination des correspondants. On a opté pour la méthode de coupure de graphe (*Graph Cuts*) qui a été introduite pour la première fois par S. Roy [2] pour résoudre le problème de la mise en correspondance en stéréovision.

Dans la section 2 nous allons observer les principaux travaux existants sur le problème de mise en correspondance à base de coupure de graphe. En section 3 nous décrivons comment est transformé le problème de la mise en correspondance stéréoscopique en un problème de minimisation d'une fonction d'énergie globale. Nous décrivons dans la section 4 notre approche et les deux méthodes que nous avons évaluées: coupure sur un graphe complet et coupure sur un graphe de taille réduite. Nous finirons par la section 5 où l'on présente notre implémentation et quelques résultats obtenus et par la section 6 qui conclue cet article.

2 Travaux existants

Si certaines méthodes utilisant la programmation dynamique tentent d'imposer une certaine régularité dans la fonction de disparité dans toutes les directions, la plupart n'exploitent pas totalement la cohérence bidimensionnelle. Ce défaut vient du fait que la majorité des méthodes utilisant la programmation dynamique apparient les pixels appartenant à la même ligne épipolaire sur les deux images sans prendre en compte une éventuelle continuité de l'image tridimensionnelle à reconstruire. La théorie des graphes permet de généraliser cette technique en deux dimensions. Roy et Cox [9] sont les premiers a appliqué une méthode globale fondée sur la coupure de graphe pour résoudre la mise en correspondance en stéréovision, ils ont introduit leur technique comme une généralisation des méthodes travaillant sur les lignes épipolaires pour construire des cartes de disparités. S.Roy a proposé une contrainte de cohérence locale qui suggère que la fonction de disparité est localement continue, ce qui veut dire que les pixels proches dans toutes les directions ont des disparités similaires. Roy intègre cette contrainte de cohérence locale avec une contrainte de ressemblance qui dépend de la variation des intensités des pixels appariés.

L'étape suivante de la méthode proposée par Roy et Cox, est de résoudre la fonction de disparité optimale sur toute l'image. Ceci peut être visualisé comme une maille 3D composée de plans, eux-mêmes constitués d'une image de nœuds. Il existe un plan pour chaque niveau de disparité, et chaque nœud représente un appariement entre deux pixels dans les images originales comme il est illustré à la figure 1(a).

Le maillage 3D est composé de points (*i, j, d*), où (*i,* j) sont les coordonnées des pixels d'une des images et *d* les disparités possibles, d'une ***source s*** et d'un ***puits t***. En interne, chaque point est connecté avec ses quatre voisins dans le plan par des arêtes dites de voisinage (appelé *n-link*), et avec les deux points dans les plans voisins avec des arêtes dites de disparité (appelé *t-link*). Roy et Cox cherchent dans ce graphe une coupe minimum séparant la source du puits. La carte de profondeur est construite à partir de la coupe minimum en associant pour chaque point (*i, j*) la plus grande disparité de l'arête associée le long de la coupe minimum choisie, comme il est illustré à la figure 1(a et b).

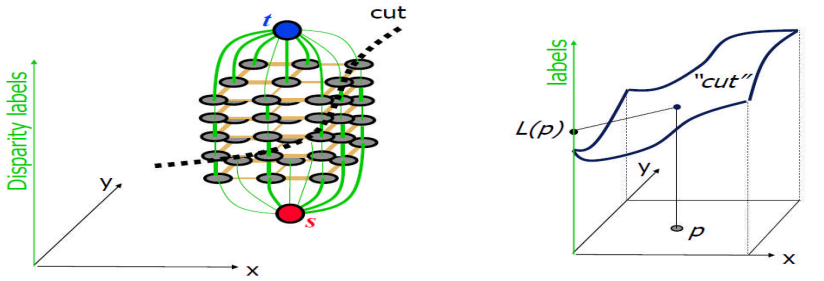


Fig. 1. (a) Le maillage 3D comme proposée par Roy et Cox, (b) la coupure minimale qui représente la surface de profondeur.

La technique de Roy et Cox a été par la suite formalisée par Veksler [8] puis Kolmogorov et Zabih [10, 11, 12] sous la forme d'un problème de labellisation visant à construire une carte de disparité. Chaque label est une valeur de disparité possible. Un label est associé à chaque pixel de manière à minimiser une énergie qui tient compte du voisinage du pixel pour avoir un résultat final régulier. Cette approche a un spectre d'application beaucoup plus large que celle proposée par S.Roy. En revanche, ces méthodes ont deux limitations, la première est qu'elles construisent des cartes de disparité qui ont toujours le défaut d'aplatir les objets ; la seconde est que, comme la fonction de pénalité entre deux voisins de labels différents n'est pas nécessairement convexe, la minimisation de l'énergie est un problème NP-complet dont on obtient finalement qu'une approximation. Par contre, Ishikawa [4] propose l'étude du cas où cette fonction est convexe et propose une méthode générale pour obtenir une fonction de pénalité convexe à une constante près sur la pénalité, ce qui peut être gênant dans le cas d'un échantillonnage irrégulier.

3 Minimisation d'énergie dans la vision

Plusieurs problèmes en vision peuvent être formulés comme un problème d'étiquetage (*Labeling Problem*). Dans un tel problème, on distingue un ensemble de sites et un ensemble d'étiquettes. Les sites représentent les primitives de l'image (*image features*), pour lesquels on veut estimer une quantité: des pixels, des segments…etc. Les étiquettes représentent les quantités associables à ces sites: l'intensité, la disparité, un numéro de région…etc.

Soit $P = 1, 2,\ldots, n$ un ensemble de n sites, et soit $L = \{l_1, ..., l_k\}$ un ensemble de k étiquettes. L'étiquetage est défini par une application de P dans L

$$f: \; P \rightarrow L$$
$$s_p \rightarrow f_p = f(s_p) = l_i$$

Donc $f(P) = \{f_1,\ldots,f_n\}$

On affecte une fonction d'énergie à cette fonction d'étiquetage, une forme générale des fonctions d'énergies est donnée par:

$$E(f) = E_{data}(f) + \lambda E_{prior}(f) \qquad \textbf{(1)}$$

Le premier terme $E_{data}(f)$ représente l'énergie intrinsèque aux données (*data energy*), qui traduit les contraintes de l'association des étiquettes aux données. Le deuxième terme $E_{prior}(f)$ regroupe les énergies extrinsèques (*prior energy*) qui traduisent les contraintes définies par des connaissances a priori. La constante λ peut contrôler l'importance relative des deux termes ; plus λ est grand plus on donne de l'importance aux informations a priori. La fonction d'énergie $E_{data}(f)$ doit être choisie pour affecter un coût important aux associations donnée/étiquette qui sont les moins pertinentes:

$$E_{data}(f) = \sum_{p \in P} D_p(f_p) \qquad \textbf{(2)}$$

où $D_p(f_p) \geq 0$ mesure la qualité de l'associations de l'étiquette f_p avec le site p.

La fonction d'énergie a priori $E_{prior}(f)$ doit attribuer un coût important aux associations f_p non compatibles avec l'information a priori. Le choix de cette fonction dépend du type de problème, mais en général, une des fonctions d'énergie a priori exprime des contraintes de lissage (*smoothing*). Cette contrainte est très connue en vision par ordinateur, et elle est bien adaptée quand la qualité à estimer varie lentement partout où presque partout: en 3D, cela correspond à l'hypothèse que le monde est continu par morceaux. Une telle hypothèse est prise en compte en introduisant une énergie d'information a priori de type lissage E_{smooth}.

Pour formuler l'énergie de lissage, on a besoin de modéliser comment les pixels interagissent entre eux: souvent il est suffisant d'exprimer comment un pixel interagit avec ses voisins. Notons N_p l'ensemble des pixels voisins du pixel p, et N l'ensemble de paires voisines $\{p, q\}$: N est dit un système de voisinage. L'énergie de lissage peut s'écrire:

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) \quad (3)$$

où $V_{\{p,q\}}(f_p, f_q)$ est une fonction d'interaction de voisinage: cette fonction doit attribuer des pénalités aux paires $\{p, q\}$ si les pixels p et q ont des étiquettes différentes.

La forme de $V_{\{p,q\}}(f_p, f_q)$ détermine le type de lissage a priori. Avec ces notations, l'énergie de lissage globale est la somme des fonctions d'interaction de voisinage de tous les pixels voisins. Souvent on choisit d'écrire $V_{\{p,q\}}(f_p, f_q)$ sous la forme:

$$V_{\{p,q\}}(f_p, f_q) = u_{\{p,q\}} V(f_p, f_q) \quad , \quad u_{\{p,q\}} \in \mathrm{R}^+ \quad (4)$$

où V est un potentiel homogène, et $u_{\{p,q\}}$ est un terme multiplicateur dépendant de la clique considérée. Comme potentiel linéaire on a choisi ce potentiel, $V(f_p, f_q) = |f_p - f_q|$. En stéréovision le terme $u_{\{p,q\}}$ est souvent une fonction décroissante de la norme du gradient entre les sites p et q, ce qui permet de favoriser la coïncidence des discontinuités de la disparité avec les contours de l'image de référence. Ce choix est traduit par la fonction $u_{\{p,q\}}$: $u_{\{p,q\}} = U(|I_p - I_q|)$. Le terme $u_{\{p,q\}}$ représente la pénalité d'affecter des disparités différentes pour les pixels voisins p et q ; la valeur de cette pénalité doit être petite pour une paire $\{p, q\}$ qui est sur un contour, donc avec une large différence d'intensité $|I_p - I_q|$. En pratique, on utilise une fonction empirique décroissante.

4 Fonction d'énergie et coupure de graphe

4.1 Coupure sur un graphe complet

En reprenant la formulation générale des problèmes d'étiquetage (eq:1) et en considérant le potentiel linéaire décrit précédemment, on obtient l'énergie globale suivante:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{\{p,q\} \in N} u_{\{p,q\}} |f_p - f_q| \quad (5)$$

Nous montrons ci-dessous que l'on peut déterminer le minimum global d'une telle fonction en résolvant le problème de la détermination de la coupure minimale dans un graphe. Cette approche est due à S.Roy [2], mais elle a profité de la reformulation proposée par Olga Veksler [8].

Rappelons qu'un graphe est une paire (V, E), où V est l'ensemble de sites et E est l'ensemble des arêtes entre les sites $E = \{(u, v) | u, v \in V\}$.

On considère le graphe pondéré $G = (V, E)$ où V contient deux nœuds particuliers: une source s et un puits t. Soit k le nombre d'appariements possibles (en stéréovision, k est donné par la plage admissible de disparités, liée à la profondeur minimale et maximale à laquelle se trouvent des objets dans la scène perçue). À chaque pixel p on associe une chaîne de nœuds $p_1, p_2, \ldots, p_{k-1}$. Ces nœuds sont connectés par des arêtes

appelées *t-link* et notées t_1^p , t_2^p ,..., t_k^p avec $t_1^p = [s,\ p_1]$, $t_j^p = [p_{j-1},\ p_j]$ et $t_k^p = [p_{k-1},\ t]$. À chaque t_j^p on affecte une capacité $K_p + D_p(l_j)$ où D_p est le coût d'appariement du pixel considéré pour la valeur de disparité correspondante, généralement ce coût est égal à $(I_p - I_{(p+j)})^2$ et K_p est une constante qui satisfait la contrainte 6. À chaque paire de pixels voisins p et q les chaînes correspondantes sont reliées par des arêtes appelées *n-link*, au niveau $j \in \{1, 2,\ldots, k-1\}$: le *n-link* $\{p_j, q_j\}$ est de capacité $u_{\{p,q\}}$.

$$K_p > (K-1) \sum_{q \in N_p} u_{\{p,q\}} \qquad \textbf{(6)}$$

La capacité d'une coupure *s-t* de ce graphe est la somme des capacités de toutes les arêtes coupées. Vue la méthode de construction du graphe, la capacité de la coupure est constituée de deux parties : la première est la somme des capacités des arêtes *t-link*, et la deuxième est la somme des capacités des arêtes *n-link*. En effet l'addition de la constante K_p sert à assurer l'unicité de la coupure de chaque *t-link*, voir [2] pour une preuve de cette unicité.

Une autre méthode pour assurer l'unicité de la coupure de chaque *t-link* est proposée par Ishikawa et Geiger [4]. Leur graphe est très proche mais il est orienté et ils ajoutent à la chaîne des sites associés au pixel *p*, une chaîne inverse mais à capacité infinie.

Une coupure de graphe consiste à diviser le graphe en deux parties. Les *t-link* coupés forment la surface de profondeur recherchée (figure 1(b)), ce qui permet d'associer une disparité à chaque pixel. Le problème de coupure de graphe peut se résoudre par le flot maximum. Ford et Fulkerson [6] ont montré que le flot de la source *s* vers le puits *t* fait saturer un ensemble d'arêtes divisant les sites en deux parties *S* et *T*. Le problème majeur de cette méthode est son temps de calcul énorme, c'est pour cela qu'on a opté pour une nouvelle méthode basée sur un graphe réduit qu'on construit, comme nous le montrons dans la section suivante.

4.2 Coupure sur un graphe réduit

Pour résoudre le problème du temps de calcul qui est énorme, Zureiki et al. [3] propose de construire un graphe réduit: pour chaque pixel on garde dans le graphe seulement quelques disparités potentielles, issues d'une méthode locale de mise en correspondance, alors que la méthode générale proposée notamment par S.Roy, considère toutes les valeurs possibles de disparité. Partant du graphe complet et pour chaque *t-link*, on supprime tous les nœuds sauf les *N* choisis. La figure 2 détaille la construction du graphe réduit. Les arêtes pointillées sont les arêtes du graphe complet. Les nœuds non supprimés sont en rouge, les nouveaux *t-links* sont en bleu et les nouveaux *n-links* sont en vert. La figure 3(a) illustre une projection frontale du graphe réduit. Par une méthode d'appariement local (stéréo corrélation), on calcule pour chaque pixel, les coûts de mise en correspondance du pixel *p* avec toutes les valeurs possibles de disparité dans la plage de disparité $[d_{min}, d_{max}]$. Puis on choisit les *N* meilleures valeurs (pour notre exemple, on va considérer $N = 4$ sans manquer de généralité). Ce choix peut être fait selon différents critère, par exemple, avec un score

ZNCC très classique en stéréo, on peut garder les valeurs de disparité autour du sommet, ou les N (s'ils existent) maximaux locaux. On va noter ces quatre disparités choisies pour le pixel p, comme $d_{1,p}$, $d_{2,p}$, $d_{3,p}$, $d_{4,p}$, et les coûts correspondants par $D_{\{p,d_{1,p}\}}$, $D_{\{p,d_{2,p}\}}$, $D_{\{p,d_{3,p}\}}$, $D_{\{p,d_{4,p}\}}$.

Pour réduire la taille du graphe, nous faisons la simplification suivante: au lieu de garder pour chaque pixel une chaîne (*t-link*) à k +1 arêtes, on supprime toutes les arêtes sauf (à titre illustratif) N = 4 arêtes. Ainsi, à chaque pixel p on associe N -1 niveaux (pour l'exemple c'est 3). À chaque *t-link* on affecte une capacité $C + D_{\{p,d_{i,p}\}}$, où C est une constante qui satisfait la contrainte suivante:

$$C > N * \max_{\{p,q\} \in N} (u_{\{p,q\}}) * |d_{max} - d_{min}| \quad (7)$$

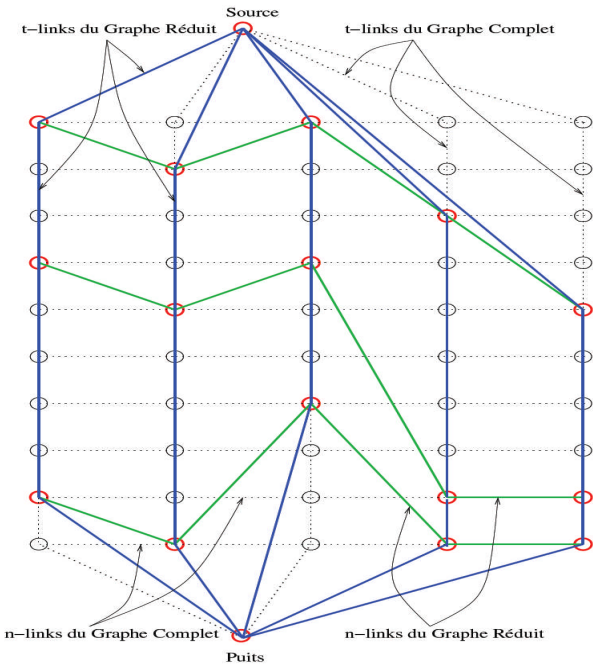


Fig. 2. Construction du graphe réduit [3].

À chaque paire de pixels voisins p et q, les chaînes correspondantes sont reliées par des arêtes (*n-link*) aux ($N-1$) niveaux, et de capacité selon l'équation (8).

Capacité de *n-link* au niveau i =

$$u_{\{p,q\}} * (|d_{i,p} - d_{i,q}| + 1) \quad (8)$$

4.3 L'approche proposée

L'approche proposée par Zureiki et al. [3] ne permet pas de traduire la coupure du graphe réduit par une fonction d'énergie analytique et elle ne respecte pas la contrainte de lissage exprimée par le deuxième terme de l'énergie globale (eq: 5), car la capacité des *n-links* n'est pas cohérente avec la formulation de cette énergie globale.

La structure du graphe utilisée dans [3] est à l'origine de ces deux faiblesses. Pour illustrer cela observons l'exemple de la figure 3(a) où la coupure (représentée par un trait pointillé de couleur violette) coupe trois *t-links* (les trois arêtes verticales). Maintenant concentrons nous sur les deux premiers *t-links* (celui qui est le plus à gauche et celui du milieu), ces deux *t-links* représentent des niveaux de disparité différents. Contrairement au graphe complet, dans le graphe réduit les arêtes verticales de même abscisse ne réfèrent pas forcément la même valeur de disparité, car la disparité n'est plus calculée à travers l'indice (*zero-based index*) de l'arête coupée mais plus tôt il y a une correspondance directe entre le coût et la valeur de disparité, comme il a été expliqué dans la section précédente.

Malgré cette différence de disparité il n'y a aucune pénalité qui a été affectée à la coupure, c'est-à-dire que cette structure de graphe ne pénalise pas correctement les variations de disparité entre des pixels voisins. Nous proposons par conséquent une nouvelle structure de graphe introduisant une pénalité qui prend en considération ce cas.

La nouvelle structure du graphe est construite comme ceci: on construit un graphe tel qu'indiqué sur la figure 3(b). La base du graphe est une grille dont les axes verticaux représentent chaque pixel p et les axes horizontaux chaque niveau de disparité possible d_i, par exemple pour $N = 4$ on a quatre niveaux. A chaque position (p, d_i) on associe une arête verticale avec la capacité $C + D_{\{p,d_{i,p}\}}$, où C est une constante qui satisfait la contrainte (7). Les arêtes horizontales correspondent à la fonction de pénalité (notons qu'elle diffère de l'équation (8) dans les indices):

$$u_{\{p,q\}} * (|d_{i,p} - d_{i+1,q}| + 1) \qquad (9)$$

ces arêtes horizontales sont appelées *n-link.*

On ajoute à cela des nœuds auxiliaires (couleur orange) entre les nœuds principaux (couleur noire), et des arêtes auxiliaires (couleur rouge) appelées *a-link* qui relient certains nœuds principaux aux nœuds auxiliaires. Ces arêtes auxiliaires sont ajoutées afin de répondre aux problèmes survenus par l'utilisation du graphe dans [3]. On affecte à ces arêtes auxiliaires une capacité selon l'équation (8). Les arêtes qui relient la source à la grille et la grille au puits ont une capacité infinie.

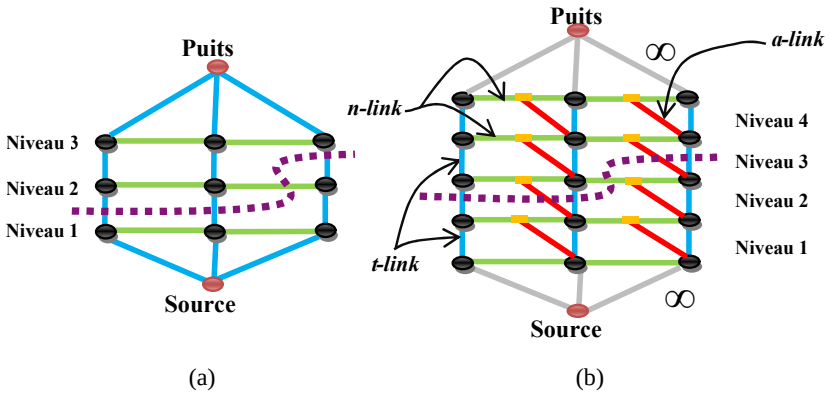


Fig. 3. (a) Graphe réduit selon [3] avec un exemple de coupure, (b) notre graphe réduit avec un exemple de coupure.

Donc si vous remarquez bien, dans la figure 3(b) la coupure est obligée de couper les arêtes de pénalité (*n-links* et *a-links*) dont le coût est proportionnel à la différence des niveaux de disparité entre les pixels voisins.

On dit qu'une coupure est *potentiellement minimale*, lorsque la coupure satisfait la remarque suivante : elle ne coupe aucune arête infinie et ne coupe qu'une et une seule arête d'abscisse p_i.

L'énergie totale minimisée par la coupure de notre nouveau graphe réduit peut être écrite:

$$E(f) = \sum_{\substack{p \in P \\ f_p \in \{d_{p,1},\ldots,d_{p,N}\}}} D_p(f_p) + \lambda \sum_{\substack{\{p,q\} \in N \\ f_p \in \{d_{p,1},\ldots,d_{p,N}\} \\ f_q \in \{d_{q,1},\ldots,d_{q,N}\}}} u_{\{p,q\}} |f_p - f_q| \qquad \textbf{(10)}$$

5 Implémentation et résultats expérimentaux

Dans la littérature on trouve deux approches pour résoudre le problème de flot maximal [5]. La première est l'algorithme du *chemin augmentant* dû à Ford et Fulkerson [6], et la deuxième est de type *Push-Relabel*. On a choisi une implémentation du type *chemin augmentant* et plus précisément un nouvel algorithme proposé par Boykov et Kolmogorov [7], ils ont montré que sur des graphes typiques pour la vision, leur algorithme est 2 à 5 fois plus rapide que les autres algorithmes tel que *Push-Relabel.*

Pour comparer l'importance de notre méthode exploitant un graphe réduit, soit W et H la largeur et la hauteur de l'image, [0, d_{max}] la plage de disparités, N le nombre des meilleurs candidats donnés par une méthode locale de mise en correspondance. Soit v le nombre des nœuds et e le nombre des arêtes. Dans le graphe original, on a:

$v = W * H * (d_{max} - 1) + 2$, $e \simeq 6 * v = 6 * W * H * (d_{max} - 1)$. La complexité théorique de l'algorithme du *chemin augmentant* est de $O(e*|MaxFlot|)$ [5], où *MaxFlot* est la valeur du flot maximal. Dans le cas du graphe complet, la complexité est de l'ordre $O(6v*|MaxFlot|)$, et puisque dans le graphe réduit on a supprimé un nombre important d'arêtes et de nœuds ceci aura comme conséquence une valeur de complexité beaucoup plus inferieure à celle obtenue avec le graphe complet. Rappelons nous une chose est que les algorithmes de coupure de graphe sont proportionnels (d'ordre 2 ou 3) en nombre de nœuds et d'arêtes, le temps de calcul pour trouver une coupure minimale dans le graphe réduit est très inferieur au temps pour trouver cette coupure dans le graphe complet. Le tableau 1 donne des résultats expérimentaux en temps de calcul. Malgré que notre structure contienne plus de nombre de nœuds et d'arêtes par rapport à [3], nos temps de calcul sont très proches. C'est principalement dû à l'algorithme [7] qu'on a utilisé. Contrairement à [3] où ils ont utilisé l'algorithme de *Push-Relabel*. Mais on a obtenu une nette amélioration en terme de qualité d'image de disparité, ceci est illustré par la figure 4.

