
10e Colloque sur l'Optimisation et les Systèmes d'Information

# COSI 2013

09-11 juin 2013, CDTA, Alger, Algérie

Organisé par :
le Centre de Développement des Technologies Avancées

# Proceedings

Cité 20 août 1956 Baba Hassen, Alger, Algérie Tel.:+213 (0) 21 35 10 Fax:+213 (0) 21 35 10 39 e-mail cosi2013@cdta.dz

**Actes du Dixième Colloque sur l'Optimisation et les Systèmes d'Information- COSI'2013**

9-11 Juin 2013, Alger, Algérie

Centre de Développement des Technologies Avancées (CDTA)

## Table des matières

## Mot du Directeur du CDTA
## Président d'honneur de COSI'2013

Le Centre de Développement des Technologies Avancées, en accueillant et en organisant COSI'2013, joint sa contribution à celles des comités de pilotage et de programme pour faire de celle-ci une conférence d'excellence. COSI'2013 souffle sa dixième bougie cette année avec un long parcours à travers tout le territoire d'Algérie. Beaucoup de chemins parcourus, mais également et surtout, COSI a gagné en maturité tant sur le plan organisationnel ainsi que sur celui de la qualité scientifique. Ces attributs placent COSI aujourd'hui au niveau du standard international.

Le Centre de Développement des Technologies Avancées, classé aujourd'hui comme premier centre de recherche en Algérie, est honoré en organisant et, en particulier, en accueillant la tenue de COSI'2013 au niveau de son siège central, sis à Baba Hassen, Alger.

COSI'2013 se distingue cette année par un programme très riche et bien équilibré. Il s'articule autour de sessions orales et posters, et des plénières. Les thèmes abordés sont également des plus pertinents qui couvrent un spectre assez large allant des techniques d'optimisation aux systèmes d'information en passant par des applications connexes. En plus du forum d'échanges scientifique et technique, COSI'2013 offre également aux jeunes chercheurs cette année une école d'été consacrée à un thème essentiel pour la communauté scientifique qu'est la fouille de données.

La contribution du Centre de Développement des Technologies Avancées à la réussite de la tenue de COSI'2013 a été rendue possible grâce aux efforts soutenus, une année durant, du comité d'organisation formé de chercheurs et de personnel de soutien de recherche du CDTA. Ainsi, je leur adresse un hommage particulier pour le labeur accompli. Je ne termine pas ce mot sans remercier les rôles essentiels des comités de pilotage et de programme dans la réussite de cette conférence sans oublier, bien sûr, nos sponsors qui nous ont permis de transformer ce rendez-vous en fête scientifique.

Baba Hassen, le 29 mai 2013,
Dr. Brahim Bouzouia
Directeur du CDTA
Président d'honneur de COSI'2013

## Préface

Ces actes regroupent les articles présentés lors de la $10^{\text{ème}}$ édition du Colloque sur l'Optimisation et les Systèmes d'Information (COSI 2013) qui s'est déroulé au CDTA à Alger, du 9 au 11 juin 2013. Les articles contenus dans ces actes représentent de façon tout à fait homogène l'ensemble des thématiques couvertes par COSI depuis son origine.
En efffet COSI est une manifestation scientifique pluridisciplinaire qui rassemble des chercheurs travaillant dans les domaines de: Théorie des Graphes et Combinatoire, Recherche Opérationnelle, Traitement d'Images et Vision Artificielle, Intelligence Artificielle et Systèmes d'Information. Les précédentes éditions de COSI ont eu lieu à : Tlemcen (2012), Guelma (2011), Ouargla (2010), Annaba (2009), Tizi-Ouzou (2008), Oran (2007), Alger (2006), Béjaia (2005) et Tizi-Ouzou (2004).
C'est un honneur pour moi de rédiger cette préface, car j'avais rédigé celle de la première conférence en 2004. 10 années c'est pour une conférence l'âge de raison, comme en témoignent les chiffres suivants: cette année nous avons eu 212 intentions de soumissions (venant de 10 pays) dont 172 ont été confirmées et ont fait l'objet d'une évaluation par le comité de programme. Parmi ceux là nous avons acceptés 30 papiers longs et 15 posters. Le taux de sélection des articles réguliers a donc été de 17%.
Nous sommes particulièrement heureux que trois chercheurs de très haut niveau aient accepté de nous présenter une conférence invitée :

- Takeaki Uno (National Institute of Informatics (NII), Japan ),
- Alexandre Dolgui (Ecole des Mines de Saint-Etienne, France), et
- Nicholas Schabanel (LIAFA, CNRS et Université Paris Diderot, France).

Tout ceci confirme la montée en puissance de COSI qui doit devenir une conférence internationale.
Enfin je remercie les auteurs pour leurs excellentes contributions, et pour leur travail bénévole :

- les membres seniors du comité: Méziane Aïder (Vice-Chair : Théorie des Graphes et Combinatoire), Mourad Baiou et Jin Kao Hao (Vice-Chairs : Recherche Opérationnelle), Nacéra Benamrane, Djamel Ziou (Vice-Chairs : Traitement d'Images et Vision Artificielle), Frédérique Saubion (Vice-Chair : Intelligence Artificielle), Hassina Seridi et Michel Schneider (Vice-Chairs : Systèmes d'Information),
- tous les membres du comité de programme (liste complète page suivante : http://www.isima.fr/cosi/cosi2013/comites.php),
- les relecteurs externes,
- et bien sûr les membres du comité d'organisation ainsi que les sponsors.

Enfin mes derniers remerciements vont à Mohamed Aidene, Lhouari Nourine et Bachir Sadi, les pionniers de 2004, puis à tous ceux qui les ont soutenus tout au long de ces dix années et tout particulièrement au comité de pilotage de COSI.

Paris, le 27 mai 2013, Michel Habib

# Organisation

Centre de Développement des Technologies Avancées (CDTA), Alger

## Président d'honneur

Brahim Bouzouia,
Directeur du Centre de Développement des Technologies Avancées (CDTA), Alger

## Comité d'Organisation

### Présidente

Samia Ourari, Centre de Développement des Technologies Avancées (CDTA), Alger

### Vice-Présidents

Ali Abbassene, Centre de Développement des Technologies Avancées (CDTA), Alger
Ali Mahdoum, Centre de Développement des Technologies Avancées (CDTA), Alger

### Membres

Faroudja Abid, Centre de Développement des Technologies Avancées (CDTA), Alger
Kahina Aissani, Centre de Développement des Technologies Avancées (CDTA), Alger
Mourad Aoudar, Centre de Développement des Technologies Avancées (CDTA), Alger
Nacer Benzaba, Centre de Développement des Technologies Avancées (CDTA), Alger
Mabrouk Boumaraf, Centre de Développement des Technologies Avancées (CDTA), Alger
Amar Badreddine Cherchali, Centre de Développement des Technologies Avancées (CDTA), Alger
Oualid Djekoune, Centre de Développement des Technologies Avancées (CDTA), Alger
Amel Derradji, Centre de Développement des Technologies Avancées (CDTA), Alger
Mahdi Gaham, Centre de Développement des Technologies Avancées (CDTA), Alger
Rafik Guerbas, Centre de Développement des Technologies Avancées (CDTA), Alger
Salim Laroussi, Centre de Développement des Technologies Avancées (CDTA), Alger
Moussa Merioua, Centre de Développement des Technologies Avancées (CDTA), Alger
Faycel Mokrani, Centre de Développement des Technologies Avancées (CDTA), Alger
Lamia Sekkai, Centre de Développement des Technologies Avancées (CDTA), Alger
El-hadi Zouaoui, Centre de Développement des Technologies Avancées (CDTA), Alger

### Publicité

Nacima Labadie, Université de Technologie de Troyes (France)
Brahim Oukacha, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)

# 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
Hamid Séridi, Université de Guelma, Algérie

# Comité de Programme

## Président

Michel Habib, LIAFA, Université Paris 7 (France)

## Vice-Chairs

Méziane Aider, USTHB (Algérie)
Mourad Baiou, LIMOS (France)
Nacéra Benamrane, USTO (Algérie)
Hao Jin-Kao, Université d'Angers (France)
Frédéric Saubion, Université d'Angers (France)
Michel Schneider, ISIMA (France)
Hassina Seridi, Université Annaba (Algérie)
Djemel Ziou, Unversité de Sherbrooke (Canada)

## Membres

Amine Abdelmalek, Université de Saida (Algérie)
Mohamed Ahmed-Nacer, USTHB (Algérie)
Rachid Ahmed-Ouamer, Université de Tizi-Ouzou (Algérie)
Mohamed Aidene, Université de Tizi-Ouzou (Algérie)
Hacène Ait Haddadene, USTHB Alger (Algérie)
Otmane Ait Mohamed, Université Concordia (Canada)
Hassan Aït-Kaci, Université Claude Bernard Lyon 1 (France)
Zaia Alimazighi, USTHB (Algérie)
Chaoui Allaoua, Université Mentouri Constantine (Algérie)
Lallouet Arnaud, Université de Caen (France)
Nadjib Badache, CERIST (Algérie)
Kamel Barkaoui, CNAM-Paris (France)
Ladjel Bellatreche, ENSMA (France)
Boualem Benatallah, UNSW (Australie)
Salima Benbernou, Université Paris Descartes (France)
Salem Benferhat, Université d'Artois, Lens (France)
Belaid Benhamou, Université d'Aix-Marseille I (France)
Abdelhafid Berrachedi, USTHB Alger (Algérie)
Stéphane Bessy, Université Montpellier II (France)
Mohand Ouamer Bibi, Université de Béjaia (Algérie)
Djamel Bouchaffra, CDTA (Algérie)
Isma Bouchemakh, USTHB (Algérie)
Mourad Boudhar, USTHB (Algérie)
Mahmoud Boufaida, Université Mentouri Constantine (Algérie)
Zizette Boufaida, Université Mentouri, Constantine (Algérie)
Mohand Boughanem, IRIT, Toulouse (France)
Kamel Boukhalfa, USTHB (Algérie)
Amel Bouzeghoub, Telecom Sud, Paris (France)
Mustapha Chellali, Université Saad Dahlab, Blida (Algérie)
Laurent D'Orazio, Université Blaise Pascal, (France)
Fedoua Didi, Abou Bekr Belkaid Tlemcen (Algérie)
Nouredine Djedi, Université Biskra, (Algérie)
Haytham Elghazel, Université Claude Bernard Lyon 1 (France)

Jean-Paul Gauthier, Université de Toulon Mohand Said Hacid, Université Claude Bernard, (Lyon)
Allel Hadjali, ENSSAT, (Lannion)
Mamadou Kante, Université Blaise Pascal (France)
Okba Kazar, Université de Biskra (Algérie)
Tahar Kechadi, UCD (Irlande)
Zoubida Kedad, UVSQ (France)
Omar Kermia, CDTA (Algérie)
Hamamache Kheddouci, Université Claude Bernard Lyon 1 (France)
Nacima Labadie, Université de Technologie de Troyes (France)
Philippe Lacomme, Université Blaise Pascal (France)
Yacine Lafifi, Université de Guelma (Algérie)
Yahia Lebbah, Université D'Oran Es-Sénia (Algérie)
Alain Leger, France Télécom (France)
Mohamed Lehsaini, Université Abou Bekr Belkaid Tlemcen (Algérie)
Vincent Limouzy, Université Blaise Pascal (France)
Ali Mahdoum, CDTA (Algérie)
Ridha Mahjoub, Université Paris Dauphine (France)
Engelbert Mephu, Université Blaise Pascal (France)
Hayett Merouani, Université Badji Mokhtar, Annaba (Algérie)
Rokia Missaoui, Université de Quebec en Outaouais (Canada)
Lhouari Nourine, Université Blaise Pascal (France)
Rachid Nourine, Université d'Oran Es-Sénia (Algérie)
Mohand Ouanes, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Brahim Oukacha, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Samia Ourari, CDTA, Algérie Jean-Marc Petit, INSA de Lyon (France)
Fethi Rabhi, Université du New South Wales, Sydney (Australie)
Mohand Said Radjef, Université Abderrahmane Mira de Bejaia (Algérie)
Michael Rao, ENS Lyon (France)
André Raspaud, Universite Bordeaux I (France)
Djamal Rebaine, Université du Québec, Chicoutimi (Canada)
Bachir Sadi, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Lakhdar Sais, Université d'Artois, Lens (France)
Yacine Sam, Université de Tours (Algérie)
Hamid Seridi, Université de Guelma (Algérie)
Yahya Slimani, Université Al Manar (Tunisie)
Pierre Spiteri, INP- Toulouse (France)
Yehia Taher, Tilburg University (Netherlands)
Abdelmalik Taleb-Ahmed, Université de Valenciennes (France)
Tatiana Tchemisova, University of Aveiro (Portugal)
Farouk Toumani, Université Blaise Pascal (France)
Ouerdane Wassila, Ecole Centrale Paris (France)
Farouk Yalaoui, Université de technologie de Troyes (France)

## Relecteurs externes

**A**
Aboura, Radia
Aliane, Hassina
Allili, Mohand Said
Amir, Samir
Aridhi, Sabeur
**B**
Barki, Hichem
Baudon, Olivier
Bauer, Henri
Beheshti, Seyed Mehdi Reza
Boughaci, Dalila
Bouker, Slim
Boukhris, Imen
Boutemedjet, Sabri
**C**
Chakroun, Chedlia
**F**
Fertin, Guillaume
Fournier-Viger, Philippe
Frihi, Ibtissem
**G**
Gerard, Fleury
**H**
Hao, Jin-Kao
Hicham, Reguieg
Hore, Alain
**J**
Jean, Stephane
Jenhani, Ilyes
**K**
Khalissa, Derbal Amieur
Khelifati, Si Larabi
**L**
Ladjal, Hamid
Lagares, Angel
Lakhdar, Akroun
Loiseau, Yannick
**M**
Melouah, Ahlem
Mohamad, Baraa
Mounir, Hemam
**N**
Niang, Cheikh
Nourine, Rachid
**O**
Ostrowski, Richard
Ouhammou, Yassine

# Théorie des graphes

# A note on $k$-Roman graphs*

Ahmed Bouchou[1], Mostafa Blidia[2] and Mustapha Chellali[2]
[1]University of Médéa, Algeria

[2]LAMDA-RO Laboratory, Department of Mathematics
University of Blida
B.P. 270, Blida, Algeria
Email: bouchou.ahmed@yahoo.fr; m_blidia@yahoo.fr; m_chellali@yahoo.com

April 21, 2013

**Abstract**

Let $G = (V, E)$ be a graph and let $k$ be a positive integer. A subset $D$ of $V(G)$ is a $k$-dominating set of $G$ if every vertex in $V(G) \setminus D$ has at least $k$ neighbors in $D$. The $k$-domination number $\gamma_k(G)$ is the minimum cardinality of a $k$-dominating set of $G$. A Roman $k$-dominating function on $G$ is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ such that every vertex $u$ for which $f(u) = 0$ is adjacent to at least $k$ vertices $v_1, v_2, ..., v_k$ with $f(v_i) = 2$ for $i = 1, 2, ..., k$. The weight of a Roman $k$-dominating function is the value $f(V(G)) = \sum_{u \in V(G)} f(u)$ and the minimum weight of a Roman $k$-dominating function on $G$ is called the Roman $k$-domination number $\gamma_{kR}(G)$ of $G$. A graph $G$ is said to be a $k$-Roman graph if $\gamma_{kR}(G) = 2\gamma_k(G)$. In this note we study $k$-Roman graphs.

**Keywords**: Roman $k$-domination, $k$-Roman graph.
**AMS Subject Classification:** 05C69

*This research was supported by "Programmes Nationaux de Recherche: Code 8/u09/510".

# 1 Introduction

We consider finite, undirected, and simple graphs $G$ with vertex set $V(G)$ and edge set $E(G)$. The *open neighborhood* $N_G(v)$ of a vertex $v$ consists of the vertices adjacent to $v$, and $N_G[v] = N_G(v) \cup \{v\}$ is the *closed neighborhood.* The *degree* of $v$ is $|N_G(v)|$. A *leaf* is a vertex of degree one. By $\Delta(G) = \Delta$ we denote the *maximum degree* of a graph $G$. A graph is *bipartite* if its vertex set can be partitioned in two independent sets. A *d-regular* graph is a graph with a degree $d$ for each vertex of $G$. A graph is called a *d-semiregular bipartite graph*, if its vertex set can be partitioned in such a way that every vertex in one of the partite sets has degree $d$. The *subdivision graph* of a graph $G$ is the graph obtained from $G$ by replacing each edge $uv$ of $G$ by a vertex $w$ and edges $uw$ and $vw$. A graph $G$ is called a *cactus graph* if each edge of $G$ is contained in at most one cycle. A *unicyclic graph* is a connected graph containing exactly one cycle. A *tree* is a connected graph with no cycle. We denote by $K_{1,t}$ a *star* of order $t + 1$.

Let $k$ be a positive integer. A subset $D \subseteq V(G)$ is a *k-dominating set* of a graph $G$, if $|N_G(v) \cap D| \geq k$ for every $v \in V(G) \backslash D$. The *k-domination number* $\gamma_k(G)$ is the minimum cardinality among the $k$-dominating sets of $G$. The concept of $k$-domination was introduced by Fink and Jacobson in [2].

A *Roman k-dominating function* on $G$ is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ such that every vertex $u$ for which $f(u) = 0$ is adjacent to at least $k$ vertices $v_1, v_2, \ldots, v_k$ with $f(v_i) = 2$ for $i = 1, 2, \ldots, k$. The *weight* of a Roman $k$-dominating function is the value $f(V(G)) = \sum_{v \in V(G)} f(v)$. The minimum weight of a Roman $k$-dominating function on a graph $G$ is called the *Roman k-domination number* $\gamma_{kR}(G)$. Note that if $k \geq \Delta + 1$, then clearly $\gamma_{kR}(G) = |V|$. Hence we may assume in the whole paper that $k \leq \Delta$. Also, if $f : V(G) \longrightarrow \{0, 1, 2\}$ is a Roman $k$-dominating function on $G$, then let $(V_0, V_1, V_2)$ be the ordered partition of $V(G)$ induced by $f$, where $V_i = \{v \in V(G) \mid f(v) = i\}$ for $i = 0, 1, 2$. Note that there is a one to one correspondence between the functions $f : V(G) \rightarrow \{0, 1, 2\}$ and the ordered partitions $(V_0, V_1, V_2)$ of $V(G)$. The Roman 1-domination number $\gamma_{1R}$ corresponds to the well-known *Roman domination number* $\gamma_R$, which was given implicitly by Steward in [5] and by ReVelle and Rosing in [4].

## 2 Known results

We begin by listing some known results that will be useful here. The first one gives a relation between the Roman $k$-domination and $k$-domination numbers for any graph.

**Proposition 1 (Kämmerling and Volkmann [3])** *For any graph $G$,*

$$\gamma_k(G) \leq \gamma_{kR}(G) \leq 2\gamma_k(G).$$

According to [3], a graph $G$ is said to be a *$k$-Roman graph* if $\gamma_{kR}(G) = 2\gamma_k(G)$. Kämmerling and Volkmann gave a necessary and sufficient condition for a graph to be $k$-Roman.

**Proposition 2 (Kämmerling and Volkmann [3])** *A graph $G$ is a $k$-Roman graph if and only if it has a $\gamma_{kR}$-function $f = (V_0, V_1, V_2)$ with $V_1 = \emptyset$.*

The following two results give sufficient conditions $G$ to have $\gamma_{kR}(G) = n$.

**Proposition 3 (Kämmerling and Volkmann [3])** *If $G$ is a graph with at most one cycle and $k \geq 2$, or $G$ is a cactus graph and $k \geq 3$, then $\gamma_{kR}(G) = n$.*

**Proposition 4 (Kämmerling and Volkmann [3])** *If $G$ is a graph of order $n$ and maximum degree $\Delta \geq 1$, then $\gamma_{\Delta R}(G) = n$.*

In [2], Fink and Jacobson have established a lower bound on the $k$-domination number of a graph.

**Theorem 5 (Fink and Jacobson [2])** *If $G$ has $n$ vertices and $m(G)$ edges, then*

$$\gamma_k(G) \geq n - \frac{m(G)}{k} \quad \text{for } k \geq 1.$$

*Furthermore, if $m(G) \neq 0$, then $\gamma_k(G) = n - \dfrac{m(G)}{k}$ if and only if $G$ is a $k$-semiregular bipartite graph.*

**Corollary 6 (Fink and Jacobson [2])** *If $G$ is a graph with $n$ vertices and $m(G) \neq 0$ edges, then*

$$\gamma_2(G) = n - \frac{m(G)}{2}.$$

*if and only if $G$ is the subdivision graph of another multigraph (graph with possibly parallel edges).*

## 3 Main Results

We begin by giving a necessary condition for a graph to be $k$-Roman.

**Theorem 7** *If $G$ is a $k$-Roman graph with $k \geq 2$, then every vertex of $G$ is adjacent to at most $k-1$ leaves.*

**Proof.** Let $G$ be a $k$-Roman graph with $k \geq 2$. Suppose that $v$ is a vertex of $G$ adjacent to at least $k$ leaves. Let $L_v$ be the set of leaves adjacent to $v$. Clearly, for every $\gamma_{kR}$-function every leaf is assigned a positive value. Also, by Proposition 2, $G$ has a $\gamma_{kR}$-function $f = (V_0, V_1, V_2)$ with $V_1 = \emptyset$. Hence $f(w) = 2$ for every leaf $w \in L_v$. Now if $f(v) \neq 0$, then we can decrease the weight of $f$ by assigning the value 1 instead of 2 to every leaf, contradicting the fact that $f$ is a $\gamma_{kR}$-function. Thus $f(v) = 0$. Since $k \geq 2$, we can change $f(w) = 2$ to $f(w) = 1$ for every vertex $w \in L_v$ and $f(v) = 0$ to $f(v) = 1$. Clearly we obtain a Roman $k$-dominating function with weight less than $f(V(G))$, a contradiction too. Therefore, $|L_v| \leq k-1$. ■

We now give a characterization of $k$-Roman graphs when $k = \Delta$.

**Theorem 8** *A graph $G$ is $\Delta$-Roman if and only if $G$ is a bipartite regular graph.*

**Proof.** Let $G$ be a graph with $\gamma_{\Delta R}(G) = 2\gamma_{\Delta}(G)$. Then by Proposition 4, $\gamma_{\Delta R}(G) = n = 2\gamma_{\Delta}(G)$, and so $\gamma_{\Delta}(G) = n/2$. Let $S$ be a minimum $\Delta$-dominating set of $G$. Clearly, since every vertex of $V \backslash S$ has $\Delta$ neighbors in $S$, the set $V \backslash S$ is independent. Now let $m(S, V \backslash S)$ be the number of edges between $S$ and $V \backslash S$. Then $m(S, V \backslash S) = \Delta |V \backslash S| = \Delta n/2$. Using the fact that $\Delta n \geq 2m(G)$, it follows that $\Delta n = 2m(G) = 2m(S, V \backslash S) = \Delta n$, and so $m(G) = m(S, V \backslash S)$. Thus, every vertex of $G$ has degree $\Delta$ and hence $S$ is also independent. Therefore, $G$ is a bipartite $\Delta$-regular graph.

Conversely, assume that $G$ is a bipartite $\Delta$-regular graph. We know by Proposition 4 that $\gamma_{\Delta R}(G) = n$. Thus, it suffices to show that $\gamma_{\Delta}(G) = n/2$. By Proposition 1 we have $\gamma_{\Delta}(G) \geq n/2$. The equality is obtained from the fact that every partite set of $G$ is a $\Delta$-dominating set. ■

Next we improve the upper bound in Proposition 1 for the class of trees. Moreover, we characterize all trees attaining this upper bound.

**Theorem 9** *Let $T$ be a tree of order $n \geq 3$ with $\Delta(T) \geq k \geq 2$. Then $\gamma_{kR}(T) \leq 2\gamma_k(T) - k + 1$, with equality if and only if*

*i) $k = 2$ and $T$ is the subdivision graph of another tree, or*

*ii) $k = n - 1$ and $T$ is a star.*

**Proof.** We first prove the upper bound. Since $m = n - 1$ for trees, it follows from Theorem 5 that for every tree $T$ and every positive integer $k$ we have $\gamma_k(G) \geq ((k-1)n + 1)/k$. Also, one can easily check that $((k-1)n + 1)/k \geq (n + k - 1)/2$ for $2 \leq k \leq \Delta(T) \leq n - 1$. Now using the fact that $\gamma_{kR}(T) = n$ (by Proposition 3) we obtain $\gamma_k(G) \geq ((k-1)n + 1)/k \geq (n + k - 1)/2 = (\gamma_{kR}(T) + k - 1)/2$, and the bound is proved.

Now assume that $\gamma_{kR}(T) = 2\gamma_k(T) - k + 1$. Then we have equality throughout the previous inequality chain. In particular, $((k-1)n + 1)/k = (n + k - 1)/2$ and $\gamma_k(G) = ((k-1)n + 1)/k$. The first equality implies that $k = 2$ or $k = n - 1$. Now, if $k = 2$, then $\gamma_2(G) = (n + 1)/2$ and by Corollary 6 we obtain (i). If $k = n - 1$, then $T$ is the star $K_{1,n-1}$.

The converse is easy to show and we omit the details. ■

The following corollary is an immediate consequence of Theorem 9.

**Corollary 10** *There are no $k$-Roman trees for $k \geq 2$.*

Next we show that there are no $k$-Roman cactus graphs for $k \geq 3$. We need the following lemma, which can be found in [7] on p. 30.

**Lemma 11** *If $G$ is a cactus graph on $n$ vertices and $m$ edges, then*

$$2m \leq 3n(G) - 3.$$

**Proposition 12** *There are no $k$-Roman cactus graph for $k \geq 3$.*

**Proof.** Suppose that $G$ is a $k$-Roman cactus graph for some $k \geq 3$. By Proposition 3 and Theorem 5 we have $n = \gamma_{kR}(T) = 2\gamma_k(G) \geq 2(n - m/k)$. Hence $kn \leq 2m$. Now, by Lemma 11 we get $kn \leq 3n - 3$, which is impossible since $k \geq 3$. ■

Next we improve the upper bound in Proposition 1 for unicyclic graphs. We denote by $K_{1,p} + e$ the graph obtained from the star $K_{1,p}$ by adding an

edge between two leaves of $K_{1,p}$. Let $P_5$ be the path on five vertices labeled in order $x_1, x_2, x_3, x_4, x_5$. Let $F$ be the graph obtained from $P_5$ by adding a new vertex $y$ and edges $yx_2$ and $yx_4$. Let $G_1$, $G_2$ and $G_3$ be three graphs obtained from $P_5$ by adding the edges $x_2x_4$, $x_3x_5$ and $x_2x_5$, respectively.

**Theorem 13** *Let $G$ be a unicyclic graph and $\Delta(G) \geq k \geq 3$. Then*

$$\gamma_{kR}(G) \leq 2\gamma_k(G) - k + 1,$$

*with equality if and only if either $k \in \{3, 4, n-1\}$ and $G = K_{1,k} + e$, or $k = 3$ and $G = F$.*

**Proof.** We first note that $n \geq 4$ since $\Delta \geq 3$. If $n = 4$, then $k = \Delta = 3$, $G = K_{1,3} + e$ and $\gamma_{kR}(G) = 2\gamma_k(G) - k + 1$. If $n = 5$, then $k \in \{3, 4\}$. If $k = 3$, then clearly $G \in \{G_1, G_2, G_3, K_{1,4} + e\}$ and $\gamma_{kR}(G) < 2\gamma_k(G) - k + 1$. If $k = 4$, then $G = K_{1,4} + e$ and $\gamma_{kR}(G) = 2\gamma_k(G) - k + 1$. Also if $n = k + 1$, then $k = \Delta$, $G = K_{1,n-1} + e$ and $\gamma_{kR}(G) = 2\gamma_k(G) - k + 1$.

Now let us suppose that $n \geq \max\{6, k + 2\}$. It can be seen that

$$\frac{(k-1)\,n}{k} \geq \frac{n + k - 1}{2} \qquad (*)$$

and the upper follows from Proposition 3 and Theorem 5.

Now assume that $\gamma_{kR}(G) = 2\gamma_k(G) - k + 1$. Clearly, if $n \in \{4, 5, k + 1\}$, then $G = K_{1,n-1} + e$. Hence we can assume that $n \geq \max\{6, k + 2\}$. Then we have equality in (*), in particular $\gamma_k(G) = (n + k - 1)/2 = (k - 1)\,n/k$. It follows that $n = 6$, $k = 3$, $\gamma_3(G) = 4$, and so $G = F$. ■

**Theorem 14** *A unicyclic graph $G$ is a 2-Roman graph if and only if $G$ is the subdivided graph of another unicyclic graph (possibly with a cycle on two vertices).*

**Proof.** If $\gamma_{2R}(G) = 2\gamma_2(G)$, then by Proposition 3 we have $n = 2\gamma_2(G)$, and so $\gamma_2(G) = n/2$. By Corollary 6, $G$ is the subdivided graph of another unicyclic graph. Now assume that $G$ is the subdivided graph of another unicyclic graph. By Theorem 6, $\gamma_2(G) = n/2$ and by Proposition 3, $\gamma_{2R}(G) = n$. Therefore, $\gamma_{2R}(G) = 2\gamma_2(G)$. ■

# References


[1] E.J. Cockayne, P.A. Dreyer, S.M. Hedetniemi and S.T. Hedetniemi, Roman domination in graphs. *Discrete Mathematics* 278 (2004) 11–22.