Tableau 1. Temps de calcul en secondes pour la mise en correspondance par coupure de graphe complet et réduit.

Taille de l'image	Graphe Complet	Notre Graphe Réduit		Graphe Réduit de [3]	
		N= 4	N= 5	N= 4	N= 5
Tsukuba 384∗288	75 s [10]	15 s	30 s	/	/
Sawtooth 434∗380	>130 s	16 s	48 s	15 s	50 s

La figure 4 illustre des résultats obtenus sur deux images *Tsukuba* et *Sawtooth* issues de la base d'images de [1] mise à disposition sur leur site http://cat.middlebury.edu/stereo/. Les cartes de disparité (vérité terrain) sont aussi tirées de [1]. Notons que la qualité des images de disparité obtenue avec notre approche est supérieure à celle obtenue dans [3] (on n'a pas pu comparer avec l'image *Tsukuba* car les auteurs n'ont présenté leurs résultats qu'avec l'image *Sawtooth*).

Pour *Tsukuba* la taille est de 384∗288 et 16 niveaux de disparité, et pour *Sawtooth*, la taille est de 434∗380 et 20 niveaux de disparité. Le test a été effectué sur un P4 à 3.2 GHz avec 512 Mo de RAM.

Le tableau 2 présente des résultats quantitatifs (comme cela est fait dans [1]). On note l'absence des résultats de [3] qui n'ont pas présenté des résultats quantitatifs dans leur article.

Tableau 2. Erreurs statistiques.

Méthode	*% d'erreurs totales*	*% d'erreurs>±1*
Notre méthode	6.9	1.4
Graphe Complet [8]	10.1	5.9
Graphe Complet [10]	6.7	1.9
Corrélation	28.5	12.8

6 Conclusion

Nous venons de décrire dans cet article, une amélioration d'une méthode fondée sur la coupure de graphe. La combinaison d'une méthode locale, capable de sélectionner pour chaque pixel gauche, un ensemble réduit de correspondants possibles dans l'image droite a permis d'éviter l'explosion combinatoire des méthodes de *graph cuts* exécutées sans élagage préalable du graphe. Notre contribution a consisté à développer un graphe original pour mieux respecter la contrainte de lissage, et permettre de traduire la coupure du graphe réduit par une fonction d'énergie analytique.

Nous pensons que la méthode peut encore être améliorée pour prendre en compte des disparités *sous-pixelliques*, et par la suite construire des cartes de disparité réelles. Nous souhaitons aussi adapter cette méthode globale pour traiter les occultations.

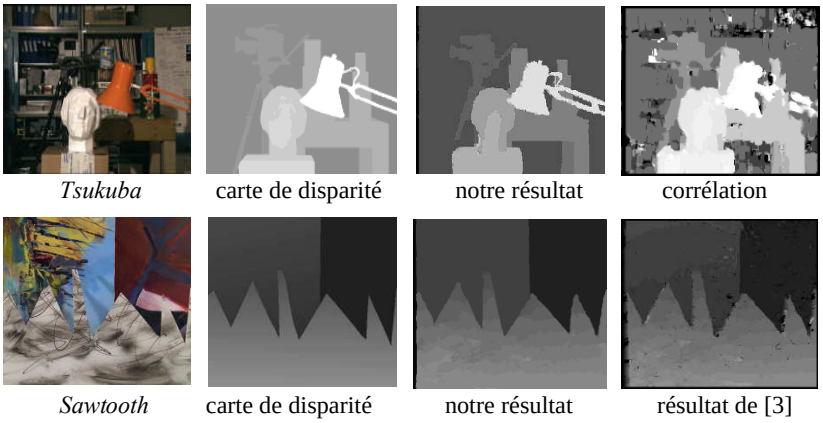


Fig. 4. Les résultats de notre méthode.

Références

1. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, vol. 47, no. 1-3, pp. 7–42, (2002)
2. Roy, S.: Stereo without epipolar lines: A maximum flow formulation, IJCV, vol. 34, no. 2-3, pp. 147–161, (1999)
3. Zureiki, A., Devy, M., and Chatila, R.: Stereo matching using reduced-graph cuts. In IEEE International Conference on Image Processing (ICIP), San Antonio, Texas (USA), September (2007)
4. Ishikawa, H.: Global Optimization Using Embedded Graphs, PhD thesis, New York University, (2000)
5. Cormen, T., and Rivest, R., and al.: Introduction to Algorithms, Second Edition, the MIT Press, (2001)
6. Ford, L. and Fulkerson, D.: Flows in Networks, Princeton University Press, (1962)

7. Boykov, Y., and Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In IEEE Trans. on PAMI , vol. 26, pp. 1124–1137, (2004)
8. Veksler, O.: Efficient graph-based energy minimization methods in computer vision, Ph.D. thesis, Cornell University, (1999)
9. Roy, S., Cox, I.: A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem, International Conference on Computer Vision, (1998)
10. Kolmogorov, V., Zabih, R.: Computing Visual Correspondence with Occlusions using Graph Cuts, International Conference on Computer Vision, (2001)
11. Kolmogorov, V., Zabih, R.: Multi-camera Scene Reconstruction via Graph Cuts, European Conference on Computer Vision, (2002)
12. Kolmogorov, V., Zabih, R.: What Energy Functions can be Minimized via Graph Cuts? European Conference on Computer Vision, (2002)

Weak pseudo-invexity and Fritz-John type optimality in nonlinear programming

Hachem Slimani[1] and Mohammed Said Radjef[2]

Laboratory of Modelling and Optimization of Systems (LAMOS)
Operational Research Department, University of Bejaia 06000, Algeria,
haslimani@gmail.com [1], radjefms@gmail.com [2]

Abstract. In this paper, we study Fritz-John type optimality conditions for a constrained nonlinear programming. New necessary and sufficient conditions for a feasible point to be an optimal solution are obtained under weak invexity with respect to different η_i. Moreover, a new concept of Fritz-John type stationary point is introduced and a characterization of solutions is established under suitable generalized invexity assumption.
keywords: Nonlinear programming; Global minimizer point; Weak pseudo-invexity; Weak FJ-pseudo-invex problem / η and $(\theta_j)_j$; Generalized Fritz-John condition; Generalized Fritz-John stationary point.

1 Introduction

Optimality criteria in mathematical programming are important both theoretically as well as computationally. In most cases optimality criteria form the basis of computational procedures. The best-known necessary optimality criterion for a mathematical programming problem is due to Karush [17] and Kuhn-Tucker [21]. However, the Fritz-John criterion [16] is in a sense more general. In order for the Karush-Kuhn-Tucker criterion to hold, one must impose a constraint-qualification on the constraints of the problem. On the other hand, no such qualification need be imposed on the constraints in order that the Fritz John criterion hold. Moreover, the Fritz John criterion itself can be used to derive a form of the constraint qualification for the Karush-Kuhn-Tucker criterion [10, 23]. Thus, in case of necessary optimality criteria, the only restriction on a constrained program is that the constraints should satisfy certain qualification [22] but for sufficient optimality criteria and duality results to hold, the objective and constraints functions are required to satisfy certain convexity or generalized convexity requirements, see, for example, Mangasarian [22] and Bazaraa et al. [7].

Several classes of functions have been defined for the purpose of weakening the hypothesis of convexity in mathematical programming. Hanson [13] introduced the concept of invexity for the differentiable functions, generalizing the difference $(x - x_0)$ in the definition of convex function to any function $\eta(x, x_0)$. He proved that if, in a mathematical programming problem, instead of the convexity assumption, the objective and constraint functions are invex with respect

to the same vector function η, then both the sufficiency of Karush-Kuhn-Tucker conditions and weak and strong Wolfe duality still hold. Further, Ben Israel and Mond [9] considered a class of functions called preinvex and also showed that the class of invex functions is equivalent to the class of functions whose stationary points are global minima, see also Craven and Glover [12]. Hanson and Mond [14] introduced two new classes of functions called type I and type II functions for the scalar optimization problem, which were further generalized to pseudo-type I and quasi-type I by Rueda and Hanson [27] and sufficient optimality conditions are obtained involving these functions. Kaul and Kaur [18] showed that the Karush-Kuhn-Tucker (Fritz-John) necessary conditions are sufficient for optimality under the hypothesis of the pseudo-invexity and the quasi-invexity (the invexity and the strict invexity) with respect to a same function η for the objective and constraint functions respectively. Martin [24] introduced a weaker invexity called Kuhn-Tucker invexity or KT-invexity which is necessary and sufficient for every Kuhn-Tucker stationary point to be a global minimiser of a given mathematical programming problem. Further properties and applications of invexity for some more general problems were studied by Craven [11], Bector et al. [8], Jeyakumar and Mond [15], Pini and Singh [26] Antczak [1, 4–6], and others.

However, one major difficulty in this extension of convexity is that invex problems require a same function $\eta(x, x_0)$ for the objective and constraint functions. This requirement turns out to be a major restriction in applications. In [28], a constrained nonlinear programming is considered and KT-invex, weakly KT-pseudo-invex and type I problems with respect to different η_i are defined (each function occurring in the studied problem is considered with respect to its own function η_i instead of a same function η). A new Kuhn-Tucker type necessary condition is introduced for nonlinear programming problems and duality results are obtained, for Wolfe and Mond-Weir type dual programs, under this generalized invexity assumptions.

In this paper, we define new concepts of weak pseudo-invexity and weak FJ-pseudo-invexity and we study Fritz-John type optimality in the classical constrained nonlinear programming. Necessary and sufficient conditions for a feasible point to be an optimal solution are obtained under weak invexity with respect to different η_i. Moreover, we introduce a new concept of Fritz-John type stationary point and a characterization of solutions is established under suitable generalized invexity assumption.

2 Preliminaries and definitions

Invex functions were introduced to optimization theory by Hanson [13] (and called by Craven [11]) as a very broad generalization of convex functions.

Definition 1. *[13] Let D a nonempty open set of $\mathbb{R}^n$. A function $f : D \to \mathbb{R}$ is said to be (def) at $x_0 \in D$ with respect to η, if the function f is differentiable at x_0 and there exists a vector function $\eta : D \times D \to \mathbb{R}^n$ such that, for each $x \in D$, (cond) holds.*

(i) def: invex,
cond:

$$f(x) - f(x_0) \geq [\nabla f(x_0)]^t \eta(x, x_0). \tag{1}$$

(ii) def: pseudo-invex,
cond:

$$[\nabla f(x_0)]^t \eta(x, x_0) \geq 0 \Rightarrow f(x) - f(x_0) \geq 0. \tag{2}$$

(iii) def: quasi-invex,
cond:

$$f(x) - f(x_0) \leq 0 \Rightarrow [\nabla f(x_0)]^t \eta(x, x_0) \leq 0. \tag{3}$$

If the second (implied) inequality in (3) is strict ($x \neq x_0$), we say that f is strictly quasi-invex at x_0 with respect to η. f is said to be invex (resp. pseudo-invex or (strictly) quasi-invex) on D with respect to η, if f is invex (resp. pseudo-invex or (strictly) quasi-invex) at each $x_0 \in D$ with respect to the same η.

Definition 2. *[2] Let D be a nonempty subset of $\mathbb{R}^n$, $\eta : D \times D \to \mathbb{R}^n$ and let x_0 be an arbitrary point of D. The set D is said to be invex at x_0 with respect to η, if for each $x \in D$,*

$$x_0 + \lambda \eta(x, x_0) \in D, \ \forall \ \lambda \in [0, 1]. \tag{4}$$

D is said to be an invex set with respect to η, if D is invex at each $x_0 \in D$ with respect to the same η.

Definition 3. *[9] Let $D \subseteq \mathbb{R}^n$ be an invex set with respect to $\eta : D \times D \to \mathbb{R}^n$. A function $f : D \to \mathbb{R}$ is called pre-invex on D with respect to η, if for all $x, x_0 \in D$,*

$$\lambda f(x) + (1 - \lambda) f(x_0) \geq f(x_0 + \lambda \eta(x, x_0)), \ \forall \ \lambda \in [0, 1]. \tag{5}$$

Definition 4. *[3] Let $D \subseteq \mathbb{R}^n$ be an invex set with respect to $\eta : D \times D \to \mathbb{R}^n$. A m-dimensional vector valued function $\Psi : D \to \mathbb{R}^m$ is pre-invex with respect to η, if each of its components is pre-invex on D with respect to the same function η.*

Further general discussions of invexity, other extensions of this concept and their applications may be found in [2, 3, 9, 14, 15, 19, 20, 24, 25, 28, 29] and others.

Craven and Glover [12] and Ben-Israel and Mond [9] stated that the class of invex functions are all those functions whose stationary points are global minima. Moreover, Ben-Israel and Mond [9] proved that the class of invex and pseudo-invex functions coincide and every function with $\nabla f(x) \neq 0$ is invex.

Proposition 1. *[9] Any differentiable function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ at a point $x_0 \in D$, with $\nabla f(x_0) \neq 0$, is invex at x_0 with respect to $\eta(x, x_0) = [f(x) - f(x_0)] \frac{[\nabla f(x_0)]}{[\nabla f(x_0)]^t [\nabla f(x_0)]}$, $\forall \ x \in D$.*

Now, we give others η for which a given scalar function is pseudo-invex.

Proposition 2. *Any differentiable function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ at a point $x_0 \in D$, with $\nabla f(x_0) \neq 0$, is pseudo-invex at x_0 with respect to $\eta(x, x_0) = [f(x) - f(x_0)][\nabla f(x_0)]$, $\forall\ x \in D$ or $\eta(x, x_0) = [f(x) - f(x_0)]t(x_0)$, $\forall\ x \in D$ where $t(x_0) \in \mathbb{R}^n$ with $t_i(x_0) = \begin{cases} 1, & \text{if } \frac{\partial f}{\partial x_i}(x_0) \geq 0, \\ -1, & \text{otherwise,} \end{cases}$ for all $i = 1, ..., n$.*

Example 1. The function $f_3 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_3(x_1, x_2) = x_2^2 - x_1$ is pseudo-invex on $\mathbb{R} \times \mathbb{R}_+$ with respect to $\eta_3(x, \tilde{x}) = (\tilde{x}_2^2 - \tilde{x}_1 - x_2^2 + x_1, x_2^2 - x_1 - \tilde{x}_2^2 + \tilde{x}_1)$ and it is pseudo invex on $\mathbb{R} \times (\mathbb{R}_- \backslash \{0\})$ with respect to $\bar{\eta}_3(x, \tilde{x}) = (\tilde{x}_2^2 - \tilde{x}_1 - x_2^2 + x_1, \tilde{x}_2^2 - \tilde{x}_1 - x_2^2 + x_1)$ because $\nabla f_3(\tilde{x}) = (-1, 2\tilde{x}_2) \neq (0, 0), \forall\ \tilde{x} \in \mathbb{R}^2$.

In Slimani and Radjef [28], a new concept of weak KT-pseudo-invexity is introduced and duality results have been obtained for a constrained nonlinear programming. Now, in the same line as in [28], we define a concept of weak pseudo-invexity for scalar functions given as follows.

Definition 5. *A function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be weakly pseudo-invex at $x_0 \in D$ with respect to η, if the function f is differentiable at x_0 and there exists a vector function $\eta : D \times D \to \mathbb{R}^n$ such that for each $x \in D$:*

$$f(x) - f(x_0) < 0 \Rightarrow \exists\ \bar{x} \in D,\ [\nabla f(x_0)]^t \eta(\bar{x}, x_0) < 0. \tag{6}$$

f is said to be weakly pseudo-invex on D with respect to η, if f is weakly pseudo-invex at each $x_0 \in D$ with respect to the same η.

Remark 1. In the definition 5, if $\bar{x} = x$, we obtain the pseudo-invexity of scalar function given in the definition 1.

If a function f is pseudo-invex at x_0 with respect to η, then it is weakly pseudo-invex at x_0 with respect to the same η (with $\bar{x} = x$) but the converse is not true.

Example 2. • The function $f_1 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_1(x) = -x_1^2 - x_2$ is weakly pseudo-invex at $x_0 = (1, 0)$ with respect to $\eta_1(x, x_0) = (x_0 - x) \in \mathbb{R}^2$ (take $\bar{x} = [(f_1(x) - f_1(x_0), f_1(x) - f_1(x_0)) + x_0] \in \mathbb{R}^2$). But f_1 is not pseudo-invex at x_0 with respect to the same η_1 because for $x = (2, -1)$, $f_1(x) - f_1(x_0) < 0$ and $[\nabla f_1(x_0)]^t \eta_1(x, x_0) > 0$.
• The function $f_2 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_2(x) = x_1 + \sin x_2$ is weakly pseudo-invex at $x_0 = (\frac{\pi}{6}, \frac{\pi}{3})$ with respect to $\eta_2(x, x_0) = x - x_0 \in \mathbb{R}^2$ (take $\bar{x} = (f_2(x) - f_2(x_0), f_2(x) - f_2(x_0)) \in \mathbb{R}^2$). But f_2 is not pseudo-invex at x_0 with respect to the same η_2 because for $x = (\frac{\pi}{3}, 0)$, $f_2(x) - f_2(x_0) < 0$ and $[\nabla f_2(x_0)]^t \eta_2(x, x_0) = 0$.

Consider the following constrained nonlinear programming problem (P):

$$(P) \quad \begin{array}{l} \text{Minimize } f(x), \\ \text{subject to } g_j(x) \leq 0,\ j = 1, ..., k, \end{array}$$

where $f, g_j : D \to \mathbb{R},\ j = 1, ..., k$, D is an open set of $\mathbb{R}^n$; $X = \{x \in D : g_j(x) \leq 0,\ j = 1, ..., k\}$ is the set of feasible solutions for (P). For $x_0 \in D$, we denote $J(x_0) = \{j \in \{1, ..., k\} : g_j(x_0) = 0\}$, $J = |J(x_0)|$ is the cardinal of the set $J(x_0)$.

The concept of Kuhn-Tucker (Frit-John) stationary point for (P) is very used in the literature for establishing optimality conditions for the problem (P). It is defined as follows.

Definition 6. *[22] A feasible point $x_0 \in X$ is said to be a Kuhn-Tucker (resp. Fritz-John) stationary point for (P), if the functions f and g are differentiable at x_0 and there exists $\lambda \in \mathbb{R}^J_{\geq}$ (resp. $(\mu, \lambda) \in \mathbb{R}^{1+J}_{+},\ (\mu, \lambda) \neq 0$) such that:*

$$\nabla f(x_0) + \sum_{j \in J(x_0)} \lambda_j \nabla g_j(x_0) = 0, \tag{7}$$

$$(resp. \quad \mu \nabla f(x_0) + \sum_{j \in J(x_0)} \lambda_j \nabla g_j(x_0) = 0). \tag{8}$$

The problem (P) is said to be HC-invex at $x_0 \in X$ if f and $g_j,\ j = 1, ..., k$ are invex at x_0 (with respect to the same function η). Thus, if the problem (P) is HC-invex, then every Kuhn–Tucker point is a minimizer of (P) [13]. Martin [24] remarked that the converse is not true in general, and he proposed a weaker notion, called KT-invexity, which assures that every Kuhn–Tucker point is a minimizer of problem (P) if and only if problem (P) is KT-invex.

Definition 7. *[24] The problem (P) is said to be KT-invex on the feasible set X with respect to η, if the functions f and g are differentiable on X and there exists $\eta : X \times X \to \mathbb{R}^n$ such that for each $x, x_0 \in X$:*

$$f(x) - f(x_0) \geq [\nabla f(x_0)]^t \eta(x, x_0), \tag{9}$$

$$-[\nabla g_j(x_0)]^t \eta(x, x_0) \geq 0,\ \forall\ j \in J(x_0). \tag{10}$$

The following result established by Martin [24] is considered as an optimality criterion for problem (P) and as a characterization of the KT-invexity notion with respect to η.

Theorem 1. *[24] Every Kuhn-Tucker stationary point of problem (P) is a global minimizer if and only if (P) is KT-invex on X with respect to η.*

Weir and Mond [29] have proved the following alternative lemma which will be used for establishing a characterization of optimal solutions for (P).

Lemma 1. *Let S be a nonempty invex set in $\mathbb{R}^n$ with respect to $\eta : S \times S \to \mathbb{R}^n$ and let $\psi : S \to \mathbb{R}^m$ be a pre-invex function on S with respect to the same η. Then either*

(i) $\psi(x) < 0$ has a solution $x \in S$,
or
(ii) $p^t \psi(x) \geq 0$ for all $x \in S$, for some $p \in \mathbb{R}^m_{+},\ p \neq 0$,

but both alternatives are never true.

Now, before establishing optimality conditions for (P), we give the following simple propositions that we have need.

Proposition 3. *Let S be a nonempty subset of $\mathbb{R}^n$. If a function $h : S \to]-\infty, 0]$ is strictly quasi-invex at $x_0 \in S$ with respect to $\theta : S \times S \to \mathbb{R}^n$ and $h(x_0) = 0$, then $[\nabla h(x_0)]^t \theta(x, x_0) < 0,\ \forall\ x \in S$.*

Proof. For $x \in S$, we have $h(x) \leq 0 = h(x_0)$, which by strict quasi-invexity of h at x_0 with respect to θ implies $[\nabla h(x_0)]^t \theta(x, x_0) < 0$. ■

Corollary 1. *Let $x_0 \in X$ be a feasible solution of (P). For each $j \in J(x_0)$, if g_j is strictly quasi-invex at x_0 with respect to $\theta_j : X \times X \to \mathbb{R}^n$, then $[\nabla g_j(x_0)]^t \theta_j(x, x_0) < 0,\ \forall\ x \in X$.*

Proposition 4. *Let x_0 be a feasible solution of (P). For each $j \in \{1, ..., k\}$, if $\nabla g_j(x_0) \neq 0$ and the components of $\theta_j : X \times X \to \mathbb{R}^n$ are defined by* $\theta_j^l(x, x_0) = \begin{cases} g_j(x) - \delta, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0) \geq 0, \\ -g_j(x) + \delta, & \text{otherwise,} \end{cases}$ *for all $l = 1, ..., n$ with $\delta \in \mathbb{R},\ \delta > 0$, then $[\nabla g_j(x_0)]^t \theta_j(x, x_0) < 0,\ \forall\ x \in X$.*

Proof. we have $[\nabla g_j(x_0)]^t \theta_j(x, x_0) = \sum_{l=1}^{n} \frac{\partial g_j}{\partial x_l}(x_0) s_l^j(x_0)[g_j(x) - \delta] < 0,\ \forall\ x \in X,$
$\forall\ j \in \{1, ..., k\}$ with $s_l^j(x_0) = \begin{cases} 1, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0) \geq 0, \\ -1, & \text{otherwise,} \end{cases}$ for all $l = 1, ..., n$. ■

3 Optimality conditions

In this section, we give Fritz-John type necessary and sufficient optimality conditions for a feasible point to be an optimal solution of (P). For the sufficiency conditions, we use the weak invexity with respect to different functions η and $(\theta_j)_j$.