[2] J.F. Fink and M.S. Jacobson, $n$-domination in graphs. *Graph Theory with Applications to Algorithms and Computer Science.* John Wiley and Sons. New York (1985), 282-300.

[3] K. Kämmerling and L. Volkmann, Roman $k$-domination in graphs. *J. Korean Math. Soc.* 46 (2009) 1309–1318.

[4] C. S. ReVelle, K. E. Rosing, Defendens imperium romanum: a classical problem in military strategy, Amer Math. Monthly 107 (2000), 585-594.

[5] I. Steward, Defend the Roman Empire!, Sci. Amer. 281 (1999), 136-139.

[6] L. Volkmann, Some remarks on lower bounds on the $p$-domination number in trees. *J. Combin. Math. Combin. Comput.* 61 (2007), 159-167.

[7] L. Volkmann, Graphen an allen Ecken und Kanten, RWTH Aachen 2006, XVI, 377 pp.

# Codes identifiants dans le Produit Cartésien de deux Cliques

Anissa Hissoum and Ahmed Semri

USTHB, Laboratoire LaROMaD, Faculté des mathématiques,
BP32, 16111 EL Alia, ALGER
nissahissoum@gmail.com ahmedsemri@yahoo.fr

**Résumé** Les codes identifiants ont été définis pour la première fois par KARPOVSKY et al. en 1998 pour modéliser un problème de détection et de localisation de processeurs défectueux dans des réseaux multiprocesseurs. Depuis, d'autres applications ont été développées comme dans les systèmes de localisation et de détection dans les environnements fermés munis de capteurs sans fil. Un code identifiant dans un graphe est un sous ensemble de sommets tel que deux sommets quelconques du graphe ont leurs ensembles de voisinage fermé différents et non vides dans le code. Trouver la cardinalité minimum d'un code identifiant dans un graphe, lorsqu'il existe, est un problème $NP$-difficile. Ce papier traite le problème des codes identifiants dans deux classes de graphes : l'hypercube et le produit cartésien de deux cliques de dimensions différentes.

**Mots clés** : Graphes, Domination, Code identifiant, Graphe de Hamming, Hypercube, clique, Produit Cartésien des graphes

## 1 Introduction

La notion des codes identifiants a été introduite par Karpovsky, Chakrabarty and Levitin en 1998 dans [8] pour modéliser un problème de détection et de localisation de processeurs défectueux dans des réseaux multiprocesseurs. Un code identifiant dans un graphe simple non orienté $G = (V, E)$ est un sous ensemble de sommets tel que deux sommets quelconques du graphe ont leurs ensembles de voisinage fermé dans le code non vides et différents, c'est à dire un code à la fois couvrant et séparateur, les élément du code sont appelés les mots de codes. On parle de code $r$-identifiant dans le cas où l'ensemble de voisinage fermé pour un sommet $v \in V$ est remplacé par $B_r(v)$ la boule de centre $v \in V$ et de rayon $r$. Une condition nécessaire et suffisante pour qu'un graphe $G = (V, E)$ admette un code $r$-identifiant est qu'il ne contient pas de sommets jumeaux (deux sommets avec le même ensemble de voisinage fermé à distance $r \geq 1$). Dans ce cas, l'ensemble des sommets $V$ est toujours un code $r$-identifiant de $G$. Le problème est donc est de trouver le nombre minimum des mots de codes nécessaires pour $r$-identifier chaque sommet de $G$.

Il a été démontré dans [3] que ce problème était $NP$-difficile pour tout $r \geq 1$. Si $C$ est un code identifiant de cardinalité minimum d'un graphe à $n$ sommets,

alors $|C| \geq \lceil \log_2(n+1) \rceil$ [8, Th1]. Cette première borne inférieure vient du fait qu'avec les éléments de $C$, il est possible de construire $2^{|C|} - 1$ sous ensembles distincts et non vides, donc pour pouvoir identifier l'ensemble des sommets du graphe, il faut que $n \leq 2^{|C|} - 1$.

Dans la première partie de notre travail, nous proposons des bornes supérieures sur la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques de tailles différentes.

Dans la deuxième partie, en utilisant une métaheuristique basée sur la Recherche Tabou, on construit des codes $r$-identifiant, $r \geq 1$ dans l'espace binaire de Hamming de dimension $n \geq 3$.

## 2 Code identifiant de $K_n \square K_m, m \geq n \geq 2$

Etant donnés deux graphes $G$ et $H$, le *produit cartésien* $G \square H$ est le graphe ayant pour ensemble de sommets $V(G) \times V(H)$ et dont deux sommets $(x_1, y_1)$ et $(x_2, y_2)$ sont reliés par une arête si et seulement si, soit $x_1 x_2 \in E(G)$ et $y_1 = y_2$, soit $y_1 y_2 \in E(H)$ et $x_1 = x_2$. Le produit cartésien de deux cliques $K_n \square K_m$, est une matrice de $n * m$ sommets tel que chaque ligne $R_x, x \in \{1, ..., n\}$ (resp colonne $C_y, y \in \{1, ..., m\}$ est une clique $K_m$ (resp $K_n$).

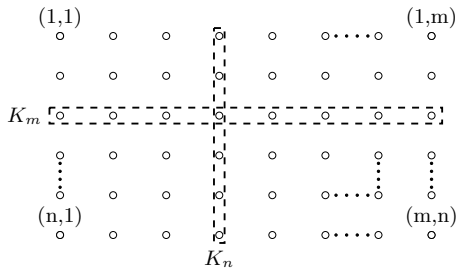



**Fig. 1.** Produit cartésien de deux cliques $K_n \square K_m$

Dans [7, Théorème 1], il a été démontré que la cardinalité minimum d'un code identifiant du produit cartésien $K_n \square K_n$ est égale à $\lfloor \frac{3n}{2} \rfloor$. La preuve de ce théorème est basée sur la détermination d'un code identifiant de cardinalité $\lfloor \frac{3n}{2} \rfloor$ :

$$D = \{(x, x) | x = 1, ...n\} \text{ et } A = \begin{cases} \{(n - x + 1, x) | x = 1, ..., \frac{n-1}{2}\}, & \text{si } n \text{ est impair} \\ \{(n - x + 1, x) | x = 1, ..., \frac{n}{2}\}, & \text{si } n \text{ est pair} \end{cases}$$

L'ensemble des sommets $D \cup A$ est un code identifiant de $K_n \square K_n$.

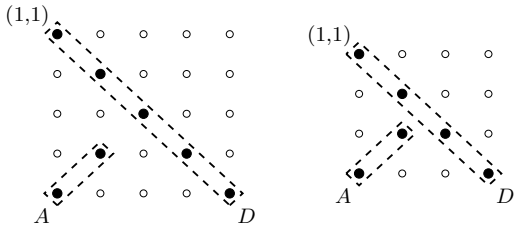



**Fig. 2.** Le code $D \cup A$ suivant la parité de $n$

Nous allons établir des bornes supérieures sur la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques de tailles différentes $K_n \square K_m, m > n$.
Soient deux cliques $K_n$ et $K_m$ tels que $m > n \geq 2$ et soient $a$ et $b$ respectivement le quotient et le reste de la division euclidienne de $m$ par $n$.

### 2.1 Cas où $a \geq 2$ et $0 \leq b \leq n-1$

**Proposition 1.** *Si $C$ est un code identifiant de cardinalité minimum de $K_n \square K_{an+b}$ avec $n, a \geq 2$ et $0 \leq b \leq n-1$, alors :*

$$|C| \leq n(2a-1) + 2b$$

*Démonstration.* Le code $C = A \cup B \cup P$ est un code identifiant de $K_n \square K_{an+b}$, $n, a \geq 2$ et $0 \leq b \leq n-1$ de cardinalité $n(2a-1)+2b$ tel que :

$$A = \{(i, i+kn)/i \in \{1,...,n\}, k \in \{0,...,a-1\}\} \cup \begin{cases} \emptyset & \text{si } b = 0 \\ \{(i, i+an)/i \in \{1,...,b\}\} & \text{si } b \neq 0 \end{cases}$$

$$B = \{(i, kn-i+1)/i \in \{1,...,n\}, k \in \{1,...,a-1\}\} \cup \begin{cases} \emptyset & \text{si } b = 0 \\ \{(n-k+1, (a-1)n+k)/k \in \{1,...,b\}\} & \text{si } b \neq 0 \end{cases}$$

$$P = \begin{cases} \left\{\left(n, \lceil \frac{n}{2} \rceil + jn\right)/j \in \{0,...,a-2\}\right\} & \text{si } 0 \leq b \leq \lfloor \frac{n}{2} \rfloor \text{ et } n \text{ est impair} \\ \left\{\left(n, \lceil \frac{n}{2} \rceil + jn\right)/j \in \{0,...,a-1\}\right\} & \text{si } \lceil \frac{n}{2} \rceil \leq b \leq n-1 \text{ et } n \text{ est impair} \\ \emptyset & \text{si } n \text{ est pair} \end{cases}$$

$C$ est un code couvrant car chaque ligne de $K_n \square K_{an+b}$ contient au moins un sommet de $C$, donc il reste à prouver que $C$ est un code séparateur. Soient deux sommets distincts $(x,y), (x',y') \in K_n \square K_{an+b}$ tel que $b \neq 0$ et $a \geq 2$. Sans perte de généralités, on suppose que $y \leq y'$ :

1. Si $(x,y)$ et $(x',y')$ ne sont ni dans la même colonne ni dans la même ligne :
   (a) Si $y \leq n$ alors
      i. si $y' \leq n$ alors $(x, (a-1)n+x) \in N[(x,y)] \backslash N[(x',y')]$
      ii. si $y' \geq n+1$ alors $(x,x) \in N[(x,y)] \backslash N[(x',y')]$

(b) Si $y \geq n+1$ alors $(x,x) \in N[(x,y)] \backslash N[(x',y')]$

2. Si $(x,y)$ et $(x',y')$ sont dans la même colonne :
   (a) si $y \leq (a-1)n$ alors $(x,(a-1)n+x) \in N[(x,y)] \backslash N(x',y)$
   (b) si $y \geq (a-1)n+1$ alors $(x,x) \in N[(x,y)] \backslash N[(x',y)]$
3. Si $(x,y)$ et $(x',y')$ sont dans la même ligne. Soient $k,k' \in \{0,...,a\}$ et $j,j' \in \{0,...,n-1\}$ tel que $y = kn+j$ et $y' = k'n+j'$
   (a) si $y \leq (a-1)n$
      i. si $x \leq \lceil \frac{n}{2} \rceil$
         A. si $j=0$ alors $(n,y) \in N[(x,y)] \backslash N[(x,y')]$
         B. si $n$ est impair et $j = \lceil \frac{n}{2} \rceil$ alors $(n,y) \in N[(x,y)] \backslash N[(x,y')]$
         C. si $j \leq \lfloor \frac{n}{2} \rfloor$ alors $(n-j+1,y) \in N[(x,y)] \backslash N[(x,y')]$
         D. si $j \geq \lceil \frac{n}{2} \rceil + 1$ alors $(j,y) \in N[(x,y)] \backslash N[(x,y')]$
      ii. si $x \geq \lceil \frac{n}{2} \rceil + 1$
         A. si $j=0$ alors $(1,y) \in N[(x,y)] \backslash N[(x,y')]$
         B. si $n$ est impair et $j = \lceil \frac{n}{2} \rceil$ alors $(\lceil \frac{n}{2} \rceil, y) \in N[(x,y)] \backslash N[(x,y')]$
         C. si $j \leq \lfloor \frac{n}{2} \rfloor$ alors $(j,y) \in N[(x,y)] \backslash N[(x,y')]$
         D. si $j \geq \lceil \frac{n}{2} \rceil + 1$ alors $(n-j+1,y) \in N[(x,y)] \backslash N[(x,y')]$
   (b) si $(a-1)n+1 \leq y < y' \leq an$, soient $j$ et $j'$ tel que $y = (a-1)n+j$ et $y' = (a-1)n+j'$ :
      i. si $j' \leq x$ alors $(j,(a-1)n+j) \in N[(x,y)] \backslash N[(x,y')]$
      ii. si $j' > x$ alors $(j',(a-1)n+j') \in N[(x,y')] \backslash N[(x,y)]$
   (c) si $an+1 \leq y < y' \leq an+b$, soient $j$ et $j'$ tel que $y = an+j$ et $y' = an+j'$
      i. si $j' \leq x$ alors $(j,an+j) \in N[(x,y)] \backslash N[(x,y')]$
      ii. si $j' > x$ alors $(j',an+j') \in N[(x,y')] \backslash N[(x,y)]$
   (d) si $1 \leq y \leq an$ et $an+1 \leq y' \leq an+b$. Soient $k \in \{0,..,a\}, j \in \{0,..,n-1\}$ et $j' \in \{1,..,b\}$ tels que $y = kn+j$ et y'=an+j'
      i. si $x \geq b+1$ alors $(j',y') \in N[(x,y')] \backslash N[(x,y)]$
      ii. si $x \leq b$ alors
         A. si $x \leq \lceil \frac{n}{2} \rceil$
            - si $j=0$ alors $(n,y) \in N[(x,y)] \backslash N[(x,y')]$
            - si $n$ est impair et $j = \lceil \frac{n}{2} \rceil$ alors $(n,y) \in N[(x,y)] \backslash N[(x,y')]$
            - si $j \leq \lfloor \frac{n}{2} \rfloor$ alors $(n-j+1,y) \in N[(x,y)] \backslash N[(x,y')]$
            - si $j \geq \lceil \frac{n}{2} \rceil + 1$ alors $(j,y) \in N[(x,y)] \backslash N[(x,y')]$
         B. si $x \geq \lceil \frac{n}{2} \rceil + 1$
            - si $j=0$ alors $(n,y) \in N[(x,y)] \backslash N[(x,y')]$
            - si $n$ est impair et $j = \lceil \frac{n}{2} \rceil$ alors $(\lceil \frac{n}{2} \rceil, y) \in N[(x,y)] \backslash N[(x,y')]$
            - si $j \leq \lfloor \frac{n}{2} \rfloor$ alors $(j,y') \in N[(x,y')] \backslash N[(x,y)]$

– si $j \geq \lceil \frac{n}{2} \rceil + 1$ alors $(n-j+1, y) \in N[(x,y)] \backslash N[(x,y')]$

□

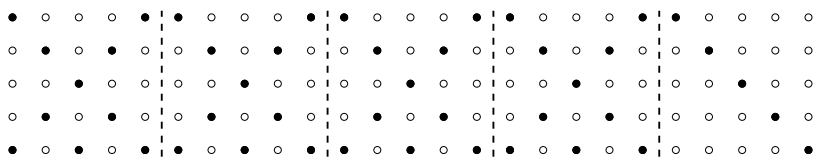


**Fig. 3.** Code identifiant de $K_5 \square K_{25}$

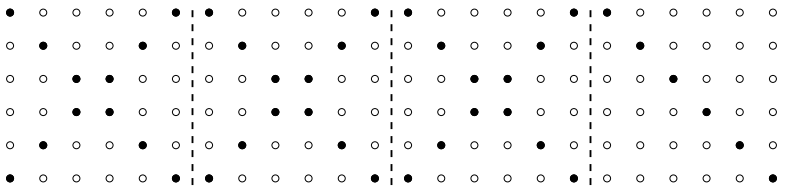


**Fig. 4.** Code identifiant de $K_6 \square K_{24}$

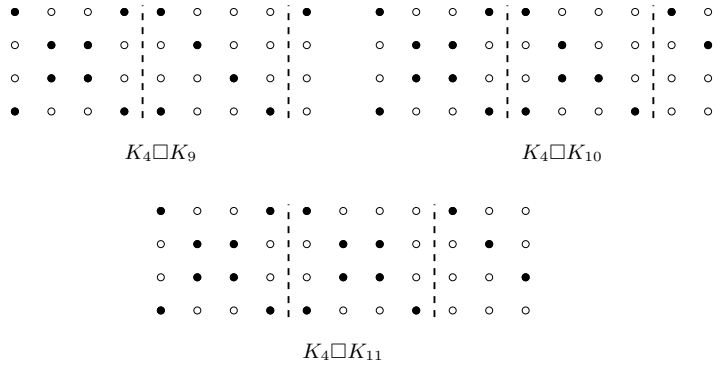



**Fig. 5.** Codes identifiants de $K_n \square K_{an+b}$, $n = 4, a = 2$ et $b \in \{1, 2, 3\}$

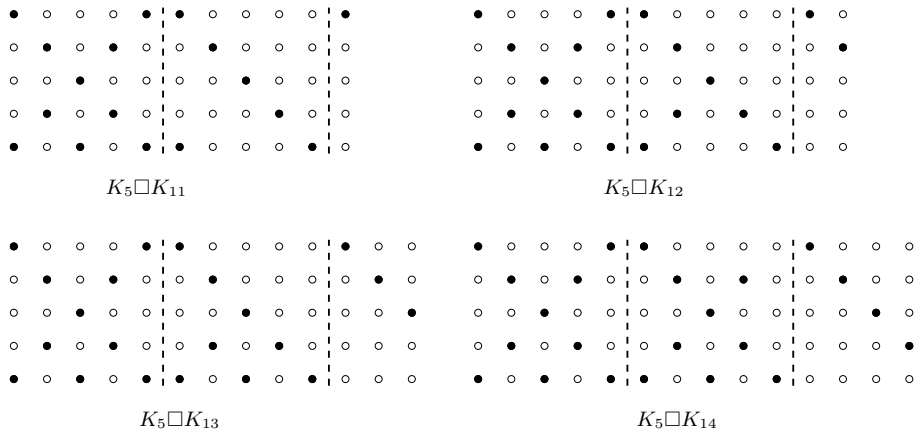



**Fig. 6.** Codes identifiants de $K_n\Box K_{an+b}$, $n = 5, a = 2$ et $b \in \{1, 2, 3, 4\}$

### 2.2 Cas où $a = 1$ et $b \neq 0$

Dans ce cas, on propose deux codes identifiants suivant les valeurs de $b$

**Si $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$**

**Proposition 2.** *Si $C$ est un code identifiant de cardinalité minimum de $K_n\Box K_{n+b}$ avec $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$, alors :*

$$|C| \leq \left\lfloor \frac{3n}{2} \right\rfloor + b$$

*Démonstration.* Si $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$, alors le code $C = A \cup B$ est un code identifiant de $K_n\Box K_{n+b}$, $n \geq 2$ de cardinalité $\lfloor \frac{3n}{2} \rfloor + b$ tel que :

$A = \{(i, i)/i \in \{1, ..., n\}\} \cup \{(i, n + i)/i \in \{1, ..., b\}\}$

$B = \{(n - i + 1, i)/i \in \{1, ..., \lceil \frac{n}{2} \rceil\}\}$

□

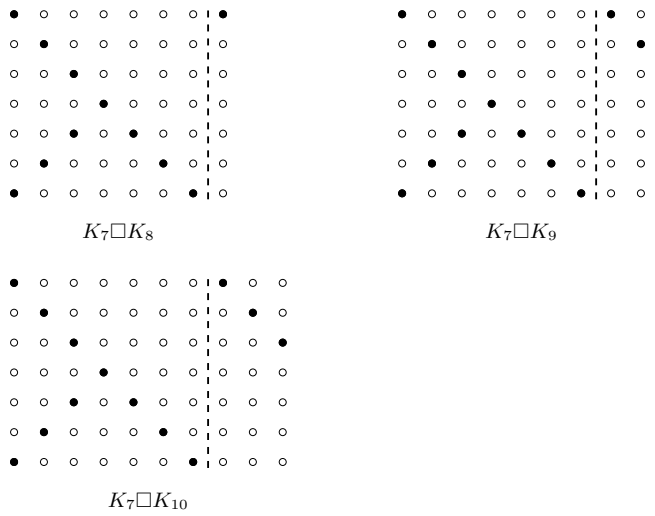



**Fig. 7.** Codes identifiants de $K_n\Box K_{n+b}$, $n = 7$ et $b \in \{1, 2, 3\}$

**Si $\left\lfloor \frac{n}{2} \right\rfloor + 1 \leq b \leq n - 1$**

**Proposition 3.** *Si $C$ est un code identifiant de cardinalité minimum de $K_n\Box K_{n+b}$ avec $\left\lfloor \frac{n}{2} \right\rfloor + 1 \leq b \leq n - 1$, alors :*

$$|C| \leq n + 2b$$

*Démonstration.* Si $\left\lfloor \frac{n}{2} \right\rfloor + 1 \leq b \leq n - 1$, alors le code $C = A \cup B \cup P$ est un code identifiant de $K_n\Box K_{n+b}$, $n \geq 2$ de cardinalité $n + 2b$ tel que :

$$A = \{(i, i)/i \in \{1, ..., n\}\} \cup \{(i, n + i)/i \in \{1, ..., b\}\}$$

$$B = \{(n - i + 1, i)/i \in \{1, ..., b\}$$

$$P = \begin{cases} \{\left(n, \left\lceil \frac{n}{2} \right\rceil\right)\} & \text{si } n \text{ est impair} \\ \emptyset & \text{si } n \text{ est pair} \end{cases}$$

□

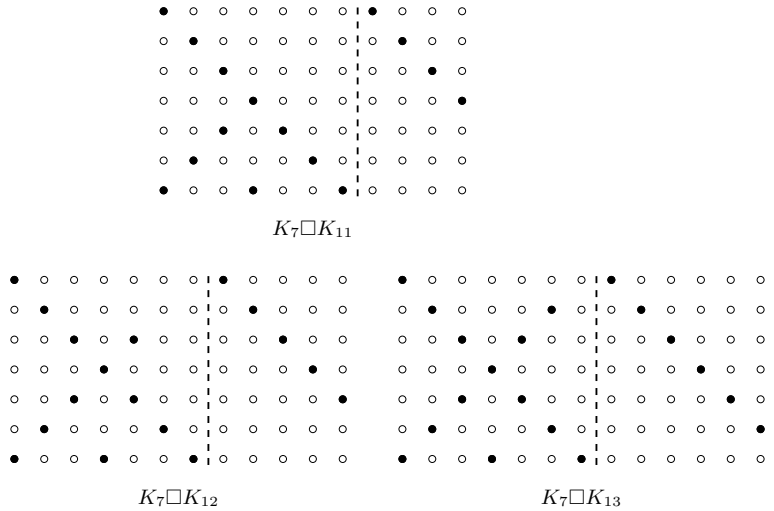



**Fig. 8.** Codes identifiants de $K_n\Box K_{n+b}$, $n = 7$ et $b \in \{4, 5, 6\}$

Nous conjecturons que les bornes supérieures trouvées dans les propositions 1, 2 et 3 coincident avec les valeurs exactes de la cardinalité minimum d'un code identifiant de $K_n\Box K_{an+b}$.

**Conjecture 1 (Anissa HISSOUM, Ahmed SEMRI).**
*Soient $K_n$ et $K_m$ deux cliques tels que $m > n \geq 2$ et soient $a$ et $b$ respectivement le quotient et le reste de la division euclidienne de $m$ par $n$. Si $C$ est un code identifiant de $K_n\Box K_m$ de cardinalité minimum alors :*

$$|C| = \begin{cases} n(2a-1)+2b & \text{si } a \geq 2 \text{ et } 0 \leq b \leq n-1; \\ \left\lfloor \frac{3n}{2} \right\rfloor + b & \text{si } a = 1 \text{ et } 1 \leq b \leq \left\lfloor \frac{n}{2} \right\rfloor; \\ n+2b & \text{si } a = 1 \text{ et } \left\lceil \frac{n}{2} \right\rceil \leq b \leq n-1. \end{cases}$$

# 3 Générer des codes $r$-identifiants dans $Q_n$ par une Recherche Tabou

L'hypercube ou l'espace binaire de Hamming de dimension $n$, noté $Q_n$, est le graphe dont les sommets représentent les $n$-uplets de $\mathbb{F}^n = \{0,1\}^n$, tels que deux vecteurs de $\mathbb{F}^n$ sont adjacents si et seulement si ils diffèrent en exactement une coordonnée.

Soit $M_r(n)$ la cardinalité minimun d'un code $r$-identifiant dans l'espace binaire de hamming $Q_n$. Dans [4], Charon et al. ont combiné deux méthodes les heuristiques (la méthode de bruitage et l'algorithme glouton) et les constructions théoriques afin de générer de bons codes $r$-identifiant dans $Q_n$ et par la suite améliorer les bornes existants dans la littérature.

Nous reprenons la même démarche théorique en la combinant avec la méthode de recherche tabou, une méthaheuristique parmi les plus étudiées dans la littérature, pour construire des codes $r$-identifiant dans $Q_n$ afin d'améliorer les bornes sur $M_r(n)$.

## 3.1 Principe de la méthode

On fixe d'abord les valeurs du rayon $r$, la dimension de l'hypercube $n$ et le nombre des mots de code $M$. Par la suite, on génère un code initial $C \in \mathbb{F}^n$ avec $M$ mots de code puis on calcule $NC(C)$ le nombre de sommets non $r$-couverts par $C$, $NS(C)$ le nombre de sommets non $r$-séparés par $C$ et la fonction d'évaluation

$$f(C) = NC(C) + NS(C)$$

qu'on essaye de rendre nulle.

A chaque itération, on modifie le code courant $C$ par une *transformation élémentaire* qui consiste à remplacer un mot de code par un sommet hors du code tout en gardant $|C| = M$. Pour se faire, on passe cycliquement par tous les mots de code, c'est à dire après la visite du dernier mot de code, on recommence avec le premier mot de code.

Supposons qu'on visite un mot de code $m$, pour tout vecteur $s$ de $\mathbb{F}^n \backslash C$, si $s$ n'est pas dans la liste tabou $T$ on pose

$$C_{m,s} = C \backslash \{m\} \cup \{s\}$$

On garde le sommet $s^*$ qui minimise la fonction $f(C_{m,s}), s \in \mathbb{F}^n \backslash C, s \notin T$, on effectue une transformation élémentaire tel que :

$$C := C_{m,s} = C \backslash \{m\} \cup \{s\}$$

Après chaque transformation élémentaire, on vérifie la fonction d'évaluation pour le code courant :

• Si $f(C) \neq 0$, on ajoute $s^*$ à la fin de la liste tabou, on enlève le sommet en tête de la liste tabou et on passe au mot de code suivant.

• Si $f(C) = 0$, alors $C$ est un code $r$-identifiant de dimension $M$, on initialise le processus en enlevant de $C$ le mot de code $m$ qui minimise $f(C\backslash\{m\})$, et on applique la procédure encore une fois sur un code de dimension $M - 1$.

Les étapes de la méthode sont résumés par l'algorithme suivant :

**Algorithm 1** Recherche Tabou pour trouver un code $r$-identifiant dans un espace binaire de Hamming de dimension $n$.

**Require:** $n >= 3$ et $1 \leq r \leq n - 1$
  Fixer la dimension de l'espace de Hamming $n$, et le rayon $r$
  Fixer le nombre de mots de code $M$, et générer un code initial $C \subseteq \mathbb{F}^n$ de taille $M$
  Calculer $NC(C)$ le nombre de vecteurs de $\mathbb{F}^n$ non $r$-couverts par le code $C$
  Calculer $NS(C)$ le nombre de vecteurs de $\mathbb{F}^n$ non $r$-séparés par le code $C$
  Calculer *la fonction d'évaluation* $f(C) = NC(C) + NS(C)$
  Initialiser $MaxIter$ le nombre d'itérations maximums
  T :=$\emptyset$ {Initialiser la liste tabou}
  $NIter := 0$
  j :=1 {sélectionner le premier mot de code $C$ }
  arret :=false
  **while** $(j \leq M)$ **and** (arret=false) **do**
    **for all** vecteur $s \in \mathbb{F}^n\backslash C$ **do**
      **if** $s \notin T$ **then**
        $C_{j,s} := C\backslash\{j\} \cup \{s\}$
      **end if**
    **end for**
    Soit $s^*$ tel que $C_{j,s^*} = \min_{s \notin T, s \in \mathbb{F}^n \backslash C} C_{j,s}$,
    $C := C_{j,s^*}$ {Mise à jour de code}
    **if** $f(C) = 0$ **then**
      $arret := true$
    **else**
      **if** la liste tabou n'est pas pleine **then**
        Ajouter $s^*$ à la fin de la liste
      **else**
        Enlever un vecteur en tête de T
        Ajouter le vecteur $s^*$ en fin de T
      **end if**
      $NIter := NIter + 1$
      **if** $NIter > MaxIter$ **then**
        $arret := true$
      **end if**
    **end if**
    **if** $j = M$ **then**
      $j := 1$
    **else**
      $j := j + 1$
    **end if**
  **end while**

### 3.2 Expérimentation et Résultat

Cette application nous a permis de générer des codes $r$-identifiant dans l'espace binaire de hamming pour $2 \leq n \leq 21$ et $r \in \{1, ..., 5\}$, mais sans avoir pu améliorer les bornes déjà existants dans la littérature.

**Tab. 1.** Bornes inférieures et supérieures de $M_r(n)$ pour $n \in \{1, ..., 21\}$ et $r \in \{1, ..., 5\}$

| | $r$=1 | | $r$=2 | | $r$=3 | | $r$=4 | | $r$=5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | Bor Inf | Bor Sup | Bor Inf | Bor Sup | Bor Inf | Bor Sup | Bor Inf | Bor Sup | Bor Inf | Bor Sup |
| 2 | a 3 | B 3 | - | - | - | - | - | - | - | - |
| 3 | b 4 | A 4 | f 7 | B 7 | - | - | - | - | - | - |
| 4 | d 7 | C 7 | g 6 | G 6 | f 15 | B 15 | - | - | - | - |
| 5 | b 10 | A 10 | a 6 | G 6 | l 9 | H 10 | f 31 | B 31 | - | - |
| 6 | m 19 | D 19 | a 8 | G 8 | a 7 | H 7 | l 18 | H 18 | f 63 | B 63 |
| 7 | e 32 | E 32 | h 14 | F 14 | a 8 | H 8 | l 13 | H 14 | l 31 | H 32 |
| 8 | c 56 | H 61 | a 17 | F 21 | a 10 | H 13 | a 9 | H 13 | l 19 | H 21 |
| 9 | c 101 | H 112 | a 26 | H 32 | a 13 | H 17 | a 10 | H 14 | l 12 | H 17 |
| 10 | c 183 | H 208 | i 41 | H 57 | a 18 | H 25 | a 12 | H 16 | a 11 | H 16 |
| 11 | c 337 | H 352 | i 67 | H 100 | a 25 | H 36 | a 15 | H 20 | a 12 | H 17 |
| 12 | c 623 | H 684 | i 112 | H 177 | a 39 | H 67 | a 19 | H 33 | a 14 | H 22 |
| 13 | c 1158 | H 1280 | i 190 | H 318 | a 61 | H 109 | a 27 | H 47 | a 17 | H 26 |
| 14 | c 2164 | H 2550 | i 326 | H 566 | a 95 | H 180 | a 38 | H 76 | a 21 | H 43 |
| 15 | c 4063 | H 4787 | i 567 | H 1020 | a 151 | H 305 | a 54 | H 123 | a 28 | H 64 |
| 16 | c 7654 | H 9494 | i 995 | H 1844 | a 241 | H 530 | a 77 | H 192 | a 37 | H 94 |
| 17 | c 14169 | H 18558 | i 1761 | H 3476 | a 383 | H 901 | a 121 | H 305 | a 53 | H 136 |
| 18 | c 27434 | H 35604 | i 3141 | H 6430 | a 608 | H 1628 | a 190 | H 511 | a 77 | H 210 |
| 19 | c 52155 | H 65536 | i 5638 | H 12458 | a 959 | H 2846 | a 304 | H 835 | a 112 | H 326 |
| 20 | c 99329 | H 131072 | i 10179 | H 25401 | k 1593 | H 5813 | a 489 | H 1710 | a 161 | H 663 |
| 21 | c 189829 | H 262144 | i 18471 | H 50342 | j 2722 | H 11477 | a 792 | H 3358 | a 229 | H 1310 |

| Borne inférieure | | | Borne supérieure | |
|---|---|---|---|---|
| a [8, Th 1(iii)] | f [1, Th 5] | j [9, Cor 5] | A [8] | F [5, Tableau 3 et 4] |
| b [8, Th 1] | g [1, Th 6] | k [9, Cor 7] | B [1, Th 5] | G [1, Th 6] |
| c [8, Th 3] | h [5, Table 4] | l [4] | C [2, Th 4] | H [4] |
| d [2, Th 4] | i [9, Cor 4] | m [6, Th 11] | D [2, Th 5] | I [6] |
| e [2, Th 11] | | | E [2, Th 6] | |

## 4 Conclusion

Dans la première partie de ce papier, nous avons établi des bornes supérieures sur la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques de tailles différentes $K_n \square K_m$ suivant les valeurs de $a$ et $b$ où $m = an + b$ avec $b \leq n - 1$. Tous les cas possibles sont résumés dans le tableau 2.

**Tab. 2.** Récapitulatif

| **a** | **b** | **Borne supérieure** |
|---|---|---|
| $a \geq 2$ | $0 \leq b \leq n-1$ | $n(2a-1)+2b$ |
| $a = 1$ | $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$ | $\lfloor \frac{3n}{2} \rfloor + b$ |
| $a = 1$ | $\lceil \frac{n}{2} \rceil \leq b \leq n-1$ | $n + 2b$ |
| $a = 1$ | $b = 0$ | $\lfloor \frac{3n}{2} \rfloor$ (valeur exacte [7, Théorème 1] |

Dans la deuxième partie, nous avons appliqué une métaheusitique recherche tabou pour construire des codes identifiant à distance $r \geq$ dans l'espace binaire de Hamming de dimension $n \geq 3$, mais sans avoir pu améliorer les bornes déjà existants.

Comme perspective de recherche, la détermination du valeur exacte de $M_r(n)$ et la recherche de bornes pour d'autres types de produit des graphes.

## Références


1. Blass U., Honkala I., Litsyn S. : On binary Codes for Identification. Journal of Combinatorial Designs, Vol. 8,(2000) 151–156, .
2. Blass U., Honkala I., Litsyn S. : Bounds on identifying codes. Discrete Math., 241(1-3) (2001)119–128 .
3. Charon I., Hudry O., Lobstein A. : Minimizing the Size of an Identifying or Locating-Dominating Code in a Graph is NP-Hard. Theorical Computer Science, vol. 290. Issue 3(2003) 2109–2120.
4. Charon I., Cohen G., Hudry O., Lobstein A. : New identifying codes in the binary Hamming space. European Journal of Combinatorics, vol 31 (2010) 491–501.
5. Exoo G., Laihonen T., Ranto S. : Improved upper bounds on binary identifying codes. IEEE Transactions on Information Theory, vol 53 (2007) 4255–4260 .
6. Exoo G., Laihonen T., Ranto S. : New Bounds on Binary Identifying Codes. Discrete Applied Mathematics, vol 156 (2008) 2250–2263.
7. Gravier S., Moncel J., Semri A. : Identifying codes of Cartesian product of two cliques of the same size. The Electronic Journal of Combinatorics, (2008) 15 N4.
8. Karpovsky M.G, Chakrabarty K., Levitin L.B. : On a new class of codes for identifying vertices in graphs, IEEE Transactions on Information Theory, Vol. 44(2) (1998) 599–611.
9. Laihonen T., Ranto S. : Codes identifying sets of binary words with large radii. in : Proc. Workshop on Coding and Cryptographie, Versailles, France, (2007) 215–224.

# Approximation du problème de KDHPP dans un graphe cubique

Kheffache Rezika[1],Giannakos Aristotelis[2], Hifi Mhand [2] and Ouafi Rachid[3]

[1] Mouloud Mammeri University, Tizi Ouzou, Algeria. [2] Picardie University, Amiens, France. [3] University of Technology and Sciences, Algiers, Algeria.
kheffache.rezika@yahoo.fr, aristotelis.giannakos@u-picardie.fr, hifi@u-picardie.fr, rouafi@usthb.dz

**Résumé** Dans cet article, nous proposons un algorithme d'approximation pour résoudre le problème du chemin hamiltonien à k dépôts dans un graphe métrique, cubique et 2-sommet-connexe. Le problème étudié peut être considéré comme une variante du problème du chemin hamiltonien (cf., Demange [1] et Malik et al [2]). Pour ce problème, nous montrons l'existance d'un algorithme d'approximation qui retourne une solution avec au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes. L'algorithme proposé est basé sur l'utilisation du couplage parfait pour supprimer des arêtes contrairement à celui de Christofides qui ajoute les arêtes du couplage au graphe.

**Mots Clés :** Algorithme d'approximation, graphe 2-arête connexe, Couplage parfait, Chemin Hamiltonien.

## 1 Introduction

Le problème du voyageur de commerce (TSP) est l'un des problèmes les plus étudiés en optimisation combinatoire et en particulier en approximation. Dans la version la plus standard du problème, on nous donne une métrique (V,d) et l'objectif est de trouver un circuit du coût minimum qui visite chaque point de V exactement une fois. Ce problème est APX-dur et la meilleure approximation connue est $\frac{3}{2}$ qui a été obtenue par Christofides.

Dans une version plus générale du problème appelé le problème du chemin de voyageur de commerce (TSPP), en plus d'une métrique (V,d), deux points sont donnés $s, t \in V$ et le but est de trouver un chemin de s à t visitant chaque point exactement une fois. Pour ce problème, le meilleur algorithme d'approximation connu est celui de Hoogeveen [3] avec un facteur d'approximation de $\frac{5}{3}$.

Très récemment, des progrès significatifs ont été accomplis en approximation du TSP et TSPP. Tout d'abord, Oveis Gharan et al [4] ont donné un algorithme avec un rapport d'approximation de $\frac{3}{2} - \epsilon$ pour le TSP. Par la suite, Momke et Svensson [5] ont obtenu un facteur nettement mieux de $\frac{14(\sqrt{2}-1)}{12\sqrt{2}-13} \approx 1.461$, ainsi que le facteur de $3 - \sqrt{2} + \epsilon \approx 1.586 + \epsilon$ pour le TSPP, pour tout $\epsilon > 0$.

La nouvelle approche est basée sur l'ajout et la suppression des arêtes appariés. Ce processus est guidé par une paire d'arêtes dite amovible qui code essentiellement les informations sur lesquelles les arêtes peuvent être retirées du graphe sans le déconnecter.

Dans cet article, nous étudions le problème du chemin Hamiltonien à k dépôts (kDHPP) qui est une généralisation du problème du voyageur de commerce (TSP). Nous montrons l'existence d'un algorithme polynomial qui donne une solution avec au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes. L'algorithme proposé est principalement basé sur l'utilisation d'un couplage parfait pour supprimer des arêtes amovibles. Le résultat ci-dessus est établi en particulier lorsque le graphe est métrique, 2-arête-connexe et cubique.

## 2 Définition du problème

Soit G(V,E) un graphe complet. $D = \{d_1, d_2, ..d_k\}$ un ensemble de k dépôts distincts. $U = \{1, 2, 3, ...n\}$un ensemble de n destinations avec $(n \geq 2)$,
$V = D \bigcup U$. A chaque arête (i, j) est associé un coût C(i,j).

Un chemin parcouru par un voyageur l est une séquence de sommets
$P_l = \{d_l, v_1^l, v_2^l .....v_{m_l}^l\}$, l=1,2,..k où $m_l$ est le nombre de sommets visités par le $l^{ème}$ voyageur et $v_{lj} \in U$ pour tout $j \in \{1.....m_l\}$.

L'objectif du problème est de trouver les chemins $P_1, P_2, ...P_k$ tel que :

(1) Chaque destination de U soit visitée une seule fois par un seul voyageur.
(2) Chaque voyageur visite au plus une destination.
(3) La somme totale des coûts $\sum_{l=1}^{k} C(P_l)$ soit minimum.

On note ce problème par kDHPP.

Dans cette article, nous résolvons le problème de kDHPP dans un graphe cubique, 2-sommet connexe et métrique.

## 3 Préliminaires

### 3.1 Graphe 2 sommet-connexe :

Un graphe G est 2 sommet-connexe (ou 2-connexe) si lorsqu'on lui supprime un sommet quelconque le graphe reste connexe.

### 3.2 Arête amovible

Une arête e d'un graphe 2-connexe G est dite arête amovible si $G - e$ est encore 2-connexe où $G - e$ désigne le graphe obtenu de G en supprimant e.

**Lemme 1** *Le graphe obtenu en éliminant les arêtes amovibles tel que au plus une arête de chaque paire est retirée est connexe.*

**Théorème 1** *[6] Dans un graphe cubique, 2-sommet-connexe G =(V,E), on peut trouver un couplage parfait M tel que :*

$$Pr_M[e \in M] = \frac{1}{3}, \forall e \in E.$$

**Théorème 2** *Si G est un graphe 2-connexe, cubique et M un couplage parfait de G alors le graphe obtenu par suppression d'arêtes amovibles du couplage M est de degré pair.*

**Preuve :**
Soit $G = (V, E)$ un graphe 2 sommet connexe et cubique, soit e une arête amovible de G.
On a d'après le théorème cité si dessus :

- $\forall e \in E$, $e \in M$ avec une prob de $\frac{1}{3}$
- au plus une arête de chaque paire amovible est dans M.

$\forall v \in V$, v est de degré pair dans le multi graphe $(V, E \bigcup M)$

On suppose que tous les graphes sont 2-sommet-connexe. Soit G un tel graphe. Soit F un DFS (Depth-first search) forêt couvrante de G avec une racine r arbitraire. Toutes les arêtes appartenant à F appelées arête-arbre et toutes les autres arêtes $e \in G \setminus F$ appelées arêtes-retour.

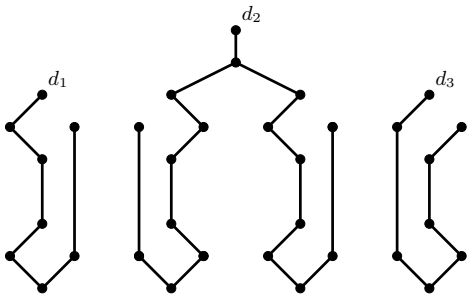
## 4 Algorithme d'approximation :

1. **Entrée** : G=(V,E) un graphe métrique, cubique, 2- connexe.
   $D = \{d_1, d_2, d_3, ..., d_k\}$ un ensemble de k dépôts.
2. Soit F une DFS (depth first search) forêt contenant k arbres.
3. On définit une paire amovible comme suit : Chaque arête retour va dans un sommet appelé v(si v n'est pas racine) est couplé avec une arête arbre de v vers le successeur de v.
4. Soit M un couplage aléatoire parfait de G tel que chaque arête est dans M avec la probabilité de $\frac{1}{3}$.

5. Pour chaque arête e de M qui appartient à la paire amovible, supprimer e de G. Pour chaque arête e de M qui n'appartient pas à la paire amovible, ajouter e à G, créer une arête en double. On obtient une composante connexe Eulerienne.
6. Appliquer le shortcutting pour le graphe obtenu.
7. **Sortie** : k chemins Hamiltoniens.

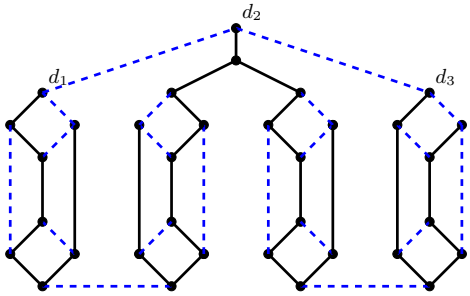
## 5 Exemple : k=3

Dans cette section, On montre comment appliquer les différentes étapes de l'algorithme pour construire la solution finale de l'algorithme d'approximation.



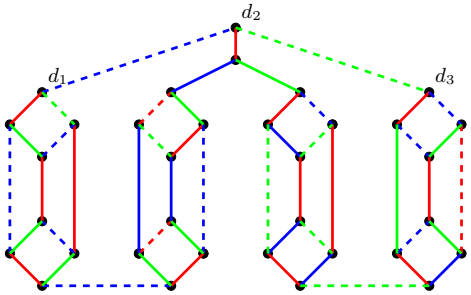

DFS forêt de trois arbres

**Fig. 1.**



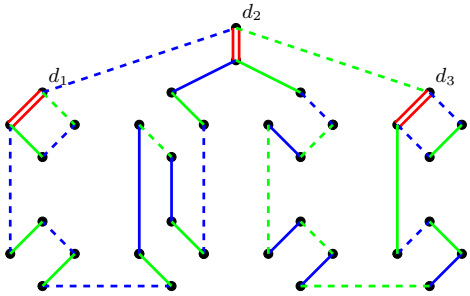

**En gras** : Arête-arbre et
**en pointillés** : Arête-retour

**Fig. 2.**



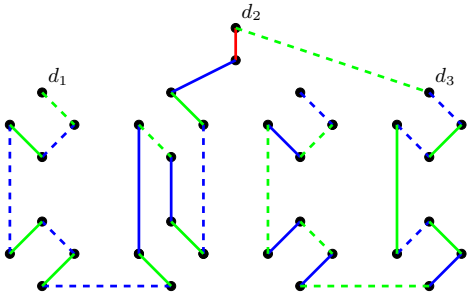

Couplage parfait avec la probabilité
de chaque arête est $\frac{1}{3}$

**Fig. 3.**



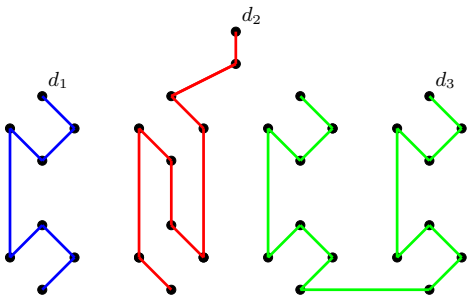

Suppression des arêtes amovibles

**Fig. 4.**




Shortcutting pour avoir un chemin Hamiltonien

**Fig. 5.**




La solution finale obtenue par l'algorithme

**Fig. 6.**

## 6 Résultat

L'algorithme présente une approximation polynomiale et donne la solution avec au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes, dans le cas d'un graphe cubique et 2-arête-connexe.

### 6.1 Preuve :

a) Montrons que l'algorithme est de complexité $O(n^3)$

Dans l'algorithme, on construit une forêt, un couplage, un chemin Eulerien et un chemin Hamiltonien mais l'algorithme est dominé par deux étapes : recherche de forêt et de couplage parfait.

soit G(V,E) un graphe tel que $|V| = n$, $|E| = m$. Le parcours en profondeur est de complexité $O(m + n)$ et comme dans un graphe cubique $|E| = \frac{3}{2}n$ donc la complexité est de $O(\frac{5}{3}n)$.

Le problème du couplage parfait qui est un couplage maximum est été résolu par Edmonds [8] en $O(n^3)$ d'où la complexité de l'algorithme proposé est de $O(n^3)$.

b) Montrons maintenant que la solution contient au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes

Soit H un ensemble contenant toutes les arêtes-arbres et arêtes-retour.
G=(V, E) est cubique donc $|E| = \frac{3}{2}n$.
Soit b le nombre d'arêtes-retour de G :

$$b = |E| - (n-k) = \frac{3}{2}n - n + k = \frac{1}{2}n + k$$

avec $k$ : nombre d'arbres qui forment la forêt.

Comme toutes les amovibles sont des paires donc pour chaque arête retour, on a exactement une autre arête amovible sauf celles reliant les dépôts d'où :

$$|H| = 2b - (k-1) = 2(\frac{1}{2}n + k) - k + 1 = n + k + 1$$

Donc on a $\frac{n+k+1}{2}$ arêtes amovibles et les autres sont inamovibles.
Le nombre d'arêtes inamovibles :

$$\frac{3}{2}n - \frac{n+k+1}{2} = \frac{2n-k-1}{2}$$

Chaque arête peut être dans le couplage parfait pris en considération avec une probabilité de $\frac{1}{3}$, si elle est dans le couplage, soit c'est une arête amovible ou inamovible.

D'où : le nombre d'arêtes du chemin Eulerien obtenu est :

$$\textit{arêtes de } G + \textit{arêtes inamovible} * \textit{prob} \in M - \textit{arêtes amovibles} * \textit{prob} \in M$$

$$= \frac{3}{2}n + \frac{1}{3}(\frac{2n-k-1}{2}) - \frac{1}{3}(\frac{n+k+1}{2}) = \frac{5}{3}n - \frac{k+1}{3}$$

Soit K le nombre d'arêtes des chemins Hamiltoniens obtenus :

$$K \leq \frac{5}{3}n - \frac{k+1}{3} - (k-1) = \frac{5}{3}n - \frac{4k-2}{3}$$

## Références


1. M. Demange. Algorithme d'approximation : un petit tour en compagnie d'un voyageur de commerce, Mathématiques et Informatique, Gazette 102, pp. 53-90, 2004.
2. W. Malik, S. Rathinam, S. Darbha. An approximation algorithm for a symmetric generalized multiple depot, multiple traveling salesman problem, Operations Research Letters, vol. 35, pp. 747-753, 2007.
3. J.A. Hoogeveen. Analysis of Christifides' heuristic : Some paths are more difficult than cycles. Operations Research Letters, vol.10, pp. 291-295, 1991.
4. S.O. Gharan, A. Saberi et M. Singh. A randomized rounding approach to the travelling salesman problem. In FOCS'11, pp. 550-559, 2011.
5. T. Momke and O. Svensson. Approximating graphic TSP by matchings. In FOCS'11, pp. 560-569, 2011.
6. B.S. Munson Clyde, L. Monma and R.W. Pulleyblank. Minimum-weight two-connected spanning networks. Mathematical Programming, pp. 153–171, 1990.

7. M. Held and R.M. Karp. The travelling salesman problem and minimum spanning trees'. Operations Research, vol. 18(6) pp. 1138-1162, 1970.
8. J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices, journal of Research of the National Bureau of Standards vol. 69B, pp. 125-130, 1965.

# Optimal Identifying Codes in Oriented Paths and Circuits

Ahmed SEMRI[1] and Hillal TOUATI[1]

[1] LaROMaD, Faculty of Mathematics, USTHB,
BP 32 Bab Ezzouar,El-Alia 16111, Algiers, Algeria.
ahmedsemri@yahoo.fr touatih@gmail.com

**Abstract.** Identifying codes in graphs are related to the classical notion of dominating sets [9]. Since there first introduction in 1998 [7], they have been widely studied and extended to several application, such as: detection of faulty processor in multiprocessor systems, locating danger or threats in sensor networks.
Let G=(V,E) an unoriented connected graph. The minimum identifying code in graphs is the smallest subset of vertices C, such that every vertex in V have a unique set of neighbors in C. In our work, we focus on finding minimum cardinality of an identifying code in oriented paths and circuits.

**Keywords:** Identifying code, Oriented paths, Circuits.

## 1 Introduction

The theories and the applications of identifying code attracted the attention of many researchers since their first introduction by Karpovsky et al in [7]. This led to many results that have been obtained in hypercubes [8, 10], grids [6, 4], paths and cycles [2, 3, 5].
Let $G = (V, E)$ a simple, connected and undirected graph, where $V$ is the set of vertices and $E$ the set of edges. We call a *code* any noempty subset of vertices and its elements a *codewords*. We define $B_r(v)$, a *ball* of center $v$ and radius $r$ by $B_r(v) = \{u \in V | d(u, v) \leq r\}$, where $d(x, y)$ denotes the length (number of edges) of the shortest path between the vertices $x$ and $y$.
Thus, an $r$-identifying code is any subset $C \subseteq V$ such that:

i. $\forall\ v \in V, B_r(v) \cap C \neq \emptyset$,
ii. $B_r(u) \cap C \neq B_r(v) \cap C$, for all $u, v \in V, u \neq v$.

Therefore, the first condition ensures that every vertice of the graph is covered by at least one codeword, and the second one ensures that every pair of different vertices is separated. In other words, each vertex of the graph $G$ is covered by a unique set of codewords. The set $B_r(v) \cap C$, denoted also by $I_r(v)$, is called the *r-identifying set* of $v$ (simply identifying set when $r = 1$).
For an oriented graph $G = (V, A)$, we just replace $B_r(v) \cap C$ by $\Gamma_r^-(v) \cap C =$

$I_r^-(v)$, where the set $\Gamma_r^-[x] = \{y \in V \mid d(y,x) \leq r\}$ contains all the predecessors at distance at most $r$ from $x$ ($x$ within).

The problem with identifying code is finding one with the fewest elements. This problem is known to be an NP-complete problem [1].
Our work studies this problem in oriented graphs, particularly in oriented paths and circuits. Thus, some partial results were obtained.

## 2 Identifying Code in Oriented Paths

As mentioned before, we are interested in finding an optimal identifying code in oriented paths and circuits. First, we give some notations that will be used in the next paragraphs.
We denote by $\mathcal{P}_n$ an oriented path of length $n$, ie it contains exactly $n+1$ vertices, and $\mathcal{C}_n$ a circuit of length $n$. Let $M_r^-(G)$ denotes the minimum cardinality of an $r$-identifying code in graph $G$.

First, we investigate the 1-identifying code (or simply identifying code, if there's no ambiguity) then the 2-identifying code.

### 2.1 1-Identifying Code

**Lemma 1.** *A subset $C \subseteq V$ is an identifying code in $\mathcal{P}_n$ if and only if:*

1. *The two vertices $x_0$ and $x_1$ belong to the code $C$,*
2. *For every pair of consecutive vertices $x_i$ and $x_{i+1}$, $i \in \{2, 3, \ldots, n-1\}$, $x_i$ or $x_{i+1}$ is a codeword.*
3. *For every triplet of consecutive vertices $x_i$, $x_{i+1}$ and $x_{i+2}$, $i \in \{2, 3, \ldots, n-2\}$, $x_i$ or $x_{i+2}$ is a codeword.*

*Proof.* For (1), $x_0$ is covered by itself, then $x_0$ must be a codeword. In addition, $x_1$ must belong to code to separate the pairs of vertices $(x_0, x_1)$.
For the second condition, suppose that $x_i \notin C$ and $x_{i+1} \notin C$. Then $\mathcal{I}^-(x_{i+1}) = \emptyset$ ($x_{i+1}$ isn't covered). Then either $x_i$ or $x_{i+1}$ must belong to the code.
For (3), suppose that neither $x_i$ nor $x_{i+2}$ belong to the code. Then we have two cases:

**Case 1** If $x_{i+1} \in C$, then $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = \{x_{i+1}\}$, ie the two vertices $x_{i+1}$ and $x_{i+2}$ aren't seperated.
**Case 2:** If $x_{i+1} \notin C$, necesseraly the two vertices $x_{i+1}$ and $x_{i+2}$ will not be covered because $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = \emptyset$.

Thus, in the two cases either $x_i$ or $x_{i+2}$ must be a codeword.
One can see the necessity and the sufficiency of the three conditions to cover all the vertices of $\mathcal{P}_n$, this comes from the fact that every semi-ball contains exactly two consecutive vertices.
Now, let's show the sufficiency of the three conditions for the separation.
Let $x_i$ and $x_j$ be two vertices, then we have two cases:

**Case 1** The vertices are neighbours. Without loss of generality, we put $j = i+1$. Above, we have shown that Condition (1) separates the vertices $x_0$ and $x_1$. Therefore, by Condition (3), we known that $x_{i-1} \in C$ or $x_{i+1} \in C$, then we have $\mathcal{I}^-(x_i) \neq \mathcal{I}^-(x_j)$. Thus, $x_i$ and $x_j$ were separated.

**Case 2** $x_i$ et $x_j$ are not neighbours, ie the distance $d(x_i, x_j) \geq 2$. Suppose, without loss of generality, that $j = i + 2$. Then, we have $\Gamma_1^-[x_i] = \{x_{i-1}, x_i\}$ and $\Gamma_1^-[x_j] = \{x_{i+1}, x_{i+2}\}$, but by Condition (2), we have $\mathcal{I}^-(x_i) \neq \mathcal{I}^-(x_j)$. Then $x_i$ and $x_j$ are separated.

This completes the proof of the lemma. □

By the following theorem we give a minimum cardinality of an identifying code in oriented pathsFirst, we investigate the 1-identifying code (or simply identifying code, if there's no ambiguity) then the 2-identifying code.

**Theorem 1.** *For an oriented path $\mathcal{P}_n$, we have:*

$$M_1^-(\mathcal{P}_n) = \begin{cases} 2p & \textit{if } n{=}3p, \\ 2p+1 & \textit{if } n{=}3p{+}1, \\ 2p+2 & \textit{if } n{=}3p{+}2. \end{cases}$$

*Proof.* Let $V$ the set vertices of $\mathcal{P}_n$. If we denote by $L$ the set of vertices identified by one codeword (or covered by one codeword). Then, the other vertices ($|V| - |L|$) are covered by at least two codewords. In other words, $C$ double covers these vertices. Thus, using the fact that $|L| \leq |C|$ (at most $|C|$ vertices are covered by one codeword), therefore we have the following inequality

$$2(|V| - |L|) + |L| \leq \sum_{x_i \in C} |\Gamma_1^-[x_i]| \leq 2.|C|$$

so

$$\begin{aligned} 2|V| - |L| \leq 2|C| &\Leftrightarrow 2|V| - |C| \leq 2|C| \\ &\Leftrightarrow \tfrac{2}{3}|V| \leq |C| \end{aligned}$$

which leads to

$$|C| \geq \left\lceil \frac{2n}{3} \right\rceil$$

Let $n = 3p + q$, with $q \in \{0, 1, 2\}$. Thus we obtain:

$$\left\lceil \frac{2n}{3} \right\rceil = \left\lceil \frac{2(3p+q)}{3} \right\rceil = 2p + \left\lceil \frac{2q}{3} \right\rceil$$

Therefore: If $q = 0$, then $\lceil \frac{2q}{3} \rceil = 0$. If $q = 1$, then $\lceil \frac{2q}{3} \rceil = 1$. And finaly, if $q = 2$, then $\lceil \frac{2q}{3} \rceil = 2$.