In the same line as in [28], we give the following Fritz-John type necessary condition for (P).

Theorem 2. *(Fritz-John type necessary optimality condition) Suppose that*

(i) x_0 is a local or global solution for (P);
(ii) the functions $f,\ g_j,\ j \in J(x_0)$ are differentiable at x_0.

Then there exist vector functions $\eta : X \times D \to \mathbb{R}^n,\ \theta_j : X \times D \to \mathbb{R}^n,\ j \in J(x_0)$ and $(\mu, \lambda) \in \mathbb{R}_+^{1+J},\ (\mu, \lambda) \neq 0$ such that $(x_0, \mu, \lambda, \eta, (\theta_j)_j)$ satisfies the following generalized Fritz-John condition

$$\mu[\nabla f(x_0)]^t \eta(x, x_0) + \sum_{j \in J(x_0)} \lambda_j [\nabla g_j(x_0)]^t \theta_j(x, x_0) \geq 0,\ \forall\ x \in X. \tag{11}$$

Proof. Suppose that x_0 is a local solution of (P). Then there exists, a neighborhood of x_0, $v(x_0) \subset X$ such that for all $x \in v(x_0),\ f(x) - f(x_0) \geq 0$. Thus, it suffices to take $\eta,\ \theta_j,\ j \in J(x_0),\ \mu$ and λ as follows:

- $\eta(x,x_0)=\begin{cases}[f(x)-f(x_0)]t(x_0), & \text{if } x\in v(x_0),\\ t(x_0), & \text{otherwise}\end{cases},\ t(x_0)\in\mathbb{R}^n$
  with $t_i(x_0)=\begin{cases}1, & \text{if } \frac{\partial f}{\partial x_i}(x_0)\geq 0,\\ -1, & \text{otherwise,}\end{cases}$ for all $i=1,...,n$;
- $\theta_j(x,x_0)=-g_j(x)s^j(x_0),\ x\in X,\ s^j(x_0)\in\mathbb{R}^n$
  with $s_i^j(x_0)=\begin{cases}1, & \text{if } \frac{\partial g_j}{\partial x_i}(x_0)\geq 0,\\ -1, & \text{otherwise,}\end{cases}$ for all $i=1,...,n$;
- $\mu=1$;
- $\lambda_j=\frac{1}{J}$, for all $j\in J(x_0)$.

If x_0 is a global solution of (P), we take $\eta(x,x_0)=[f(x)-f(x_0)]t(x_0),\ \forall\ x\in X$ instead of the one given above. ■

Now, using the generalized Fritz-John condition (11), we establish sufficient conditions for a feasible point to be an optimal solution of (P) under weak invexity with respect to different η and $(\theta_j)_j$.

Theorem 3. *Let $x_0\in X$ and suppose that:*

(i) f is (weakly) pseudo-invex at x_0 with respect to $\eta: X\times X\to\mathbb{R}^n$;
(ii) g is differentiable at x_0 and for all $j\in J(x_0)$, there exists a function $\theta_j: X\times X\to\mathbb{R}^n$ such that $[\nabla g_j(x_0)]^t\theta_j(x,x_0)<0,\ \forall\ x\in X$.

If there exists a vector $(\mu,\lambda)\in\mathbb{R}_+^{1+J},\ (\mu,\lambda)\neq 0$ such that the generalized Fritz-John condition (11) is satisfied, then the point x_0 is an optimal solution of (P).

Proof. Let us suppose that x_0 is not an optimal solution of (P). Then there exists a feasible point x such that $f(x)-f(x_0)<0$.
Since f is weakly pseudo-invex at x_0 with respect to η, it follows that

$$\exists\ \bar{x}\in X,\ [\nabla f(x_0)]^t\eta(\bar{x},x_0)<0. \tag{12}$$

By hypothesis, we have

$$[\nabla g_j(x_0)]^t\theta_j(\bar{x},x_0)<0,\ \forall\ j\in J(x_0). \tag{13}$$

As $(\mu,\lambda)\geq 0,\ (\mu,\lambda)\neq 0$ and from (12) and (13), it follows that

$$\mu[\nabla f(x_0)]^t\eta(\bar{x},x_0)+\sum_{j\in J(x_0)}\lambda_j[\nabla g_j(x_0)]^t\theta_j(\bar{x},x_0)<0,$$

which contradicts (11), and therefore, x_0 is an optimal solution of (P). ■

Remark 2. According to the corollary 1, we can replace in the above theorem the hypothesis $\forall\ j\in J(x_0),\ [\nabla g_j(x_0)]^t\theta_j(x,x_0)<0,\ \forall\ x\in X$ by: $\forall\ j\in J(x_0)$, g_j is strictly quasi-invex at x_0 with respect to θ_j.

In order to illustrate the obtained result, we shall give an example of nonlinear programming problem in which the optimal solution will be obtained by the application of theorem 3.

Example 3. We consider the following nonlinear programming problem

$$\begin{array}{c} \text{Minimize } f(x) = -x_1, \\ \text{subject to } \ g_1(x) = x_1^3 - x_2 \leq 0, \\ g_2(x) = x_2 \leq 0, \end{array} \tag{14}$$

where $f : \mathbb{R}^2 \to \mathbb{R}$ and $g = (g_1, g_2) : \mathbb{R}^2 \to \mathbb{R}^2$. The set of feasible solutions of problem is $X = \{x = (x_1, x_2) \in \mathbb{R}^2 : x_1^3 - x_2 \leq 0 \text{ and } x_2 \leq 0\}$.
We have $x_0 = (0, 0) \in X$ and

- f is pseudo-invex at x_0 with respect to $\eta(x, x_0) = (x_1, -x_1)$ using the proposition 2;
- g is differentiable at x_0 , g_1 and g_2 are active constraints at x_0 and using the proposition 4, for $\theta_1(x, x_0) = (x_1^3 - x_2 - 1, -x_1^3 + x_2 + 1)$ and $\theta_2(x, x_0) = (x_2 - 1, x_2 - 1)$, we obtain that $[\nabla g_j(x_0)]^t \theta_j(x, x_0) < 0, \ \forall\ x \in X, \ \forall\ j \in J(x_0) = \{1, 2\}$.

The generalized Fritz-John condition (11) at x_0 for $\mu = 1$ and $\lambda_1 = \lambda_2 = 0$ takes a form $[\nabla f(x_0)]^t \eta(x, x_0) = -x_1 \geq 0, \ \forall\ x \in X$. It follows that, by theorem 3, x_0 is an optimal solution for the given nonlinear programming problem.

As particular case of theorem 3, if the functions θ_j are equal to $\eta, \ \forall\ j \in J(x_0)$ and on using the usual Fritz-John condition, we obtain the following theorem.

Theorem 4. *Let $x_0 \in X$ and suppose that:*

(i) f is (weakly) pseudo-invex at x_0 with respect to $\eta : X \times X \to \mathbb{R}^n$;
(ii) g is differentiable at x_0 and $\forall\ j \in J(x_0)$, $[\nabla g_j(x_0)]^t \eta(x, x_0) < 0, \ \forall\ x \in X$.

If there exists a vector $(\mu, \lambda) \in \mathbb{R}_+^{1+J}, \ (\mu, \lambda) \neq 0$ such that the Fritz-John condition (8) is satisfied, then the point x_0 is an optimal solution of (P).

Proof. It suffices to multiply the relation (8) by $\eta(x, x_0)$ and use the theorem 3. ■

Remark 3. Kaul and Kaur [18, theorem 3.2] proved that the Fritz-John condition (8) is sufficient for x_0 to be an optimal solution of (P), if the objective and active constraint functions, f and $g_j, \ j \in J(x_0)$, are invex and strictly invex, respectively, at x_0 with respect to the same η. It is shown, in the theorem 4 (see remark 2), that the result is also true under weak invexity, when f is (weak) pseudo-invex and $\forall\ j \in J(x_0), \ g_j$ is strictly quasi-invex at x_0 with respect to η.

Using the proposition 3 and while reasoning in the same manner as in the proof of theorem 3, we can prove the following result.

Theorem 5. *Let $x_0 \in X$ and suppose that f is (weakly) pseudo-invex at x_0 with respect to $\eta : X \times X \to \mathbb{R}^n$. If there exists a vector $(\mu, \lambda) \in \mathbb{R}_+^{1+J}, \ (\mu, \lambda) \neq 0$ such that the scalar function $\Psi(x) = \sum_{j \in J(x_0)} \lambda_j g_j(x)$ is strictly quasi-invex at x_0 with respect to $\theta : X \times X \to \mathbb{R}^n$ and*

$$\mu[\nabla f(x_0)]^t \eta(x, x_0) + \sum_{j \in J(x_0)} \lambda_j [\nabla g_j(x_0)]^t \theta(x, x_0) \geq 0, \ \forall\ x \in X, \tag{15}$$

then the point x_0 is an optimal solution of (P).

Remark 4. In the above theorem, the strict quasi-invexity assumption of the scalar function $\Psi(x) = \sum_{j \in J(x_0)} \lambda_j g_j(x)$ at x_0 with respect to θ can be replaced by the relation $\sum_{j \in J(x_0)} \lambda_j [\nabla g_j(x_0)]^t \theta(x, x_0) < 0, \ \forall \ x \in X$.

4 Characterization of solutions

In this section, we define new class of Fritz-John type stationary points for (P) and we establish new characterization of solutions under suitable generalized invexity requirement.

Definition 8. *A feasible point x_0 for (P) is said to be a generalized Fritz-John stationary point with respect to η and $(\theta_j)_{j \in J(x_0)}$, if the functions f and g are differentiable at x_0 and there exist functions $\eta : X \times X \to \mathbb{R}^n$, $\theta_j : X \times X \to \mathbb{R}^n, \ j \in J(x_0)$ and a vector $(\mu, \lambda) \in \mathbb{R}_+^{1+J}, \ (\mu, \lambda) \neq 0$ such that $(x_0, \mu, \lambda, \eta, (\theta_j)_{j \in J(x_0)})$ satisfies the generalized Fritz-John condition (11).*

Remark 5. The concept of generalized Kuhn-Tucker stationary point can be defined by setting $\mu = 1$ and $\lambda \in \mathbb{R}_+^J$ in the definition 8.

Martin [24] has characterized the optimal solutions for (P) by using the concept of KT-invexity (with respect to the same η) given in the definition 7. Now we characterize the optimal solutions for (P) by using the concept of generalized Fritz-John stationary point and new kind of invex functions which we define as follows.

Definition 9. *The problem (P) is said to be weakly FJ-pseudo-invex at $x_0 \in X$ with respect to η and $(\theta_j)_{j \in J(x_0)}$, if the functions f and g are differentiable at x_0 and there exist functions $\eta : X \times X \to \mathbb{R}^n$ and $\theta_j : X \times X \to \mathbb{R}^n, \ j \in J(x_0)$ such that for each $x \in X$:*

$$f(x) - f(x_0) < 0 \Rightarrow \exists \ \bar{x} \in X, \begin{cases} [\nabla f(x_0)]^t \eta(\bar{x}, x_0) < 0, \\ [\nabla g_j(x_0)]^t \theta_j(\bar{x}, x_0) < 0, \ \forall \ j \in J(x_0). \end{cases} \tag{16}$$

If $\bar{x} = x$, in the relation (16), we say that (P) is FJ-pseudo-invex at x_0 with respect to η and $(\theta_j)_{j \in J(x_0)}$. The problem (P) is said to be (weakly) FJ-pseudo-invex on X with respect to η and $(\theta_j)_j$, if it is (weakly) FJ-pseudo-invex at each $x_0 \in X$ with respect to the same η and $(\theta_j)_{j \in J(x_0)}$.

Theorem 6. *Let $X \subseteq \mathbb{R}^n$ be a nonempty invex set with respect to $\phi : X \times X \to \mathbb{R}^n$. Further, let $\eta : X \times X \to \mathbb{R}^n$ and $\theta_j : X \times X \to \mathbb{R}^n, \ j = \overline{1,k}$ be functions, such that for all $x_0 \in X, [\nabla f(x_0)]^t \eta(x, x_0)$ and $[\nabla g_j(x_0)]^t \theta_j(x, x_0), \ j \in J(x_0)$ are preinvex of x on X with respect to ϕ. Then, every generalized Fritz-John stationary point with respect to η and $(\theta_j)_j$ of problem (P) is a global minimizer if and only if (P) is weakly FJ-pseudo-invex on X with respect to η and $(\theta_j)_j$.*

Proof. **(1)** Let $x_0 \in X$ be a generalized Fritz-John stationary point with respect to η and $(\theta_j)_{j\in J(x_0)}$ for (P). If (P) is weakly FJ-pseudo-invex at x_0 with respect to η and $(\theta_j)_{j\in J(x_0)}$, then, from theorem 3, we obtain that x_0 is a global minimizer of (P).
(2) For the converse, suppose that every generalized Fritz-John stationary point with respect to η and $(\theta_j)_j$ of problem (P) is a global minimizer.
Let us suppose that there exist two feasible points $\tilde{x}$ and x_0 such that

$$f(\tilde{x}) - f(x_0) < 0. \tag{17}$$

This means that x_0 is not a global minimizer of (P), and by using the initial hypothesis, x_0 is not a generalized Fritz-John stationary point with respect to η and $(\theta_j)_{j\in J(x_0)}$ for (P), ie:

$$\left(\mu[\nabla f(x_0)]^t\eta(x,x_0) + \sum_{j\in J(x_0)} \lambda_j[\nabla g_j(x_0)]^t\theta_j(x,x_0) \geq 0,\ \forall\ x \in X.\right)$$

is not satisfied for all $(\mu,\lambda) \in \mathbb{R}^{1+J}_+,\ (\mu,\lambda) \neq 0$. Therefore, by lemma 1, the system

$$\begin{cases} [\nabla f(x_0)]^t\eta(x,x_0) < 0, \\ [\nabla g_j(x_0)]^t\,\theta_j(x,x_0) < 0, \quad \forall\ j \in J(x_0). \end{cases}$$

has the solution $x = \bar{x} \in X$. In consequence, (P) is weakly FJ-pseudo-invex on X with respect to η and $(\theta_j)_j$. ■

5 Conclusion

In this paper, we have defined new concepts of weak pseudo-invexity and weak FJ-pseudo-invexity to study Fritz-John type optimality for the classical constrained nonlinear programming. New necessary and sufficient conditions for a feasible point to be an optimal solution are obtained under weak invexity with respect to different η and $(\theta_j)_j$. We have established simple propositions which helped us to construct these different functions (η and $(\theta_j)_j$) to verify the optimality of a feasible point (see example 3). Moreover, a new concept of Fritz-John type stationary point is introduced and a characterization of solutions is established under suitable generalized invexity assumption.

References

1. Antczak, T.: A class of B-(p,r)-invex functions and mathematical programming. J. Math. Anal. Appl. **286**, 187–206 (2003)
2. Antczak, T.: Mean value in invexity analysis. Nonlinear Analysis **60**, 1473–1484 (2005)
3. Antczak, T.: Multiobjective Programming under d-invexity. Eur. J. Oper. Res. **137**, 28–36 (2002)

4. Antczak, T.: On (p,r)-Invexity-Type Nonlinear Programming Problems. J. Math. Anal. Appl. **264**, 382–397 (2001)
5. Antczak, T.: (p,r)-invex sets and functions. J. Math. Anal. Appl. **263**, 355–379 (2001)
6. Antczak, T.: r-preinvexity and r-invexity in mathemetical programming. Comp. Math. with Appl. **50**, 551–566 (2005)
7. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. Wiley, New York, Third Edition, 2006.
8. Bector, C.R., Suneja, S.K., Lalitha, C.S.: Generalized B-vex functions and generalized B-vex programming. J. Optim. Theory Appl. **76**, 561–576 (1993)
9. Ben-Israel, A., Mond, B.: What is Invexity ?. J. Austral. Math. Soc. Ser. **B 28**, 1–9 (1986)
10. Cottle, R.W.: A theorem of Fritz John in mathematical programming. RAND Memorandum RM-3858-PR, October, 1963.
11. Craven, B.D.: Invex Functions and Constrained Local Minima. Bull. Austral. Math. Soc. **24**, 357–366 (1981)
12. Craven, B.D., Glover B.M.: Invex Functions and Duality. J. Austral. Math. Soc. Ser. **A 39**, 1–20 (1985)
13. Hanson, M.A.: On Sufficiency of the Kuhn-Tuker Conditions. J. Math. Anal. Appl. **80**, 445-550 (1981)
14. Hanson, M.A., Mond, B.: Necessary and Sufficient Conditions in Constrained Optimization. Math. Programming **37**, 51–58 (1987)
15. Jeyakumar, V., Mond, B.: On generalized convex mathematical programming. J. Austral. Math. Soc. Ser. **B 34**, 43–53 (1992)
16. John, F.: Extremum problems with inequalities as side conditions. In K.0. Friedrichs, 0.E. Neugebauer and J.J. Stoker (eds.), Studies and Essays, Courant Anniversary Volume, Wiley (Interscience), New York, pp. 187–204, 1948.
17. Karush, W.: Minima of functions of several variables with inequalities as side conditions. Dissertation, Department of Mathematics, University of Chicago, 1939
18. Kaul, R.N., Kaur, S.: Optimality Criteria in Nonlinear Programming Involving Nonconvex Functions. J. Math. Anal. Appl. **105**, 104–112 (1985)
19. Kaul, R.N., Suneja, S.K., Srivastava, M.K.: Optimality Criteria and Duality in Multiple-Objective Optimization Involving Generalized Invexity. J. Optim. Theory Appl. **80 (3)**, 465–482 (1994)
20. Khanh, P.Q.: Invex convexlike functions and duality. J. Optim. Theory Appl. **87**, 141–165 (1986)
21. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In J. Neyman (ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Univ. of Calif. Press, Berkeley, Calif., pp. 481–492, 1951
22. Mangasarian, O.L.: Nonlinear Programming. McGraw–Hill, New York, 1969
23. Mangasarian, O.L., Fromovitz, S.: The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. J. Math. Anal. Appl. **17**, 37–47 (1967)
24. Martin, D.H.: The Essence of Invexity, J. Optim. Theory Appl. **47**, 65–76 (1985)
25. Osuna-Gomez, R., Beato-Morero, A. Rufian-Lizana, A.: Generalized Convexity in Multiobjective Programming. J. Math. Anal. Appl. **233**, 205–220 (1999)
26. Pini, R., Singh, C.: A survey of recent [1985-1995] advances in generalized convexity with applications to duality theory and optimality conditions. Optim. **39 (4)** (1997) 311–360.
27. Rueda, N.G., Hanson, M.A.: Optimality criteria in mathematical programming involving generalized invexity. J. Math. Anal. Appl. **130**, 375–385 (1988)

28. Slimani, H., Radjef, M.S.: Duality for nonlinear programming under generalized Kuhn-Tucker condition. Journal of Optimization: Theory, Methods and Applications (IJOTMA), in press (2009)
29. Weir, T., Mond, B.: Pre-invex Functions in Multiple Objective Optimization. J. Math. Anal. Appl. **136**, 29–38 (1988)

Estimation de la Confiance en des Hôtes Potentiellement Malicieux pour la Protection des Agents Mobiles

Meriem Zaïter[1], Salima Hacini[1], et Zizette Boufaïda[1],
[1] : Laboratoire Lire, Université Mentouri, Constantine, Algérie.
{meriem.zaiter, salimahacini, zboufaida}@gmail.com

Résumé. Les études concernant la technologie d'agent mobile ont fait couler beaucoup d'encre. La raison réside dans les diverses avantages offerts par ce paradigme dont l'impact sur l'optimisation de l'utilisation des réseaux est certain. Néanmoins, l'assurance d'une exécution sûre de l'agent mobile est un problème crucial qui doit être résolu afin que cette technologie soit acceptée et puisse être employée par différentes applications telles que le e-commerce. Notre objectif est de proposer une technique de protection de l'agent mobile basée sur l'estimation de la confiance en l'hôte d'accueil. L'environnement de confiance étant une des protections les plus sûres, l'agent mobile se trouve ainsi à l'abri de nombreuses attaques.

Mot clefs: agent mobile, sécurité des agents mobiles, confiance, métriques de la confiance, hôtes malicieux, sécurité de transactions.

1 Introduction

L'échange d'information sur internet nécessite un degré de confiance assez important qui est accumulé au fur et à mesure du déroulement de l'interaction de l'internaute avec le site visité. Par ailleurs, ces dernières années de nombreuses recherches se sont intéressées aux applications liées au e-commerce afin de garantir des transactions plus efficaces. L'une des tendances qui vise l'amélioration du déroulement de ces transactions est l'utilisation des agents mobiles. Avec l'expansion des attaques internet, les critères de choix du fournisseur d'un produit/service ont évolué pour englober d'autres caractéristiques telles que la réputation du fournisseur ou la sûreté de la transaction. L'approche de protection, présentée dans cet article, s'appuie sur l'exécution de l'agent mobile dans un environnement de confiance [1].
L'objectif de ce travail est, en premier lieu, de préciser l'ensemble des métriques nécessaires à l'évaluation de la confiance lors d'une transaction en ligne. Chacune de ces métriques est influencée par un ensemble de paramètres et facteurs [2]. En second lieu, une phase de simulation s'avère indispensable afin de valider ces métriques. Cet objectif permet d'exploiter les caractéristiques de l'agent mobile qui est capable

d'interagir avec l'hôte visité (interaction, observation et inspection [3]) et de s'adapter.
La seconde section de cet article présente les différents travaux ayant trait au calcul de la confiance dans les systèmes distribués. La troisième section constitue le cœur de notre travail puisqu'elle identifie l'ensemble des métriques utilisées pour estimer la confiance de l'hôte d'accueil. La validation des métriques établies à travers une simulation d'une application commerce électronique est décrite au niveau de la section 4. La section 5 souligne l'apport de la prise en compte de l'aspect multidimensionnel de la confiance. Enfin, la section 6 conclut ce travail.

2 Les métriques de la confiance

La question de la confiance est fréquemment posée dans différents domaines de la vie et une vaste littérature existe sur ce sujet. L'hétérogénéité des définitions et l'absence d'une définition simple et commune ne doit pas surprendre puisqu'il s'agit d'un phénomène traité par différentes disciplines. Ainsi, les points de vue recensés sont à replacer dans un contexte bien spécifique [4]. La confiance est donc utilisée sous plusieurs formes selon le domaine et le contexte de son utilisation. Elle peut être vue comme une métrique influencée par un ensemble de paramètres [5]. C'est le cas de Lee [6] qui l'a subdivisée en confiance du partenaire et celle du service offert. De leur côté, McKnight et *al.* [4] ont considéré que la confiance est supportée par un ensemble de métriques. Ils l'ont définie comme la croyance en la bonne foi, la loyauté, la sincérité, la fidélité d'autrui (ou en ses capacités), la compétence et la qualification professionnelles. Ils ont réalisé une classification qui leur a permis d'établir cinq catégories d'estimation de la confiance représentées par la compétence, la prédictibilité, la bienveillance, l'intégrité, et d'autres attributs tels que la personnalité attrayante ou communicative.
La confiance peut donc être évaluée sur la base d'un ensemble de métriques relatives à des paramètres la caractérisant. A titre d'exemple, Lin et *al.* [5], ont estimé la confiance sur la base de trois métriques : l'intégrité, la bienveillance, et la compétence. Chacune d'elle étant influencée par un ensemble de paramètres.
Les métriques de la confiance peuvent aussi englober la réputation [7], [8] ou le risque [9], [10]. La réputation est liée à l'historique du comportement d'une entité. Elle peut être directe ou indirecte selon qu'elle est directement collectée ou fournie par une tierce partie de confiance [11], [12].
Le risque, quant à lui, est proportionnel à la confiance. Une telle situation est rencontrée dans le domaine du e-commerce dans lequel l'augmentation du risque impose un besoin important de confiance. Dans ce cas, deux paramètres permettent de mesurer le risque : la qualité du produit et le montant échangé entre les parties [13].
L'estimation de la confiance résulte donc de l'agrégation de différentes métriques; sa valeur dépend du choix de ces métriques ainsi que des paramètres intervenant dans leur évaluation. L'objectif de ce travail est donc de définir les différentes métriques de la confiance afin de protéger l'agent mobile en lui permettant de s'exécuter dans des environnements de confiance.

3 Les métriques proposées

La sécurité de l'agent mobile est liée de manière intrinsèque à la confiance placée en l'hôte visité [3]. Par ailleurs, la confiance est considérée comme une notion multi-facettes influencée par diverses métriques telles que la réputation ou le risque. Cependant ces deux métriques ne peuvent pas, à elles seules, décider un agent mobile à s'engager dans une transaction. En effet, d'autres éléments interviennent tels que la disponibilité ou la compétence de l'hôte d'accueil. La décision de l'agent mobile doit donc reposer sur une estimation rigoureuse de la confiance afin d'éviter des réactions inadaptées pouvant léser aussi bien l'hôte d'accueil que l'agent mobile activant pour le compte d'un organisme. Cette étude a révélé [14], [4] l'implication d'un ensemble de métriques dans l'évaluation de la confiance de l'hôte visité:

- l'identification de l'hôte visité (I),
- sa bienveillance (B),
- sa reputation (R),
- sa disponibilité (D),
- le risque encouru (K)

Chacune des métriques définies fait intervenir un certain nombre de paramètres pouvant servir à son évaluation.

3.1 L'identification (I)

L'intérêt de l'identification entre l'agent et l'hôte visité est fondamental pour assurer la sureté de la transaction électronique. Cette identification permet l'authentification de l'hôte visité. L'estimation de cette métrique se base sur un ensemble de paramètres tel que :

- l'identité et/ou le mot de passe,
- l'adresse géographique complète,
- les coordonnées téléphoniques et électroniques (email),
- le numéro d'immatriculation au registre de commerce,
- etc

Son évaluation est établie à l'aide de la formule suivante:

$$I = \frac{\sum_{k=1}^{n} V_k W_k}{\sum_{k=1}^{n} W_k} \qquad (1)$$

V_k: est une valeur binaire correspondant à la validité de la valeur du k^{ème} paramètre.
W_k : désigne le poids exprimant l'importance du paramètre.
n : représente le nombre de paramètres intervenant dans l'estimation de l'identification.

3.2 La bienveillance (B)

Elle concerne les bonnes intentions de l'hôte visité. Cette métrique est interceptée durant l'interaction de l'agent avec l'hôte visité. La table suivante (*Cf.* Table1)

résume l'ensemble de paramètres (b_1, b_2, b_4, b_4) qui intervient lors de l'évaluation de cette métrique :

Table1. Les paramètres et les facteurs nécessaires à l'évaluation de la bienveillance

Evaluation	Paramètres ou facteurs
$b_1 = \frac{\sum_{i=1}^{n} \frac{2}{p}}{n}$ (2) $b_2 = \frac{\sum_{i=1}^{k} S_i}{k}$ (3) $b_3 = \begin{cases} 1 & \text{si le contrat est valide} \\ 0 & \text{sinon} \end{cases}$ $b_4 = \alpha \times \beta \times f$ (4) $\alpha = \begin{cases} 0.25 & si\ la\ taille\ de\ la\ clé < 128bits \\ 0.5 & si\ 128 \leq taille\ de\ la\ clé < 256bits \\ 0.75 & si\ 256 \leq taille\ de\ la\ clé < 512bits \\ 1 & si\ la\ taille\ de\ la\ clé \geq 256bits \end{cases}$ $\beta = \begin{cases} 0.5\ si\ l'algorithme\ de\ chiffrement\ est\ symétrique \\ 1\ \ si\ l'algorithme\ de\ chiffrement\ est\ asymétrique \end{cases}$ $f = \begin{cases} 0.5 & \text{si l'algorithme de chiffrement est faible(a déja été cassé)} \\ 1 & \text{si l'algorithme de chiffrement est fort} \end{cases}$ $b_5 = \begin{cases} 1 & \text{si le certificat est valide} \\ 0 & sinon \end{cases}$ $B = \frac{\sum_{i=1}^{nb} b_i}{nb}$ (5)	p : est le nombre de fois de saisie pour chaque paramètre évalué. n: représente le nombre de paramètres à tester. Si = 1 si la valeur du ième facteur correspond bien aux offres (remise, baisse de TVA…) proposées, Si=0 sinon. Dans ce cas une exception peut être déclenchée. K : est le nombre des facteurs considéré.

3.3 La réputation (R)

La notion de la réputation concerne l'historique des comportements de l'hôte visité. Elle est subdivisée en deux catégories dénotant la réputation directe et la réputation indirecte. Elle est dite indirecte lorsqu'elle est révélée par une tierce partie de confiance et elle est dite directe quand l'émetteur de l'agent mobile a déjà interagi avec cet hôte et possède donc une idée sur la manière dont il gère ses transactions.

- La réputation directe PK (Personal Knowledge) indique une expérience personnelle. Cette valeur est calculée à partir des notes détenues par l'émetteur de l'agent mobile relatives à des interactions antérieures. Elle est établie par la formule :

$$PK = \frac{\sum_{q=1}^{\theta} NCa_q(j)}{\theta} \quad (6)$$

Où NCaq(j) exprime une note de confiance associée à l'interaction q avec l'hôte j et
θ: détermine le nombre d'interactions antérieures considéré avec $\theta \leq \theta_{max}$

- La réputation indirecte EK (External Knowledge) est relative à une expérience externe. Elle est calculée sur la base des notes reçues à partir d'un ensemble d'hôtes. La réputation externe est évaluée à l'aide de la formule:

$$EK = \frac{\sum_{q=1}^{m} NC_{kq}(j)}{m} \qquad (7)$$

Où NCkq(j) exprime une note de confiance associée à l'interaction q avec l'hôte j.
Sachant que les valeurs des notes de confiance prises satisfont l'ensemble des conditions suivantes :

- La crédibilité dénotant la confiance que possède l'hôte i en l'hôte k doit être importante. Par exemple: CRi(k)>0.8.
- Les notes négatives sont prises en considération. Elles expriment, par exemple, que NCk(j)<0.5.
- La valeur de la crédibilité associée à la note de confiance fournie par l'hôte k doit être acceptable. Par exemple CNCk(x) ≥0.6.

m: indique le nombre des hôtes qui satisfont l'ensemble des conditions.
Nous précisons que si plusieurs valeurs de l'ensemble des notes de confiance fournies par l'hôte k vérifient les conditions citées, la valeur de la note fournie qui possède une crédibilité importante est retenue et la valeur la plus récente est favorisée en cas d'égalité des valeurs.
En vertu des formules (6) et (7), celle de la réputation finale est établie :

$$R = \frac{\varepsilon[\theta \times PK] + \xi[\theta_{max} \times EK]}{(\varepsilon \times \theta) + (\xi \times \theta_{max})} \qquad (8)$$

ε, ξ : représentent les poids accordés respectivement aux valeurs des réputations directe et indirecte.
La valeur de ξ est plus petite que ε pour favoriser l'expérience personnelle de l'hôte émetteur de l'agent mobile.

3.4 Le risque (K)

Dans une transaction dans le domaine du commerce électronique le risque est relié à deux facteurs:

- le risque lié au produit/service : il peut concerner la limite de validité du produit ou encore sa sensibilité telle qu'un produit chimique dangereux, ou un produit alimentaire dont la période de péremption est courte (par exemple les laitages). Ce dernier cas est principalement rattaché au délai de livraison du produit. De plus, cette métrique est liée au degré d'implication révélant les limites de responsabilité.

La durée de validité du produit/service est évaluée par la formule suivante:

$$Dv = d_{lm} - (d_a + d_{lv}) \qquad (9)$$

Où d_{lm} indique la date limite, da désigne la date actuelle et d_{lv} renseigne sur la durée de livraison.
Le facteur τ de vérification de la durée de validité prend la valeur 1 si D_v >0 et 0 sinon. Il est multiplié par un autre facteur dénoté S afin d'ajuster la valeur du risque en fonction d'une durée t soit de péremption du produit/service ou bien choisie par

l'émetteur de l'agent mobile pour des raisons précises. L'estimation finale du paramètre r1 est donnée par la formule ci-dessous :

$$r_1 = \tau \times S \qquad (10)$$

Sachant que S prend l'une des deux valeurs suivantes :

$$S = \begin{cases} 1, & Dv < t \\ 0.5, & Dv \geq t \end{cases}$$

La valeur t est attachée au degré d'implication de l'hôte d'accueil ainsi qu'à la sensibilité et au type du produit/service (alimentaire ou autre).

- le risque financier est proportionnel au montant mis en jeu. La fonction r2 d'estimation de ce paramètre est calculée par la formule:

$$r_2 = \alpha \times x \qquad (11)$$

Où *x* désigne le montant mis en jeu et α est un facteur relatif à la sensibilité du produit/service et au mode de paiement. A titre d'exemple, le paiement par le biais d'une banque diminue le risque.

Nous définissons les trois variantes suivantes de α:

- α est égale à £/x en cas de risque minime,
- α est égale à £/2x en cas de risqué moyen, et
- α est égale à £/3x en cas d'un risque élevé.

£ appartient à l'intervalle [0,1]; cette valeur est spécifiée par l'émetteur de l'agent.

La métrique finale du risque est évaluée selon la formule :

$$K = \frac{r_1 + r_2}{2} \qquad (12)$$

3.5 La disponibilité (D)

La disponibilité est une métrique importante qui intervient en dernier lieu pour confirmer ou infirmer la crédibilité de l'hôte visité. Cette métrique englobe deux paramètres :

- la sûreté des logiciels: l'utilisation des logiciels et des codes sûrs influe sur la disponibilité des systèmes informatiques.
- la compétence d'un hôte: elle est liée à sa capacité de calcul et au débit de connexion. La capacité de calcul est relative à l'infrastructure utilisée.

L'évaluation de cette métrique est directement liée au temps d'exécution:

$$D = \frac{D_{HA}}{D_{EA}} \qquad (13)$$

Où D_{HA} exprime la durée d'exécution du code de l'agent mobile relatif à la phase de collecte des valeurs des paramètres participant à l'estimation de la crédibilité de l'hôte visité, et D_{EA} indique la durée de cette exécution estimée par l'émetteur de l'agent.

À ce niveau, cinq métriques contribuant à l'estimation de la confiance sont définies. La confiance T (Trust) est calculée selon la formule :

$$T = (IW_1S_1) + (BW_2S_2) + (KW_3S_3) + (RW_4S_4) \quad (14)$$

Le paramètre $S_i \in \{0,1\}$ permet de déterminer la cause du manque de confiance. Ainsi, la valeur 0 de S_i indique que la valeur de la confiance estimée pour la $i^{ème}$

métrique est insuffisante. W_i concernent les poids devant être attribués à chaque métrique pour évaluer la confiance.
On remarque qu'au niveau de l'estimation de la confiance la métrique de la disponibilité (D) ne figure pas dans la formule 14, la raison est que cette métrique est impliquée en cas de manque de confiance pour un éventuel traitement en appliquant l'algorithme présenté au niveau de laTable2.
Table2. L'algorithme montrant l'implication de la métrique de la disponibilité

Si (D >1) et (b_1>=1) **alors**
Le degré de disponibilité de l'hôte d'accueil est bas
et donc, tolérer la réexécution de l'agent à nouveau
si ce n'est pas un déni de service.
Sinon
Si (S_1=1) et (S_2=1) et (S_3=1) et (S_4=0) **alors**
Tolérer l'exécution de l'agent et envoyer la
décision si ce n'est pas un déni de service.
Sinon
Annulez l'exécution
Fin si
Fin si

4 La simulation de l'estimation de la confiance

Notre simulation repose sur un ensemble d'hypothèses qui concernent la gestion des relations entre l'émetteur de l'agent mobile, l'agent mobile et l'hôte visité :

- Un contrat est établi entre l'émetteur de l'agent et l'hôte visité.
- L'émetteur de l'agent sauvegarde un ensemble d'informations (les informations initiales) propres à l'hôte visité au niveau d'une base de données locale. Elles sont nécessaires à l'estimation de la confiance. Elles concernent des informations sur le produit ainsi que des informations privées de l'hôte visité (elles sont bien évidement liées à la transaction en cours).
- L'agent mobile ne fait confiance à aucun hôte. La décision de l'accomplissement ou non de la transaction provient de son hôte émetteur considéré comme sûr.
- La réputation est calculée localement par l'émetteur de l'agent en parallèle avec la collecte des valeurs des paramètres par l'agent mobile au niveau de l'hôte d'accueil.
- Les valeurs des métriques obtenues seront sauvegardées au niveau d'une base de données. Elles vont servir au calcul de la réputation lors des transactions futures.

Cette simulation a été modélisée à travers un ensemble de tests. Dés l'arrivée de l'agent mobile au niveau de la plate forme de l'hôte visité, il commence à récupérer l'ensemble des informations nécessaires à l'estimation de la confiance. Cette collecte

se base essentiellement sur l'interaction de l'agent mobile avec l'hôte visité. Elle englobe principalement quatre étapes :

1. Identification initiale (le protocole de sécurité de bas niveau : vérification de l'adresse IP…).
2. Demande de la valeur d'un paramètre par l'agent mobile à l'hôte visité exprimée à travers un questionnaire.
3. Réponse de l'hôte visité à l'agent mobile.
4. Observation de l'hôte visité par l'agent mobile afin de calculer la valeur du paramètre b_1 de la métrique de la bienveillance.

Les trois dernières étapes sont répétées autant de fois qu'il existe de paramètres. Le calcul final de la confiance est réalisé au niveau de l'hôte émetteur qui prend alors la décision de l'accomplissement ou non de la transaction.

Au niveau de cette partie, une application de e-commerce est conçue et déroulée à travers un ensemble de tests à partir desquels nous observons l'influence des différentes métriques sur la valeur finale de la confiance. Quelques cas expérimentaux sont présentés en commençant par celui où l'hôte se comporte honnêtement.

Test1- cas d'un comportement normal: l'hôte d'accueil saisit les informations demandées d'une manière très correcte suivant les exigences de l'agent mobile.

La Table 3 (*Cf.* Table3) présente un exemple d'informations stockées au niveau de la base donnée locale de l'hôte émetteur de l'agent, ainsi qu'un exemple d'informations émises par l'agent mobile vers son hôte d'origine.

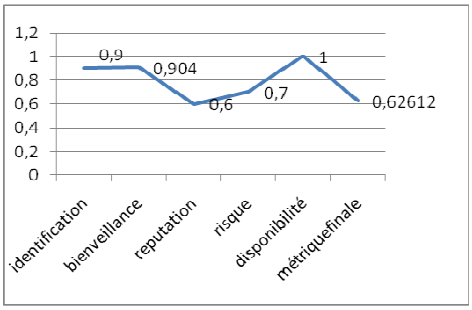
Table3. Comparaison des données stockées au niveau de la base de données locale de l'émetteur de l'agent mobile avec celles émises par l'agent

Paramètres	Valeurs détenues par l'émetteur dr l'agent	Valeurs reçues par l'agent
Numéro Identité	1 Algérie télécom	343 Algérie télécom
Acronyme Mot de passé	Actel 1lata	Actel 1lata
Adresse N° registre	Constantine 85	Constantine 85
Nid TVA tel	89 21321660201	89 21321660201
e-mail Domaine	actel@alg.com Commercial	actel@alg.com Commercial
Prix HT Transport	100 3	100 3
TVA Prix TTC	7 110	7 110
Durée de l'offre Limite géographique	Illimitée Illimitée	Illimitée Illimitée
Hash du contrat Taille de la clé		26640373 0.75
Algo. De chiffrement Algo force		0,5 1
Certificat		1
Date de validité Durée de livraison		02 04 2009 0
Durée réelle		21

L'opération de vérification des paramètres reçus se base principalement sur la comparaison des valeurs de ces paramètres avec les informations stockées au niveau de l'émetteur, en appliquant le diagramme d'activités présenté par la Figure suivante (*Cf.* Fig1).

Le calcul de la valeur de la confiance exige l'attribution de poids aux différentes métriques et paramètres. À titre d'exemple, les valeurs 2, 1, 28 et 25 représentent respectivement les poids associés au mot de passe, à l'acronyme, à la bienveuillance et au risque.

Le Graphe 1 (*Cf.* Graphe 1) montre la nouvelle valeur de la confiance placée en l'hôte visité. Elle est égale à 0.67[1].Dans ce cas, l'hôte visité est considéré comme un hôte de confiance car il possède un degré de confiance supérieur à 0.50.



Graphe1. La représentation graphique des résultats du premier cas de test

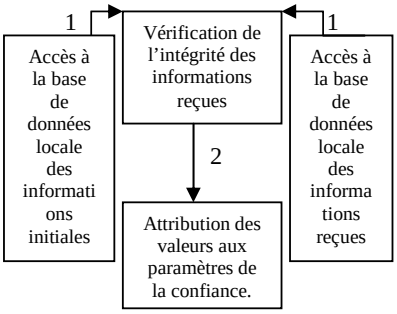


Fig1. L'opération de vérification des informations reçues.

Test2 – cas d'une identification erronée: Dans ce test le problème réside au niveau de la métrique de l'identification, l'hôte visité donne en effet de fausses informations concernant son authentification et essaye de désorienter l'agent pour déduire les valeurs des paramètres demandées. Ce test se caractérise par le fait qu'au moins l'identité de l'hôte visité est correcte grâce au protocole de sécurité de bas niveau.

En évaluant les différentes métriques, on obtient les résultats présentés par le Graphe2 (*Cf.* Graphe 2).

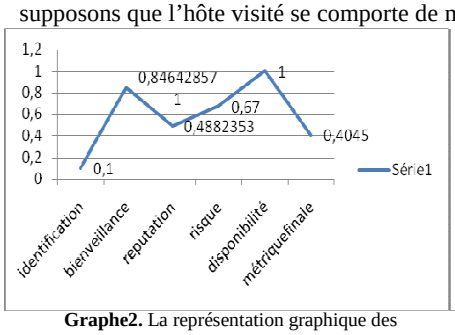
Lorsque l'hôte essaye d'effectuer une mascarade, il tente de trouver les bonnes informations soit par un temps de réponse élevé ou bien par plusieurs saisies des informations demandées. La métrique de la bienveillance se trouve alors influencée et l'attaques est détectée. Dans ce test, malgré que l'hôte essaye de répondre à temps aux requêtes de l'agent mobile, l'attaque est détectée et l'hôte est considéré comme malicieux.

Test3 – cas d'une mauvaise réputation: Dans ce cas si la valeur de la confiance est suffisante, on peut se trouver dans le cas du test1. Toutefois, si la valeur de confiance est insuffisante, on considère l'algorithme de la disponibilité (*Cf.* Table2) en appliquant la seconde condition qui permet l'émission d'une décision positive si le manque de confiance est le résultat d'une mauvaise réputation (le facteur S de chaque métrique est égal à 1 sauf celui de la réputation). Ce cas est illustré par le Graphe3.

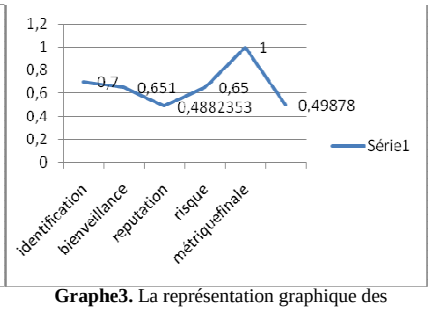
La réputation de l'hôte visité est considérée en tant que métrique secondaire car une forte réputation de l'hôte visité n'implique pas obligatoirement qu'il se comportera d'une manière honnête. De plus, si elle est vue comme une métrique de base, elle augmente la valeur de la confiance calculée. Cette valeur peut donc désorienter le propriétaire de l'agent mobile qui peut prendre une décision erronée.

[1] Pour des raisons de clarté d'échelle la valeur de la confiance est divisée sur 100.

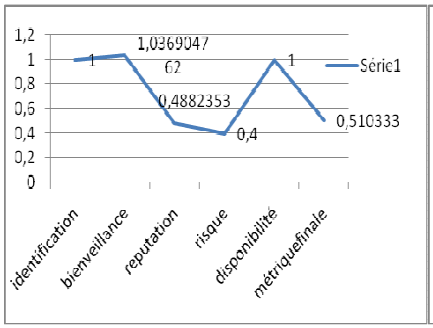
Test4 – cas d'un risque élevé: Un montant important (un paramètre de la métrique du risque) est proposé pour examiner son impact sur la confiance. Dans ce cas, nous supposons que l'hôte visité se comporte de manière honnête.



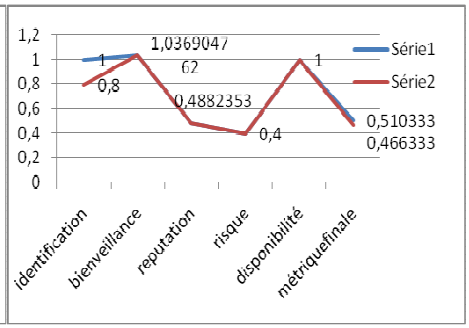
Graphe2. La représentation graphique des résultats du test2



Graphe3. La représentation graphique des résultats du test3



Graphe4. La représentation graphique des résultats du test4



Graphe5. Un graphe mettant en évidence la sensibilité des métriques à une altération des informations reçues.