To conclude, we exhibit an identifying code which reaches the bound for each case. Thus, we can choose $C = \{x_i | i \equiv 0[3] \text{ and } i \equiv 1[3]\}$ for all cases (see figure 1). □

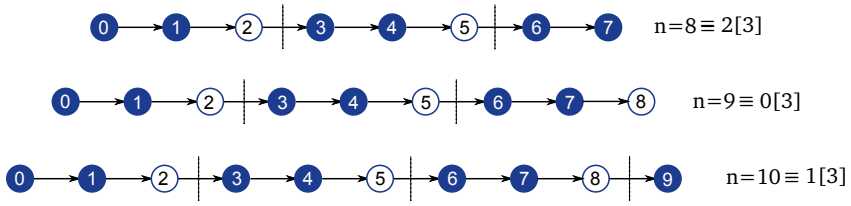



**Fig. 1** – An Example of identifying code in oriented paths of length 7,8 and 9

### 2.2 2-Identifying Code

Before proceding to the proof of our results we need the following result:
Let $\mathcal{P}_n = \{x_0, x_1, \ldots, x_n\}$ an oriented path of length $n$, and $C$ a code in $\mathcal{P}_n$.

**Lemma 2.** *A subset $C$ is a 2-identifying code in $\mathcal{P}_n$ if and only if the following three conditions are satisfied:*

1. *The vertices $x_0$,$x_1$ and $x_2$ must belong to $C$,*
2. *For every group of three consecutive vertices, $x_i, x_{i+1}, x_{i+2}$, $i \in \{3, 4, \ldots, n-2\}$, at least one belong to the code $C$,*
3. *For every group of four consecutive vertices, $x_i, x_{i+1}, x_{i+2}, x_{i+3}$, $i \in \{3, 4, \ldots, n-3\}$, we can't have $x_i \notin C$ and $x_{i+3} \notin C$.*

*Proof.* For the condition (1), if $x_0 \notin C$, then we have $\Gamma_2^-[x_0] = \emptyset$ (the vertex is not covered). Thus $x_0$ must be a codeword.
For (2), we can see that if any of the three vertices $x_i, x_{i+1}, x_{i+2}$, for all $i \in \{3, 4, \ldots, n-2\}$, is not a codeword then $\mathcal{I}^-(x_{i+2}) = \emptyset$ which contradicts the fact that $C$ is a covering code.
Finaly for the condition (3), suppose that neither $x_i$ nor $x_{i+3}$ is in $C$ then the two vertices $x_{i+2}$ and $x_{i+3}$ will not be separated because $\mathcal{I}_2^-(x_{i+2}) = \mathcal{I}_2^-(x_{i+3}) = \{x_{i+1}, x_{i+2}\}$. Thus $x_i \in C$ or $x_{i+3} \in C$, for all $i \in \{3, 4, \ldots, n-2\}$.
We remark that the condition (1) and (2) are necessary and sufficient for the condition that $\Gamma_2^-[x_i] \cap C \neq \emptyset$ for all $i \in \{0, 1, \ldots, n\}$.
We need to show that the three conditions of the previous lemma are sufficient for the separation. Let $x_i$ and $x_j$ be two distinct vertices. Thus, two cases appear:

**Case 1** The two vertices are neighbours, $j = i + 1$. In this case, by (1) we have the pairs $(x_0, x_1)$, $(x_1, x_2)$, $(x_2, x_3)$ seperated, and by the condition (3), we have $x_{i-3} \in C$ and $x_i \in C$ for all pairs $(x_i, x_{i+1})$, where $i \in \{3, 4, \ldots, n-1\}$. We can observe that $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_{i+1})$ for all pairs of consecutive vertices. Thus, the vertices $x_i$ and $x_j$ are separated by the code $C$.

**Case 2** The two vertices $x_i$ and $x_j$ are at distance at least 2, ie $d(x_i, x_j) \geq 2$. In this case, if $d(x_i, x_j) > 2$ $(j > i + 2)$, then by (2) we have $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_j)$, and if $d(x_i, x_j) = 2$, then by the conditions (2) and (3) we have also $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_j)$. Thus, in this case, also $x_i$ and $x_j$ are separated by the code $C$. □

For more clearness, we will denote each vertex $x_i$ by its subscript $i$.
We know that each 2-identifying code is 2-separator in $\mathcal{P}_n$. Also, we know that for each vertex $i \in \mathcal{P}_n$ $(i \geq 2)$ we have $\Gamma_2^-[i] \Delta \Gamma_2^-[i+1] = \{i-2, i+1\}$[1], then $\forall\ i \in \mathcal{P}_n$ one of the vertices $i-2$ and $i+1$ must belong to $C$ (condition (3) of lemma 2). Thus, we have $i-2 \in C$ or $i+1 \in C$ for each vertex $i \in \{2, n-1\}$ from $\mathcal{P}_n$ (see figure 2). Such disjunction will be called (ie $i-2 \in C$ or $i+1 \in C$) *Elementary Constraint* (EC), so it is abbreviated as $i-2 \vee i+1$.

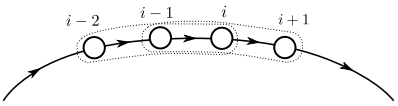



**Fig. 2** – One of the two vertices $i$ or $i+3$ belong to a code to separate $i+2$ and $i+3$

Next, we introduce an example which clear up some notations that we will use in the rest of this paper.

**Example 1** *Let $\mathcal{P}_{10} = x_0, x_1, \ldots, x_9$, an oriented path of length 9. To obtain a 2-identifying code we have to separate nine pairs of consecutive vertices. Thus, we have nine ECs to satisfy those we enumerate as follows :* $0 \vee 3,\ 3 \vee 6,\ 6 \vee 9,\ 1 \vee 4,\ 4 \vee 7,\ 2 \vee 5,\ 5 \vee 8$. *Note that we omit the two ECs which separate the pairs (0,1) et (1,2), because they are separated (condition (1) of lemma 2). The set of these constraints will be called General Constraint (GC). This set of elementary constraints (or GC) can partitioned in three subsets of constraints, called Partial Constraints (PC), such as :*

$$0 \vee 3, 3 \vee 6, 6 \vee 9$$
$$1 \vee 4, 4 \vee 7$$
$$2 \vee 5, 5 \vee 8$$

*However, in order to get a general formulation, we give some adaptation for this notation. Thus, we can write the above PCs as follows:*

$$0 \vee 0+1\times 3, 0+1\times 3 \vee 0+2\times 3, 0+2\times 3 \vee 0+3\times 3 \quad (C_0)$$
$$1 \vee 1+1\times 3, 1+1\times 3 \vee 1+2\times 3 \quad (C_1)$$
$$2 \vee 2+1\times 3, 2+1\times 3 \vee 2+2\times 3 \quad (C_2)$$

[1] $A \Delta B = A \cup B \setminus A \cap B$, called symmetric difference

*we call* $C_i,\ i = 0, 1, 2$*, the partial constraint* $i$*.*

In general, if the number of vertices is $n$, then we suppose that $n = 3p + q$, with $p \in \{1, 2, \ldots, [\frac{n}{3}]\}$ and $q \in \{0, 1, 2\}$. Thus the PC $i$ has the following form:

$$i \vee i + 1 \times 3, i + 1 \times 3 \vee i + 2 \times 3, \ldots, i + (s_i - 1) \times 3 \vee i + s_i \times 3$$

where $s_i$ is the greatest integer for which the following inequality checked:

$$i + s_i \times 3 \leq n.$$

In the above example, we have $s_0 = 3$ for the PC $C_0$, .

Let $V_i$ be the set of vertices in the PC $i$. We remark that $V_i \cap V_j = \emptyset$, for all $i \neq j | i, j \in \{0, 1, 2\}$. In other word, all the PCs have disjoint sets of vertices.
Thus, satisfying the GC, to obtain an 2-identifying code, amounts to satisfy all PCs.
Using this notation, we get the following result:

**Theorem 2.** *Given an oriented path* $\mathcal{P}_n$ *of length* $n$*, where* $n = 3p + q$ *and* $q \in \{0, 1, 2\}$*. If* $C$ *is an* 2*-identifying code in* $\mathcal{P}_n$*. Then:*

**(1)** *If* $p = 0$*, then* $M_2^-(\mathcal{P}_n) = q + 1$

**(2)** *If* $q = 0$*,* $p \geq 1$*,* $M_2^-(\mathcal{P}_n) = \begin{cases} \frac{3p}{2} + 1 & if\ p\ is\ even \\ \frac{3(p+1)}{2} & if\ p\ is\ odd \end{cases}$

**(3)** *If* $q = 1$*,* $p \geq 1$*,* $M_2^-(\mathcal{P}_n) = \begin{cases} \frac{3p}{2} + 2 & if\ p\ is\ even \\ \frac{3(p+1)}{2} & otherwise \end{cases}$

**(4)** *If* $q = 2$*,* $p \geq 1$*,* $M_2^-(\mathcal{P}_n) = \begin{cases} \frac{3p}{2} + 2 & if\ p\ is\ even \\ \frac{3(p+1)}{2} + 1 & otherwise \end{cases}$

*Proof.* If $p = 0$ ($n \leq 2$) the minimum cardinality of a 2-identifying code in $\mathcal{P}_n$ is deduced from the first condition of lemma 2.

For the second case, ie $p \geq 1$ and $q = 0$, we know, by the condition (1) of lemma 2, that the vertices $0, 1$ and 2 belong to the code, which satisfies the first EC of the PCs 0,1 and 2.
On the other hand, we have, by condition (2) of lemma 2, necessary at least one codeword between $3, 4$ and 5, thus we have one EC between the PCs 0,1 and 2 for which two vertices are a codeword. Without loss of generality, let 3 this vertex, then this satisfies two ECs in the PC 0. In this case, we need to satisfy $(p - 3)$ ECs, then at least $\lceil \frac{p-3}{2} \rceil$ codeword are needed to satisfy the rest of ECs in PC 0.
In addition, for each of the partial constraints 1 and 2 we have one elementary constraint satisfied (since $1, 2 \in C$), then $(p - 2)$ ECs aren't satisfied for each one. Thus, at least $\lceil \frac{p-2}{2} \rceil$ codewords are needed to satify the rest of ECs for PCs

1 and 2.
We conclude that we need, totally, at least:

$$4 + 2\left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-3}{2} \right\rceil$$

codewords to satisfy the general constraint.

If $p$ is even, then $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 1$. Else, $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$.
Finally, we construct a 2-identifying code that reaches the bound to conclude. Indeed, we use the following construction:
We take all vertices $i \in V$, where $i$ is even and adding vertices 0 and 1.


**Fig. 3** – 2-identifying code ($C = \{2, 4, 6, 8\} \cup \{0, 1\}$) for an oriented path of length 8 ($n = 3 \times 3 + 0$ vertices)

The proof of the case $p \geq 1$ and $q = 1$ is similar. Indeed, concerning the first EC of CP 0, we have $(p-1)$ ECs to satisfy, since $0 \in C$. Thus, we need at least $\lceil \frac{p-1}{2} \rceil$ codewords. For the PCs 1 and 2 we need, respectively, at least $\lceil \frac{p-3}{2} \rceil$ and $\lceil \frac{p-2}{2} \rceil$ codewords to satisfy the rest of elementary constraints. Thus, we need at least :

$$4 + \left\lceil \frac{p-3}{2} \right\rceil + \left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-1}{2} \right\rceil$$

codeword.
Then, if $p$ is even, then $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 2$. Else, $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$.

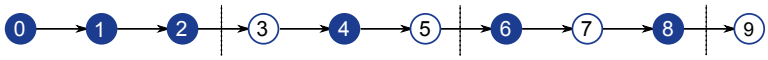



**Fig. 4** – A 2-identifying in an oriented path having $n = 3 \times 3 + 1$ vertices ($p = 3$, $q = 1$)

To conclude, we exhibit a code reaching these bound. Indeed, we remark that for the code $C = \{i | i \text{ is even}\} \cup \{0, 1\}$ the bound is attained (see figure 4).

Finally, the proof for the last case ($p \geq 1$ and $q = 2$) is also similar, we have $(p-1)$ ECs to satisfy for the PCs 0 and 1, and we need respectivey at least $\lceil \frac{p-1}{2} \rceil$ and $\lceil \frac{p-2}{2} \rceil$ codewords. For the PC 2 we have $(p-1)$ EC, since $2 \in C$ then

we have $(p-2)$ ECs to satisfy, then at least $\lceil \frac{p-2}{2} \rceil$ codewords are needed. Thus, we need at least :

$$4 + 2\left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-1}{2} \right\rceil$$

codewords to satisfy the GC.
If $p$ is even, then $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 2$. If $p$ is odd, then $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$. To conclude, we just consider the same construction as the previous cases to exhibit a 2-identifying code reaching the bound. □

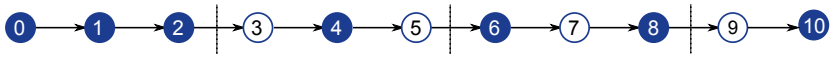



**Fig. 5** – An Example of 2-identifying code in an oriented path with $3 \times 3 + 2 = 11$ vertices ($p = 3$, $q = 2$)

## 3 Identifying Code in Circuits

In circuits we give an optimal 2-identifying code.

### 3.1 2-Identifying Code

In the case of circuit, the two conditions of the lemma 2 are still valid. Then:

**Lemma 3.** *Let $\mathcal{C}_n = \{1, 2, \ldots, n, 1\}$ a circuit of length $n$. $C$ is a 2-identifying code for $\mathcal{C}_n$ if and only if the following conditions are satisfied:*

1. *For all group of three consecutive vertices $x_i, x_{i+1}$ and $x_{i+2}$ at least one of them is a codeword,*
2. *For all group of four consecutive vertices $x_i, \ldots, x_{i+3}$ we could not have $x_i \notin C$ and $x_{i+3} \notin C$.*

*Proof.* The proof is similar to lemma 2. Except adding the condition that the distance between $i$ and $j$ isn't greater than that between $j$ and $i$ to show the sufficiency and the necessity of the conditions (1) and (2). □

In the case of circuit, Although the reasoning is similar as in oriented path, there is, however, some differences. Thus, we define a partial constraint $i$ ($i \in \{1, 2, 3\}$) as follow:

$$i \vee i + 1 \times 3, i + 1 \times 3 \vee i + 2 \times 3, \ldots, i + (s_i - 1) \times 3 \vee i + s_i \times 3, i + s_i \times 3 \vee h_i$$

where $s_i$ is the greatest integer such:

$$i + s_i \times 3 \leq n$$

and $h_i$ is such that $i + (s_i + 1) \times 3 \equiv h_i(\mod [n])$, (ie $h_i \in \{1, 2, 3\}$).

**Example 2** *Let $\mathcal{C}_n = \{1, 2, \ldots, n, 1\}$ be a circuit of length $n$. Suppose that $n = 10$, thus $p = 3$ and $q = 1$. Then the PCs, 1,2 et 3, can be written as follows:*

$$\begin{array}{l} 1 \vee 4, 4 \vee 7, 7 \vee 10,\ 10 \vee 3, \quad (i = 1) \\ 2 \vee 5, 5 \vee 8,\ 8 \vee 1 \quad (i = 2) \\ 3 \vee 6, 6 \vee 9,\ 9 \vee 2 \quad (i = 3) \end{array}$$

*denoting by $i|j$ the elementary constraint $i \vee j$. Then the GC is written:*

$$1|4|7|10|3|6|9|2|5|8|1$$

*If, for example, $n = 12$, then the GC will be:*

$$1|4|7|10|1, 2|5|8|11|2, 3|6|9|12|3$$

Thus, we have determined the optimal 2-identifying code. The result is given by the following theorem:

**Theorem 3.** *Let $C_n$ be a circuit of length $n$. Then:*

$$M_2^-(C_n) = \begin{cases} \emptyset & \text{if } n \leq 3, \quad (1) \\ 3 & \text{if } n = 4, \quad (2) \\ k & \text{if } n = 2k, k \geq 3, \quad (3) \\ k + 1 & \text{if } n = 2k + 1, k \geq 2, \quad (4) \end{cases}$$

*Proof.* For (1), it is clear that, if $n \leq 3$, then $\mathcal{C}_n$ can't admit a 2-identifying code because there are twin vertices[2].
For (2), we show that there is no 2-identifying code of cardinality 2 in a circuit of length 4. Indeed, suppose that there are only two vertices as codeword. Without loss of generality, let 1 and 3 be these vertices, then $I_2^-(1) = I_2^-(3) = \{1, 3\}$. Therefore at least three vertices must belong to a code. Finally, it suffices to exhibit a code with cardinality 3 to conclude (see figure 6).

[2] We call twin vertices every two vertices $u, v$ such that $\Gamma_r^-(u) = \Gamma_r^-(v)$

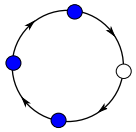


**Fig. 6** – Exemple de code 2-identifiant dans $\mathcal{C}_4$

Concerning (3), ie the case where the length of the circuit is even ($n = 2k$), we know that there is $\frac{n}{2} = k$ ECs to satisfy, therefore we need at least $k$ codewords. It suffices to exhibit a 2-identifying code of cardinality $k$ to conclude. Thus, we can take as a code the set $C = \{i | i \text{ even}, \ 1 \leq i \leq n\}$ (see the figure 7) hence the result.

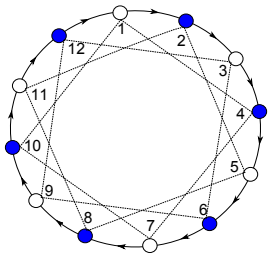



**Fig. 7** – An example of 2-identifying code in circuit of length 12. Six codewords are needed to cover and separate all the vertices of the circuit.

Similarly to the previous case, when $n = 2k + 1$ (n is odd) we need at least $k + 1$ codewords to satisfy all the ECs.
Thus, there is at least one EC which has its vertices belong to the code. We want to show that $|C| > k + 1$. To do it, we suppose that we can find a 2-identifying code $C$ of cardinality $k+1$ in a circuit of length $n$, and we get to a contradiction. Since we have $k + 1$ vertices as codewords, then necessarily two codewords are adjacent. Without loss of generality, let 1 and 2 these two vertices, or one of the two vertices $n$ and 3 must be a codeword by the condition (2) of lemma 3. Thus, for every 2-identifying code at least three consecutive vertices are codewords.
Now, there are two cases:

**Case 1:** Suppose that the length of the circuit is equal to $n = 4p + 3$ ($k = 2p+1$). Since at least three consecutive vertices are codewords ( as mentioned previously), then we need to cover and separate $4p$ vertices. But by conditions (1) and (2) of lemma 3, we know that for every four consecutive vertices,

at least two of them are codewords. Thus, we need at least $2p$ vertices as codeword, therefore $2p+3 = k+2$ vertices belong to a code (see the example of the figure 8). Hence the contradiction.

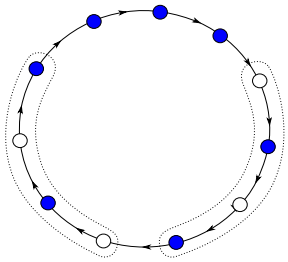


**Fig. 8** – An example of optimal 2-identifying code of cardinality $2 \times 2 + 3$ in circuit of length $4 \times 2 + 3$

**Case 2:** In this case, we have $n = 4p + 1$ ($k = 2p$). Observing this case, we see that it's similar to the first one. We have $n = 4p+1 = [4(p-1)+3]+2$ ($k = 2p$). Thus, $2(p-1)+3 = k+1$ vertices are codewords among the $4(p-1)+3$ vertices that a circuit contains (by condition (2) of lemma 3) adding the three consecutive vertices belong to the code.

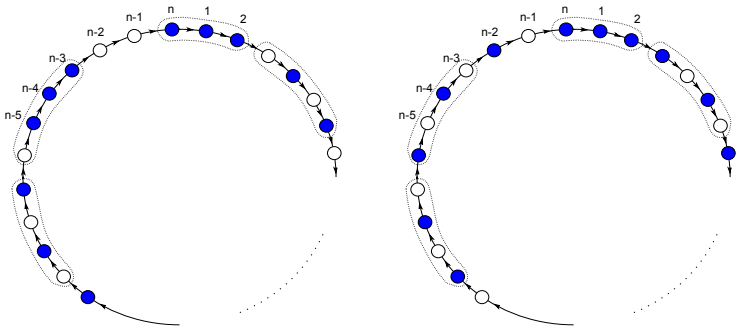



**Fig. 9** – An example of two optimal 2-identifying code

In the case where the remaining two vertices don't belong to the code, saying, without loss of generality, $n-1$ and $n-2$, then necessarely the three vertices $n-3$,$n-4$ and $n-5$ belong to the code (see the left representation in figure 9). Thereby, we have $4(p-2)$ vertices that are couvered and separated by $2(p-2)$ codewords plus $n-3$,$n-4$ et $n-5$ and the three consecutive vertices. In total there is $2(p-2)+6 = 2p+2 = k+2$ codewords.

If, either $n-1$ or $n-2$ belong to the code then, we will need $2(p-1)$ codewords for covering and separating the $4(p-1)$ vertices (conditions of lemma 3) adding the three consecutive vertices (see the left representation in figure 9). Thus, in total, we will have $1+2(p-1)+3 = 2p+2 = k+2$ codewords.

Therefore, in the two cases, we will have at least $k+2$ codewords.
We conclude by exhibiting a 2-identifying code reaching this bound. The latter, constituted of the set of vertices $C = \{2\} \cup \{i \equiv 1[2], 1 \leq i \leq n\}$. □

## 4 Conclusion

In this work we gave some results about identifying code in oriented paths and circuits. It remains to determine the minimum cardinality for the case of a 1-identifying code in circuit. In addition, the question of the general case, ie $r$-identifying code is also an open problem.


## Acknowledgements

The authors wish to thank anonymous reviewers for careful reading and detailed comments which improved the quality of the paper.


## References


[1] I. Charon, O. Hudry et A. Lobstein. Minimizing the size of an identifying or locating-dominating code in graph is NP-Hard. *Theoretical Computer Science*, 290:2109–2120, 2003.

[2] N. Bertrand, I. Charon, O. Hudry et A. Lobstein. Identifying and locating-dominating codes on chains and cycles. *European Journal of Combinatorics*, 25:969–987, 2004.

[3] S. Gravier, J. Moncel et A. Semri. Identifying codes of cycles. *European Journal of Combinatorics*, 27:767–776, 2006.

[4] G. Cohen, L. Honkala, S. Gravier, A. Lobstein, M. Mollard et C. Payan. Improved Identifying Codes For the Grid. *Electronic Journal of Combinatorics*, 1999.

[5] D. L. Roberts et F. S. Roberts. Locating sensors in paths and cycles: The case of 2-identifying codes. *European Journal of Combinatorics*, 29:72–82, 2008.

[6] M. Daniel, S. Gravier et J. Moncel. Identifying code in some subgraphs of the square lattice. *Theoretical Computer Science*, 319:411–421, 2004.

[7] M. G. Karpovsky K. Chakrabarty et L. B. Levitin. On New Class of Codes for Identifying Vertices in Graphs. *IEEE Transactions On Information Theory*, 44(2):599–611, 1998.

[8] U. Blass, L. Honkala et S. Litsyn. Bounds on Identifying Codes. *Discret Mathematics*, 241:119–128, 2001.

[9] M. Laifenfeld, A. Trachtenberg et T.Y. Berger-Wolf. Identifying Codes and the Set Cover Problem. *Annual Allerton Conf. on Comm, Ctrl and Comput* $44^{th}$, 2006.

[10] G. Exoo, T. Laihonen, S. Ranto et V. junnila. Upper Bounds For Binary Identifying Codes. *Advances in Applied Mathematics*, (42):277–289, 2009.

# Fouille de données et apprentissage

# Organisation Sémantique des Métadonnées des applications de Datamining Haute Performance dans un Système de Stockage de Cloud Computing

Sonia IKKEN[1], M-Tahar KECHADI[2] , A.Kamel TARI[1]

[1] Université Abderrahmane Mira Bejaia
[2] Université College Dublin, Ireland

**Résumé**.Le Cloud Computing (CC) est un nouveau paradigme informatique qui a suscité beaucoup d'intérêt au sein de la communauté des chercheurs et de l'entreprise. Son objectif est de fournir de puissantes capacités de stockage et de calcul, et une gestion de ressources excellente basée sur la virtualisation de manière à fournir des services en ligne à la demande pour différents types d'utilisateurs. Actuellement, le CC représente un environnement idéal pour le stockage et le traitement des applications de datamining haute performance (DMHP). Par ailleurs, les difficultés actuelles du CC les plus pertinentes sont la congrégation des quantités massives de données venant du Web et des applications scientifiques telles que le datamining distribué. Ils doivent maintenir ces énormes quantités de données hétérogènes tout en fournissant une recherche d'information efficace. Ainsi, le CC ne sera pas en mesure d'utiliser les méthodes actuelles de gestion de données pour répondre à la croissance de la demande et aux exigences de telles applications. Ce document analyse fondamentalement l'utilité d'une nouvelle approche pour résoudre la situation ci-dessus. Ainsi, nous proposons dans ce papier les premiers éléments de notre approche pour l'organisation des métadonnées et des données des applications de DMHP dans un environnement de CC.

**Mots Clés**: Cloud Compuing, DMHP, Système de fichiers Distribué, Métadonnée, Ontologie, Knowledge Map.

## 1 Introduction

Actuellement, la majorité des plateformes de calcul pour le datamining sont des systèmes de types grille, grappe distribuée et maintenant le Cloud. Ces systèmes sont dotés de plusieurs processeurs et mémoires (noeud) qui doivent être partagés parmi les utilisateurs. Dans un environnement de Cloud, les algorithmes de datamining sont distribués et exécutés sur ces plateformes en utilisant les services de bibliothèque de Cloud tout en minimisant les communications qui sont souvent très couteuses, ainsi que les traitements parallèles doivent être les plus autonomes et effectués avec le moins de synchronisation possible [1]. Par le biais de ces plateformes de calcul et de stockage de données les applications de DMHP ont vu le jour, et mises en oeuvre par les utilisateurs ou les fournisseurs de services pour traiter d'énormes ensembles de données distribuées et stockées sur

des différents sites. Ces applications font recours au calcul parallèle et distribué et/ou à l'utilisation de nouvelles structures de données, qui apparaît comme une solution naturelle à ce problème [2]. Cependant, dans un tel traitement parallèle et distribué, il reste toujours quelques problèmes comme: l'organisation et la localisation des données, la mise en échelle des algorithmes; et plus précisément la précision des résultats des techniques de datamining dans un environnement de CC. En effet, l'une des difficultés de CC les plus pertinentes auxquelles les entreprises et les utilisateurs personnels sont confrontés est la congrégation des quantités massives de données structurées et non-structurées venant du Web et des applications scientifiques telles que le datamining parallèle et distribué. Ils doivent maintenir ces énormes objets de données hétérogènes tout en fournissant une recherche d'information efficace. Le DMHP génèrent beaucoup de résultats (appelés connaissances) dans le Cloud, le problème, est que le Cloud à travers le monde commence à exiger des quantités toujours croissantes pour le stockage des données et des résultats de ces applications. D'autre part, le Web sémantique est un domaine émergent pour augmenter le raisonnement humain, et représenter pertinemment les données structurées hétérogènes. Il existe plusieurs propriétés décrivant les applications de datamining, qui représentent des métadonnées de haut niveau. Ces métadonnées peuvent être exprimées par un modèle du Web sémantique. Dans un environnement distribué, un système de fichiers prend en charge la gestion et le stockage des métadonnées sur les données brutes, mais pas nécessairement au niveau de l'application, notamment celle de datamining. En outre, le CC ne sera pas en mesure d'utiliser les méthodes actuelles pour organiser les métadonnées au niveau de l'application, et accéder de manière efficace aux résultats de ces applications. En s'appuyant sur une vision élargie des métadonnées d'une manière plus intelligente, de nouvelles méthodes permettront aux individus et aux organisations d'extraire encore plus de nouvelles chaînes de valeur à partir des données brutes.

Motivé par les remarques ci-dessus, ce papier tente de proposer une future vision pour la "gestion de données des applications de DMHP dans le Cloud". Ainsi, ce document traite globalement différents aspects, et analyse fondamentalement l'utilité d'une nouvelle approche pour répondre aux besoins des applications de DMHP, et de proposer les premiers éléments de notre approche.

La suite de ce papier est organisée comme suit: la section 2 présente un état de l'art sur le DMHP et le CC. La section 3 présente les travaux antérieurs. La section 4 annalyse l'utilité d'intégration de nouveaux concepts pour les appliactions de datamining dans le Cloud. La section 5 décrit l'approche proposée. Enfin, la section 6 conclut ce papier et suggère des travaux futurs.

## 2 Etat de l'art

### 2.1 Datamining Haute Performance

Le datamining est l'extraction automatique de modèles représentants des connaissances implicitement stockées à partir d'énormes quantités de données [3].