Nous remarquons au niveau de ce test que la valeur de la confiance diminue de 0.6785 à 0.5103 pour le même comportement de l'hôte visité (test 1) mais cette fois-ci avec un montant important (un risque financier important). De plus, la diminution de la valeur du risque (la valeur 0.59 de la métrique risque n'est pas suffisamment grande) rend la valeur finale de la confiance plus sensible à la détection d'une altération au niveau des informations reçues. À cet effet, une décision totalement différente de la première peut être prise par l'émetteur de l'agent mobile (*Cf.* Graphe5[2]). Ce qui permet d'affirmer que l'emploi des métriques est efficace pour détecter les comportements malicieux des hôtes visités.

Ceci peut d'ailleurs être observé en reprenant le test4 mais cette fois-ci avec une valeur de la métrique de l'identification plus petite (cette dernière se trouve diminuée à cause de la réception de deux paramètres erronés au niveau de la métrique de l'identification).

[2] : Série1 représente le scenario 4 (l'hôte visité est de confiance). Série2 représente le scenario 4 modifié (l'hôte visité est malicieux).

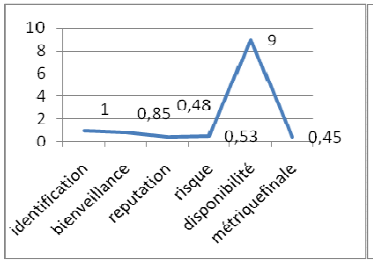
5 L'apport de l'aspect multidimensionnel de la confiance

Notre travail se base sur celui réalisé par Hacini [1] au niveau duquel la confiance est employée comme critère d'adaptation de l'agent mobile afin de le protéger contre des hôtes malicieux. Notre travail se trouve dans le prolongement du travail de Hacini [1] puisqu'il détaille le calcul de la confiance en identifiant les différentes métriques ainsi que leur évaluation. La formule du calcul de la confiance n'est plus linéaire mais multidimensionnelle.

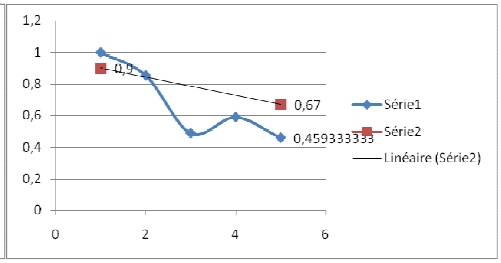
L'apport des métriques dans la technique de protection de l'agent mobile peut être observé au niveau du test 5 suivant :

- L'hôte visité ne pas possède pas de certificat.
- Après plusieurs tentatives, l'hôte a réussi à découvrir les informations exactes exigées par l'agent mobile telles que les données de l'identification.

Au niveau du travail de Hacini et *al* [1] le calcul de la confiance est linéaire et se base sur les valeurs de paramètres pour calculer la confiance de l'hôte visité. Le résultat de ce test peut se résumer comme suit: après la réception des informations envoyées par l'agent à son hôte d'origine, ce dernier estime la confiance et considère l'hôte d'accueil comme un hôte qui possède un degré de confiance appartenant à l'intervalle ambigu (*Cf.* série 2 - Graphe7). Dans ce cas, une décision d'exécution d'un service dégradé (ou nul quand l'hôte est jugé malicieux) est prise. Par contre, le résultat obtenu en appliquant notre métrique de la confiance permet de décider directement que l'hôte d'accueil est un hôte malicieux (*Cf.* Graphe 6) en passant par l'algorithme de traitement du manque de confiance (*Cf.* Table2) (en appliquant la première condition) et la transaction n'est pas effectuée (*Cf.* Série 1 - Graphe7).



Graphe6. Représentation graphique des résultats du test 5



Graphe7. Un graphe montrant l'apport de la prise en compte de l'aspect multi dimensionnelle de la confiance

Nous remarquons qu'au niveau du traitement du manque de confiance la disponibilité (la valeur de D est égale à 9) de l'hôte d'accueil est élevée (*Cf.* Graphe6). Ainsi, ce dernier profite de cette haute capacité de calcul pour déduire les informations exigées par l'agent mobile.

Finalement, l'apport peut être observé au niveau de la prise en compte de l'aspect multidimensionnel de la confiance d'un côté ainsi que dans l'exploitation des métriques au niveau de l'adaptabilité de l'agent mobile du fait que le risque et la réputation peuvent être considérés comme des facteurs d'adaptation.

Nous observons graphiquement (*Cf.* Graphe7) l'apport intéressant de la prise en compte de la caractéristique multidimensionnelle de la confiance.

6 Conclusion

Ce papier englobe la spécification de différentes métriques exploitant les facteurs de confiance permettant l'estimation de la crédibilité de l'environnement d'exécution. Par ailleurs, une validation de ces métriques par le biais d'une simulation a été réalisée. Elle a permis de vérifier l'impact des différentes métriques sur la valeur de la confiance et par voie de conséquence leur influence sur la décision d'accomplissement de la transaction. L'apport de cette estimation de la confiance a été fourni par son caractère multidimensionnel. Ce travail rajoute non seulement un jalon à la protection de l'exécution des agents mobiles mais aussi à la sécurité des transactions dans le e-commerce. Il peut, cependant, être amélioré par le renforcement des métriques. Ceci peut être réalisé par l'implication des paramètres permettant la détection des modifications illégales de l'agent mobile.

Références

1. Hacini, S., Guessoum, Z. and Boufaïda, Z.: "TAMAP: A New Trust-based Approach for Mobile Agent Protection". Journal in computer virology. Springer Paris. ISSN 1772-9890 (print) 1772-9904 (online). Volume 3, N°4/November 2007, p. 267-283 (2007).
2. Meriem, Z., Hacini, S. and Boufaida Z.: " Trust metrics identification for mobile agents protection ". In Proceedings of the 9th International Arab Conference on Information Technology (ACIT'2008), 16-18 December, Hammamet, Tunisia (2008).
3. Hacini, S., Guessoum, Z. and Boufaïda, Z.: "A Trust Based Environment Key for Mobile Agents Code Protection", International Journal of Applied Mathematics and Computer Sciences (IJAMCS), Volume 3, Number 2, p. 68-73 (2006).
4. McKnight, D. H. and Chervany, N. L.: "Conceptualizing Trust: A Typology and E-Commerce Customer Relationships Model ". In Proceedings of the 34th Hawaii International Conference on System Sciences (2001).
5. Lin, F.R, Lo, Y.P. and Sung, Y.W.:" Effects of Switching Cost, Trust, and Information Sharing on Supply Chain Performance for B2B e-Commerce: A Multi-agent Simulation Study". Proceedings of the 39th Hawaii International Conference on System Sciences (2006).
6. Lee, H.Y., Ahn and H., Han, I.: "Analysis of Trust in the E-commerce Adoption". In Proceedings of the 39th Hawaii International Conference on System Sciences (2006).
7. Gomez, M., Earle, C. B., Carbo, J.: "Trust dynamics, motivational attitudes and epistemic actions" Copyright 2007, American Association for Artificial Intelligence (2007).
8. Li Xiong and Ling Liu: "A ReputationBased Trust Model for PeertoPeer eCommerce Communities". USA. ACM 158113679X/03/0006. *EC'03*, San Diego, California, June 9–12, (2003).
9. Carbone, M., Nielsen, M. and Sassone, V.:"A Formal Model for Trust in Dynamic Networks". In Proceeding of 1st International Conference on Software Engineering and Formal Methods (SEFM'03), p. 54–63, Brisbane, Australia (2003).
10. Grandison, T. and Sloman, M. :"Specifying and Analysing Trust for Internet Applications". 2nd IFIP Conference on e-Commerce, e-Business, e-Government, I3e2002, Lisbon Oct. (2002).
11. Josang, A. and Presti, S. L.: "Analysing the relationship between risk and trust". In: Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings Volume LNCS 2995/2004. P. 135–145 (2004).
12. Resnick, P., Zeckhauser, Friedman, R. E. and Kuwabara, K.: "Reputation systems". Communications of the ACM 43 p. 45–48 (2000).
13. Bhatnagar, A., Misra, S. and Rao, H.R.: "On Risk, Convenience and Internet Shopping Behavior, Association for Computing Machinery", Communications of the ACM, 43, 11, p. 98-108 (2000).
14. Ruohomaa, S. and Kutvonen, L.: "Trust Management Survey" P. Herrmann et al: iTrust 2005, LNCS 3477, p. 77–92, 2005.Springer-Verlag Berlin Heidelberg (2005).

Distributed Feature Selection: benchmarking collaboration protocol

M. A. Esseghir[1,2], Y. Slimani[1], and G. Goncalves[2]

[1] Faculty of Sciences of Tunis
Department of Computer Science
Campus Universitaire, 1060 Tunis, Tunisia.
mohamedemir@gmail.com,yahya.slimani@fst.rnu.tn,gilles.goncalves@univ-artois.fr
[2] Université d'Artois, Faculté des Sciences Appliquées
Laboratoire de Génie Informatique et Automatique de l'Artois (LGI2A)
Technoparc Futura, 62400 Béthune, France

Abstract. Feature selection (FS) refers to the process of identification of salient features for a specific/ensemble learning scheme. Numerous studies were proposed to achieve such a goal. Nevertheless, the search effetiveness for the targeted solutions might struggle particularly when they are trapped within local minimas. In this paper, we are interested by, diversification and collaborative search. To this end, we propose a distributed collaborative model for feature selection and we study, in depth the effect of collaboration on both search evolution and final results. Some of distribution aspects are figured out empirically and encourage such a modeling perspective.
Keywords: Data mining, Feature selection, Machine learning, Parallel genetic algorithms, Combinatorial optimization.

1 Introduction

Machine learning refers to the process entailing the extraction of useful knowledge from the available large volumes of data by applying analytical, numerical as well as symbiotic methods and tools to derive patterns, statistical or predictive scheme[1]. Various kind of models have been proposed under the label of *learning from data*. They could be, typically, categorized within three learning paradigms : *(i)* Supervised learning or classification, *(ii)*unsupervised learning so-called clustering, and *(ii)* semi-supervised learning. In this paper, we are concerned with supervised classification problems. From this point of view, we can define a classification technique as a process predicting the classes (*i.e* models outputs, target variables) of unseen data based on patterns learned from available instances [2].
The increasingly large data sets provided by numerous applications have been posing unprecedented challenges to knowledge discovery community and especially in data mining, machine learning and statistical analysis. Furthermore, the emerging data high throughput is being increasing the need to a widespread areas of applications for a such learning schemes as analytical/predictive tools and decision support systems as well [1]. We can cite, for example, massive data provided by the web for a particular context, bioinformatics and especially micro-arrays data. Experimental studies have confirmed that data mining techniques are unable to handle such amounts of data [3]. Several data mining learning tasks are faced with the problem of selecting the most relevant features to pilot learning or/and classification processes. In this case, human expertise is often required to convert raw data into a set of useful features. Our emphasis is on automatic feature selection methods that consolidates the trade-off between prediction accuracy and generalization[1] for the selected subset of features that would drive the learning stage.
Research on feature selection is motivated by the followings [3, 4]: (i) removing redundant, irrelevant, and noisy features; (ii) avoiding over-fitting in learning;(iii) providing comprehensive models; (iv) speeding-up modeling techniques (*i.e.* classifiers) and improving yielding knowledge quality (*i.e.* improving learning accuracy and generalization).

Nevertheless, looking for such a relevant subset require effectiveness in search and the selection of suitable criterion. Consequently, the search that implies the exploration of the space of feasible sets could yield a prohibitive computational cost, especially when we refer to the NP-hard problem property (*see* Section 2). Besides, the alleviating stochastic heuristics are not optimal, and might be trapped in some of the search space local minimas.
The main contributions of this paper can be summarized as follows: (i) the design of a distributed model involving heuristic collaboration and parallel evolution (ii) the exploration of the collaboration mechanism as well as associated parameters. Distribution aspects and design consideration are detailed on the next sections.
The paper is organized as follows. Section 2 formalizes the feature selection problem and reviews representative approaches. Section 3 describes our proposed distributed model. Section 4 compares and assesses the model's empirical results. Section 5 concludes this paper and presents some directions of future research.

2 Feature selection :Background

2.1 The FS problem

Consider a data set D with N attributes such that $\parallel N \parallel = n$, and let M ($M \subseteq N$) be a subset of N. Let $J(X)$ the relevance assessment function applied to the subset X. The $j((.)$ should represent the selected criterion measure for the evaluation. The problem of feature selection states the selection of a subset Z such that:

$$J(Z) = max_{X \subseteq N} J(X) \tag{1}$$

In other words, the retained features should be compact and representative of the dataset objects or the underlying context as well. This can be done by ignoring redundant and/or irrelevant attributes by keeping the minimal information loss.
One important question that might be addressed here, is how to categorize features into the following groups: relevant, irrelevant, redundant and noisy features ? To reply to this question, we need to recall definitions of relevance property [5].
Definition 1.
Strong Relevance: An attribute x_i is strongly relevant if its removal yields a deterioration of the performance in the classification.
Definition 2.
Weak Relevance: An attribute x_i is weakly relevant if it's not strongly relevant and there exists a subset of features X such that the performance on $X \bigcup \{x_i\}$ is better than the performance on X.

Therefore features that are neither strongly relevant nor weakly relevant are irrelevant.
Solutions that have been proposed to tackle this problem have investigated a variety of both evaluation metrics and search approaches [6, 7, 3].
If we consider a given dataset of n features, the exploration would require the examination of 2^n possible subsets. Consequently, the search through the feasible solutions search space is a combinatorial problem [4]. An exhaustive exploration of the feature space seems to be impractical, especially, when the n became large. In fact, several heuristic strategies have been proposed to reduce data dimensionality. Among these heuristics, we can mention the successful application of the Branch and Bound, randomized search and genetic algorithms [8–11].
In this paper, we consider the feature selection problem as an optimization problem guided by the accuracy of the selected features. Consequently, we consider that the goal of feature selection is to maximize the accuracy of selected features. A number of relevance measures have been proposed to identify the features having the highest predictive power for a given target [4, 7]. These measures and techniques are presented in the following sections.

2.2 State of the art approaches

Filters as local search

Considered as the earliest approach to feature selection, filter methods discards irrelevant features, without any reference to a data mining technique, by applying independent search criterion to find appropriate features [7]. FOCUS [12] and RELIEF [13] are considered as the pioneer filter approaches. FOCUS, was originally designed for boolean variables, exhaustively explores the whole set of possible feature subsets by discarding one feature at one time in turn. FOCUS looks for the minimal set of features able to predict pure classes. Nevertheless, RELIEF is a randomized search procedure that attempts to find weak relevant features. Each feature is assigned to a weight that reflects its ability to distinguish among the class values. Features are ranked by weight and those that exceed a threshold parameter are selected to form the final subset. This method is not well sweeten for datasets with redundant or highly correlated features [14]. The main advantage of the filter methods is their reduced computational time which is due to the *simple* independent criterion used as an evaluation function. However, the issues of the induction algorithm are not taken into account, hence the features selected are, in most of the time, independent of the *classifier* which will pick up these features to use them as inputs. Since the relation between variables does not matters for filter they are usually blamed for efficiency concerns.

Wrappers : search based on classification scheme

Comparatively to filters methods, when feature selection is based on a wrapper, the exploration of the feature space is driven by both classification accuracy returned by the selected subset of features and the involved search technique. Typically, a classifier is used as a part of the evaluation process by awarding the retained subsets according to their target predictive power. The wrapper methods often have better results than filter ones because they are tuned to the specific interaction between an induction algorithm and its training data [5, 15]. Kohavi *et al.* [16] were the firsts to advocate the wrapper as a general framework for feature selection in machine learning. ID3 [17] and C4.5 [18] were, separately, have used wrappers and both forward selection and backward elimination as heuristic search. On the same way, numerous studies have used the above framework by either changing the wrapper or the search technique. We can remark that recent proposed feature selection techniques have focused on both genetic algorithms and neural networks [8, 5, 10, 19].
Nevertheless, feature selection methods based on wrappers are computationally expensive compared to filters, due to the cost of running the classification algorithm [1]. Some heuristic strategies have been investigated to address the problem of finding the best subsets of features [7, 3].

Distributed approaches

The parallel scatter search model [20] and parallel tabu search [21] could be considered as the earliest attempts of feature selection distribution. Both approaches, could be considered as distributed wrappers relying on local search. The first applies a greedy search while the second uses the tabu search as enhanced heuristic local search.

3 The proposed model

When widely accepted wrapper approaches succeed to outperform the existing classical filter approaches, particularly when relationships among feature and subset selection matters, they did not succeed to afford the expected efficiency, especially in search space exploration. In most of the cases the whole process either suffer form local minima or struggle to avoid premature convergence. In both cases, final results are affected. At the same time, we thought that global modeling and especially distributed approaches would be interesting

to investigate. In our case, anf for the problem of feature selection the nature of the search space endowed with a large number of solutions (subset of features) having providing similar classification performance, could motivate such modeling orientation. we propose an approach based on collaborative wrapper model and designed as distributed system. In fact the prohibitive exploratory computational burden of the subset search space could be both alleviated by distribution over the different distributed wrappers. Algorithms 1 and 2 despcribe respectively the main island program and the behavior of a wrapper instance assigned to an island. The following subsections detail design aspects in relation with model distribution and collaboration.

3.1 Distributed search

In this paragraph, we emphasize on the nature and the organization of distribution scheme adopted for the management of the involved FS resolver. Since the search effectiveness through the feasible feature subset solutions requires a robust exploratory process, we chooses a distribution scheme based on one the most successful search technique in optimization and particularly for the feature selection problem: Genetic algorithms[3, 4]. Indeed, the exploration within the distributed framework will be enhanced by both parallel exploration and collaboration throw search. Numerous Distributed genetic approaches have been proposed for similar problems and combinatorial optimization in general. Each of which differs for the others, by the workload distribution and the underlying collaboration protocol.
As mentioned above, the wrapper instances will be distributed according to the island model scheme[22]. In fact, the whole model defines a set of islands evolving subpopulations.In our case, an wrapper instance based on a genetic algorithm will be assigned to each island. The genetic evolutionary process evolves solutions represented as binary strings. Each bit state corresponds to the presence of the associated attribute in the considered set of feature ($i.e$ the i^{th} bit is set to 1 implies the selection of the i^{th} attribute in the proposed solution). The evaluation procedure assigns to each solution a fitness level that will be used by selection procedures to compare and assess solution. Since wrapper relies on the classification metrics to asses the subsets performances, we combine two measures for the fitness evaluation. The fitness function is presented by the equation 2:

$$fitness = (ICI + TMSE)/2 \tag{2}$$

where ICI and $TMSE$ denotes, respectively, the instances misclassification rate and the mean square error over the test set. The iterative genetic process entails evolution of the initial population in such a way that a part of the population will be renewed using a set of evolutionary operators. The regeneration starts with the selection of a subpopulation. Here, we opted for a tournament selection[23] and this is to keep an equilibrated selection pressure over the hole process. Once the solutions selected the reproduction operator will be applied using respectively one point crossover and the classical mutation operators. The resulting new solutions will replace some of the existing solution using a reverse tournament procedure. An additional stage should be provided when we move to distributed islands. This stage handles solution exchange and communications aspects. The migration policy is detained by the next subsection.

3.2 Collaboration protocol

When multi-instances evolves in parallel or in a distributed scheme, the search space is being explored in parallel by sub-populations. Indeed, each population explores a region of the search space. In such a context, the information exchange among islands might guide the search through each GA instance. Each instance might also, even, compare their results toward exchanged solution or use it within its evolutionary process. Consequently, the collaboration would constitute a new factor of diversification -in the case where the where foreign solution are accepted- that will be added to the classical evolutionary operators

Algorithm 1: Main Distributed Program

Input:
Nb_{is}: Number of islands
N: Number of solution to generate
$Collab_P$: Collaboration protocol parameters
GA_{params}: Initial GA parameter values
D: Dataset
Output: S' : Population of the last generation

```
Begin
    S= GenerateInitialSolutionSet(N)
    i=0
    While i< Nb_is do
        s_i=GetSubpopulation(S,i)
        Island_i=createGAInstance (s_i, GA_params, Collab_P)
        Island_i.Start()
        i = i + 1
End
```

Algorithm 2: GA Instance

Input:
S: Initial solutions set
Cla: Classifier
$Collab_P$: Collaboration protocol parameters
$Maxgen$: Total number of iterations
GA_{params}: Initial GA parameter values
D: Dataset
Output: S' : Population of the last generation

```
Begin
    Population P=S, P_tmp=Ø

    i=0
    While i<Maxgen do
        // Evolution Stage
        P_tmp=Select (P, GA_params)
        Crossover(P_tmp, GA_params)
        Mutate(P_tmp, GA_params)
        Evaluate(P_tmp, Cla, D)
        Replace(P_tmp, P, GA_params)
        // Collaboration Stage
        M=SelectMigrantSolutions(P, Collab_P)
        Migrate(M, Collab_P)
        F=GetForeign(Collab_P)
        Integrate(F, P, Collab_P)
        i=i+1
    Return (S'=P)
End
```

(*i.e* crossover, mutation,local search,*etc*). The collaboration could be affected by different aspects (factors) in relation with the exchange nature. In fact, it is ranging from totally isolated islands (no communication) to the solution exchange in the multi-cast mode where each island is connected to all remaining distributed component. In our proposal the islands are organized in a ring scheme. This structure assigns to each island a sender which provide migrant solution and a receiver that could accept selected solution from the current island.By this way each wrapper is both sender and receiver The choice could be argued by the reduction of the communication overhead. the collaboration stage is represented in Algorithm 2 by four functions ranging from line 13 to line 16.
After the evolution stage, we should pick up a migrant solution which might be duplicated on another island. In our case, only one randomly selected solution will be elected for migration. We thought that the selection of the best solution could increase the selection pressure to a high level and consequently might led to premature convergence. The frequency of migration might also have a direct impact of the whole process evolution The replacement applies the same scheme as for internal population regeneration: reverse tournament. In a such scheme the migrant solution could replace the worst solution provided by a set of a randomly selected solution from the island current population.

4 Empirical study

The aim of this section is to evaluate our distributed model on different well known benchmarks. Assessment procedure will takes into account the classification accuracy over unseen data (generalization) as well as the degree of dimensionality reduction. In the following sections, we will start by shedding some light on the aspects in relation with model implementation. Afterwards, the model will be assessed, through three steps. With each step, we show how the distributed model affects results according to the considered criterion.

4.1 Benchmarks

Our empirical study, is driven by three benchmark data sets provided by the UCI repository [24]. We should note, that's for evaluation needs two datasets are derived from each dataset by dividing its instances between training and test data sets proportionally to the class distribution. the proposed data sets for evaluation uses search spaces ranging from middle sized (*i.e.* 2^{22} possible combinations) to almost large search spaces (i.e. *i.e.* 2^{69} possible combinations).

4.2 Distribution analysis

In the first experiments set, we assess empirically the distributed model and we compare it to the centralized genetic algorithm. Since wrappers and specially those based on genetic algorithms outperform almost filter approaches, we opted for the genetic algorithm and the Naive Bayes as a wrapper reference for feature selection evaluation. By this way, the classical genetic wrapper for feature selection, is compared to a set of distributed island models. The number of wrapper instances (islands) is ranging from 4 to 16. Figure 1 shows for each model, the evolution of the fitness over iterations on each of the selected benchmarks. We should note that the reported fitness values correspond to the best solution fitness for the population in each iteration. As a first result we can see clearly, the domination of the distributed approaches over the centralized wrapper even when the centralized GA start at relatively good level.. In fact, the final results of distributed islands are improved for all the benchmarks.
Furthermore, another more interesting result can be pointed out. The final result for centralized wrapper can be obtained more rapidly by the distributed wrappers. For example, with audiology benchmark, the final fitness level could be easily outperformed after after thirty iterations. When we could reach advanced level of search after relatively small number of iterations, this would alleviate computational workload by early stopping. When we

compare only distributed islands, we could depicts the superiority of 4 and 8 islands final results. This property, is particularly interesting when we refer the respective computational burden generated b each model. Consequently, four and eight islands we could get a good trade-off between computational cost and results levels. By this way, discovered effectiveness in search property for distributed modeling, could encourage tackling more difficult and large feature selection problems.

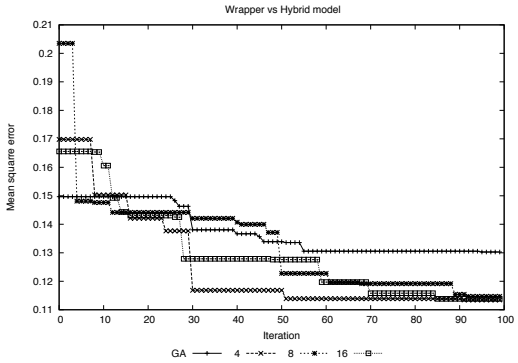
The second set of experiments, studies the impact of population size on the final results for island models. Table 4.2 summarizes islands performances on each benchmark , when varying the population size. In this case the evaluation is based on a couple of criterion: effectiveness of the selected subsets in classification (error %) and the number of the selected features(feat.). Besides, the classification performance of the original set of feature (whole set) before feature selection, and total number of features are provided. we should note that the reported classification performance are expressed in terms of generalization accuracy and correspond to islands of 8 wrappers. When considering best results we could report the effectiveness of large populations (i.e. populations of 30 and 50 solutions). Nevertheless, when we compare dimensionality reduction levels, the size of the population have not a great impact on feature subset criterion.

Population size	**20 sol.**		**30 sol.**		**50 sol.**		**Whole set**	
Data	$\#feat.$	$Error(\%)$	$\#feat.$	$Error(\%)$	$\#feat.$	$\%Error$	$\#feat.$	$\%Error$
Audiology	**36**	26.42	40	**25.48**	41	25.93	69	29.06
Horse_colic	8	22.58	9	21.87	**7**	**21.79**	22	32.84
Ionosphere	9	11.15	**8**	11.31	**8**	**10.92**	34	39.81

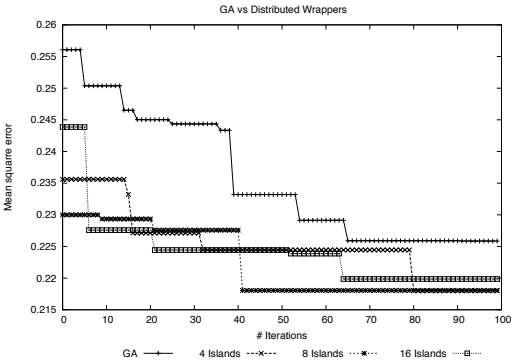
Table 1. population size effect

4.3 Migration impact

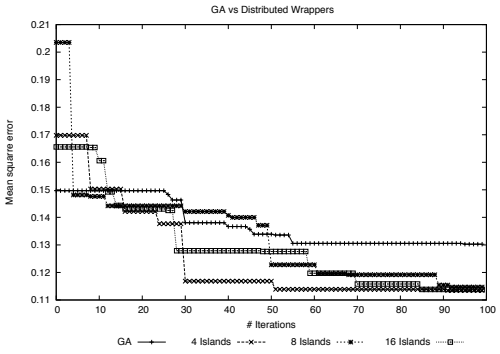
As the the distributed models have confirmed empirically theatrical assumptions, we will try to figure out one of the important aspect of distribution: the migration of solution between islands. For this purpose, we will compare distributed wrappers with different communication scheme. The number of solutions elected for migration is set to one. This choice could be argued b the fact that when we refer to similar distributed modeling approaches designed for combinatorial optimization problem this parameter is suggested to set to 1. Also, we taught, that the effect of this factor should be compared to the whole distributed model parameters, and this could be done in further study. In these experiments, we are only concerned by the study of the migration frequency impact on the evolution of search space exploration as well as on the final results. Figure 2 compares isolated island with a different level of exchange frequency frequencies (i.e. the migrant solution is sent after 8, 16 and 32 iterations) for the three benchmarks. With isolated islands the wrappers can't exchange any solution. With a such model the wrapper relies only of distribution aspects (*i.e.* exploration of more than one region of the search space). When we compare isolated wrapper to the remaining collaboration scheme, we can clearly see the effect of collaboration. In fact, in most of the cases the collaborative wrappers outperform isolated ones. This, can be explained in part, by the fact that the collaboration scheme enhance evolution process diversity b the new added solutions. When the we compare the different collaboration scheme, the best results are , in most of the cases provided by the exchange frequency done after 8 and 16 iterations. The largest exchange frequency did not bring enough diversity to island, and this could be explain its relative enhancement.



(a) Audiology (100it.)

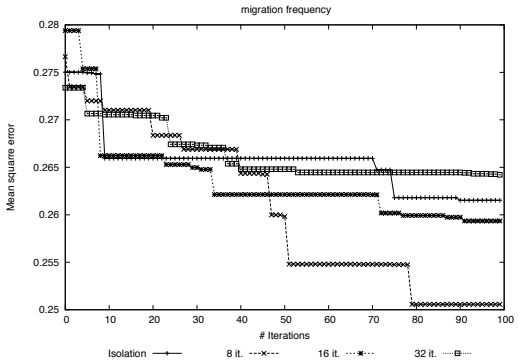


(b) Horse_colic (100it.)

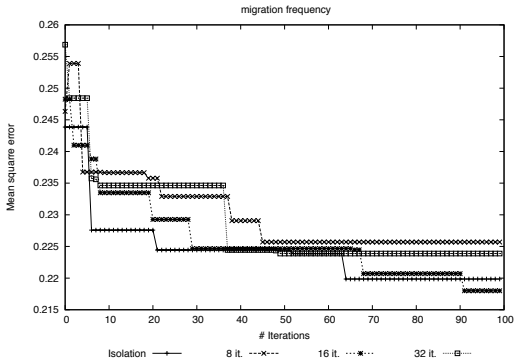


(c) Ionosphere (100it.)

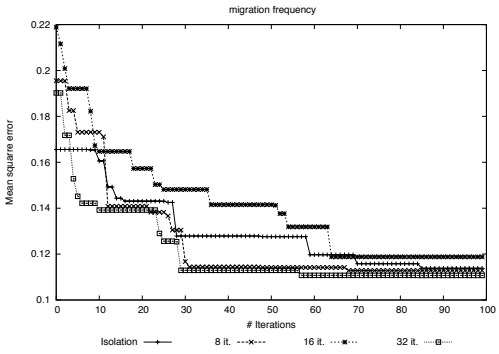
Fig. 1. Distributed Genetic Algorithms



(a) Audiology



(b) Horse_colic



(c) Ionosphere

Fig. 2. Migration frequency analysis

5 Conclusion and perspectives

In this paper, we proposed a distributed model based on both wrapper for the feature selection problem. Three features characterize this proposed model: (i) it's distributed; (ii) it endows classical wrappers by a collaboration behavior (iii) it is based on a successful distribution scheme . With this distributed model, we have empirically assessed both distribution and collaboration aspects on three benchmarks. From the results of our experimentations, we showed how distributed and collaborative modeling could enhance the feature selection efficiency. In the future, we will study the scalability as the hybridization of our distributed model.

References

1. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann (1999)
2. da Silva, J.C., Gianella, C., Bhargava, R., Kargupta, H., Klush, M.: Distributed data mining and agents. Engeneering Application of Artificial Intelligence **18** (2005) 791–807
3. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction, Foundations and Applications. Series Studies in Fuzziness and Soft Computing. Springer (2006)
4. Liu, H., Motoda, H.: Computational methods of feature selection. Chapman and hall/CRC Edtions (2008)
5. Yu, S.: Feature Selection and Classifier Ensembles: A Study on Hyperspectral Remote Sensing Data. PhD thesis, The University of Antwerp (2003)
6. Liu, H., Dougherty, E.R., Dy, J.G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Parsons, L., Zhao, Z., Yu, L., Forman, G.: Evolving feature selection. IEEE Intelligent Systems **20** (2005) 64–76
7. Guyon, I., Elisseff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
8. Huang, C., Wang, C.: A ga-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications **31** (2006) 231–240
9. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data Engineering **15** (2003)
10. Hong, J., Cho, S.: Efficient huge-scale feature selection with speciated genetic algorithm. Pattern Recognition Letters **27** (2006) 143–150
11. Liu, H., Li, J., Wong, L.: A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Informatics **13** (2002) 51–60
12. Almuallim, H., Dietterich, T.G.: Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91). Volume 2., Anaheim, California, AAAI Press (1991) 547–552
13. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the Ninth International Conference on Machine Learning (ICML-92) San Francisco, Morgan Kaufmann (1992) 249–256
14. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324
15. Hall, M.A.: Correlation-based feature selection for machine learning. PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand (1998)
16. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: International Conference on Machine Learning. (1994) 121–129
17. Quinlan, J.R.: Simplifying decision trees. Int. Journal of Man Machine Studies (1987) 221–234
18. Quinlan, J.R.: Induction of decision trees. In Buchanan, B.G., Wilkins, D.C., eds.: Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems. Kaufmann, San Mateo, CA (1993) 349–361
19. Esseghir, M.A., BenYahia, S., Abdelhak, S.: Localizing compact set of genes involved in cancer diseases using an evolutionary conectionist approach. In: European Conferences on Machine Learning and European Conferences on Principles and Practice of Knowledge Discovery in Databases. ECML/PKDD Discovery Challenge. (2005)

20. López, F.G., Torres, M.G., Batista, B.M., Pérez, J.A.M., Moreno-Vega, J.M.: Solving feature subset selection problem by a parallel scatter search. European Journal of Operational Research **2** (2006) 477–489
21. Cerveròn, V., Fuertes, A.: Parallel random search and tabu search for the minimal consistent subset selection problem. In Springer, ed.: Randomization and Approximation Techniques in Computer Science. Volume 1518. (1998) 248–259
22. Engelbrecht, A.P.: Computational Intelligence: An Introduction. John Wiley & Sons,InterEditions (2007)
23. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison Wesley (1989)
24. Blake, C., Merz, C.: UCI repository of machine learning databases. (1998). (http :/www.ics.uci.edu/ mlearn/MLRepository.html)

Optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème linéaire stochastique multi-objectifs

Kahina Ghazli and Mustpha Moulaï

Institut des Mathématiques
Université des Sciences et de la Technologie Houari Boumediane (USTHB)
Laboratoire LAID 3, BP 32, Bab Ezzouar 16111, Alger, Algerie

Résumé Cette etude porte sur le problème d'optimisation dans l'ensemble des solutions efficaces d'un problème linéaire stochastique multi-objectif (MOSLP). Après avoir converti le MOSLP en un problème équivalent déterministe (MOLP) en adaptant l'approche de recours à 2-niveaux, une technique de pivotage est appliquée pour générer une solution efficace optimale globale sans énumérer toutes les solutions efficaces du MOSLP. Cette méthode combine la méthode d'Ecker et Song et la méthode L-Shaped. Un exemple numérique est donné pour illustrer la méthode.

Mots clés : Programmation multi-objectifs, Programmation stochastique, Modèle de recours à 2-niveaux, Optimisation sur l'ensemble des solutions efficaces.

1 Introduction

Nous considérons le problème linéaire stochastique multi-objectif (**MOSLP** ; **M**ultiple Objective **S**tochastic **L**inear **P**rogramming) suivant :

$$\left|\begin{array}{rl} \text{“min”} & Z_k = C_k(\xi)\, x \quad k = 1, \ldots, p \\ s.\grave{a}. & T(\xi)x = h(\xi), \\ & Ax = b, \\ & x \geq 0, \end{array}\right. \tag{1}$$

avec
$C_k(\xi),\ T(\xi),\ h(\xi)$ sont des vecteurs aléatoires de dimensions respectives $(1 \times n), (m_0 \times n),\ (m_0 \times 1)$, définis sur l'espace de probabilités $(\Xi, \mathcal{F}, P)$, $S = \{Ax = b\ :\ x \geq 0\}$, un polyèdre convexe et déterministe des décisions x, A et b sont des vecteurs déterministes de dimensions $(m \times n),\ (m \times 1)$ respectivement.

La plupart des méthodes de résolution des MOSLPs transforment d'abord le problème (1) en un problème déterministe et puis le résoudre par une méthode interactive, dans ce contexte, nous citons la méthode PROTRADE (Goicoechea et al. , 1976), STRANGE (Teghem et al. , 1986) et

PROMISE (Urli et Nadeau , 1990).
La méthode STRANGE-MOMIX développée par Teghem (1990) traite le problème MOSLP à variables entières (MOSILP), à notre connaissance, ce cas a été aussi considéré par Moulaï et Amrouche (2006) et par Abbas et Bellahcene (2006).

Classiquement, la résolution de ces problèmes passe par la détermination de l'ensemble des solutions efficaces. Néanmoins, dans la pratique, il peut s'avérer que l'ensemble des solutions efficaces soit très grand et il devient impossible pour le décideur de choisir le meilleur compromis en termes de ses préférences. L'optimisation d'un critère, qui explique les préférences du décideur, sur l'ensemble des solutions efficaces constitue, dès lors, un sujet de recherche essentiel dans ce domaine. Cependant, à notre connaissance, aucune méthode dans la littérature n'a été proposée à ce sujet.

Connaissant a priori la structure des préférences du décideur, nous proposons, dans ce travail, une méthode exacte pour résoudre le problème d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces du MOSLP. Cette méthode combine la méthode de Ecker et Song (1994) et la méthode L-Shaped (Van Slyke et Wets , 1969) qui traite les problèmes linéaires stochastiques à deux niveaux avec recours.

Ce problème est définit mathématiquement comme suit :

$$(P)\left|\begin{array}{ll}\max \phi(x) = d^T x \\ s.à. \quad x \in E\end{array}\right.$$

où
d, un vecteur de dimension $(1 \times n)$ $(d \in \mathbb{R}^n)$.
$\phi : \mathbb{R}^n \longrightarrow \mathbb{R}$, une fonction linéaire déterministe continue à maximiser.
E, l'ensemble des solutions efficaces du **MOSLP**.

difficulté du problème : Une complication additionnelle, à savoir l'incorporation de l'aspect stochastique introduit par le MOSLP, est superposée au problème classique à savoir, l'optimisation sur l'ensemble des solutions efficaces d'un problème linéaire multi-objectifs (MOLP) qui apparient à la classe des problèmes d'optimisation globale (non convexe) dû à l'ensemble des solutions efficaces qui est non convexe constitué de l'union de certaines faces du polyèdre S.

Notre méthode comporte deux grandes phases : la phase de modélisation qui consiste à transformer le MOSLP en un problème équivalent déterministe en adaptant l'approche de recours à deux niveaux (Dantzig , 1955; Beale , 1955) et la phase de la résolution qui consiste, principalement à :

1. Dans un premier temps, déterminer une solution optimale du problème relaxé de (P) avec **la méthode L-Shaped** et tester l'efficacité du point retourné.

2. En second lieu, étant donné une solution efficace courante, utiliser **la technique de pivotage d'Ecker et Song**, qui inclue **la technique d'Ecker et Kouada** (Ecker et Kouada , 1978) pour l'identification des arêtes efficaces incidentes à une solution donnée, pour la recherche de la meilleure solution efficace en termes des préférences du décideur.

2 Problème déterministe équivalent du MOSLP

Dans notre travail, nous avons adapté le modèle de recours à 2-niveaux pour transformer le problème (1) en un problème déterministe équivalent. Cette approche suppose que la stochasticité du MOSLP est contenue dans un vecteur aléatoire ξ avec une distribution de probabilités discrète et finie, $\{(\xi_\imath, p_\imath),\ \imath = 1, \ldots, N\}$. Chaque scénario $\xi_\imath$ est caractérisé par une composante $(T(\xi_\imath), h(\xi_\imath), q(\xi_\imath))$.
Nous supposons que la matrice de recours $W(\xi)$, qui correspond aux coefficients des variables de recours y dans le problème de recours est fixe, ($W(\xi) = W$, les coefficients sont les mêmes pour tout les scénarios).

1. Nous associons à chaque scénario $\xi_\imath$, un critère Z_{ki}, une matrice $T_\imath$ et un vecteur $h_\imath$, pour mettre en évidence toutes les conséquences de toutes les réalisations possibles.
2. Revenant à l'idée de l'approche de recours, nous supposons que le décideur peut préciser les coûts des pénalités, $q_\imath = q(\xi_\imath)$ des contraintes violées $y_\imath$ (le coût de l'action corrective menée afin de maintenir la faisabilité du modèle à titre d'exemple, en pratique, cela peut être le coût d'achat d'une quantité d'un article pour couvrir la demande d'un client au cas ou la quantité que l'on produit ne satisfait pas la demande). Nous ajoutons, à chaque critère Z_{ki}, la fonction de recours $Q(x, \xi_\imath)$,

$$\begin{aligned} Q(x, \xi_\imath) = \min\ & q_\imath^T\ y_\imath \\ & s.à.\ W y_\imath = h_\imath - T_\imath\ x,\ \ y_\imath \geq 0. \end{aligned}$$

On aura, ainsi, à minimiser l'espérance mathématique du coût total, $\widetilde{Z}_k = \mathbb{E}[Z_k + Q(x, \xi)]$, $k = 1, \ldots, p$.

Le problème stochastique multi-objectifs à 2-niveaux avec recours (**TSMOSPR** : **T**wo **S**tage **M**ultiple **O**bjective **S**tochastic **P**rogramm with **R**ecorse) résultant en termes des décisions du premier niveau est décrit par :

$$\left| \begin{array}{lll} \text{“min”} & \widetilde{Z}_k = Z'_k + Q(x), & k = 1, \ldots, p, \\ s.à. & Ax = b, & \\ & x \geq 0, & \end{array} \right. \tag{2}$$

avec

$$Z'_k = \mathbb{E}[Z_k] = \sum_{\imath=1}^{N} p_\imath Z_{ki} = \sum_{\imath=1}^{N} p_\imath C_k(\xi_\imath) x = \mathbb{E}[C_k(\xi) x], \tag{3}$$

et

$$Q(x) = \mathbb{E}[Q(x,\xi)] = \sum_{\imath=1}^{N} p_\imath \; Q(x,\xi_\imath). \tag{4}$$

En appliquant la reformulation de L-Shaped par l'introduction d'une variable auxiliaire θ pour estimer la valeur de la fonction de recours $Q(x)$, nous obtenons la formulation équivalente du TSMOSPR suivante que l'on note $(MOLP)$:

$$(MOLP) \left| \begin{array}{ll} \text{“min”} & \widetilde{Z}_k = Z'_k + \theta, \quad k = 1, \ldots, p, \\ s.\grave{a}. & x \in \widetilde{S}, \\ & \theta \geq Q(x), \end{array} \right. \tag{5}$$

avec
$\widetilde{S} = \{x \in S \mid \sigma^T T_\imath x \geq \sigma^T h_\imath, \; \imath \in \{1, \ldots, N\}\}$.

3 Test et coupe de faisabilité

Pour verifier la faisabilité des problèmes du second niveau $Q(x,\xi)$, on doit déterminer le vecteur σ, en traitant le problème

$$\sigma \max_{\sigma} \{\sigma^T (h_\imath - T_\imath x^0) \;\mid \sigma^T W \leq 0, \; \| \sigma \|_1 \leq 1\}. \tag{6}$$

Si pour un certain $\xi_\imath, \; \imath = 1, \ldots, N$, $(h_\imath - T_\imath x^0)^T \sigma > 0$, alors on a trouvé un scénario $\xi_\imath$ pour lequel la décision du premier niveau $x = x^0$ ne génère pas un problème de second niveau réalisable, par conséquent, on doit exclure $(h_\imath - T_\imath x^0)$ (sans exclure des solutions réalisables), on crée alors la *coupe de faisabilité*

$$\sigma^T (h_\imath - T_\imath x) \leq 0. \tag{7}$$

4 Test et coupe d'optimalité

Soit x^0 une décision du premier niveau réalisable et θ^0 (initialement on se fixe θ^0 à $-\infty$). La valeur de $Q(x^0)$ est calculée à partir du problème dual (8)

$$\pi \max_{\pi} \{(h_\imath - T_\imath x^0)^T \pi \;\mid W^T \pi \leq q_\imath^T\} \tag{8}$$

$$Q(x^0) = \sum_{\imath=1}^{N} p_\imath \; Q(x^0, \xi_\imath) = \sum_{\imath=1}^{N} p_\imath \; \pi_\imath^T (h_\imath - T_\imath x^0), \tag{9}$$

avec $\pi_\imath^T (h_\imath - T_\imath x^0)$ la valeur optimale du problème $Q(x^0, \xi_\imath)$. Si $Q(x^0) < \theta^0$, alors x^0 est optimale, sinon, on exclut le point x^0, on crée alors la *coupe d'optimalité*

$$\theta \geq \sum_{\imath=1}^{N} p_\imath \; \pi_\imath^T (h_\imath - T_\imath x). \tag{10}$$

5 Regions de faisabilité

Il est souvent commode de définir les ensembles réalisables associés au différents niveaux du problème stochastique avec recours. L'ensemble réalisable du TSMOSPR est divisé en deux ensembles.
S, l'ensemble réalisable du premier niveau déterminé par les contraintes déterministes, à savoir celles qui ne dépendent pas du vecteur aléatoire ξ (S est un polytope).
K, l'ensemble réalisable du second niveau, à savoir l'ensemble des décisions qui admettent des décisions de recours (décisions de second niveau) réalisables indépendamment de S après l'occurrence d'un scénario. Cet ensemble est donné par :

$$K = \{x \mid Q(x) < \infty\}$$

Le problème linéaire stochastique multi-objectifs à 2-niveaux avec recours peut être reformulé simplement comme suit :

$$\left|\begin{array}{l}\min \widetilde{Z}_k = Z'_k + Q(x), \quad k = 1, \ldots, p \\ s.à.\ x \in S \cap K.\end{array}\right. \tag{11}$$

Rappelons que le problème central que l'on désire résoudre est le problème d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces du TSMOSPR (TSMOSPR est l'équivalent déterministe de MOSLP). Pour caractériser les solutions efficaces, nous avons adapté le concept suivant :

définition 1. *$\widehat{x} \in S$ est une solution efficace du TSMOSPR ssi il n'existe pas un x tel que $\widetilde{Z}_k(x) \leq \widetilde{Z}_k(\widehat{x})$ et $\widetilde{Z}_\imath(x) < \widetilde{Z}_\imath(\widehat{x})$ pour au moins un $\imath \in \{\imath = 1, \ldots, p\}$ et pour tout scénario $\xi_\imath$, $\imath = 1, \ldots, N$.*

Donc, l'ensemble des solutions efficaces du MOSLP est généré à partir de la résolution du TSMOSPR.