Généralement, le datamining possède deux objectifs de haut niveau de description et de prédiction, qui peuvent êtres réalisés en utilisant une variété de méthodes de datamining, par exemple, les règles d'association, les motifs séquentiels, la classification, la régression et bien d'autres méthodes. D'un point de vue de l'utilisateur, l'exécution d'un processus de datamining et la découverte d'un ensemble de modèle peut être considérée soit comme une réponse à une requête d'une base de données sophistiquée ou bien d'un résultat obtenu lors de l'exécution d'un ensemble de tâches de datamining. La première est appelée approche descriptive, tandis que la seconde est une approche procédurale. Pour soutenir la première approche, plusieurs langages de requête de datamining ont été développés [4][5]. Dans cette dernière approche, les applications de datamining sont considérées comme un processus de découverte de connaissances complexes composées de différentes tâches de traitement de données. Le datamining traitant de grandes quantités de données peut grandement bénéficier de l'utilisation d'environnement informatique parallèle et distribué, mais aussi d'autres techniques pour atteindre la haute performance et améliorer la précision des modèles découverts. Ces environnements permettent un calcul parallèle intensif pour le datamining sur des données distribuées, ce qui est dénommé le DMHP. Les algorithmes développés dans ce domaine abordent le problème pour obtenir efficacement les résultats d'exploration de données à travers des sources distribuées. Ainsi, des efforts de recherche importants ont été investis dans la mobilisation des infrastructures informatiques distribuées pour mettre en oeuvre des systèmes de DMHP.

### 2.2 Cloud Computing

Le CC est une forme de traitement et de stockage basé sur des services massivement évolutifs liés aux IT, qui sont fournis à la demande à travers l'Internet à de multiples clients externes [6]. Cela signifie que le calcul se fait dans un endroit éloigné et inconnu (dans les nuages d'Internet) plutôt que sur un poste de travail local. Les clients utilisent des ressources informatiques virtualisées sur un modèle de paiement à l'utilisation [7] comme un service, et sont libérés de tous problèmes de fourniture de matériel et de logiciel. Les applications peuvent être déployées sur des Clouds publics, privés et hybrides [8]. La décision concernant le modèle de Cloud doit être sélectionnée qui dépend de nombreux facteurs, parmi lesquels le coût, la confidentialité, le contrôle de données, la sécurité et la qualité de service (QoS). Le CC implique l'utilisation de l'infrastructure en tant que service sur le réseau. La question est de savoir comment les différentes infrastructures constituent une base de ces services. La virtualisation jette les bases pour le partage sur une infrastructure à la demande, à laquelle trois catégories de base de Cloud [8] sont offertes:

– **Infrastructure as a Service (IaaS)**: le niveau le plus bas de la pile de CC, fournissant aux utilisateurs un accès à la demande aux ressources virtuelles, qu'ils peuvent configurer entièrement. Les utilisateurs peuvent louer des ressources de calcul, de stockage ou de mise en réseau pour des périodes

prédéfinis. Ils ont accès aux services de gestion d'infrastructure tels que le déploiement automatique des ressources et l'échelle dynamique du nombre de noeuds loués.
- **Platform as a Service (PaaS)**: les services de PaaS consistent en un environnement intégré à un niveau élevé qui permet à des utilisateurs d'établir, d'examiner et d'exécuter leurs propres applications (système d'exploitation, middleware, environnement de développement et logiciels d'application).
- **Software as a Service (SaaS)**: les services de SaaS permet aux applications complètes d'un utilisateur final d'être déployer, gérer et livrer comme un service généralement par le biais d'un navigateur sur Internet. Le SaaS prend en charge uniquement les applications des fournisseurs sur les infrastructures et les plateformes de Cloud.

Les infrastructures basées sur les concepts de CC permettent la réalisation du développement et de l'utilisation des tâches de datamining portant sur des données distribuées à grande échelle. Par conséquent, le CC est adapté à la résolution des problèmes de DMHP.

## 3 Travaux connexes

### 3.1 Infrastructures distribuées pour le DMHP

Des efforts de recherches importants ont été investis dans le développement des systèmes de DMHP, d'intégration et d'accès aux données. Des infrastructures qui exploitent le middleware de grille ont été mise en oeuvre pour fournir des services de haut niveau d'exploration de données qui combinent les ressources matérielles et logicielles de nombreux sites dispersés. Des systèmes comme Knowledge Grid [9] et Admire [10] ont été proposés. L'architecture de Knowledge Grid se base sur les services fournis par Globus Toolkit. L'architecture se compose de deux couches : la couche noyau de Knowledge Grid, qui se concentre principalement sur la gestion de métadonnées. Elles utilisent un service de répertoire de connaissances KDS (Knowledge Directory Service) pour gérer les métadonnées notamment sur les répertoires et les sources de données, les outils et les algorithmes d'exploration de données, les plans d'exécution distribués et les résultats du calcul, à savoir les modèles de datamining. La seconde couche est le haut niveau de Knowledge Grid, qui inclut des services pour composer, valider et exécuter des calculs pour la découverte de connaissances. Dans les travaux de [11], un modèle XML est proposé pour l'architecture de Knowledge Grid, décrivant toutes les métadonnées pour la gestion et l'exécution des applications de découverte de connaissance distribuées. Ce modèle permet aussi d'associer une sémantique aux métadonnées de ces applications, et de gérer automatiquement les informations représentées dans les documents de ces métadonnées. Le projet ADMIRE est un cadre pour l'exploration et la découverte de connaissance, qui possède deux principaux niveaux hiérarchiques : le niveau datamining et le niveau Knowledge Map. Il utilise un service de grille de données basé sur DGET, qui accède aux données et aux ressources à travers des plateformes hétérogènes. Il met l'accent

sur l'intégration des résultats d'extraction et des connaissances distribuées existantes au niveau de son composant Knowledge Map (KM) présenté dans [12].
En effet, les projets cités-ci haut se sont concentrés beaucoup plus sur un l'aspect exploration et intégration de données. Par ailleurs, les infrastructures de CC abordent beaucoup plus les problèmes liés à l'accès aux masses de données distribuées en exploitant les systèmes de stockage distribués. Un système de stockage joue un rôle prépondérant pour la localisation et l'accès aux données dans un environnement de CC. Grossman et al. [13] ont développé une infrastructure basée sur le CC pour l'analyse de grands ensembles de données distribuées. L'infrastructure se compose d'un nuage de stockage appelé Sector et d'un nuage de calcul appelé Sphere. Le Sector a été conçu pour les réseaux haute performance à large zone, et emploie des protocoles spécialisés pour utiliser la bande passante disponible. Le Sector fournit un stockage persistant à long terme à de grands ensembles de données, qui sont gérées comme des fichiers distribué indexés. La Sphere permet des fonctions prédéfinies par l'utilisateur. Les travaux dans [14] explorent le cadre MapReduce de Google [15] pour le traitement des algorithmes de DMHP, à savoir l'algorithme de clustering distribué nommé K-means parallèle et l'algorithme d'extraction de règles d'association. Les deux implémentations utilisent la plateforme Hadoop [16] pour fournir un cadre robuste, scalable et parallèle dans un environnement distribué. D'autres infrastructures de CC pour les applications de données intensives comprennent BigTable [17], Google File System (GFS) [18] et Hadoop Distributed File System (HDFS) [19] qui couple les données et les ressources de calcul pour accélérer le traitement de ces applications.

### 3.2 Système de Stockage Hadoop-HDFS

Les systèmes de stockage dans le Cloud fournissent un support pour l'accès toujours disponible, omniprésent aux données hébergées. Une entité de données virtualisée est disponible en ligne et hébergée sur une variété de plusieurs serveurs virtuels. Ces approches de stockage dans le Cloud sont plus utilisées dans les scénarios du monde réel pour faciliter l'accès aux données brutes hautement disponibles à l'échelle d'Internet. Cela inclut le service de stockage de Amazon S3 "Simple Storage Service" [20] et le système de stockage de Hadoop. Ce dernier est un projet de Apache,qui fournit un cadre java libre destiné aux applications distribués et à la gestion intensive des données. Il permet aux applications de travailler avec des milliers de noeuds et des pétaoctets de données. Tous les composants de Hadoop sont disponibles via une licence open source de Apache. Hadoop a été inspiré par les publications MapReduce, GoogleFS et BigTable de Google. HDFS est le le système de fichiers distribué de Hadoop.
HDFS possède une architecture de type maître/esclave sur un cluster dans un réseau. HDFS stocke les métadonnées du système de fichiers et les données des applications séparément. Comme dans d'autres systèmes de fichiers distribués, tel que PVFS [21] et Lustre [22], HDFS stocke les métadonnées sur un serveur dédié, appelé le namenode (serveur maitre). Les données des applications sont

stockées sur d'autres serveurs appelés datanodes. Tous les serveurs sont entièrement connectés et communiquent entre eux en utilisant les protocoles basés sur TCP. Pour rendre les données persistantes, le contenu d'un fichier est répliqué sur plusieurs datanodes pour un maximum de fiabilité. La figure 1 illustre le fonctionnement de HDFS. L'espace de noms de HDFS est une hiérarchie de fichiers et de répertoires. Les fichiers et les répertoires sont représentés sur le namenode par des inodes, ce dernier s'occupe de la cartographie des blocs des datanodes pour gérer toutes les métadonnées locales. Les fichiers de données étant volumineux, ils sont découpés en blocs typiquement de taille de 128MB et stockés sur plusieurs datanodes. Les datanodes se chargent des requêtes de lecture et d'écriture. Ils ont également comme tâche les opérations de création, replication ou suppression de fichiers. Les clients communiquent à la fois avec le namenode et les datanodes. Les fichiers dans HDFS peuvent êtres récupérer en utilisant leur métadonnées. Celles-ci contiennent des informations comme par exemple la taille du fichier, le numéro d'inode, le nom associé à un inode, les droits d'accès et l'adresse IP d'un bloc de données. Ces métadonnées permettent à l'utilisateur de structurer ses données et au système de fichier de les traiter et de mettre en application des politiques d'accès sur ces données. Les métadonnées (dépendants du système de fichier) sont principalement mises en oeuvre par tous les systèmes de fichiers existants en raison de sa sémantique de bas niveau, ainsi il est difficile pour les applications de niveau supérieur d'accéder aux métadonnées, et les manipulations doivent être effectuées uniquement par le système de fichiers pour des raisons évidentes.

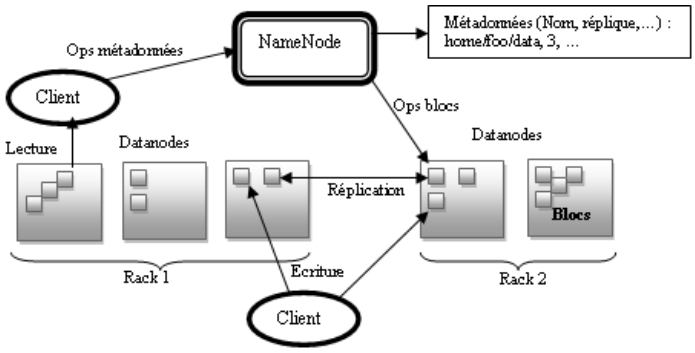



**Fig. 1.** *Architecture de HDFS*

## 4 Métadonnées sémantiques pour le DMHP dans le CC

La découverte de nouvelles connaissances et l'accès aux données des applications de DMHP à partir d'un centre de données et d'un environnement de CC sont des

aspects difficiles. Le concept de CC ne fournit pas d'installations spécifiques pour l'accès aux connaissances distribuées et à la localisation d'information de manière efficace pour les applications de datamining. Bien que HDFS et autre système de stockage dans le Cloud offrent un service de stockage hautement disponible qui ne coûte pas cher à entretenir, ils ciblent principalement les objets de données volumineux qui ne sont pas souvent modifiés et ne sont pas sensible au contexte. Ces système de stockage ne traitent pas spécifiquement le cas où ces données sont en fait des connaissances, à savoir les données structurées sémantiques exprimées par un modèle logique formelle, ou bien les modèles de datamining (clustering, classification, règle...etc.). Ainsi, il est nécessaire que la structure, le schéma et l'architecture doivent être en harmonie avec la découverte et l'accès aux connaissances de datamining dans une infrastructure de CC. Les systèmes de stockages distribués dans le Cloud se tournent pour adopter une architecture qui découple les opérations sur les métadonnées et les données afin d'obtenir un impact significatif sur la scalabilité dans les chemins de données. Bien que la taille des métadonnées soit généralement faible par rapport à la capacité de stockage globale d'un tel système, 50% à 80% de tous les accès au système de fichiers se consentrent sur les métadonnées [23]. Par ailleurs, l'utilisation efficace de Cloud pour les applications de DMHP nécessite l'intégration des métadonnées pour gérer l'hétérogénéité des services de CC, notamment les métadonnées sur les outils et les algorithmes de datamining, les plans d'exécution, les modèle ou patterns de datamining, les informations liées à la provenance de l'environnement de construction et au contrôle de version des fichiers sources de ces applications, qui sont actuellement gérés dans des régimes distincts pour chacun. En outre, il n'existe pas de méthode établie pour un développeur ou un utilisateur de gérer ces types de métadonnées. L'idée est de réfléchir à une méthode pour organiser et accéder d'une manière efficace aux connaissances extraites en considérant les propriétés standards des fichiers sur les métadonnées riches pour les applications de datamining.

Par ailleurs, le manque de description des métadonnées peuvent causer une faible structure pour les données, et avoir une mauvaise interchangeabilité et accessibilité, ainsi une difficulté de maintenir toutes ces données hétérogènes et ces connaissances produites par les techniques de datamining. En effet, le noyau du Web sémantique est les métadonnées. Le Web sémantique est une extension du World Wide Web actuelle (WWW). L'idée de fusionner le CC et le Web sémantique peut parvenir à de nouvelles générations d'applications plus spécifiques et favorables pour celles de datamining. L'ontologie est considérée comme l'une des principales composantes du Web sémantique, utilisée pour représenter, acquérir et réutiliser des connaissances [24] afin d'assister les machines à comprendre le sens du contenu des différentes ressources Web. Les types les plus spécifiques de l'ontologie pour le Web sont la taxonomie et l'ensemble de règles d'inférence. L'ontologie fournit un vocabulaire qui définit les différentes ressources hétérogènes [25]. Pour représenter les métadonnées, il existe différents langages d'ontologie, XML (eXtensible Markup Language) et RDF (Resource Description Framework), mais le choix revient au langage OWL (Web Ontology

Language) pour la construction des ontologies liées aux ressources dans le Cloud [26][27].

## 5 Approche proposée

Á travers les différents concepts et travaux étudiés dans ce papier, la solution que nous proposons est l'adoption d'une couche sémantique pour les métadonnées des applications de DMHP au niveau d'un système de fichier, qui gère des métadonnées d'une sémantique de bas niveau. Pour définir et caractériser les premiers éléments de notre approche, nous utilisons le système de fichiers distribué HDFS pour le stockage des données et des résultats de datamining, et le cadre MapReduce pour le traitement des algorithmes de datamining, tout ceci sur une couche de base d'une infrastructure de Cloud. Par ailleurs, l'efficacité de traitement des problèmes de datamining requiert d'obtenir au préalable un partitionnement intelligent des données de manière à effectuer le plus indépendamment possible le traitement des ensembles de données en tenant compte des caractéristiques spécifiques des services de CC (scalabilité, simplicité d'utilisation, efficacité). Dans notre approche, le partitionnement dépend des connaissances extraites par ces techniques et la relation entre ces connaissances et les données des sources (brutes). Pour faciliter ce processus nous intégrons dans notre archi-

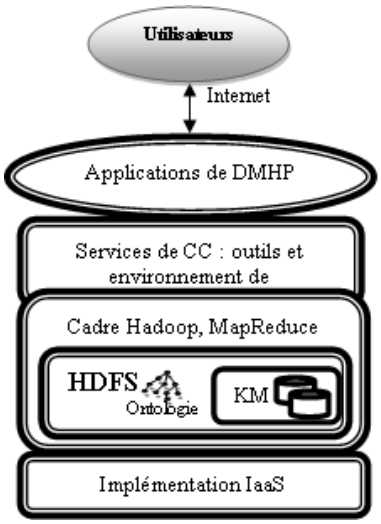



**Fig. 2.** *Architecture du système de traitement et de stockage des applications de DMHP basée sur le CC*

tecture le système de KM, qui intègre plusieurs composants pour récupérer rapidement et efficacement les connaissances extraites de plusieurs sites. Le noyau de

KM manipule les métadonnées sur les connaissances extraites de chaque noeud d'Hadoop. Les connaissances préalables contenues dans KM aiguilleront les données brutes aux nouvelles connaissances en fonction de leur relation et ceci en utilisant une ontologie globale. Les algorithmes de datamining sont utilisés par les applications de DMHP, qui est le niveau supérieur de l'architecture. La figure 2 illustre l'architecture de base.

- Niveau "Infrastructure de CC": concerne les ressources physiques intégrées par une infrastructure de Cloud: machines physiques, équipements de traitement, de stockage et de contrôle, machines virtuelles, système d'exploitation.
- Niveau "Stockage et traitement de données": ce niveau se décompose en deux sous-niveaux : la plateforme Hadoop et le système de KM guidé par une ontologie globale. Le cadre Hadoop résout le problème de l'efficacité pour l'exécution et le stockage des applications de DMHP. Le déploiement d'un cluster Hadoop est exécuté au dessus des machines virtuelles fournies par l'infrastructure de Cloud, sur lequel les programmes Java et les algorithmes des applications de DMHP basés sur le modèle MapReduce peuvent être exécutés. Toutes les données d'entrées de datamining sont stockées dans HDFS. Après exécution des tâches de MapReduce, les résultats aussi seront chargés dans HDFS. Tandis que les utilisateurs des applications de DMHP soumettent de nouvelles requêtes, le système KM récupère les résultats traités et contenus dans chaque noeud d'un cluster Hadoop. Les répertoires de KM représentent une couche logique (répertoires virtuels qui indique les requêtes des utilisateurs) pour le NameNode dans HDFS. Ces répertoires manipulent les métadonnées des connaissances extraites comme: les propriétés et la description des connaissances, et utilise une ontologie pour l'organisation sémantiques de ces métadonnées.
- Niveau "Service de CC" : notre approche n'est pas seulement considérée pour un utilisateur final (entreprise ou utilisateur individuel), mais aussi aux concepteurs des applications de datamining qui développent leur propres programmes sur le Cloud. Ainsi, ce niveau fournit des API de programmation Web pour créer de nouvelles applications basées sur un navigateur, des environnements de programmation et des outils de composition qui facilitent la création, le déploiement et l'exécution de ces applications dans le Cloud.
- Niveau "applications de DMHP": représente les différents types d'applications pour le datamining sous forme de service dans le Cloud.

### 5.1 Organisation sémantique des métadonnées en couche

Le namenode dans HDFS maintient seulement les métadonnées sur les données brutes distribuées sur les datanodes, et ne considère pas les informations au niveau de l'application. L'ajout d'une couche de KM dans HDFS va permettre de maintenir les informations et les propriétés liés aux résultats de DMHP, pour chaque noeud d'un cluster Hadoop. Ainsi les requêtes d'accès et de recherches de connaissances pour HDFS seront plus efficaces et plus rapides. La figure 3 montre cette structuration. Le namenode considère dans son espace de nommage les

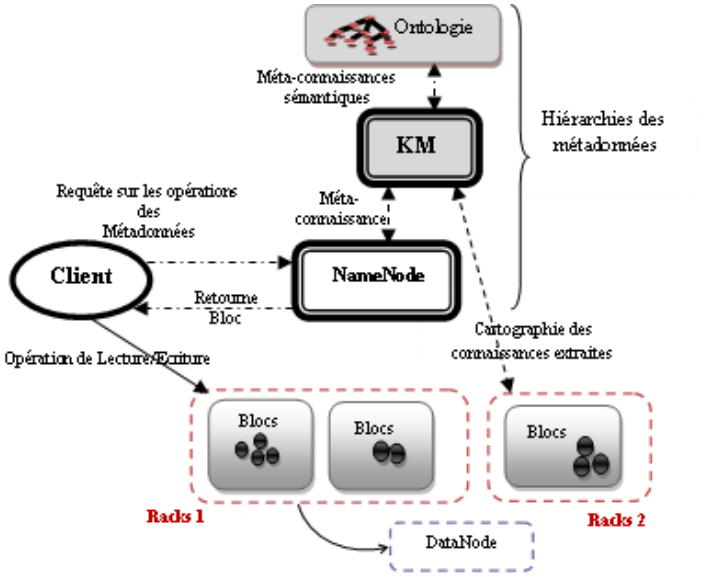



**Fig. 3.** *Organisation sémantique des métadonnées dans HDFS*

répertoires des connaissances de KM, qui sont similaires à des répertoires traditionnels. La recherche dans KM est basée sur une ontologie, qui présente les mots clés du contenu des connaissances et les domaines d'application auxquels les connaissances appartiennent. Pour enrichir l'ontologie, nous utilisons les techniques de datamining qui vont extraire les métadonnées dans le système de KM, et construire les règles pour définir la hiérarchie de l'ontologie. Le développement d'une ontologie au niveau de KM permettra d'ajouter aussi de la sémantique aux connaissances, d'identifier la relation entre les connaissances et les données brutes et de regrouper les métadonnées des connaissances selon leur similarité sémantique. La relation sémantique peut considérer par exemple les fichiers de mêmes résultats (connaissances similaires sur la classification ou sur le clustering), un domaine commun (fichiers liés à un même domaine d'application, fichiers liés à un programme ou à une application) et un ensemble de fichiers de données pour une même requête.

## 6 Conclusion et travaux futurs

Dans ce papier nous avons présenté une nouvelle approche pour l'organisation des métadonnées d'un système de stockage pour accéder aux données des applications de DMHP dans un environnement de CC. L'architecture de base présentée dans ce papier utilise la plateforme Hadoop pour le stockage et le traitement des techniques de datamining. Nous avons scindé le fonctionnement

de notre approche sur le système de fichier HDFS, qui manipule les métadonnées, en lui associant d'autres informations pertinentes par l'exploitation du système KM. Pour rechercher et représenter les connaissances sémantiques de KM nous avons ajouté une ontologie au niveau de ce système. Cependant, il reste beaucoup de travaux qui doivent être réalisés afin 1) de construire l'ontologie, et 2) d'implémenter et d'évaluer l'approche sur une infrastructure de CC.

## References


1. Tannir K., Kadima H., Malek M.: Application de K-Means à la définition du nombre de VM optimal dans un Cloud. In: the 12th edition of the International Francophone Conference EGC. Bordeaux, France (2012).
2. BENAISSA L. KADDAR, MOKEDDEM D.: Construction Parallèle des Arbres de Décision. In: 5th International Conference Sciences of Electronic, Technologies of Information and Telecommunications. TUNISIA University of Science, Technology USTO- Algérie (2009).
3. Han J., Kamber M., Pei J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, University of Illinois at Urbana-Champaign (2006).
4. Hastings S., Langella S., Oster S., Kurc T., Pan T., Catalyurek U., Janies D., Saltz J.: Grid-based management of biomedical data using an XML-based distributed data management system. In: Proceedings of the ACM Symposium on Applied Computing-SAC'05',pp. 105-109. ACM Press. New York (2005).
5. Netz, A., Chaudhuri, S., Fayyad, U. M. and Bernhardt, J.: Integrating data mining with SQL databases: OLE DB for data mining. In: Proceedings of the 17th International Conference on Data Engineering, pp. 379-387 (2001).
6. Coutty A., Vonfelt T.: L'évolution maitrisée vers le IaaS/PaaS. Dans: le livre blanc produit par EuroCloud France, Novembre (2011).
7. Brock M., Goscinski A.: Toward ease of discovery, selection and use of clusters within a cloud. In: IEEE International Conference on Cloud Computing, pp. 289-296 (2010).
8. Brock M., Goscinski A.: A technology to expose a cluster as a service in a cloud. In: Proceedings of the Eighth Australasian Symposium on Parallel and Distributed Computing - vol. 107, pp. 3-12. Australia (2010).
9. Cannataro M., Talia D., Trunfio P.: KNOWLEDGE GRID: High Performance Knowledge Discovery Services on the Grid. In: Second International Workshop Denver, CO, USA, Proceedings (2001).
10. Le-Khac N., Kechadi T., Carthy J.: ADMIRE FRAMEWORK: DISTRIBUTED DATA MINING ON DATA GRID PLATFORMS. In: School of Computer Science and Informatics, University College Dublin (2008).
11. Mastroianni C., Talia D., Trunfio P.: Metadata for Managing Grid Resources in Data Mining Applications. In: Journal of Grid Computing (2004).
12. Nhien An Le Khac, Lamine M. Aouad and M-Tahar Kechadi. Distributed Knowledge Map for Mining Data on Grid Platforms. In: IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.10, October (2007).
13. Grossman R., Gu Y.: Data mining using high performance clouds: Experimental studies using sector and sphere. In: Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2008).
14. C. Chu, S.K. Kim, Y. Lin, Y.Y. Yu, G. Bradski, A.Y. Ng, K.: Mapreduce for machine learning on multicore. In: Proceedings of Neural Information Processing Systems Conference (NIPS) (2007).

15. Dean, J., and Ghemawat, S.: Mapreduce: Simplied data processing on large clusters. In USENIX OSDI (2004).
16. Apache Hadoop. http://hadoop.apache.org/.
17. Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Burrows, D. A. W. M., Chandra, T., Fikes, A., and Gruber, R. E.: Bigtable: A distributed storage system for structured data. In: USENIX OSDI (2004).
18. Ghemawat, S., Gobioff, H., and Leung, S.-T.: The google File system. In: ACM SIGOPS Operating Systems Review - SOSP (2003).
19. Shvachko K., Kuang H., Radia S., Chansler R.: The Hadoop Distributed File System. In: Proceeding of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies-MSST, pp. 1-10 (2010).
20. Amazon Simple Storage Service (Amazon S3), http://aws.amazon.com/fr/s3/.
21. Carns P. H., Ligon III W. B., Ross R. B., Thakur. R.: PVFS: A parallel file system for Linux clusters. In: Proceedings of 4th Annual Linux Showcase and Conference, pp. 317-327 (2000).
22. Lustre File System. http://www.lustre.org.
23. Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, Lan Xue : Efficient metadata management in large distributed storage systems. In: Proceedings of the 20 th IEEE/11 th NASA Goddard Conference on Mass Storage Systems and Technologies-MSS'03, pp. 290. IEEE Computer Society, Washington, DC, USU. (2003).
24. Qin L., Atluri V.: An ontology Guided Approach to change Detection of the Semantic Web Data. In: Journal on Data Semantics V (2006).
25. Davies J.: Prospects for Semantic Technologies. In: IEEE Intelligent Systems 23(1): 76-88 (2008).
26. Amir M. T., RodziahA., Rusli A., Masrah A. A. M.: Security Ontology Driven Multi Agent System Architecture for Cloud Data Storage Security: Ontology Development. In: International Journal of Computer Science and Network Security-IJCSNS. VOL.12 No.5, May (2012).
27. Bernstein D., Vij D.: Using Semantic Web Ontology for Intercloud Directories and Exchanges. In: International Conference on Internet Computing-ICOMP'10. Las Vegas (2010).

# A New Approach for Pretreatment of Large Multi-Dimensional Data using Sampling Methods

Rima Houari[1], Ahcène Bounceur[2] , Tahar Kechadi[3]

[1] University of Abderrahmane Mira Bejaia
[2] Lab-STICC Laboratory - European University of Britanny - University of Brest
[3] University College Dublin, Ireland

**Abstract**

Today we collect large amounts of data and we receive more than we can handle. The accumulated data are raw and often far from good qualities, they contain noise, and mostly redundant information. The presence of these data is major disadvantages for most data mining algorithms. Intuitively, the relevant information is embedded in many attributes and its extraction is only possible if the original data is cleaned. In this paper we propose a new technique which involves eliminating redundant values to reduce dimensionality.

**Key-words** : Data mining, Copulas, Redundant data, Multidimensional Sampling.

## 1 Introduction and previous work

Due to the need for handling massive and high dimensional datasets, many statistical methods, such as multivariate regression analysis [4][5], neural networks [3], Principal Component Analysis (PCA) [4][13], clustering analysis [20], Bayesian analysis [9][10][11][7] and many others [11][12][15], have been adopted in data mining to abstract useful information hidden in large datasets.

For the sake of simplicity, some of these methods assume that all random variables follow a normal distribution. However, the real word data is not homogeneous and each random variable can have different probability distribution, which is a major challenge for the modeling of multivariate analysis probability space. For this reason, we propose a new approach based on the theory of Copulas which involves eliminating Redundant variables.

The paper is organized as follows : the basic concepts of the model are presented in Section 2. Section 3 describes the proposed method, the experimental results are given in Section 4, and finally Section 5 concludes the paper.

## 2 Basic concepts

Let $X = (X_1, X_2, ..., X_m)$ denotes a random vector of $m$ random variables $X_1, X_2, ..., X_m$ . Let also $f_i(x_i)$ and $F_i(x_i)$ denote the marginal PDF and CDF of $X_i$, respectively. The usual approach to the problem is to first estimate the density $f(x_1, x_2, ..., x_m)$ and then simulate it to obtain the required number of random samples of X.

The transformations of $X_i$ according to their CDF into $U_i$, such as $U_i = F_i(X_i)$ are invertible, specifying the dependency between the random variables $X_i$ and $U_i$.