Nous associons à (P) le problème relaxé suivant :

$$(R)\left|\begin{array}{l}\max\ dx + Q(x) \\ s.à.\ \ x \in S \cap K\end{array}\right. \tag{12}$$

6 Description des grandes étapes de la méthode

Recherche d'une solution efficace initiale La détermination d'une solution efficace initiale du TSMOSPR se fait en deux étapes complémentaires : résoudre le problème relaxé (R) de (P) en appliquant la méthode L-Shaped et puis verifier l'efficacité de la solution obtenue. La reformulation L-Shaped du problème relaxé (R) nous conduit à la forme équivalente suivante (problème maître associé au problème relaxé) :

$$\left|\begin{array}{l}\max \widetilde{\phi}(x) = \phi(x) + \theta \\ s.à.\ x \in \widetilde{S}, \\ \qquad \theta \geq Q(x).\end{array}\right. \tag{13}$$

avec
$\widetilde{S} = \{x \in S \mid \sigma^T T_\imath x \geq \sigma^T h_\imath,\ \imath \in \{1, \ldots, N\}\}$.

Proposition 1. *Une solution efficace initiale du TSMOSPR, si elle existe, sera déterminée après un nombre fini d'itérations.*

Recherche d'une meilleure solution efficace Soit $\widetilde{S}_{\overline{M}}$ l'ensemble réduit induit de l'application des coupes de faisabilité et d'optimalité générées par L-Shaped et des coupes sur le critère principal :

$$\widetilde{S}_{\overline{M}} = \{x \in \widetilde{S} \mid dx \geq d\widetilde{x}\}.$$

Étant donné une solution efficace courante $\widetilde{x} \in \widetilde{S}_{\overline{M}}$, on génère la coupe hyperplane ($dx \geq d\widetilde{x}$) qui élimine tout les points $x \in \widetilde{S}_{\overline{M}}$ dont la valeur du critère principal est inférieure à $d\widetilde{x}$. Cette coupe ne réduit pas nécessairement le domaine réalisable courant $\widetilde{S}_{\overline{M}}$.

La question qui se pose est comment déterminer la meilleure solution efficace, si elle existe, dans l'ensemble réduit (après l'ajout de la coupe sur le critère principal au point $\widetilde{x} : dx \geq d\widetilde{x}$) ?
Inspirés par la technique de pivotage d'Ecker et Song, Ceci peut se faire en minimisant les objectifs individuellement dans l'ensemble réduit en appliquant la méthode L-Shaped.

On considère le problème linéaire multicritère déterministe ($\overline{MOLP}$),

$$(\overline{MOLP})\left|\begin{array}{lll}\text{“min”} & \widetilde{Z}_k = Z'_k + \theta, & k = 1, \ldots, p, \\ s.à. & x \in \widetilde{S}_{\overline{M}}, & \\ & \theta \geq Q(x), & \end{array}\right. \tag{14}$$

avec
$\widetilde{S}_{\overline{M}} = \{x \in \widetilde{S} \mid dx \geq d\widetilde{x}\}$.
On définit les problèmes mono-critères, (I_k), $k = 1, \ldots, p$, comme suit :

$$(I_k)\left|\begin{array}{l}\min \widetilde{Z}_k = Z'_k + \theta, \\ s.à.\ x \in \widetilde{S}_{\overline{M}}, \\ \qquad \theta \geq Q(x).\end{array}\right. \tag{15}$$

Étant donné le problème maître (I_k), trois situations peuvent apparaître à l'issue de la résolution de (I_k). Soit x^k le point réalisable optimal retourné par la méthode L-Shaped.

Situation 1 . x^k est efficace et ($dx^k > d\widetilde{x}$). On choisit x^k comme étant la nouvelle solution efficace optimale courante, on génère la coupe hyperplane ($\phi(x) \geq dx^k$) et on continu le processus de recherche d'une solution efficace qui augmente en valeur la fonction $\phi(x)$ dans l'ensemble réduit résultant $\widetilde{S}_{\overline{M}}$.

Situation 2 . x^k est efficace et ($dx^k = d\widetilde{x}$). La solution x^k est sur la face de découpage. On cherche dans $\widetilde{S}_{\overline{M}}$, en utilisant la technique d'Ecker et Kouada, une arête efficace, $(x^k, \overline{x})$ si elle existe, incidente à x^k qui apporte une amélioration en valeur de dx et qui admet au moins une décision de recours réalisable, sinon (si $Q(\overline{x}) = \infty$) rajouter les coupes de faisabilité et d'optimalité au problème maître (I_k) tant que nécessaire et vérifier le test d'efficacité.
Si une telle solution existe, on utilise $\overline{x}$ comme une nouvelle solution efficace courante, on génère la coupe ($\phi(x) \geq d\overline{x}$) et on poursuit le processus de recherche d'une meilleure solution si elle existe.
Sinon on passe à traiter le problème (I_{k+1}).

Situation 3 . x^k n'est pas efficace. x^k est dominée par des solutions examinées auparavant. On résout le problème (I_{k+1}).

À partir d'un point $\widetilde{x}$, si pour toute solution optimale x^k de (I_k) pour $k = 1, \ldots, p$, aucune des deux premières situations citées n'est rencontrée au cours de l'exécution de la procedure indiquant que x^k est dominée par des solutions efficaces déterminées au cours des étapes précédentes ou bien, y'a plus de solutions efficaces du $(MOSLP)$ qui améliorent la valeur de dx, alors $\widetilde{x}$ est une solution optimale de (P).

Proposition 2. *La procedure se termine avec une solution efficace du problème $(MOLP)$ qui incrémente la valeur de dx si elle existe.*

théorème 1. *Sous l'hypothèse S borné et non vide, l'algorithme de recherche d'une solution optimale du problème (P) converge en un nombre fini d'étapes.*

Algorithme. (Optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces du TSMOSPR).

⟨⟨ **Étape 0** ⟩⟩ **(Initialisation)**
Appliquer ***la procédure L-Shaped*** *au problème (R) avec la condition initiale $\theta = -\infty$ et sans aucune coupe de faisabilité et d'optimalité ($\widetilde{S} = S$) (sans aucune restriction sur θ).*

Si (R) n'est pas réalisable alors (P) est aussi non réalisable. ***Terminer****.*

Sinon, Soit $(\widehat{x}, \widehat{\theta})$ la solution optimale obtenue.

Résoudre le problème (P_{x^*}) au point $\widehat{x}$ dans l'ensemble courant $\widetilde{S}$.

Si, (P_{x^*}) n'admet pas une solution optimale finie, alors l'ensemble des solutions efficaces du TSMOSPR est vide ($E = \emptyset$).
***Terminer**.*

*Sinon, si $\widehat{x} \in E$. **Terminer**, avec (x^{opt}, θ^{opt})=$(\widehat{x}, \widehat{\theta})$ et $\phi^{opt} = \phi(\widehat{x})$. Sinon, soit $\widetilde{x}$ la solution optimale de (P_{x^*}), utiliser $\widetilde{x}$ comme solution efficace initiale du TSMOSPR. Poser $(x^{opt}, \theta^{opt}) = (\widetilde{x}, \widetilde{\theta})$ et $\phi^{opt} = \phi(\widetilde{x})$. Générer la coupe $dx \geq \phi^{opt}$. Poser $\kappa = 1$ et aller à l'étape κ.*

$\langle\langle$ **Étape κ $\rangle\rangle$(Exploration et progression)**
*Appliquer la méthode **procedure L-Shaped** et résoudre le problème (I_κ), soit $(x^\kappa, \theta^\kappa)$ la solution optimale obtenue. Évaluer $\phi(x^\kappa) = dx^\kappa$ et tester l'efficacité de x^κ, alors :*

$\langle$ $\kappa 1$ $\rangle$ *Si x^κ n'est pas efficace. Poser $\kappa = \kappa + 1$.*

$\langle$ $\kappa 2$ $\rangle$ *Si x^κ est efficace et ($dx^\kappa > dx^{opt}$). Poser (x^{opt}, θ^{opt})=$(x^\kappa, \theta^\kappa)$, $\phi^{opt} = \phi(x^\kappa)$ et générer la coupe $dx \geq \phi^{opt}$. Poser $\kappa = 1$.*

$\langle$ $\kappa 3$ $\rangle$ *Si x^κ est efficace et ($dx^\kappa = dx^{opt}$) (x^κ se trouve sur la coupe hyperplane). Utiliser la **technique d'Ecker et Kouada** (décrite dans le chapitre 1) et déterminer, si elles existent, les arêtes efficaces incidentes à x^κ. Soit $x_\jmath^\kappa$ la solution efficace obtenue en incrémentant la variable hors base $x_\jmath^N$ correspondante à x^κ.*

S'il existe $x_\jmath^\kappa$ tel que $dx_\jmath^\kappa > dx^\kappa$. Tester si ($Q(x_\jmath^\kappa, \xi_\imath) < \infty$), $\forall \xi_\imath$, $\imath = 1, \ldots, N$, ajouter les coupes de faisabilité (7) si nécessaire, évaluer $Q(x_\jmath^\kappa)$, vérifier le test d'optimalité et le test d'efficacité.
Poser $(x^{opt}, \theta^{opt}) = (x_\jmath^\kappa, \theta_\jmath^\kappa)$, évaluer $\phi^{opt} = dx_\jmath^\kappa$ et générer la coupe $dx \geq \phi^{opt}$. Poser $\kappa = 1$.

Sinon. Poser $\kappa = \kappa + 1$.

$\langle$ $\kappa 4$ $\rangle$ *Si $\kappa \leq p$, aller l'étape κ. Sinon **Terminer** avec (x^{opt}, θ^{opt}) la solution optimale de (P) avec la valeur optimale ϕ^{opt} correspondante.*

$\langle$ $\kappa 5$ $\rangle$ *Si le problème (I_κ) n'est pas réalisable, **Terminer** avec (x^{opt}, θ^{opt}) la solution optimale de (P) avec la valeur optimale ϕ^{opt} correspondante.*

---

7 Exemple illustratif

Nous considérons le problème MOSLP (d'une structure similaire à celle du problème (1)),

$$(MOSLP) \left| \begin{array}{lll} \text{“min”} & Z_k = C_k(\xi)\, x & k = 1, 2, 3 \\ s.à. & T(\xi)x \quad\quad = h(\xi), & \\ & 4x_1 - \ 2x_2 \leq 8, & \\ & x_1 + \ \ x_2 \leq 5, & \\ & x_1, \quad x_2 \geq 0. & \end{array} \right. \tag{16}$$

$C_k(\xi), T(\xi)$ et $h(\xi)$ sont les éléments aléatoires du problème dépendants de la variable aléatoire ξ à deux éventualités équiprobables, ξ_1 et ξ_2 avec les probabilités $p_1 = p_2 = \frac{1}{2}$ respectivement (ξ suit une loi de bernoulli).
Nous associons à chaque scénario $\xi_\imath, \imath = 1, 2$, les réalisations suivantes (la stochasticité du MOSLP est résumée dans le vecteur ξ) :
$C_1(\xi_1) = (-9, 4); C_2(\xi_1) = (3, -5); C_3(\xi_1) = (8, -11);$
$C_1(\xi_2) = (3, -5); C_2(\xi_2) = (7, 1); \quad C_3(\xi_2) = (-4, 9);$
$T(\xi_1) = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, \ T(\xi_2) = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}, \ h(\xi_1) = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \ h(\xi_2) = \begin{pmatrix} 6 \\ 1 \end{pmatrix};$

Nous considérons le problème central suivant :

$$(P)\left|\begin{array}{l}\max -2x_1 - x_2, \\ s.à. \quad x \in E.\end{array}\right. \tag{17}$$

E, l'ensemble des solutions efficace du (MOSLP).

Pour passer à l'équivalent déterministe du (MOSLP), nous adaptons l'approche de recours. Nous supposons connaître la matrice de recours W et les coûts de pénalité q associés aux actions de recours y,
$q(\xi_1) = \begin{pmatrix} 1 & 0 & 6 & 2 \end{pmatrix}^T, \ q(\xi_2) = \begin{pmatrix} 5 & 3 & 2 & 1 \end{pmatrix}^T; \ W(\xi) = W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix};$

Calcul de $\mathbb{E}(\mathbf{Z}_\imath(\mathbf{x}, \xi))$, $\imath = \mathbf{1}, \mathbf{2}, \mathbf{3}$:
$Z'_1 = \mathbb{E}(Z_1(x, \xi)) = \frac{1}{2}C_1(\xi_1)x + \frac{1}{2}C_1(\xi_2)x = -3x_1 + x_2.$
$Z'_2 = \mathbb{E}(Z_2(x, \xi)) = \frac{1}{2}C_2(\xi_1)x + \frac{1}{2}C_2(\xi_2)x = 5x_1 - 2x_2.$
$Z'_3 = \mathbb{E}(Z_3(x, \xi)) = \frac{1}{2}C_3(\xi_1)x + \frac{1}{2}C_3(\xi_2)x = 2x_1 - x_2.$

Nous obtenons le problème stochastique multi-objectif à 2-niveaux avec recours suivant (TSMOSPR) :

$$\left|\begin{array}{l}\min \widetilde{Z}_1 = -3x_1 + x_2 + Q(x) \\ \min \widetilde{Z}_2 = 5x_1 - 2x_2 + Q(x) \\ \min \widetilde{Z}_3 = 2x_1 - x_2 + Q(x) \\ s.à. \ 4x_1 - 2x_2 \leq 8, \\ \qquad x_1 + \ x_2 \leq 5, \\ \qquad x_1, \quad x_2 \geq 0.\end{array}\right. \tag{18}$$

$Q(x) = \frac{1}{2}Q(x, \xi_1) + \frac{1}{2}Q(x, \xi_1).$
$Q(x, \xi_1)$ et $Q(x, \xi_1)$ les deux problèmes du second niveau associés aux deux scénarios ξ_1 et ξ_1 respectivement :

$$Q(x, \xi_1) = \left|\begin{array}{ll}\min q^T(\xi_1)y(\xi_1) & \\ s.à. & Wy(\xi_1) = h(\xi_1) - T(\xi_1)x, \\ & y(\xi_1) \geq 0.\end{array}\right. \tag{19}$$

$$Q(x, \xi_2) = \left|\begin{array}{ll}\min q^T(\xi_2)y(\xi_2) & \\ s.à. & Wy(\xi_1) = h(\xi_2) - T(\xi_2)x, \\ & y(\xi_2) \geq 0.\end{array}\right. \tag{20}$$

Nous démarrons l'algorithme avec la solution efficace initiale $\mathbf{x^0} = (\frac{\mathbf{14}}{\mathbf{5}}, \frac{\mathbf{11}}{\mathbf{5}})$ avec une pénalité $\theta^{\mathbf{0}} = \frac{\mathbf{8}}{\mathbf{5}}$ et le vecteur critère correspondant $(\mathbf{Z_1'}, \mathbf{Z_2'}, \mathbf{Z_3'}) = (-\frac{\mathbf{31}}{\mathbf{5}}, \frac{\mathbf{38}}{\mathbf{5}}, \frac{\mathbf{17}}{\mathbf{5}})$. On pose $(\mathbf{x^{opt}}, \theta^{\mathbf{opt}}) = (\mathbf{x^0}, \theta^{\mathbf{0}})$ et $\phi^{\mathbf{opt}} = -\frac{\mathbf{39}}{\mathbf{5}}$.

– On génère la coupe d'Ecker et Song $-2x_1 - x_2 \geq -\frac{39}{5}$ équivalente à $-2x_4 - \frac{3}{5}x_5 + x_6 = 0$ avec x_6 la variable d'écart introduite par la contrainte. On rajoute la contrainte au problème. On pose $\kappa = 1$ et on passe l'étape 1 pour résoudre le problème maître I_1 dans $\widetilde{S}_{\overline{M}}$.

Étape 1 La résolution du problème mono-objectif (I_1) dans $\widetilde{S}_{\overline{M}}$ donne la solution $x = (\frac{14}{5}, \frac{11}{5}) = x^0$, qui est sur face correspondante à la coupe hyperplane. Et donc, on doit vérifier s'il existe une arête efficace incidente à x améliorant le critère principal $\phi(x)$. Pour ce faire, on applique la technique d'Ecker et Kouada (voir Ecker et Kouada (1978)). Cette technique montre que l'incrémentation de la variable hors base x_5 conduit à une arête efficace incidente à x^0 et entraîne une amélioration de $\phi(x)$.
Faisant renter x_5 dans la base, on aura une nouvelle solution optimale $x = (0, 5)$.

– *Test da faisabilité de* $x = (0, 5)$

En traitant les problèmes $\sigma_1^T = \sigma_2^T = 0$, cela implique que la solution optimale obtenue, $x = (0, 5)$, engendre des problèmes du second niveau, $Q(x, \xi_1)$ et $Q(x, \xi_2)$, réalisables.

– *Test d'optimalité de* $x = (0, 5)$

Pour tester l'optimalité de $x = (0, 5)$, on résout le problème (8) pour ξ_1 et ξ_2. Nous obtenons $\theta = Q(x) = \frac{13}{2}$, alors $x = (0, 5)$ est une solution optimale avec une pénalité $\theta = \frac{13}{2}$, on la note x^1.
$\mathbf{x^1} = (\mathbf{0}, \mathbf{5})$ est la deuxième solution efficace qu'on obtient avec une pénalité $\theta^{\mathbf{1}} = \frac{\mathbf{13}}{\mathbf{2}}$ et le vecteur critère correspondant $(\mathbf{Z_1'}, \mathbf{Z_2'}, \mathbf{Z_3'}) = (\mathbf{5}, -\mathbf{10}, -\mathbf{5})$ et $\phi(x^1) = -5 > \phi^{opt}$.
On pose $(\mathbf{x^{opt}}, \theta^{\mathbf{opt}}) = (\mathbf{x^1}, \theta^{\mathbf{1}})$ et $\phi^{\mathbf{opt}} = -5$.

– On génère la coupe d'Ecker et Song sur le critère principal $-2x_1 - x_2 \geq -5$, équivalente à $x_1 - x_4 + x_7 = 0$ avec x_7 une variable d'écart. On rajoute la contrainte au problème (I_1) et on résout à nouveau (I_1).

Nous obtenons la solution optimale $x = (\frac{7}{5}, \frac{11}{5})$.

– *Test de faisabilité de* $x = (\frac{7}{5}, \frac{11}{5})$

La solution optimale obtenue, $x = (\frac{7}{5}, \frac{11}{5})$, engendre des problèmes du second niveau, $Q(x, \xi_1)$ et $Q(x, \xi_2)$, réalisables.

– *Test d'optimalité de* $x = (\frac{7}{5}, \frac{11}{5})$

$\theta = Q(x) = \frac{23}{10}$, alors $x = (\frac{7}{5}, \frac{11}{5})$ est une solution optimale avec une pénalité $\theta = \frac{23}{10}$, on la note x^2.
$\mathbf{x^2} = (\frac{\mathbf{7}}{\mathbf{5}}, \frac{\mathbf{11}}{\mathbf{5}})$ est la troisième solution efficace qu'on obtient avec une pénalité $\theta^{\mathbf{1}} = \frac{\mathbf{13}}{\mathbf{10}}$ et le vecteur critère correspondant $(\mathbf{Z_1'}, \mathbf{Z_2'}, \mathbf{Z_3'}) = (-\mathbf{2}, \frac{\mathbf{13}}{\mathbf{5}}, \frac{\mathbf{3}}{\mathbf{5}})$ et $\phi(\mathbf{x^2}) = -\mathbf{5} = \phi^{\mathbf{opt}}$.
x^2 est donc sur la coupe hyperplane, on doit alors vérifier à nouveau s'il existe des arêtes efficaces incidentes à x^2.

La technique d'Ecker et Kouda propose la variable x_5 à rentrer dans la base, néanmoins, ceci n'implique pas l'amélioration du critère principal $\phi(x)$, par conséquent, aucune arête efficace incidente à x^2 ne correspond à une direction de maximisation de ϕ. Par conséquent, on pose $\kappa = \kappa + 1 = 2$, on va à l'étape 2.

Étape 2 On résout le problème (I_2), on obtient le point $x^1 = (0, 5)$ qui est sur la coupe hyperplane et on vérifie aisément qu'il n'existe pas une arête efficace incidente à x^1 qui améliore le critère principal ϕ. On pose $\kappa = \kappa + 1 = 3$ et on va à l'étape 3.

Étape 3 La résolution du problème (I_3) donne aussi le point $x^1 = (0, 5)$.

En conclusion, tout les points optimaux de (I_1), (I_2) et de (I_3) sont sur la face de découpage $\phi(x) \geq -5$, et il n'existe pas de solutions efficaces sur la coupe qui ont des arêtes efficaces incidentes qui apportent une augmentation en valeur du critère principal $\phi(x)$, par conséquent, on termine la résolution du problème (P) avec la solution optimale $\mathbf{x^{opt}} = (\mathbf{0}, \mathbf{5})$ avec une pénalité $\theta^{\mathbf{opt}} = \frac{\mathbf{13}}{\mathbf{2}}$, $(\mathbf{Z'_1}, \mathbf{Z'_2}, \mathbf{Z'_3}) = (\mathbf{5}, -\mathbf{10}, -\mathbf{5})$ le vecteur critère associé et la valeur de la fonction objectif $\phi^{\mathbf{opt}} = \phi(\mathbf{x^1}) = -\mathbf{5}$.

8 Conclusion

Le présent article propose une méthode de recherche d'une solution optimale, en termes des préférences du décideur, dans l'ensemble des solutions efficaces d'un MOSLP sans énumérer toutes les solutions efficaces. L'algorithme s'appuie sur une combinaison de la méthode L-Shaped et la méthode d'Ecker et Song.

Des suppositions doivent être maintenues pour que l'algorithme fonctionne correctement et puisse être appliqué : disposer de la forme explicite de la structure de préférences du décideur, pour une solution obtenue, on doit tester sa faisabilité et évaluer l'espérance de la fonction de recours en résolvant une série de problèmes linéaires qui correspondent aux N scénarios, on doit donc avoir un nombre raisonnable ou petit de scénarios, ce nombre croit exponentiellement avec la taille du vecteur aléatoire qui représente les paramètres incertains du problème. Il est aussi important de noter que l'algorithme est approprié aux problèmes avec un nombre modéré de fonctions objectif à maximiser (ou à minimiser), du moment qu'à une certaine étape de recherche d'une meilleure solution efficace, on doit résoudre, dans le pire des cas, tout les problèmes mono-objectifs, par contre, l'espace des décisions envisageables peut être assez large. Les coupes de faisabilité et les coupes d'Ecker et Song qu'on applique à chaque itération de l'algorithme permet d'obtenir la solution optimale après un nombre modéré d'itérations.

Bibliographie

M. Abbas, F. Bellahcene, *Cutting plane method for multiple objective stochastic integer linear programming*, European Journal of Operational Research 168, 967–984, 2006.

E. Beale, *On minimizing a convex function subject to linear inequalities*, Journal of the Royal Statistical Society Series, 173-184, 1955.

G. Dantzig, *Linear programming under uncertainty*, Management Science, 197-206, 1955.

J.G. Ecker, I.A. Kouada, *Finding All Efficient Extreme Points for Multi-objectve Linear Programs*, Mathematical Programming 14, 249-261, 1978.

J.G. Ecker, H.G. Song, *Optimizing a linear function over an efficient set*, Journal of Optimization Theory and Applications 83(3), 541–563, 1994.

A. Goicoechea, L. Dukstein, R.L. Bulfin, *Multiobjective stochastic programming the PROTRADE-method*, Operation Research Society of America, 1976.

M. Moulaï, S. Amrouche, *Optimisation linéaire stochastique multi-objectifs en nombres entiers*, Actes COSI'06, 404-420, 2006.

J. Teghem, *STRANGE-MOMIX : An interactive method for mixed integer linear programming*, in : R. Slowinski, J. Teghem (Eds.), Stochastic Versus Fuzzy Apprroaches to Multiobjective Mathematical Programming Under Uncertainty, Kluwer Academic Publishers, Dordrecht, 101–115, 1990.

J. Teghem, D. Dufrane, M. Thauvoye, P.L. Kunsch, *STRANGE : Interactive method for multiobjective linear programming under uncertainty*, European Journal of Operational Research 26 (1), 65–82, 1986.

B. Urli, R. Nadeau, *Multiobjective stochastic linear programming with incomplete information : A general methodology*, In : R. Slowinski, J. Teghem (Eds.), 131–161, 1990.

R. Van Slyke, R. J-B. Wets, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics 17, 638-663, 1969.

Compression des images médicales 3D par la quantification vectorielle algébrique

Ouddane Samira[1], Benamrane Nacéra[1]

[1]Université des Sciences et de Technologie d'Oran, Mohamed Boudiaf,
Faculté des Sciences,
Département d'informatique, BP 1505 El-Mnaouer, 31000 Oran, Algérie
[1]nacerabenamrane@yahoo.fr, samiraoud191@yahoo.fr,

Résumé. L'imagerie médicale est un domaine en plein essor du fait du développement des technologies numériques qui produisent des données 3D et même 4D. La contrepartie de la résolution offerte par les images volumiques réside dans une quantité de données gigantesque d'où la nécessité de la compression. Cet article présente un nouveau schéma de codage dédié aux images médicales 3D de type IRM. Il utilise une transformée en ondelettes suivie d'une quantification vectorielle algébrique (QVA) sur les sous bandes détails. L'originalité de notre approche réside dans la conception d'une zone morte vectorielle pendant l'étape de quantification vectorielle qui permet de prendre en compte la corrélation entre les pixels voisins. Cette approche permet d'atteindre des taux bien supérieurs, tout en conservant une qualité visuelle acceptable.

Keywords: Compression avec perte, pile d'images médicales, transformée en ondelette, quantification vectorielle algébrique avec zone morte.

1 Introduction

L'imagerie médicale a connu des progrès très importants ces dernières années avec le développement de techniques qui produisent des données 3D de plus en plus précises mais en contrepartie de plus en plus volumineuses. Certaines de ces images sont intrinsèquement volumiques alors que d'autres correspondent à une succession d'images 2D (encore appelées piles d'images). L'augmentation croissante des capacités de stockage apporte une réponse partielle à ce problème mais demeure cependant insuffisante. Outre la question de l'archivage, la transmission de ces images sur des bandes passantes par nature limitées pose également un problème.

La compression des images médicales volumiques apparaît donc incontournable, elle consiste à minimiser le nombre de bits nécessaire à une représentation fidèle de l'image originale et d´accéder uniquement à l´information requise, allégeant ainsi les transferts et autorisant un accès à distance aux données.

On distingue deux types de compression, la compression sans perte (ou réversible) qui préserve l'intégrité des données, et la compression avec perte (ou irréversible) qui génère des dégradations mais offre des performances en termes de réduction de l'information bien plus grandes que celles issues de la compression sans perte [1].

Dans cet article nous présentons une approche de compression d'images médicales 3D basée sur la décomposition en ondelettes qui est reconnue comme une transformation décorrélante très efficace pour ce type d'images et nous proposons la conception d'un nouveau dictionnaire de la quantification vectorielle algébrique contenant une zone morte. L'avantage principal du schéma proposé est sa capacité à supprimer les vecteurs non significatifs pour quantifier plus précisément les vecteurs importants. Cela fournit une amélioration notable du compromis débit-distorsion. Les résultats numériques et visuels produits par la quantification vectorielle avec zone morte (QVAZM) sur des IRMs sont prometteurs par rapport aux meilleurs codeurs actuels publiés dans la littérature.

Différentes approches de la compression des images médicales 3D ont été proposées dans la littérature.

Jean-Marie Moureaux **[1]** propose une méthode dite de «quantification vectorielle algébrique avec zone morte », qui est associée à une analyse multirésolution par ondelettes, permettant d'améliorer sensiblement les performances en termes de compromis débit-distorsion, et ainsi la qualité visuelle de l'image reconstruite.

Cette méthode est décomposée en trois étapes : l'indexage des vecteurs du dictionnaire, le réglage des paramètres du dictionnaire (facteur d'échelle et zone morte) et l'allocation des ressources binaires.

L'approche proposée par I. P.AKAM BITA **[2]** est basée sur une étude sur l'application des transformations d'ACI en compression d'images multi-composantes, en proposant deux algorithmes permettant d'obtenir d'une part la transformation minimisant le critère dans le cas général, et d'autre part celle qui minimise le critère sous la contrainte que la distorsion dans le domaine transformée est la même que celle du domaine de l'image.

Yann-Gaudeau **[3]** propose une technique qui utilise la zone morte multidimensionnelle pendant l'étape de quantification qui permet de prendre en compte les corrélations entre les voxels voisins.

2 Approche proposée

Le schéma général de notre approche de compression est présenté par la figure 1. Dans une première étape, une transformation des pixels est appliquée sur l'image. Une quantification vectorielle algébrique avec zone morte sera ensuite appliquée sur les coefficients produits par la transformée en ondelettes discrète (TOD) à l'intérieur de chaque sous-bande des détails dans une approche intra-bande.

La quantification vectorielle algébrique est une méthode des quantificateurs structurés **[4]** et qui a fait l'objet de nombreux travaux ces dernières années **[5]**, **[6]**. Elle présente le double avantage d'éviter la génération (souvent longue) et le stockage d'un dictionnaire mais également de permettre une quantification rapide des vecteurs du fait des propriétés géométriques du réseau régulier de points sur lequel s'appuie son dictionnaire **[7].** En contrepartie, ses performances sont généralement plus faibles que celles de la QV non structurée lorsque la statistique de la source n'est pas uniforme.

La dernière étape de la chaîne de compression consiste à encoder chaque vecteur quantifié *Y* en utilisant un code préfixe efficace qui associe à *Y* un unique couple *(e,pos),* où $e = Y$ correspond à la norme de *Y* et *pos* à la position sur la surface de rayon *e*. Pour être efficace en terme de débit, *e* est codé avec un code entropique alors que *pos* est codé sur une longueur fixe **[8].**

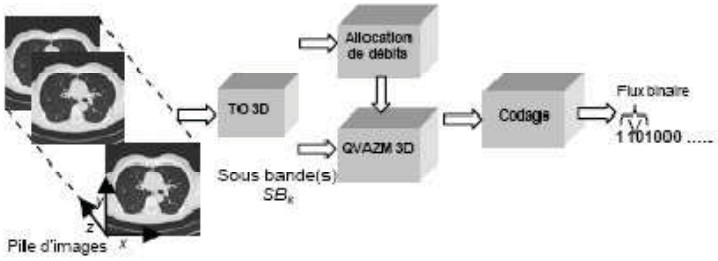


Fig.1. Schéma général de compression avec pertes proposé.

Notre schéma permet de seuiller les vecteurs de la source non significatifs en fonction d'un critère basé sur leur norme, permettant d'allouer plus de bits à ceux qui sont significatifs. Cette zone morte vectorielle peut être efficacement appliquée aux images médicales dans le domaine des ondelettes qui contient de grandes zones non significatives.

De plus, la norme *L* constitue une bonne mesure de l'activité locale dans le cas des coefficients d'ondelettes dont les plus significatifs ont tendance à s'agglutiner, sous la condition que les vecteurs sont orientés le long des détails de la sous-bande correspondante.

2.1 Transformée en ondelettes

La fonction de transformation de l'image s'inspire le plus souvent d'une opération de la décorrélation. Elle permet alors de diminuer la dynamique du signal et d'éliminer les redondances **[9]**.

Nous avons implémenté la transformée en ondelettes par l'approche multirésolution par banc de filtres numériques. En compression d'images, il est essentiel que les filtres d'analyse et de synthèse soient symétriques. De plus l'ondelette mère doit être suffisamment régulière. Nous avons employé les ondelettes de Haar et les ondelettes de Daubechies qui sont toutes les deux orthogonales.

2.1.1 L'ondelette de Haar

L'ondelette de Haar est extrêmement simple, elle a un support compact, elle est bien localisée en espace, et elle est donc facile à mettre en œuvre algorithmiquement c'est pourquoi c'est la plus utilisée dans la compression des images.

2.1.2 Les ondelettes à support compact de Daubechies

Les ondelettes de I. Daubechies, très couramment utilisées car, présentant de bonnes propriétés, sont connues dans le cadre de l'analyse multirésolution comme des ondelettes orthogonales possédant une régularité choisie. La régularité intervient sur le nombre de moments nuls de l'ondelette. Enfin les ondelettes sont à support compact et sont non symétriques. Un cas particulier des ondelettes est celle d'ordre 2 qui est l'ondelette de Haar.

2.2 Codage prédictif

L'image transformée par les ondelettes n'est pas totalement décorrélée, Un codage prédictif est appliqué sur les approximations résultantes de la transformée en ondelettes dans le but de diminuer la corrélation entre les coefficients. Etant donnée la corrélation entre les coefficients voisins, cette différence sera faible et son codage sera efficace.

2.3 Quantification vectorielle algébrique avec zone morte appliquée aux détails

La quantification vectorielle algébrique n'est autre que la généralisation de la quantification scalaire uniforme à un espace de dimension n. Elle s'appuie sur un partitionnement régulier de cet espace (en cellules de Voronoï identiques), lui-même conduisant à un réseau régulier de points. Un réseau régulier de points λ dans R^n est défini dans la formule (1) **[7]** :

$$\Lambda = \left\{ y \in R^n / y = u_1 a_1 + u_2 a_2 + ... u_n a_n \right\} \qquad \textbf{(1)}$$

où les vecteurs $a_i \in R^m (m \geq n)$ forment une base du réseau, les u_i étant des coefficients entiers.

La mise en œuvre d'un QVA dans une application de compression se déroule en cinq étapes : **[10]**

– **Choix d'un réseau** : Cette étape permet de minimiser la distorsion granulaire (c'est à dire l'erreur à l'intérieur de la cellule de quantification).

– **Troncature du réseau et normalisation de la source** : Il s'agit d'un point clé de la QVA puisqu'il conditionne ses performances en termes de compromis débit-distorsion. Le dictionnaire C est un sous-ensemble borné du réseau Λ.

Soit r le rayon de l'hypersphère $S_n^d(r)$ définie ci-dessous (2):

$$S_n^d(r) = \left\{ X \in R^n / d(0, X) = r \right\} \quad (2)$$

avec d une distance.

La troncature du réseau introduit une distorsion dite de surcharge liée à la projection en surface du dictionnaire des vecteurs sources n'appartenant pas à celui-ci. Afin de minimiser ce bruit de surcharge, on choisit généralement une forme de troncature adaptée à la statistique de la source.

La normalisation de la source par un facteur d'échelle γ permet une mise à l'échelle du réseau par rapport à la source. Elle influence fortement le débit binaire obtenu après quantification puisque, selon la valeur de γ, les vecteurs source sont projetés sur des régions plus ou moins peuplées du dictionnaire, c'est à dire plus ou moins coûteuses en terme de débit.

$P_\gamma : R^n \longrightarrow R^n$
$X \longrightarrow Y = X / \gamma$

Y est appelé le vecteur projeté.

– **Quantification :** La quantification du vecteur de la source X dans le dictionnaire est la composition entre P_γ et Q la quantification sur le réseau Λ :

$$Y = Q \circ P_\gamma(X) = Q\left(\frac{X}{\gamma}\right) = Q(\tilde{X}) \quad (3)$$

Y est un vecteur du réseau, le vecteur quantifié.

La reconstruction des vecteurs après quantification s'effectue de la manière suivante :

$$\tilde{X} = P_\gamma^{-1}\left\{Q\left[P_\gamma(X)\right]\right\} = \gamma Q\left(\frac{X}{\gamma}\right) = \gamma Q(\tilde{X}) \quad (4)$$

$\tilde{X}$ est le vecteur reconstruit.

Pour atteindre le débit cible ou le taux de compression souhaité, il suffit uniquement de régler le paramètre γ.

- **Indexage**: c'est une étape cruciale de la chaîne de compression. Elle consiste à assigner à chaque vecteur quantifié un index (ou indice, ou étiquette) unique, qui une fois codé est transmis sur le canal.

Les méthodes utilisées pour l'indexage des vecteurs quantifiés sont basées sur la propriété suivante : un vecteur Y d'un réseau est caractérisé par sa norme $e = \|Y\|_{\alpha}^{\alpha}$ et sa position **pos** sur l'hypersphère de rayon e, donc le code produit est le résultat de la concaténation de ces deux informations : **(e, pos)** c'est l'index du vecteur Y, **e** est appelé le préfixe et **pos** le suffixe de l'index.

- **Codage des index** : Afin de coder efficacement les redondances des index du fait de la non uniformité de la source, il est nécessaire de terminer la chaîne de compression par une opération de codage entropique.

Lorsque le découpage des vecteurs est effectué dans la direction du filtrage on aura tendance à observer une grande concentration de vecteurs de faible énergie (traduit sur l'histogramme des normes par un pic proche de 0) et un nombre beaucoup moins important de vecteurs de haute énergie. Ceci nous conduit à définir deux classes de vecteurs : celle des vecteurs significatifs (moyenne ou forte énergie au sens de la magnitude moyenne) et celle des vecteurs non significatifs (faible énergie au sens de la magnitude moyenne).

Ainsi, par analogie avec les quantificateurs scalaires actuels qui tirent profit de la concentration de faibles coefficients d'ondelettes en incluant un seuillage de ceux-ci (zone morte scalaire), nous avons proposé d'élargir et de déformer la cellule de Voronoï d'origine du quantificateur vectoriel algébrique (zone morte vectorielle) ce qui revient à effectuer un seuillage des vecteurs en fonction de leur énergie. Comme le montre la figure 2, la taille ainsi que la forme de la cellule d'origine sont modifiées.

Dans le schéma proposé, le processus de quantification d'un vecteur quelconque de la source X devient le suivant :

- Si $\|X\| \leq R_{ZM}$, X est remplacé par le vecteur nul 0.
- Si $\|X\| > R_{ZM}$, X est mis à l'échelle par un facteur d'échelle γ et quantifié avec des algorithmes de quantification rapides **[8]**.

Le choix du R_{ZM} dépend des valeurs des sous bandes, chaque sous bande a un R_{ZM} approprié.

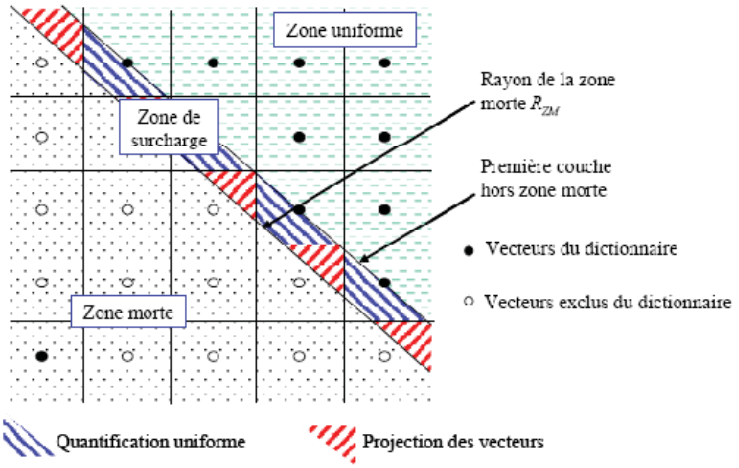


Fig.2. Zoom sur les trois zones de quantification pour un dictionnaire avec zone morte pyramidale sur le réseau Z^2.

3 Résultats expérimentés

Nous avons testé notre approche de compression sur une séquence de coupes médicales 2D.

Le tableau 1 récapitule les résultats obtenus sur des coupes en niveaux de gris. Les coupes sont de type IRM de résolution 256×256.

Taille de la carte		8×8	
Nombre de coupes	Nombre de décompositions	Compression rate (%)	PSNR(dB)
5	2	70.54	55.25
	3	89.06	52.02
	4	96.36	49.30
10	2	70.40	54.75
	3	89.00	51.34
	4	96.34	48.59
30	2	69.96	52.34
	3	96.27	47.63
	4	96.27	46.31

Tableau 1. Résultats de la quantification vectorielle algébrique avec zone morte.

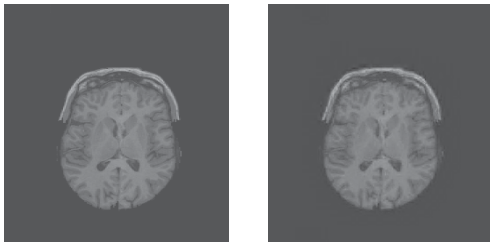


Fig.3. De gauche à droite, la coupe initiale et la coupe décompressée après la QVAZM avec Taille du dictionnaire = 64, nombre de décompositions = 3, CR=88.64, PSNR=47.30.

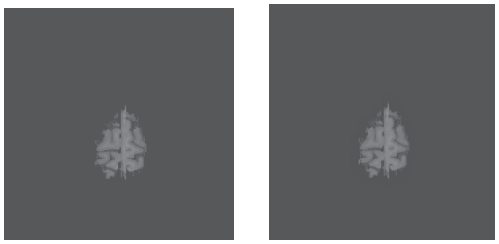


Fig.4. De gauche à droite, la coupe initiale et la coupe décompressée après la QVAZM avec Taille du dictionnaire = 64, nombre de décompositions = 2, CR= 70.65, PSNR=57.02.

4 Interprétation des résultats

L'augmentation du nombre de coupes implique une diminution du taux de compression et du PSNR. Cela peut être expliqué par la croissance des informations.

Quelque soit le nombre de coupes, la diminution de la taille de la carte entraîne une augmentation du taux, et une diminution du PSNR, c'est à dire la qualité des images reconstruites est détériorée.

L'augmentation du nombre de décompositions entraîne une diminution du PSNR.

L'augmentation du nombre de décompositions implique une augmentation remarquable du taux.

5 Conclusion

Dans cet article, nous avons proposé une approche de compression des images médicales 3D par les ondelettes et une quantification vectorielle algébrique avec zone morte.

Cette approche a été testée sur des images médicales 3D (coupes 2D), les résultats obtenus sont satisfaisants ; des taux de compression intéressants et une bonne qualité des coupes reconstruites.

De plus, en assignant plus de bits aux vecteurs significatifs suivant un critère basé sur la norme, notre méthode permet de mieux préserver les fines structures et produit des images de meilleure qualité globale, ce qui est d'importance pour les applications médicales.

Dans l'algorithme proposé, plusieurs paramètres permettent de jouer sur le compromis qualité de reconstruction - taux de compression : taille des dictionnaires, facteur de projection et le mode de partitionnement en cellules de Voronoi. Ces paramètres dépendent fortement de la nature de l'image.

6 Références

1. Jean-Marie Moureaux, « Quantification vectorielle algébrique : un outil performant pour la compression et le tatouage d'images fixes », 2007.
2. I. P.Akam Bita, « Une approche de l'analyse en composantes indépendantes à la compression des images multi composantes », Université Joseph Fourier de Grenoble, 2007.
3. Yann-Gaudeau, Contributions en compression d'images médicales 3D et d'images naturelles 2D, Doctorat de l'Université Henri Poincaré, Nancy 1, 2006.
4. A. Gersho and R.M. Gray, «Vector Quantization and Signal Compression», Kluwer Academic Publishers, 1992.
5. F. Chen, Z. Gao and J. Villasenor, «Lattice Vector Quantization of Generalized Gaussian Sources» , IEEE Transactions on Information Theory, vol.43, pp. 92-103,1997.
6. JM. Moureaux, L. Guillemot, «Data hiding in the context of lossy compression : a combined approach» , SPIE Journal of Electronic Imaging, vol. 14, n_ 3, pp. 033017-1 - 033017-12, ISSN 1017-9909, September 2005.
7. J.H Conway et N.J.A Sloane, "Fast quantizing and decoding algorithm for lattice quantizers and codes", IEEE transactions on information theory, vol. 28, 2, pp. 227-232, Mars 1982.
8. Voinson, T, Guillemot, L, Moureaux J-M, Image "compression using lattice vector quantization with code book shape adapted thresholding," *ICIP*, pages 641-644, Rochester, USA, 2002.
9. J. Ordonez, G. Cazuguel, J. Puentes, B. Solaiman, C. Roux, « L'utilisation des moments spatiaux pour la recherche d'images médicales par leur contenu dans le domaine compressé : application à la quantification vectorielle et le standard JPEG-DCT », 2003.

10. JM. Moureaux, «Lossy Compression of Volumetric Medical Images with 3D Deadzone Lattice Vector Quantization», Picture Coding Symposium, PCS.07, Lisboa, Portugal, November 7-9, 2007.

6 ème Colloque sur l'Optimisation et les Systèmes d'Information

COSI'2009

Http://www.isima.fr/cosi2009/

Thématiques:

- Base de données.
- Fouille de données.
- Intégration d'informations et d'applications.
- Systèmes d'information décisionnels.
- Applications et systèmes d'information dédiés.
- Algorithmique discrète.
- Optimisation combinatoire, programmation mathématique.
- Système à base de connaissances, TAL, ..
- Applications en imagerie médicale et vision artificielle, au traitement du signal, à la planification des réseaux de télécommunications et des transports, à l'ordonnancement de production, en économie et finance, en biologie, ...

Université Badji Mokhtar - Annaba -
Faculté des Sciences de l'Ingénieur
Département Informatique
Tel/Fax: 00 213 (0)38 87 27 56
Http://www.univ-annaba.org