The problem of estimating the density $f(x_i)$ has been converted to estimating a density $c(u_i)$ that has uniform marginal distributions, using probability Sampling on $c(u_i)$ to obtain samples of the random vector $U$ and finally making the inverse transformations $X_i = F_i^{-1}(U_i)$.

Therefore, given that nonparametric density estimation presents a lot of difficulties in high-dimensional spaces, this transformation may largely simplify the density estimation problem. The CDF $C(u_1, u_2, ..., u_m)$ associated with the density $c(u_1, u_2, ..., u_m)$ is called Copula.

### 2.1 Definition and Illustration of Copulas

A m-dimensional Copula C is a m-dimensional distribution function on $[0, 1]^m$ with standard uniform marginal distributions. Sklar's Theorem [18] states that every distribution function F with margins $F_1...F_m$ can be written as $\forall(x_1..x_m) \in R^m$

$$F(x_1..., x_m) = C(F_1(x_1), ..., F_n(x_m)) \tag{1}$$

The Copula $C$ may be extracted by evaluating

$$C(u_1..., u_m) = P(U_1 \leq u_1, ..., U_m \leq (u_m)) \tag{2}$$

**Empirical Copula**

To avoid introducing any assumptions on the marginal CDF $F_i(x_i)$, we will use the empirical CDF of $F_i(x_i)$, to transform the m samples of X into m samples of U. Empirical Copula is useful for examining the dependence structure of multivariate random vectors. Formally, the empirical Copula is given by the following equation :

$$(c_{ij}) = \frac{1}{m}(\sum_{k=1}^{m} I_{(v_{kj} \leq v_{ij})}) \tag{3}$$

The function $Iarg$ is the indicator function, which equals to 1 if $arg$ is true and 0 otherwise. Here, $m$ is used to keep the empirical $CDF$ less than 1. where $m$ is the number of observations ; $v_{kj}$ is the value of the $k^{th}$ row and $j^{th}$ column ; $v_{ij}$ is the value of the $i^{th}$ row and $j^{th}$ column.

### 2.2 Family of Copulas

Copulas provide a general structure for modeling multivariate distributions.

**Gaussian Copula**

The difference between the Gaussian Copula and the joint normal $CDF$ is that the Gaussian Copula allows having different marginal $CDF$ types from the joint $CDF$ type where as the joint normal $CDF$ does not. The Gaussian Copula is defined as :

$$C(\Phi(x_1)...,\Phi(x_m)) = \frac{1}{\mid \Sigma \mid^{\frac{1}{2}}} \exp(\frac{-1}{2} X^t(\Sigma^{-1} - I)X) \tag{4}$$

where $f_i(x_i)$ is standard Gaussian distribution, i.e. $X_i \sim N(0,1)$, and let $\Sigma$ is the correlation matrix. The resulting Copula $C(u_1,...,u_n)$ is called Gaussian Copula. The density associated with $C(u_1,...,u_n)$ is obtained by using Equation (5). using $u_i = \Phi(x_i)$ we can write

$$C(u_1...,u_m) = \frac{1}{\mid \Sigma \mid^{\frac{1}{2}}} \exp[\frac{-1}{2} \xi^t(\Sigma^{-1} - I)\xi] \tag{5}$$

where $\xi = (\Phi^{-1}(u_1)....\Phi^{-1}(u_m))^T$.

**Student Copula**

The student Copula is extracted from the multivariate Student distribution which is given by the following : $\forall (u_1...,u_m) \in [0,1]^m$

$$C(u_1...,u_m) = \frac{(f_{(v,\Sigma)}(t_v^{(-1)}u_1,...t_v^{(-1)}u_1))}{(\prod_{i=1}^{(n)}(f_{(v)}t_v^{(u_i)}))} \tag{6}$$

where $t_v^{(-1)}$ is the inverse of the t distribution centered and reduced to univariate degrees of freedom.
$f_{(v,\Sigma)}$ is the probability density function of the Student distribution which is centered and reduced.
$\Sigma$ is the correlation matrix and $f_{(v)}$ is the density uni-range of the Student distribution, centered and reduced ($\Sigma$ =1).

**Archimedean Copulas**

For an Archimedean Copula, there exists a generator such that the following relationship holds :

$$C(u_1...,u_n) = \begin{cases} \varphi^{-1}(\varphi(u_1 + .... + u_n) & \text{if } \sum_{\varphi(u_i) \leq \varphi(0)}, \\ 0 & \text{else} \end{cases}$$

$\varphi$ is called the generating function that checks the Copula : $\varphi(1) = 0$, $\varphi(\acute{u}) < 0$ and $\acute{u}\varphi(\acute{u}) > 0$, $0 \leq u < 1$ .
In the table blows (Table 1) we have summarized some examples of Archimedean Copulas.

**Table 1.** Examples of Archimedean Copulas

| Copulas | $\varphi(u)$ | C(u) |
|---|---|---|
| $\prod$ | -ln u | $\prod_{i=1}^{d} u_i$ |
| Gumbel | $(-lnu)^{\theta}, \theta \geq 1$ | $exp\{-[\Sigma_{i=1}^{d}(-lnu)^{\theta}]^{1/\theta}\}$ |
| Frank | $\frac{-lnexp((-\theta u)-1}{exp(-\theta)-1}$ | $-\frac{1}{\theta}ln(1+\frac{\prod_{i=1}^{d} exp((-\theta u_i)-1}{(exp(-\theta)-1)^{d-1}})$ |
| Clayton | $u^{-\theta}-1, \theta > 0$ | $(\Sigma_{i=1}^{d} u^{-\theta} - d + 1)^{\frac{-1}{\theta}}$ |

### 2.3 Random variable generation

One of the primary applications of Copulas is in simulation and Monte Carlo studies. To generate jointly-distributed random variable $X$ from a given joint $CDF$ $F(x)$, the first step is to generate independently uniformly-distributed random $U$ in $[0, 1]$, since the transform $U = F(X)$ results in a random variable U via the inverse $CDFs$ and $X$ is defined as $X = F^{-1}(U)$. So to generate an arbitrary large sample of $X$, we start with a uniformly distributed sample of U that can easily be generated using a standard pseudo-random generator. For each sampled value of $U$, we can calculate a value of $X$ using the inverse $CDF$ given by $X = F^{-1}(U)$.

### 2.4 Dependence

Different measures of dependence are used in order to define the relationship between random variables. For Gaussian Copula, we should estimate the correlation matrix $\Sigma$, while the Student Copula requires estimation of the correlation matrix $\Sigma$ and the degree of freedom dl. Typically, the $\Sigma$ relationship can be measured by the Pearson correlation factor $\rho_{XY} = cov(X, Y)/(\sigma_X \sigma_Y)$, where $cov(X, Y)$ is the covariance of $X$ and $Y$ while $\sigma X \sigma Y$ are the standard deviations of $X$ and $Y$ respectively, or by nonparametric statistics such as rank correlation coefficients $\tau = P[(X_i - X_j)(Y_i - Y_j) > 0] - P[(X_i - X_j)(Y_i - Y_j) < 0]$, where $(X_1, Y_1), .., (X_j, Y_j)$ are random sample of m observations. In the Archimedean Copulas, a single parameter denoted by $\Theta$, must estimated. For the estimation of $\Theta$ we use the moment method which is based on the relationship between the rate of Kendall and the parameters of Copulas. The relation between $\Theta$ and $\tau$ of Kendall is given by :

$$\tau = 1 + 4 \int_0^1 (\varphi(t)/(\dot{\varphi})(t))dt \tag{7}$$

The following table summarizes the relationship between the rate of Kendall and the parameters of Gumbel, Frank, Clayton Copulas .

**Table 2.** the relationship between the rate of Kendall and the parameters of copulas

| | |
|---|---|
| **Copulas** | $\tau$ |
| **Gumbel** | $1 - \Theta^{(-1)}$ |
| **Frank** | $1 - (4[D_1(\Theta) - 1]/\Theta)$, $D1 = \int_0^{\Theta}(t/(e^t - 1))dt$ |
| **Clayton** | $\Theta/(\Theta + 2)$ |

### 2.5 Singular value decomposition (SVD)

SVD [17] is a matrix factorization technique commonly used for producing low-rank approximations. Given an $m \times n$ matrix $A$, with rank $r$, the singular value decomposition, is defined by

$$SVD(A) = U \times S \times V^T \tag{8}$$

Where $U$, $S$ and $V$ are of dimensions $m \times m$, $m \times n$, and $n \times n$, respectively. $S$ is a diagonal matrix having only $r$ nonzero entries, which makes the effective dimensions of these three matrices $m \times r$, $r \times r$, and $r \times n$, respectively. $U$ and $V$ are two orthogonal matrices, and is called the singular matrix. The diagonal entries $(s_1, s_2, ....., s_r)$ of S have the property that $s_i > 0$ and $s_1 \geq s_2 \geq .... \geq s_r$. The first $r$ columns of $U$ and $V$ represent the orthogonal eigenvectors associated with the $r$ nonzero eigenvalues of $AA^T$ and $A^TA$, respectively. In other words, the $r$ columns of $U$, corresponding to the nonzero singular values span the column space, and the $r$ columns of $V$ the left and the right singular vectors, respectively. SVD provides the best low-rank linear approximation of the original matrix $A$. It is possible to retain only $k \ll r$ singular values by discarding other entries of low effect. Since the entries in S are sorted, the reduction process is performed by retaining the first $k$ singular values. The matrices $U$ and $V$ are also reduced to produce matrices $U_k$ and $V_k$, respectively. The matrix $U_k$ is produced by removing $(r - k)$ columns from the matrix $U$ and matrix $V_k$is produced by removing $(r-k)$ rows from the matrix $V$. When we multiply these three reduced matrices, we obtain a matrix $A_k = U_k.S_k.V_k^T$. $A_k$ is of a rank $k$ and it is the closest approximation to the original matrix $A$.

## 3 Proposed approach

### 3.1 Problem modeling

Based on our model, the original problem which is identify and remove redundant dimension from a dataset $X$. To illustrate this problem, we try to find the minimum set of data $r$ ($r \leq s$) uncorrelated linear functions, which allows detecting the maximum of the Redundant Data.

**Notation**

| | |
|---|---|
| X | is the $n * m$ data matrix (random variable) |
| $x_j^i$ | the value of the $i^{th}$ row and the $j^{th}$ column |
| $X^i$ | is the $i^{th}$ row of the matrix X. |
| $X_j$ | is the $j^{th}$ column of the matrix X. |
| $F_j(.)$ | is the distribution of $j^{th}$ column of the matrix X. |
| $f_j(.)$ | is the probability density of $j^{th}$ column of the matrix X. |
| C | is the Copula of the matrix X. |
| $C_j$ | is the distribution of $j^{th}$ column of C (uniform). |
| $\Sigma$ | is the correlation matrix of Gaussian or Student Copula. |
| $\Theta$ | is the parameter of Archimedean Copulas. |

**Decision Variables**

Let the variable $Y_j$ which takes the value 1 if the redundancy of column $k$ is detected by different Measures of dependence of the Copula $C$, and the value 0 otherwise. Let the decision variable $Z^i$ that takes the value 1 if the line $k\prime$ belongs to the maximum set of redundancies and takes the value 0 otherwise. In this model, we represent the decision variable as follows :

$$Y_j = \begin{cases} 1, & \text{if the redundancy of dimension } X_j \text{ is detected} \\ 0, & \text{otherwise} \end{cases}$$

$$Z^i = \begin{cases} 1, & \text{if the redundancy of rows } X^i \text{ is detected} \\ 0, & \text{otherwise} \end{cases}$$

**Objectives functions**

1. Maximize the number of dimensions (columns) redundant that will be eliminated in $X$.

$$Max_y(\sum_{j=1}^{m}(Y_j)) \tag{9}$$

2. Maximize the number of rows redundant that will be eliminated in $X$.

$$Max_z(\sum_{i=1}^{n}(Z^i)) \tag{10}$$

**Constraints**

1. $Y_j \in \{0,1\}^m, \forall j \in \{1..m\}$
2. $Z^i \in \{0,1\}^n, \forall i \in \{1..n\}$
3. verify the independence of bases in vector columns of the matrix X.

$$\sum_{k\epsilon Bc} \alpha_k X^k = 0 \Leftrightarrow \alpha_k = 0, \forall k\epsilon Bc \qquad Bc = \{K \ / y_k = 0\} \tag{11}$$

4. verify the independence of bases in the rows of the matrix $X$.

$$\sum_{k\epsilon Bl} \alpha_k X^k = 0 \Leftrightarrow \alpha_k = 0, \forall k\epsilon Bl \qquad Bl = \{K \ / Z^k = 0\} \tag{12}$$

To illustrate our approach, we give an overview in the following flowchart :

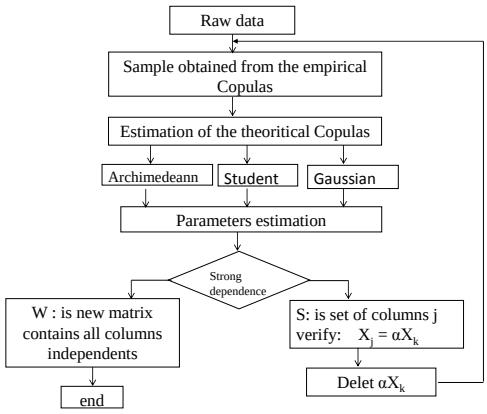



**Figure 1.** *General outline of the proposed approach : reducing the dimension*

First, we calculate the empirical Copula to better observe the dependencies between the variables of these data. According to the marginal distributions from the observed and approved empirical Copula, we can determine the theoretical Copula adjusted by the family of Copulas presented in Section 2 (2), in order to generate the theoretical Copula having the same distribution as the empirical Copula. To calculate the dependence between different variables of data, we'll estimate the parameters of the Copula which be obtained after computing the empirical Copula (presented Section 2 (3)). After having determined the appropriate Copula and having estimated the parameters in the previous steps, we will define a function that compares dependence between a subset of dimensions with the parameters estimated to eliminate attributes redundant and we will result in a reduced matrix with independent variables .

## 4 Results and discussion

To evaluate the effectiveness of our solution, we have developed two large-scale experiment on UCI machine learning repository server.

***Dataset 1 : Waveform Database Generator***

The first dataset was obtained from a study on "Waveform Database Generator" from Wadsworth International Group : Belmont, California in 11/10/1988. The number of Instances was 33367 and the number of attributes was 21. In this Dataset, we noticed that most of the Copula obtained have the form of first right scatter plot shown in Figure 2, For this we choose a bivariate Copula among 21 and we will illustrate all the results from a this Copula.

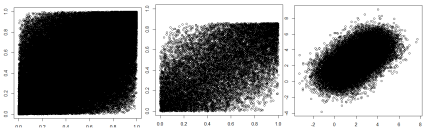


**Figure 2.** Scatter plot of the Dataset 1

Given that the statistical model of the joint distribution is not known, we can calculate the empirical Copula of this sample to see if it follows a known Copula. the seconde figure in Figure 2 shows the estimated Copula. This Copula is obtained by converting each point of the original sample by the cumulative distribution function of each marginal. The resulting Copula has an elliptical form just as a Gaussian Copula . To verify this formally, we used the fit test for Gaussian Copulas .

To generate a sample from this Copula and with the same parameters, we can calculate the inverse marginal $CDFs$ to obtain a sample of large size with the same statistical model as the empirical sample. To calculate the inverse $CDFs$, we observed the marginal distributions of variables $X^5$, $X^6$. These distributions are Gaussian. They were validated by fit test classic such as univariate

Kolmogorov-Smirnov. the last figure in Figure 2 shows the theoretical sample obtained with a million points.

According to the calculation of the parameter of Gaussian Copula, we obtained a correlation matrix of dimension 21. We noticed that after having verify the independence of the subset of all column, we noticed that dimensions $\{X_1, X_2, X_3, X_4, X_{13}, X_{19}, X_{20}, X_{21}\}$ are subset minimal independent and build a vector basis.

$$\sum_{k\in\{1,2,3,4,13,19,20,21\}} \alpha_k X_k = 0. \tag{13}$$

the coefficient of the system equal to zero :

$$\alpha_k = 0, \forall k \in \{1, 2, 3, 4, 13, 19, 20, 21\} \tag{14}$$

We obtained The new matrix of 57% reduction of dimensions $X_j$ that verify the constraint of independence of dimensions.

After computing the empirical Copula of the new matrix obtained, we have shown that we had a Gaussian Copula. The parameter estimate has the correlation matrix $\Sigma$, which all correlation coefficient are independent.

**Singular Value Decomposition** In this part, we use the method of singular value decomposition (SVD) for reduce the number of rows of data. The matrix obtained After decomposition as follows : U is a singular matrix $(33367 \times 33367)$, $S_k$ is a diagonal matrix $(33367 \times 21)$, V is a singular matrix $(21 \times 21)$. The Figure 3 shows the matrix diagonale after reduction.

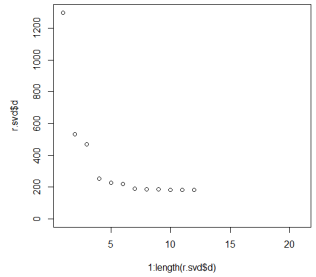



**Figure 3.** Matrix diagonale after reduction

***Dataset 2 : Pima Indians Diabetes Database***
The second dataset was from obtained from a study "Pima Indians Diabetes

Database" of the National Institute of Diabetes and Digestive and Kidney Diseases, in May 1990. The number of Instances was 768 otherwise the number of attributes was 8 for the Yeast dataset.

We will illustrate all the results from a bivariate Copula .Based Sample is shown in the scatter plot Figure 4.

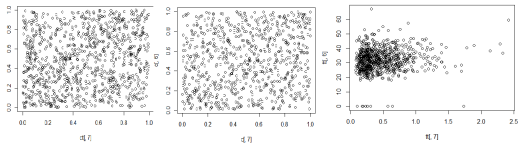


**Figure 4.** Scatter plot of Dataset 2

After calculate the inverse $CDFs$ of the data, we observed the marginal distributions of variables follows gamma distribution and, we noticed that the resulting of the empirical Copula has a Gaussian Copula.

For this reason, we have generate a theoretical Copula with the same parameters of empirical Copula in the seconde figure of Figure 4. According to the calculation of the parameter of Gaussian Copula, we obtained a correlation matrix of dimension 8. We noticed that all attributs are independent In this case we haven't a redundant values.

**Singular Values Decomposition**

We noticed with Svd that the matrix $S$ is a diagonal matrix haven't values negative or near to zero,The diagonal entries $(s_1, s_2, ....., s_8)$ are independent.

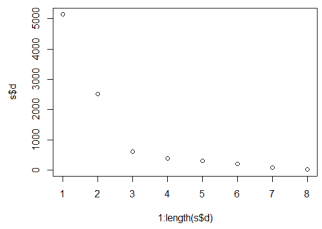



**Figure 5.** Matrix diagonale with SVD

In this case we haven't a redundant values.

The SVD is an optimal technique related types of dimensionality reduction approaches but our approach is more good and should be a practical solution.

## 5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a new method for dimensionality reduction for data preprocessing of high-dimensional data. This approach is based on Sampling techniques and Copulas.

We first reformulated the problem of reduction of data into a constrained optimization problem. The approach proposed is then designed to solve a challenging issue of redundant data for the pre-treatment process of KDD.

An experimental study on a large scale has provided very good results, those that show the effectiveness of our method and the comparison with one of the most efficient method of Datamining (SVD).

## Références


1. A.Estoup and all, Estimation of demo-genetic model probabilities with approximate bayesian computation using linear discriminant analysis on summary statistics, Molecular Ecology Resources in press, 2012.
2. Barnes ans all, Considerate approaches to achieving sufficiency for ABC model selection, Statistics and Computing, in press, 2012.
3. Blum, Choosing the summary statistics and the acceptance rate in approximate Bayesian computation. In G. Saporta and Y. Lechevallier (Eds.), COMPSTAT 2010, Proceedings in Computational Statistics, pp. $47_5$6. Springer, Physica Verlag, 2010.
4. Abdi H. and L. J. Williams, Partial least square regression, projection on latent structure regression, Wiley Interdisciplinary Reviews : Computational Statistics 2, $433_4$59, 2010.
5. Blum and O. Francois, Non-linear regression models for approximate Bayesian computation, Statistics and Computing 20, 6373, 2010.
6. Del Moral and all, An adaptive sequential Monte Carlo method for approximate Bayesian computation, Statistics and Computing 22, $1009_1$020, 2012.
7. L. You, and M. West, Bayesian learning from marginal data in bionetwork models. Statistical Applications in Genetics and Molecular Biology 10(1), 49, 2011.
8. Fearnhead and D. Prangle, Constructing summary statistics for approximate Bayesian computation : Semi-automatic ABC (with discussion, Journal of the Royal Statistical Society : Series B 74, $419_4$74, 2012.
9. Filippi and all, Contribution to the discussion of Fearnhead and Prangle, Constructing summary statistics for approx imate Bayesian computation : Semi-automatic approximate Bayesian computation,Journal of the Royal Statistical Society : Series B 74, $459_4$60, 2012.
10. Jasra and all, Filtering via approximate Bayesian computation. Statistics and Computing in press,2012.
11. Jin and all, A Robust High-Dimensional Data Reduction Method, The International Journal Of Virtual Reality 9(1), pp.55 – 60,2010.
12. Kush R and all, Linear Dimensionality Reduction for Margin Based Classification : HighDimensional Data and Sensor Networks, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL 59, NO 6, JUNE 2011

13. L. Juan and O. Gwun, A Comparison of SIFT , PCA-SIFT and SURF,International Journal of Image Processing, vol. 3, no. 4, pp. 143 − 152, 2009.

14. M. Babu Reddyand, Dimensionality Reduction : An Empirical Study on the Usability of IFE-CF Measures, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue1, No. 1, January 2010, india,2012.

15. M.Esmaeili, A.Mosavi, Variable Reduction for Multi- Objective Optimization Using Data Mining Techniques, Application to Aerospace Structures, IEEE Proc Of international conf, on On Mechanical and Aerospace Engineering, 2010.

16. Makoto Yamada and all, Computationally Efficient Sufficient Dimension Reduction via Squared-Loss Mutual Information, JMLR : Workshop and Conference Proceedings $247_262$ Asian Conference on Machine Learning, 2011.

17. Rama Devi Y and all, Fuzzy Rough Data Reduction Using SVD, International Journal of Computer and Electrical Engineering, Vol. 3, No. 3, June 2011.

18. R.B.Nielsen, an introduction to copulas, second edition, springer, 2005.

19. Sembiring and all, Alternative Model for Extracting Multidimensional Data Based On Comparative Dimension Reduction, ICSECS (2), pp. 28 − 42, 2011.

20. Sembiring and all, Clustering High Dimensional Data Using Subspace And Projected Clustering Algorithm, International Journal Of Computer Science and Information Technology (IJCSIT) Vol.2,No.4, pp.162 − 170, 2010.

21. Sedki and P. Pudlo, Contribution to the discussion of Fearnhead and Prangle, Constructing summary statistics for approximate Bayesian computation : Semi-automatic approximate Bayesian computation, Journal of the Royal Statistical Society : Series B 74, $466_467$, 2012.

22. Reuven Y. Rubinstein, Dirk P. Kroe, Simulation and the Monte Carlo method, edition willy, Canda, 2007

# Personalized Documents Ranking With Social Contextualization

Mohamed Reda Bouadjenek[1⋆], Hakim Hacid[2⋆⋆], and Mokrane Bouzeghoub[1]

[1] PRiSM Laboratory, Versailles University
{reda.bouadjenek, mokrane.bouzeghoub}@prism.uvsq.fr
[2] SideTrade, 114 Rue Gallieni, 92100 Boulogne-Billancourt, France
hhacid@sidetrade.com

**Abstract.** We present in this paper a contribution to IR modeling by proposing a new ranking function for documents while considering the social dimension of the Web. This social dimension is any social information that surrounds documents along with the social context of users. Currently, our approach relies on folksonomies for extracting these social contexts, but it can be extended to use any social meta-data, e.g. comments, ratings, tweets, etc. The evaluation performed on our approach shows its benefits for personalized search with respect to the closest state of the art methods.

## 1 Introduction

Nowadays, the Web is becoming more and more complex with the socialization and interaction between individuals and objects. This evolution is known as social Web, which includes linking people through the World Wide Web. This is mainly done through platforms such as *Facebook*, *Twitter*, or *YouTube*, where users can comment, spread, share and tag information and resources. The social Web leads to facilitate the implication of users in the enrichment of the social context of web pages. Especially, it allows users to freely tag web pages with annotations. These annotations can be easily used to get an intuition about the content of web pages to which they are related. Hence, several research works ([21,6,8,4]) reported that adding tags to the content of a document enhances the search quality, as they are good summaries for documents. In particular, tags are useful for documents that contain few terms where a simple indexing strategy is not expected to provide a good retrieval performances (e.g. the *Google homepage*[3]).

⋆ This work has been mainly done when the author was a PhD student at Bell Labs France, Centre de Villarceaux.

⋆⋆ This work has been mainly done when the author was a research scientist at Bell Labs France, Centre de Villarceaux.

[3] http://www.google.com/ There are only few terms on the page itself but a thousands of annotations available on *delicious* are associated to it. Eventually, the social information of the *Google* homepage is more useful for indexing.

In such a context, classic model of Information Retrieval (IR) should be adapted by considering (i) the social context that surrounds web pages and resources, e.g. their annotations, their associated comments, their ratings, etc. and (ii) the social context of users, e.g. their used tags, their comments, their trustworthiness, etc. Exploiting social information has a number of advantages (for IR in particular). First, feedback information in social networks is provided directly by the user. Hence, accurate information about the user interest can be learned because people actively express their opinions on social platforms. Second, exploring published information doesn't violate user privacy, since the primary goal for most of people is to share information. Finally, social resources are often publicly accessible, as most of social networks provide APIs to access their data (even if often, a contract must be established before any use).

In this paper, we are interested in improving the IR model. Especially, we propose a new ranking function for ranking documents while considering the social context of the Web. The approach we are proposing relies on social annotations as a source of social information, which are associated to documents in bookmarking systems.

### 1.1 Background

Social bookmarking websites are based on the techniques of *social tagging or collaborative tagging*. The principle behind social bookmarking platforms is to provide the user with a means to annotate resources on the Web, e.g. URIs in *delicious*, videos in *youtube*, images in *flickr*, or academic papers in *CiteULike*. These annotations (also called tags) can be shared with others. This unstructured (or better, free structured) approach to classification with users assigning their own labels is often referred to as a *folksonomy [9]*. A folksonomy is based on the notion of bookmark, which is formally defined as follow:

**Definition 1.** *Let $U, T, R$ be respectively the set of Users, Tags and Resources. A bookmark is a triplet (u,t,r) such as $u \in U, t \in T, r \in R$, which represents the fact that the user $u$ has annotated the resource $r$ with the tag $t$.*

Then, a folksonomy is formally defined as follow:

**Definition 2.** *Let $U, T, R$ be respectively the set of Users, Tags and Resources. A folksonomy $\mathbb{F}(U, T, R)$ is a subset of the Cartesian product $U \times T \times R$ such that each triple $(u, t, r) \in \mathbb{F}$ is a bookmark.*

A folksonomy can be represented by a tripartite-graph where each ternary edge represents a bookmark. In particular, the graph representation of the folksonomy $\mathbb{F}$ is defined as a tripartite graph $\mathcal{G}(V, E)$ where $V = U \cup T \cup R$ and $E = \{(u, t, r) | (u, t, r) \in \mathbb{F}\}$. Figure 1 shows example of a folksonomy with seven bookmarks.

### 1.2 Problem definition

The problem we are addressing can be formalized as follows: Let consider a folksonomy $\mathbb{F}(U, T, R)$ whose a user $u \in U$ submits a query $q$ to a search engine.

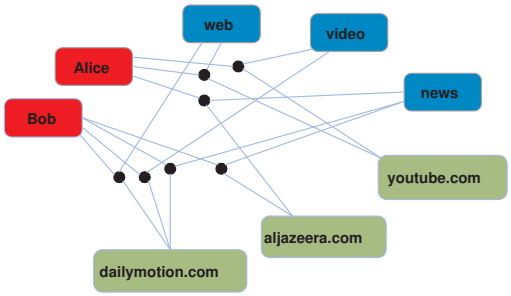



Fig. 1: Example of a folksonomy.

We would like to re-rank the set of documents $R_q \subseteq R$ (or resources) that match $q$, such that relevant documents for $u$ are highlighted and pushed to the top for maximizing his satisfaction and personalizing the search results. The ranking follows an ordering $\tau = [r_1 \geq r_2 \geq \cdots \geq r_k]$ in which $r_k \in R$ and the ordering relation is defined by $r_i \geq r_j \Leftrightarrow Rank(r_i, q, u) \geq Rank(r_j, q, u)$, where $Rank(r, q, u)$ is a ranking function that quantify similarity between the query and the resource w.r.t the user [16].

### 1.3 Contributions and paper organization

In this context of social Web, we propose the following contributions: (1) A ranking function that leverages the social context of the Web. (2) Two methods for weighing user profiles and the social representations of documents. (3) An intensive evaluation of our approach and a comparison with the closest works on a large public dataset.

The rest of this paper is organized as follows: in Section 2 we present the related works and we position our method consequently. Section 3 introduces our approach for ranking documents. The different experiments are discussed in Section 4. Finally, we conclude and provide some future directions in Section 5.

## 2 Related Work

We distinguished two categories for social results re-ranking that differ in the way they use social information. The first category uses social information by adding a social relevance to documents while the second use it for personalization.

### 2.1 Re-ranking using social relevance

Several approaches have been proposed to improve document re-ranking using social relevance. Social relevance refers to information socially created that characterizes a document from a point of view of its interest, i.e. its general interest, its popularity, etc. Two formal models for folksonomies and ranking algorithm

called *folkRank and Social PageRank* are defined in [10] and [1] respectively. Both are an extension of the well-known *PageRank* algorithm adapted for the generation of rankings of entities within folksonomies. In the same spirit, Takahashi et al. [15] propose *S-BIT* and *FS-BIT*, an extension of the well-known HITS [11] approach. Finally, Yanbe et al. [20] proposed *SBRank*, which indicates how many users bookmarked a page, and use the estimation of *SBRank* as an indicator of web search.

### 2.2 Personalized re-ranking

In general, users have different interests, different profiles, and different habits. Hence, in an IR system, providing the same documents sorted in the same way is not really suitable since relevance judgment is user-dependent [14]. Therefore, a personalized function to sort documents differently according to the each user is expected to improve search results.

Several approaches have been proposed to personalize ranking of search results using social information [7,16,17,19]. Almost all these approaches are in the context of folksonomy and follow the same idea that the ranking score of a document $d$ retrieved when a user $u$ submits a query $q$ is driven by: (i) a term matching process, which calculates the similarity between $q$ and the textual content of $d$ to generate a user unrelated ranking score; and (ii) an interest matching process, which calculates the similarity between $u$ and $d$ to generate a user related ranking score. Then a merge operation is performed to generate a final ranking score based on the two previous ranking score.

The approach we are proposing is part of this initiative. However, we enhance the ranking process by considering a new aspect, which is *the social matching score*. It measures the similarity between the query and the social representation of documents. Details of our ranking function are given in the next section.

## 3 A ranking function for personalized search

In this Section, we first define our ranking function, then we present the methods used for modeling and weighting the social representation of documents and user profiles.

### 3.1 Ranking for personalized search

On the one hand, we believe that a matching score between a document $d$ and a query $q$ should be based on (i) a textual matching score, and (ii) a social matching score. The textual matching score expresses the similarity between the textual content of $d$ and $q$. The social matching score expresses how similar the social representation of $d$ is, for $q$. This social representation is based on the annotations associated to $d$ modeled and weighted as described in Section 3.2. More formally, in this paper, we consider this two ranking scores as an independent evidence,

and we propose to merge them using the *Weighted Borda Fuse*. This merge is summarized in Equation 1:

$$Score(q,d) = \beta \times Sim(\overrightarrow{q}, \overrightarrow{d}) + (1-\beta) \times Sim(\overrightarrow{q}, \overrightarrow{S_d}) \quad (1)$$

where $\beta$ is a weight that satisfies $0 \leq \beta \leq 1$, $Sim(\overrightarrow{q}, \overrightarrow{d})$ denotes the textual matching score between $d$ and $q$ (computed using the *Apache Lucene*[4] search engine in our implementation), $\overrightarrow{S_d}$ is the vector that models the social representation of the document $d$, and $Sim(\overrightarrow{q}, \overrightarrow{S_d})$ denotes the social matching score between $d$ and $q$. Inspired by the Vectorial Space Model (VSM), we compute this similarity using the cosine measure as follows:

$$Sim(\overrightarrow{q}, \overrightarrow{S_d}) = \frac{\overrightarrow{q} \bullet \overrightarrow{S_d}}{|\overrightarrow{q}| \times |\overrightarrow{S_d}|} \quad (2)$$

On the other hand, in the non-personalized search engines, the relevance between a query and a document is assumed to be only based on the textual content of the document. However, as relevance is actually relative for each user [14], considering only a matching between a query and documents is not enough to generate satisfactory search results. Thus, we propose to estimate the interest of a user $u$ to a document $d$ by computing a similarity between the profile of $u$ and the social representation of $d$. Then, we propose to merge this interest value to the previous ranking score computed in Equation 1 for computing the matching score of a document to a query with respect to a user. Formally, the ranking score of a document $d$ that potentially match the query $q$ issued by a user $u$ is computed as follows:

$$Rank(d,q,u) = \gamma \times Sim(\overrightarrow{p_u}, \overrightarrow{S_d}) + (1-\gamma) \times \left[\beta \times Sim(\overrightarrow{q}, \overrightarrow{d}) + (1-\beta) \times Sim(\overrightarrow{q}, \overrightarrow{S_d})\right] \quad (3)$$

where, $\gamma$ is the weight that satisfies $0 \leq \gamma \leq 1$, and $Sim(\overrightarrow{p_u}, \overrightarrow{S_d})$ is the similarity between the profile of $u$ and the social representation of $d$. This similarity quantifies the interest of $u$ to $d$ and is computed using the cosine measure as follows:

$$Sim(\overrightarrow{p_u}, \overrightarrow{S_d}) = \frac{\overrightarrow{p_u} \bullet \overrightarrow{S_d}}{|\overrightarrow{p_u}| \times |\overrightarrow{S_d}|} \quad (4)$$

At the end of this process, we obtain a list of re-ranked documents according to: (i) a textual content matching score of documents and the query, (ii) a social matching score of documents and the query, and (iii) the social interest score of the user to documents. Finally, the top ranked documents are formatted for presentation to the user.

In the next two subsections, we present two methods to weight and estimate the social document representation and the user interest vectors.

[4] http://lucene.apache.org/

### 3.2 Social document modeling

In this paper, the social representations of documents are estimated by their social annotations and modeled as in the VSM. Hence, if we consider web pages as documents and annotations as terms, the above setting is right for the VSM. Even if the VSM has been developed a long time ago, it has shown its effectiveness for IR and remains very competitive and challenging. One of the key points in the VSM is the weighting of terms. Hence, we first propose to simply weight annotations using the *tf-idf* measure as follows:

$$w_t = tf_t \times log(\frac{\mid R \mid}{\mid R_t \mid}) \tag{5}$$

where $tf_t$ denotes the tag frequency, $\mid R \mid$ denotes the total number of web pages in the whole collection and $\mid R_t \mid$ denotes the number of web page tagged with $t$.

Beside this, the BM25 weighting scheme is a more sophisticated alternative, which represents state-of-the-art weighting functions used in IR. It is computed as follows:

$$w_t = log(\frac{\mid R \mid - \mid R_t \mid + 0.5}{\mid R_t \mid + 0.5}) \times \frac{tf_t \times (k_1 + 1)}{tf_t + k_1 \times (1 - b + b \times \frac{dl}{avgdl})} \tag{6}$$

where $k_1$ and $b$ are free parameters set to 2 and 0.75 respectively, $dl$ denotes number of annotations associated to the web page and $avgdl$ denotes the average number of annotations associated to web pages the collection.

### 3.3 User modeling

Folksonomies have proven to be a valuable knowledge for user profiling [7,13,16,19]. Personalization allows discriminating between individuals by emphasizing on their specific domains of interest and their preferences. Several techniques exist to provide personalized services among which the user profiling. The user profile is a collection of personal information associated to a specific user that enables to capture his interests. In this paper and in the context of folksonomies, we define a user profile as follow:

**Definition 3.** *Let $U$, $T$, $R$ be respectively the set of Users, Tags and Resources of a folksonomy $\mathbb{F}(U, T, R)$. A profile assigned to a user $u \in U$, is modeled as a weighted vector $\overrightarrow{p_u}$ of $m$ dimensions, where each dimension represents a tag the user employed in his tagging actions. More formally, $\overrightarrow{p_u} = \{w_{t_1}, w_{t_2}, ..., w_{t_m}\}$ such that $t_m \in T \wedge (\exists r \in R \mid (u, t_m, r) \in \mathbb{F})$, and $w_{t_m}$ is the weight of $t_m$.*

At this point, the main challenge is *how to define the weight of each dimension in the user profile?* Hence, we fist propose to use an adaptation of the well-known *tf-idf* measure to estimate this weight. Formally, we define the weight $w_{t_i}$ of

the term $t_i$ in a user profile as the *user term frequency, inverse user frequency (utf-iuf)*, which is computed as follows:

$$w_t = utf_t \times log\left(\frac{\mid U \mid}{\mid U_t \mid}\right) \tag{7}$$

where $utf_u$ is the user term frequency, i.e. the number of time the user $u$ used the tag $t$, $\mid U \mid$ is the total number of users in the folksonomy, and $\mid U_t \mid$ is the number of users who have used the term $t_i$.
Similarly, we can adapt the BM25 weighting scheme to weight the user profiles. It is computed as follows:

$$w_t = log(\frac{\mid U \mid - \mid U_t \mid +0.5}{\mid U_t \mid +0.5}) \times \frac{utf_t \times (k_1 + 1)}{utf_t + k_1 \times (1—b + b \times \frac{dl_u}{avgdl_u})} \tag{8}$$

where $k_1$ and $b$ are free parameters set to 2 and 0.75 respectively, $dl_u$ denotes number of annotations used by $u$ and $avgdl_u$ denotes the average number of annotations used by users in the collection.

In summary, our ranking function for ranking documents that match a query with respect to a user takes into account: (i) the textual content of documents, (ii) their social context, and (iii) the social context of the user by defining a profile and estimating his interest. The social representations of documents and the user profiles are modeled as vectors, and we proposed two methods for weighting these vectors based on state of the art weighting schemes, i.e. *tf-idf* and *BM25*.

## 4 Evaluation

In this section, we describe the dataset we used, the evaluation methodology and the evaluations we have performed.

### 4.1 Dataset

We have selected a *delicious* dataset to perform an off-line evaluation, which is public, described and analyzed in [18][5]. Before the experiments, we performed five data preprocessing tasks: (1) We remove manually several annotations that are too personal or meaningless, e.g. “toread”, “Imported IE Fa-vorites”, “system:imported”, etc. (2) Although the annotations from delicious are easy for users to read and understand, they are not designed for machine use. For example, some users may concatenate several words to form an annotation such as “java.programming” or “java/programming”. We tokenize this kind of annotations before using them in the experiments. (3) The list of terms undergoes a stemming by means of the Porter’s algorithm in such a way to eliminate the differences between terms having the same root. (4) We downloaded all the available

[5] http://data.dai-labor.de/corpus/delicious/

web pages while removing those which are no longer available using the *cURL* command line tool. (5) Finally, we removed all the non-english web pages using *Apache Tika* toolkit. Table 1 gives a description of the resulted dataset after our cleansing:

Table 1: Details of the delicious dataset

| Bookmarks | Users | Tags | Web pages | Unique terms |
|---|---|---|---|---|
| 9 675 294 | 318 769 | 425 183 | 1 321 039 | 12 015 123 |

The resulted dataset still has the same properties, i.e. it is very sparse and follows a long tail distribution [18].

### 4.2 Evaluation methodology

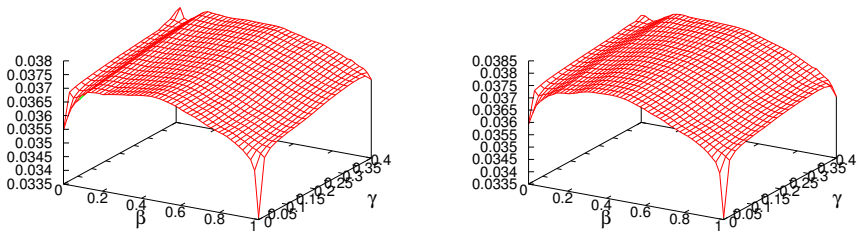
Making evaluations for personalized search is a challenge since relevance judgments can only be assessed by end-users themselves [7]. This is difficult to achieve at a large scale. However, different efforts [12,3,4] state that the tagging behavior of a user of a folksonomy closely reflects his behavior of search on the Web. In other words, if a user tags a document $d$ with a tag $t$, he will choose to access the document $d$ if it appears in the result obtained by submitting $t$ as query to the search engine. Thus, we can easily state that any bookmark $(u, t, r)$ that represents a user $u$ who bookmarked a resource $r$ with tag $t$, can be used as a test query for evaluations. The main idea of these experiments is based on the following assumption:

**Assumption 1** *For a personalized query $q = \{t\}$ issued by user $u$ with query term $t$, the relevant documents are those tagged by $u$ with $t$.*

Hence, for each evaluation, we randomly select 2000 pairs $(u, t)$, which are considered to form a personalized query set. For each corresponding pair $(u, t)$, we remove all the bookmarks $(u, t, r) \in \mathbb{F}, \forall r \in R$ in order to not promote the resource $r$ (or document) in the results obtained by submitting $t$ as a query in our algorithm and the considered baselines. By removing these bookmarks, the results should not be biased in favor of documents that simply tend to return tagged documents and making comparisons to the baseline uninformative. For each pair, the user $u$ sends the query $q = \{t\}$ to the system. Then, we retrieve and rank all the documents that match this query using our approach or a specific baseline, where documents are indexed based on their textual content using the *Apache Lucene*. Finally, according to the previous assumption, we compute the Mean Average Precision (MAP) and the Mean Reciprocal Rank (MRR) over the 2000 queries. The random selection was carried out 10 times independently, and we report the average results.

### 4.3 Evaluation of the parameters

In this Section, we propose a parameter estimation that aims to provide insights regarding the different values of the parameters used in our approach as well as their potential impact on the system. Our approach has two parameters that can be tuned ($\gamma$ and $\beta$) and two weighting models. Note that each time, we use either the *tf-idf* weighting model for weighting both the social representations of documents and the user profiles or the *BM25* weighting model, i.e. we do not merge the two weighting models. Figure 2 shows the MAP obtained for different values of $\gamma$ and $\beta$ and our two weighting models. We vary $\gamma$ from 0 to 0.4 to better show the impact of $\beta$, i.e. for high values of $\gamma$, $\beta$ has a very low impact according to our ranking function of Equation 3.




(a) tf-idf based ranking

(b) BM25 based ranking

Fig. 2: MAP for different values of $\beta$ and $\gamma$ using the different weighting models.

First, according to Figure 2, the optimal performance is achieved for $\beta \in [0.2, 0.6]$ for the different values of $\gamma$. This shows that both the textual matching score part and the social matching score part are important and are complementary. Second, Figure 2 shows that the behavior of our ranking function seems to be the same for our two ranking models while varying $\gamma$ and $\beta$. Finally, even if the *BM25* weighting model improves better the performance than the *tf-idf* weighting model, our ranking function still doesn't depends on the weighing model.

In the next section, we present the results of the comparison of our approach with several state of the art approaches.

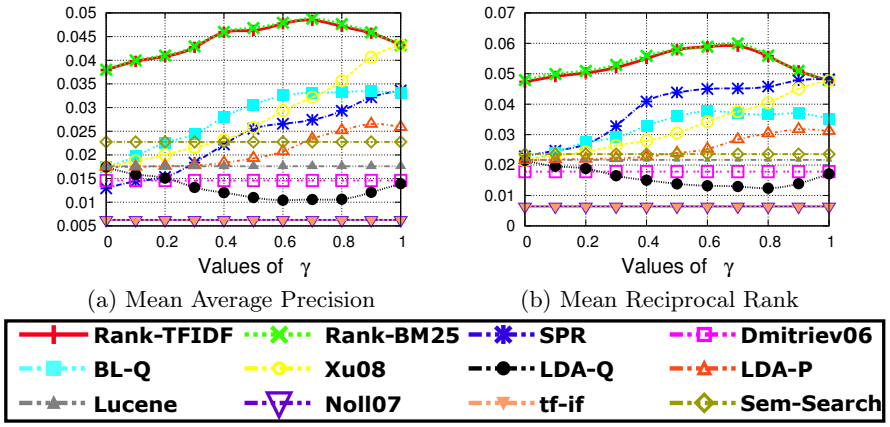
### 4.4 Comparison with baselines

We compare our approach to several baselines, in which the social enhancement score is merged with the textual based matching score using the *Weighted Borda Fuse (WBF)* with a $\gamma$ parameter. The baselines are summarized and described in Table 2.

Table 2: Summary of the baselines.

| | | Baseline | Description |
|---|---|---|---|
| Non-personalized approaches | **1** | **SPR [1]** | SocialPageRank (SPR), which captures the popularity (quality) of web pages using folksonomies. |
| | **2** | **Dmitriev06 [8]** | Combine the annotations with the content of documents to produce a new index. |
| | **3** | ***B*L-Q** | This approach use a query based ranking function where a similarity between a document and a query is computed by merging the textual based matching score and a social based matching score only. The social representation of each document is based on all its annotations weighted using the *tf-idf* measure. |
| | **4** | **Lucene** | This approach represents the Lucene naive score. |
| | **5** | **LDA-Q** | Using LDA [5], we model queries and documents. Then, for each document that match a query, we compute a similarity between its topic and the topic of the query using the cosine measure. The obtained value is then merged with the textual ranking score. |
| Personalized approaches | **6** | **Xu08 [19]** | This approach use a profile based ranking function, where documents are weighted using the *tf-idf*. |
| | **7** | **Noll07 [13]** | The approach considers only a user interest matching between a user and a document. It does not make use of the user and document length normalization factors, and only uses the user tag frequency values. The authors normalize all document tag frequencies to 1, since they want to give more importance to the user profile. |
| | **8** | **tf-if [16]** | This approach is an adaptation of **[13].** The main difference is that tf-if incorporate both the user and document tag distribution global importance factors, following the VSM principle. |
| | **9** | **Semantic Search [2]** | This approach ranks documents by considering users that hold similar content to the query, i.e., users who used at least one of the query terms in describing their content. |
| | **10** | **LDA-P** | Using LDA, we model users and documents. Then, for each document that match a query, we compute a similarity between its topic and the topic of the user profile using the cosine measure. The obtained value is then merged with the textual ranking score. |

The obtained results are illustrated in Figure 3, while varying $\gamma$. The results show that our approach is much more efficient than all the baselines for our two weighting models and for all the values of $\gamma$. Especially, our approach significantly outperform the Xu08 and LDA-P approaches, which we consider as the closest works to our. Hence, we conclude that the personalization efforts introduced by our ranking function bring a considerable improvement to the search quality. We also notice that most of the approach decease their performance for high values




(a) Mean Average Precision (b) Mean Reciprocal Rank

Fig. 3: Comparison with the baseline while varying $\gamma$ and using the optimal values of the parameters.

of $\gamma$. This is certainly due to the fact that they are not designed for personalized search, since these approaches fail in discriminating between users in spite of their preferences.

Finally, we note that the better performances are obtained for $\gamma \in [0.6, 0.8]$, a compromise between the user interest matching score and the query affinity matching score. Although its simplicity, our ranking function is very efficient compared to other state of the art approaches. However, these results should be reinforced using an on-line evaluation to give a better overview of the performance, which is an ongoing work.

## 5 Conclusion and future work

This paper discusses a contribution to the area of IR modeling while leveraging the social dimension of the web. We proposed a new documents ranking function, which uses social information to enhance and improve web search. The experiments performed show the benefit of our approach while comparing it to the closest works. This method can be improved in different way. First, the temporal dimension of social users' behavior has not been deeply investigated yet in the literature. Considering this dimension is a part of our future work, e.g. considering the evolution of the taste of users in the ranking function. Second, considering a social relevance score factor, which characterizes documents from a point of view of interest, is a possible improvement of our ranking function, e.g. their popularities. Finally, performing an on-line user evaluation in order to validate our results is also an ongoing work.

## References


1. S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW*, 2007.
2. M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, R. Schenkel, and G. Weikum. Exploiting social relations for query expansion and result ranking. In *ICDE Workshops*, 2008.
3. D. Benz, A. Hotho, R. Jäschke, B. Krause, and G. Stumme. Query logs as folksonomies. *Datenbank-Spektrum*, 10:15–24, 2010.
4. K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *CIKM*, 2008.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
6. D. Carmel, H. Roitman, and E. Yom-Tov. Social bookmark weighting for search and recommendation. *The VLDB Journal*, 2010.
7. D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. Personalized social search based on the user's social network. In *CIKM*, 2009.
8. P. A. Dmitriev, N. Eiron, M. Fontoura, and E. Shekita. Using annotations in enterprise search. In *WWW*, 2006.
9. T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools : A general review. *D-Lib Magazine*, 11(4), April 2005.
10. A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, 2006.
11. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
12. B. Krause, A. Hotho, and G. Stumme. A comparison of social bookmarking with traditional search. In *ECIR*, 2008.
13. M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *ISWC'07/ASWC'07*, 2007.
14. J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel. Personalized search. *Commun. ACM*, 2002.
15. T. Takahashi and H. Kitagawa. A ranking method for web search using social bookmarks. In *DASFAA*, 2009.
16. D. Vallet, I. Cantador, and J. M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *ECIR*, 2010.
17. Q. Wang and H. Jin. Exploring online social activities for adaptive search personalization. In *CIKM*, 2010.
18. R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *ECAI*, 2008.
19. S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *SIGIR*, 2008.
20. Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Towards improving web search by utilizing social bookmarks. In *ICWE*, 2007.
21. X. Zhang, L. Yang, X. Wu, H. Guo, Z. Guo, S. Bao, Y. Yu, and Z. Su. sdoc: exploring social wisdom for document enhancement in web mining. In *CIKM*, 2009.

# Etude expérimentale d'une approche à base d'automate cellulaire pour la détection de spam

Baya Naouel Barigou, Fatiha Barigou, Baghdad Atmani

Equipe de recherche « Simulation, intégration et fouille de données
Laboratoire d'informatique d'Oran (LIO)
Université d'Oran
{Barigounaouel,fatbarigou,atmani.baghdad}@gmail.com

**Abstract.** Le courrier électronique rend service aux usagers, c'est un moyen rapide et économique pour échanger des informations. Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de messages indésirables appelés aussi spam. Le volume croissant de ce type de courriel a engendré un besoin de filtres anti-spam fiables. L'utilisation des techniques d'apprentissage supervisé pour filtrer automatiquement ces courriels indésirables a attiré l'attention de nombreux chercheurs. Dans ce contexte, nous étudions une classification supervisée à base d'automate cellulaire pour le filtrage de spam. Dans un premier temps, et pour évaluer cette nouvelle méthode, nous menons une série d'expériences sur le corpus LingSpam. Et dans un deuxième temps, nous comparons nos meilleurs résultats avec d'autres algorithmes implémentés dans la plateforme Weka.

Keywords: spam, automate cellulaire, apprentissage supervisé, Weka.

## 1 Introduction

Le courrier électronique (ou courriel) est sans doute la technique qui a changé nos habitudes à une grande échelle. C'est un moyen rapide et économique pour échanger des informations. Si nous comparons le courrier électronique aux autres moyens de communication nous nous apercevons que les avantages des courriels surpassent ses inconvénients. Sa force réside dans le médium du transport des messages, la rapidité avec laquelle circulent les courriels, et à la possibilité de les envoyer à plusieurs personnes en même temps. La nature informatique de ces courriels offre des avantages incomparables, dont l'envoi des documents électroniques par attachement, l'archivage des messages est beaucoup plus facile à effectuer qu'avec les communications écrites ou par téléphone, ainsi que, le courriel permet d'effectuer un traitement rapide, efficace et automatique sur les messages comme la recherche par mots clés, le tri automatique par sujet. Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de messages indésirables ou non sollicités appelés aussi spam. En effet, le spam est rapidement devenu un problème majeur sur Internet. L'agence européenne ENISA (Agence Européenne de la Sécurité des Réseaux et de l'Information) vient de sortir une nouvelle étude selon laquelle 95,6% des messages électroniques seraient identifiés comme étant des spam par les chaines de filtrages des

fournisseurs d'adresses email[1]. Ainsi le volume croissant de ce type de courriel a engendré un besoin de filtres anti-spam fiables. L'utilisation des techniques d'apprentissage supervisé pour filtrer automatiquement ces courriels indésirables a attiré l'attention de nombreux chercheurs. Dans ce contexte, nous étudions une classification supervisée à base d'automate cellulaire pour le filtrage de spam. Nous analysons expérimentalement une nouvelle approche de détection de spam que nous avons proposée dans [4]. Cette approche se base sur l'induction symbolique par automate cellulaire [3].

Le reste du papier est organisé de la manière suivante : la section 2 dresse un état de l'art sur les différents travaux ayant utilisé l'apprentissage supervisé pour le filtrage de spam. Nous étudions les techniques adoptées dans ces différents travaux et analysons les points forts et les points faibles de chacune d'elles. La section 3 est consacrée à l'étude de l'approche proposée pour la détection de courriels indésirables. La section 4 présente les différentes expérimentations réalisées ainsi que les résultats obtenus avec les différentes configurations pour finalement conclure et présenter quelques orientations futures en section 5.

## 2 Travaux connexes

Le spam est un message électronique non sollicité, envoyé massivement à un grand nombre de destinataires, à des fins publicitaires ou malveillantes. Le terme spam est aussi utilisé pour désigner le même type de message transmis par d'autres moyens de communication électroniques tels que les messageries instantanées, les blogs, les forums, et plus récemment, des réseaux de téléphonie mobile, *via* les SMS ou MMS. Même si le moyen de communication est différent, les techniques d'envoi et de détection restent relativement similaires.

Au départ, le spam visait principalement des objectifs publicitaires. Aujourd'hui, il s'est considérablement développé, diversifié et complexifié, pour atteindre de plus en plus souvent des objectifs malveillants. En effet, Le spam s'est non seulement développé en termes de volume, mais également en termes de contenu. Aujourd'hui, les objectifs des spam sont très variés, il peut s'agir de publicité, d'hameçonnage, de canular, de scam ou autres.

Plusieurs techniques de lutte contre le spam sont possibles et peuvent être cumulées : analyse statistique, filtrage par mots clés ou par auteur, listes blanches, listes noires. Ces techniques de lutte doivent s'adapter en permanence car de nouveaux types de spam réussissent à les contourner. Les techniques de filtrage diffèrent selon que cette détection se fasse en amont ou en aval. En effet, deux solutions de détection de spam sont envisageables: la détection au niveau du serveur mail FAI et la détection au niveau de l'utilisateur final. Ces outils peuvent être divisés en deux groupes : le filtrage d'enveloppe, et le filtrage de contenu. Dans ce papier nous nous intéressons au filtrage de contenu à base d'apprentissage automatique. Ce type de filtrage se fait au niveau de l'utilisateur ou son contenu est analysé pour détecter les spam qui ont réussi à passer à travers le filtre d'enveloppe.

---

[1] http://www.enisa.europa.eu/act/res/other-areas/anti-spam-measures/studies/spam-slides: consulté le 16/01/2012

Plusieurs solutions à base d'apprentissage supervisé ont été proposées pour faire face à cette charge croissante de spam. Ci-après, nous présentons un panorama des techniques de filtrage de spam basées sur l'apprentissage supervisé[2] menés par la communauté de recherche et des développeurs de logiciels libres [13, 22, 23].

En 1998, dans une même conférence, apparaissent les premières publications académiques concernant le filtrage de spam [17, 18], on propose l'utilisation d'un classificateur bayésien naïf pour le classement de spam. Sahami [18] n'a retenu, pour l'opération de classement, que les 500 mots les plus significatifs de chaque message. Ce dernier a constaté que le classement binaire était plus efficace que le classement multi-classes basé sur le genre du message. Androutsopoulos *et al*. [2] ont évalué et comparé le classificateur bayésien naïf [1] avec le classificateur à mots-clés et TiMBL[3]. Les métriques d'évaluation d'efficacité ont intégré des coûts différents aux erreurs de classement selon la classe. Globalement, sauf pour le classificateur à mots-clés, les résultats étaient équivalents.

En 2002, Graham a réalisé un filtre appelé le modèle de Graham [12] qui utilise le principe de Bayes avec une représentation binaire en sac de mots d'un corpus privé et la mesure gain informationnel pour la sélection de termes. Son modèle a atteint des résultats similaires à ceux du modèle proposé dans [18]. Des variantes de cette technique de base ont été implémentées dans plusieurs travaux de recherche et produits logiciels. En 2006, Medlock, a réalisé son filtre nommé ILM [16]. Il utilise toujours le principe de Bayes mais cette fois-ci avec une représentation fréquentielle TF-IDF du corpus GenSpam[4]. Ce modèle a donné des résultats meilleurs que le classifieur SVM. Cormack et Lynam [7] ont testé des outils de filtrage de spam comme SpamAssassin, Bogofilter, SpamProbe et CRM114 avec le corpus Lingspam[5]. Ils ont constaté que ces filtres étaient en général incapables de classer les messages spam correctement.

L'utilisation des SVM pour le filtrage de spam a été proposée initialement par [10]. On a comparé l'efficacité des SVM avec RIPPER [6], Rocchio et Boosting. C'est une publication intéressante, car c'est la première qui a essayé un large ensemble de configurations d'expérimentations sur la sélection des termes et les différents modes d'apprentissage. Leurs conclusions étaient : (a) SVM et Boosting sont comparables, mais les SVM permettent d'atteindre des taux de faux positifs plus bas et plus facilement. (b) Les méthodes RIPPER et Rocchio ne sont pas performantes pour le filtrage de spam. (c) L'apprentissage avec Boosting est énormément long. D'autres résultats, par la suite, ont été publiés, et utilisent les SVM. Les auteurs dans [14] ont étudié la prise en compte des erreurs de classification spécifique à chaque classe. D'autre part, Sculley et Wachman [21] proposent l'utilisation des ROSVM (*Relaxed Online SVM*) pour l'apprentissage et le classement en ligne des spam; c'est une simplification qui limite le nombre d'itérations de l'algorithme. Son efficacité de classement reste proche de celle que l'on peut obtenir sans simplification.

---

[2] Nous avons préféré citer les travaux en suivant un classement tout d'abord selon le type d'algorithme d'apprentissage utilisé ensuite chronologiquement.

[3] une version de k plus proches voisins

[4] http://www.benmedlock.co.uk/genspam.html

[5] http://csmining.org/index.php/ling-spam-datasets.html

[19] réussissent la réalisation d'un filtre anti spam avec les K plus poches voisins (Kppv) en utilisant la méthode Gain informationnel (GI) pour la sélection des termes sur le corpus Lingspam, pour obtenir à la fin des résultats meilleurs que ceux du classifieur Bayésien naïf. Delany et Cunningham [8] réalisent le filtre «ECUE» avec Kppv en étudiant deux représentations différentes : binaire et fréquentielle. Ce filtre a obtenu une performance similaire à celle de Bayes. Ce même filtre a été amélioré par Delany et ses collègues [9] en appliquant une mise à jour aux messages mal classés, ce filtre a atteint une réduction considérable du taux des faux positifs. [11] réalisent le filtre «Spam Hunting», avec une représentation fréquentielle et une sélection de termes avec la méthode gain informationnel sur le corpus SpamAssassin[6]. Ce filtre a réussi de réduire le taux des faux positifs et des faux négatifs avec une grande rapidité de mise à jour.

Le filtrage à base de réseaux de neurones a été aussi étudié, on trouve par exemple le filtre «Linger » [5], le filtre « SBSA » [15] qui a montré des performances meilleures que celles de Bayes par la réduction des faux positifs. Par contre, les deux filtres à base de perceptron et Winnow [24] ont obtenu des résultats similaires à ceux de Bayes. Récemment, [20] ont examiné l'utilisation de la sémantique dans le filtrage de spam en représentant les courriels avec un nouveau modèle vectoriel nommé eTVSM (enhanced Topic Vector Space Model). Le eTVSM utilise une ontologie pour représenter les différentes relations entre les termes et, de cette manière, offre un modèle plus riche qui est en mesure de représenter la synonymie, l'homonymie et autres phénomènes linguistiques. Sur la base de cette représentation, ils appliquent plusieurs classifieurs bien connus (Bayes, Kppv, SVM et arbre de décision) sur le corpus Lingspam et montrent que la méthode proposée permet de détecter la sémantique interne des messages et que cette approche donne des pourcentages élevés de détection de spam, tout en gardant le nombre de messages légitimes mal classés faible.

## 3 Approche proposée

Dans cette section, nous allons étudier une nouvelle approche pour la détection de spam à base d'induction symbolique. Le principe consiste à construire un modèle booléen utilisant le principe de l'automate cellulaire CASI à partir d'un ensemble de courriels d'apprentissage. Le modèle une fois construit est censé analyser les nouveaux e-mails entrant pour détecter les spam. Avant d'aller plus loin, nous allons tout d'abord décrire brièvement l'automate cellulaire utilisé dans cette étude (pour plus d'informations voir [3]).

### 3.1 Automate cellulaire CASI

L'automate CASI (Cellular Automata for System Induction) issu des travaux de [3] est une méthode cellulaire de génération, de représentation et d'optimisation des graphes d'induction [26] générés à partir d'un ensemble d'exemples d'apprentissage. Ce système cellulo-symbolique est organisé en cellules où chacune d'elles, est reliée

[6] http://csmining.org/index.php/spam-assassin-datasets.html

seulement avec son voisinage. Toutes les cellules obéissent en parallèle à la même règle appelée fonction de transition locale, qui a comme conséquence une transformation globale du système. Pour construire le modèle booléen qui va servir par la suite dans la détection de spam, trois composants coopèrent entre eux :

1. Le module COG (Cellular Optimization and Generation) : s'occupe de la génération du graphe d'induction cellulaire et de son optimisation ;
2. Le module CIE (Cellular Inference Engine) : le moteur d'inférence cellulaire s'occupe de la génération d'un ensemble de règles cellulaires utilisées dans la phase de classification ;
3. Le module CV (cellular validation) pour la validation cellulaire.

**Base de connaissances.** Les deux composants COG et CIE utilisent une base de connaissances sous forme de deux couches finies d'automates finis : La première couche, *CelFact*, pour la base de faits et, la deuxième couche, *CelRule*, pour la base de règles. La couche *CelFact* se caractérise par trois états : état d'entrée (*EF*), état interne (*IF*) et état de sortie (*SF*). Toute cellule de *CelFact* est considérée comme un fait établi si son état EF est égale à 1, sinon, elle est considérée comme fait à établir. La couche *CelRule* se caractérise par trois états : état d'entrée (*ER*), état interne (*IR*) et état de sortie (*SR*). Toute cellule de *CelRule* est considérée comme une règle candidate si son état ER est égale à 1, sinon, elle est considérée comme étant une règle qui ne doit pas participer à l'inférence.

**Voisinage.** Le voisinage des cellules est défini par deux matrices d'incidence:
Soit l le nombre de faits dans la base et soit r le nombre de règles.

1. La relation d'entrée, notée $R_E$, est formulée comme suit :

$$\forall i \in [1,l], \forall j \in [1,r]: si\ (i\ fait\ \in prémisse(règle\ j) alors\ R_E\ (i,j) = 1 \quad (1)$$

2. La relation de sortie, notée $R_S$, est formulée comme suit :

$$\forall i \in [1,l], \forall j \in [1,r]: si\ (i\ fait\ \in conclusion\ (règle\ j) alors\ R_S\ (i,j) = 1 \quad (2)$$

**Inférence.** La dynamique de l'automate cellulaire, utilise deux fonctions de transitions [3] *δfact* qui simule les phases de sélection et de filtrage dans un système expert et *δrule* qui simule la phase d'exécution :

- $(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{fact}} (EF, IF, \boldsymbol{EF}, \boldsymbol{ER} + (\boldsymbol{R_E^T}\ ^{\times}\boldsymbol{EF}), IR, SR)$
- $(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{\text{rule}}} (\boldsymbol{EF} + (\boldsymbol{R_S} \times \boldsymbol{ER}), IF, SF, ER, IR, \overline{\boldsymbol{ER}})$

### 3.2 Architecture du système

La figure 1 illustre l'architecture du système de détection de spam à base d'automate cellulaire CASI que nous avons nommé SPAMAUT.

L'architecture globale du système est constituée principalement des étapes suivantes :

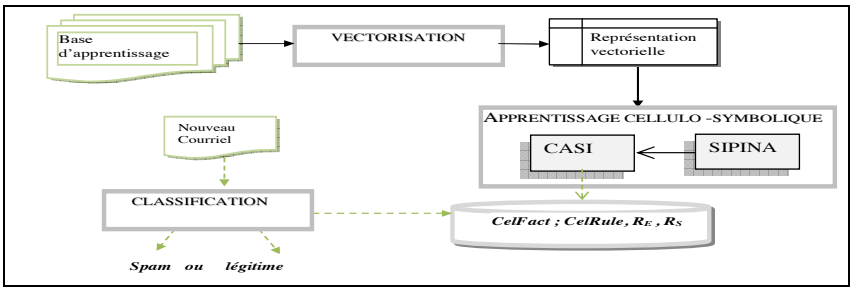



**Fig. 1.** Architecture du système SPAMAUT

**Représentation vectorielle des courriels d'apprentissage.** Les textes des courriels subissent un ensemble de traitements pour récupérer une représentation numérique exploitable par un algorithme d'apprentissage. Cette représentation est appelée représentation vectorielle. Les courriels passent donc par les étapes suivantes :

- Segmentation en mots, normalisation et racinisation
- Construction de l'index où sont sauvegardées pour chaque mot toutes les informations nécessaires à la construction de la représentation vectorielle
- Réduction de l'index par une des méthodes de réduction des attributs qui sont le gain informationnel (GI), la statistique de chi-2 et l'information mutuelle (IM) telles que définies dans [25].
- Création de l'espace vectoriel et pondération des termes représentatifs (binaire, fréquentielle (TF) ou TF-IDF[7])

L'ensemble des textes est transformé en un ensemble de vecteurs, ou un tableau croisé (individus×variables) où les individus (les lignes du tableau) représentent les courriels, et les variables (les colonnes du tableau) représentent les termes qui sont extraits des documents d'apprentissage et sélectionnés pour représenter la collection. Chaque cellule dans le tableau représente le poids du terme dans un courriel donné.

**Apprentissage cellulo-symbolique.** Nous résumons dans l'algorithme 1 suivant les différentes étapes suivies pour générer le modèle à base d'automate cellulaire utilisé par la suite dans la détection de spam.

```
Algorithm 1 :
Entrée: Format arff de la représentation vectorielle
Sortie modèle booléen : Celfact, CelRule, RE et RS
Début
  Appliquer l'algorithme de Sipina pour générer le
graphe d'induction
  Lancer le module COG de CASI pour transformer le
graphe cellulaire en un modèle booléen :
      a. Générer les règles
```

[7] Term Frequency-Inverse Document Frequency

```
        b. Représentation en couches (Celfact et CelRule)
           des règles
        c. Génération des matrices d'incidence de
           voisinage
    Lancer le moteur CIE
    Répéter
    Appliquer les deux fonctions de transitions δfact et
    δrule
    Jusqu'à stabilisation
    Sauvegarder le modèle obtenu
Fin.
```

Ce modèle booléen ainsi construit nous offre deux avantages:

1. ***une mémoire de stockage réduite*** ; une fois ce modèle élaboré, nous n'avons plus besoin de sauvegarder le graphe d'induction, celui-ci est représenté par les deux couches CelFact, CelRule et les deux matrices d'incidence d'entrée/sortie qui sont toutes booléennes et peuvent être représentées par des vecteurs de bits.
2. ***un temps de classification réduit*** : pour classifier un nouvel e-mail, nous allons tout simplement lancer l'inférence cellulaire à l'aide des deux fonctions de transitions δfact et δrule qui se basent sur des opérations booléennes (addition, produit, négation) qui sont rapides en exécution.

**Classification.** Pour classifier les nouveaux courriels, l'automate cellulaire fait appel au module CIE qui simule le fonctionnement du cycle de base d'un moteur d'inférence en utilisant comme base de connaissances le modèle élaboré depuis la phase d'apprentissage ; c'est la configuration finale de CelFact, CelRule, $R_E$ et $R_S$. Nous résumons les principales étapes dans l'algorithme 2 suivant :

```
Algorithme 2 :
   Entrée : le nouveau courriel
   Sortie type du courriel : légitime ou spam
  Début
  Charger le modèle booléen: Celfact, Celrule, R_E et R_S
     Initialiser la base de faits Celfact
     Pour chaque terme t ∈ Celfact
       Si t figure dans le courriel alors
         activer la cellule t=1 : EF(t=1)=1
       sinon
         activer la cellule t=0 : EF(t=0)=1
       Fin si
     Fin pour
     Lancer le moteur d'inférence cellulaire CIE
     Appliquer δfact•δrule
             Si la cellule « class= spam » est active :
         EF(class=spam)=1 alors le courriel est un spam
           Sinon le courriel est un légitime
 Fin.
```

## 4 Etude expérimentale et Résultats

Afin de tester les performances de la méthode cellulaire décrite dans la section précédente, nous avons choisi d'utiliser le corpus Lingspam.

LingSpam contient 2893 courriels dont 2412 sont des messages légitimes et le reste sont des spam. Nous avons entamé plusieurs expériences sur ce corpus. Et en nous appuyant sur la validation croisée (10-validation croisée), et en suivant les travaux effectués dans ce domaine nous mesurons alors plusieurs indicateurs d'évaluation de la classification : le rappel de la classe spam, la précision de la classe spam, la F-mesure de la classe spam et l'exactitude. Nous avons mené plusieurs expériences en faisant varier :

- La représentation des courriels, l'unité linguistique peut être le mot tel qu'il apparaît dans les textes des courriels ou bien sa racine, avec utilisation ou non d'une stop-liste.
- La pondération des termes peut être binaire, fréquentielle (TF) ou TFIDF,
- La mesure de sélection des termes peut être GI, MI ou $\chi 2$

Nous avons constaté que la qualité de prédiction devient de plus en plus meilleure en termes de précision, rappel et exactitude à partir de 300 termes lorsqu'il y a racinisation des mots et élimination des mots vides et ceci quelque soit le type de pondération et la mesure de sélection utilisé. La performance du système atteint son maximum lorsque le nombre de termes équivaut 500.

Nous avons constaté que la mesure de sélection gain d'information (voir figure 2) amène à une qualité de prédiction meilleure que les deux autres en terme de rappel, exactitude et F-mesure, alors que les deux autres sont meilleurs en précision (=100%). Notez, aussi que le rappel est très faible avec chi-2 car le nombre de faux négatifs est très élevé ce qui a donné une F-mesure aussi faible.

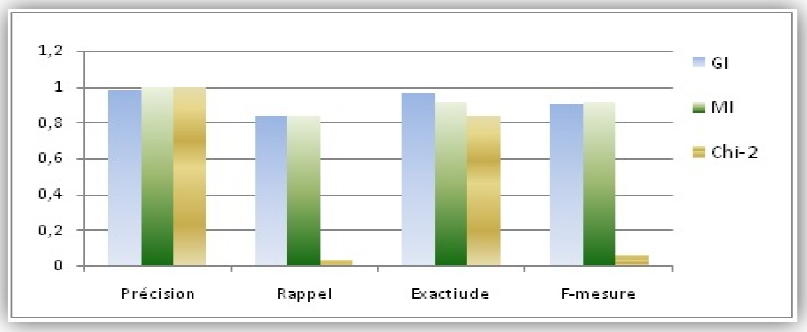



**Fig. 2.** Performance du système en fonction de la mesure de réduction

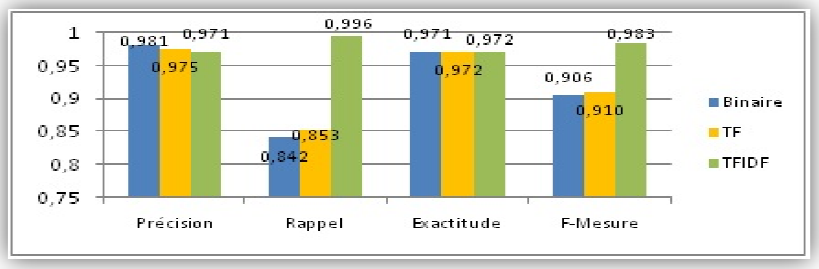



**Fig. 3.** Performance du système en fonction de la pondération des termes[8]

Les expériences ont permis aussi de vérifier que l'approche proposée se stabilise à partir de 500 termes et amène à une qualité de prédiction intéressante et encourageante. Comme illustré dans la figure 3, la précision atteint son maximum (98,1%) dans le cas d'une pondération binaire. Par contre le rappel, l'exactitude et la F-mesure sont meilleurs avec la pondération TFIDF (rappel = 99,6%, exactitude :=97,2%, F-mesure :=98,3%).

Nous avons aussi comparé notre approche avec d'autres algorithmes (les arbres de décision (J48), les réseaux bayésiens (BayesNet), les machines à vecteurs de support (SMO), l'algorithme bayésien (NB) et les K-plus proches voisins (k=1, k=3)) de la plate forme Weka[9]. Sur les figures 4, 5, 6 et 7, nous avons les résultats des expériences effectués sur le corpus Lingspam selon la configuration suivante : sélection de 500 termes avec la méthode gain informationnel et pondération binaire.

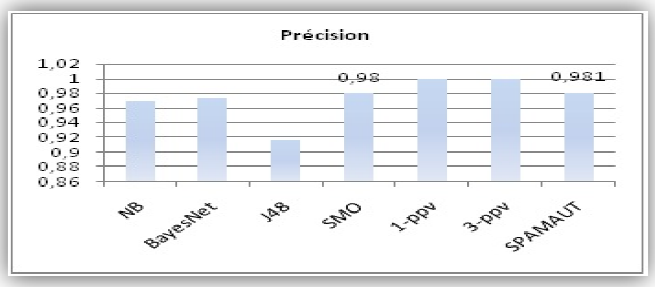



**Fig. 4.** Comparaison de la précision

[8] Résultats obtenus avec la meilleure configuration : Racinisation + élimination de mots vides + réduction avec la mesure GI

[9] www.cs.waikato.ac.nz/ml/weka

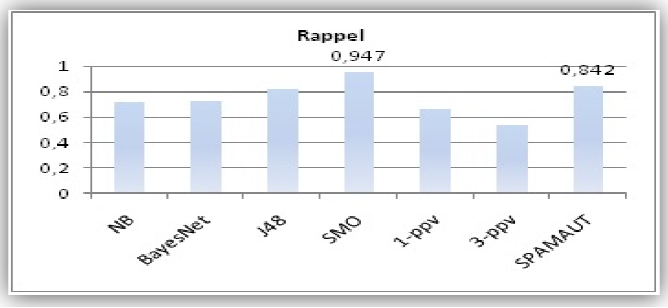



**Fig. 5.** Comparaison du rappel

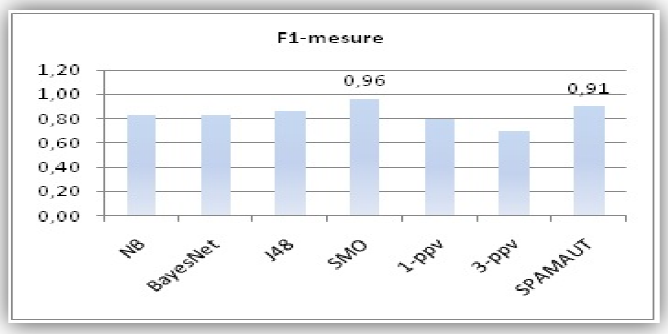



**Fig. 6.** Comparaison de la F1-mesure

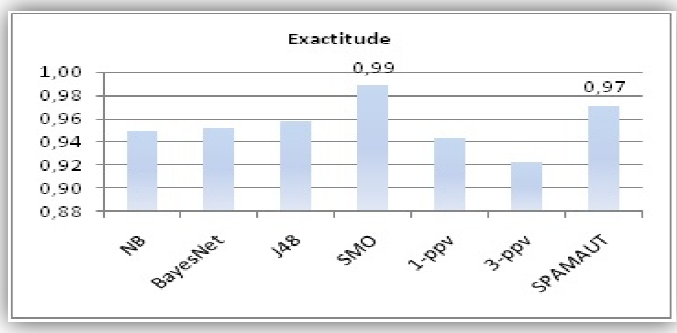



**Fig. 7.** Comparaison de l'exactitude

Nous observons que dans l'ensemble notre système est le plus performant par rapport à la majorité des autres algorithmes. Nos résultats en précision, rappel, F1-

mesure et exactitude sont les meilleurs bien que légèrement en dessous de ceux de SMO dans le cas du rappel, F1-mesure et exactitude.

Notons que notre système se base sur un modèle booléen caractérisé par un temps d'exécution et un stockage mémoire réduit et une compréhensibilité du modèle très élevée (génération des règles). Alors que SMO qui est une version de SVM est couteux en temps d'apprentissage et sa compréhensibilité est faible.

## 5 Conclusion

Dans ce papier, nous avons expérimenté une nouvelle approche de détection de spam. La méthode que nous proposons s'appuie sur l'induction symbolique par automate cellulaire CASI. Elle procède par construction d'un modèle booléen composé de deux couches finies d'automates finis (CelFact et CelRule) où le voisinage est représenté par deux matrices d'incidence d'entrée ($R_E$) et de sortie ($R_S$) et l'inférence, pour le filtrage de spam, est assurée par deux fonctions de transitions (δfact et *δrule* ).

Dans nos expériences, nous avons tout d'abord, essayé d'examiner l'impact de diverses méthodes de sélection de termes, la racinisation, et la pondération des termes sur la performance du système proposé. Nous avons donc entrepris une étude expérimentale en testant plusieurs configurations. Nos premières évaluations indiquent que l'approche proposée est intéressante et encourageante. Les résultats expérimentaux ont permis d'observer que la qualité de prédiction est meilleure avec la configuration suivante : *racinisation des termes, élimination de mots vides et sélection des termes avec la mesure GI*. La fonction de sélection GI amène à une meilleure qualité de classification, en termes de rappel, exactitude, et F-mesure. Par contre la précision est meilleure avec les fonctions de sélection MI et Chi-2.

La méthode proposée est aussi comparée à d'autres algorithmes de la plate forme Weka. Les résultats obtenus sont intéressants, ils nous permettent d'attester dans un premier temps le bon fonctionnement du système et dans un deuxième temps de le positionner par rapport aux résultats des différents algorithmes expérimentés dans cette étude.

## Références


1. Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D.: An Evaluation of Naive Bayesian Networks. In Proceeding of of the Workshop on Machine Learning in the New Information Age, pp. 9--17. Barcelona, Spain (2000a).
2. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., & Spyropoulos, C. D.: Learning to filter spam e-mail: a comparison of a naïve Bayesian and a memory based approach. Proceedings of the Workshop on Machine Learning and Textual Information Access. Lyon, France (2000b).
3. Atmani, B., Beldjilali, B.: Knowledge Discovery in Database : Induction Graph and Cellular Automaton. Computing and Informatics Journal. 26, 171--197 (2007)
4. Barigou, F., Barigou, N.: Un Automate Cellulaire pour la détection de spam. Atelier "Data Mining, Applications, Cas d'Etudes et Success Stories. 11ème conférence francophone "Extraction et Gestion de Connaissances" EGC. - Brest, France, pp. 25-28 (2011)

5. Clark, J., Koprinska, I., Poon, J.: A neural network based approach to automated e-mail classification. Proceeding of the IEEE/WIC international conference on web intelligence and Intelligent Agent Technology, pp. 702-705, Sidney university, Australia (2003)
6. Cohen, W. W., Singer, Y.: Context-sensitive methods for text categorization. Proceedings of the 19th Annual International Conference on Research and Development in Information Retrieval, pp. 307--315 (1996)
7. Cormack, G. V., Lynam, T. R.: Online supervised spam filter evaluation. *ACM Transactions on Information System. 25(3),* (2007)
8. Delany, S. J., Cunningham, P.: An Analysis of Case-Based Editing in a Spam Filtering System. 7th European Conf. on Case-Based Reasoning 3155, pp. 121--141 Springer (2004)
9. Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L.: A case based technique for tracking concept drift in spam filtering. Knowledge-Based Systems, 18, 187-195 (2006)
10. Drucker, H., Vapnik, V., Wu, D.: Support vector machines for spam categorization. IEEE Transactions on Neural Networks , 10 (5), 1048--1054 (1999)
11. Fdez-Riverola, F., Iglesias, E., Díaz, F., Méndez, J. R., Corchado, J. M. : SpamHunting: An instance-based reasoning system for spam labelling and filtering. *Decision Support Systems , 43* (3), 722--736 (2007)
12. Graham, P.: *A plan for Spam*. http://www.paulgraham.com/spam.html (2002)
13. Guzella, T. S., Caminhas, W. N. : A review of machine mearning approaches to spam filtering. *Expert systems with applications , 36(7)*, 10206--10222 (2009)
14. Kolcz, A., Alspector, J.: SVM-based filtering of e-mail spam with content-specific misclassification costs[C]. Proc. of Workshop on TM, pp.123--130 California, USA (2001)
15. Luo, X., Zincir Heywood, N.: Comparison of a SOM based sequence analysis system and naive Bayesian classifier for spam filtering. Proc. of the int. conf. on neural networks, 4, pp. 2571--2576 (2005)
16. Medlock, B.: An adaptive, semi-structured language model approach to spam filtering on a new corpus. Proc. of the3rd conference on email and anti-spam. California,USA (2006)
17. Pantel, P., Lin, D. *Spamcop :* A spam classification and organization program, learning for text categorization. Technical Repport. (1998)
18. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A bayesian approach to filtering junk email. Proc. of the Work. on Learning for Text Categorization. Madison, USA (1998)
19. Sakkis, G., Androutsopoulos, I., Paliouras, G., & Karkaletsis, V.: A memory based approach to anti spam filtering for mailing lists. Information Retrieval., *6* (1), 49--73 (2003)
20. Santos, I., Laorden, C., Sanz, B., Bringas, P. G.: Enhanced Topic-based Vector Space Model for semantics-aware spam filtering. (eTVSM). *Expert Systems With Applications, 39* (1), 437--444 (2012)
21. Sculley, D., Wachman, G. M.: Relaxed online SVMs for spam filtering. Proc. of the annual int. ACM SIGIR conf. on research and developement in information retrieval (2007)
22. Subramaniam, T., Jalab, H., Taga, A. Y.: Overview of textual anti-spam filtering techniques. International journal of the physical sciences , 5(12), 1869--1882 (2010)
23. Upasana, P. U., Chakraverty, S.: A review of text classification approaches for e-mail management. International journal of engineering technologies , 3(2), 137--144 (2011)
24. Wang, B., Jones, ,. G., Pan, W.: Using online linear classifiers to filter spam emails. Pattern Analysis and Applications. , 9 (4), 339--351 (2006)
25. Yang, Y., Pederson, J.: A comparative study on feature selection in text categorization. Proceedings of ICML'97, pp. 412--420 (1997)
26. Zighed, D.A., Rakotomalala, R.: Graphes d'induction: apprentissage et data mining. Hermes Science Publications (2000)

# Optimisation

# Feasible short-step interior point algorithm for linear complementarity problem based on kernel function

El Amir Djeffal[1] and Lakhdar Djeffal[1]

[1]Mathematics Department, University Hadj Lakhdar of Batna, Algeria

**Abstract.** In this paper we deal with the study of the polynomial complexity analysis and numerical implementation for a short-step interior point algorithm for monotone linear complementarity problems ($LCP$) based on kernel function. The analysis is based on a new class of search directions. We establish the global convergence of the algorithm. Furthermore, it is shown that the algorithm has $O(n^{2.5}L)$, iteration complexity. For its numerical tests some strategies are used and indicate that the algorithm is efficient.

## 1 Introduction

Let us consider the linear complementarity problem ($LCP$): find vectors $x$ and $y$ in real space $\Re^n$ that satisfy the following conditions:

$$x \geq 0,\ y = Mx + q \geq 0 \ and \ x^t y = 0, \tag{1}$$

where $q$ is a given vector in $\Re^n$ and $M$ is a given $n \times n$ real matrix. $LCP$ have important applications in mathematical programming and various areas of engineering. Interior-point methods ($IPMs$) for solving Linear Optimization ($LO$) problems were initiated by Karmarkar [**?**]. They not only have polynomial complexity, but are also highly efficient in practice. Feasible $IPMs$ start with a strictly feasible interior-point and maintain feasibility during the solution process. Feasible $IPMs$ require that the starting points satisfy exactly the equality constraints and are strictly positive, i.e., they lie in the interior of a region defined by constraints. Extending methods for $LO$ to $LCP$ has been successful in many cases. See, e.g.,[**?**,**?**]. Recently, Peng et al. [**?**,**?**] designed primal–dual feasible $IPMs$ by using self-regular functions for $LO$ and also extended the approach to $LCP$

In this paper we deal with the complexity analysis and the numerical implementation of a short-step interior point algorithm. This algorithm is based on the strategy of the central path and on a method for finding a new search directions, where we show that this short-step algorithm deserves the best current polynomial complexity namely $O(n^{2.5}L)$.

The paper is organized as follows. In the next section, the statement of the problem is presented, we deal with the weighted vector introduced to ensure

that the initial point $(x^0, y^0)$ verified $\delta(x^0y^0, \mu^0) = 0$, (proximity measure define bellow). In Section 3, we deal with the new search directions and the description of the algorithm. In Section 4, we state its polynomial complexity. Section 5 contains the numerical experiments. In Section 6, a conclusion and remarks are given.

## 2 Statement of the problem

The feasible set, the strictly feasible set and the solution set of (1) are denoted, respectively by

$$F = \left\{(x, y) \in \Re^{2n} : y = Mx + q, x \geq 0, y \geq 0\right\},$$

$$oF = \left\{(x, y) \in F : x > 0, y > 0\right\},$$

and

$$\Omega = \left\{(x, y) \in : x \geq 0, y \geq 0, x^t y = 0\right\}.$$

In this paper, we assume that the following assumptions hold.

**Assumption 1**. $oF \neq \emptyset$.

**Assumption 2**. $M$ is a positive semidefinite matrix.

In addition (1), is equivalent to the following convex quadratic problem, see, e.g., [**?**].

$$\min\left\{x^t y : x \geq 0, y \geq 0, y = Mx + q\right\}. \tag{2}$$

Hence, finding the solution of (1) is equivalent to find the minimizer of (2) with its objective value is zero.

In order to introduce an interior point method to solve (2), we associate with it the following barrier minimization problem

$$\min\left\{f_{\mu r}(x, y) : y = Mx + q, x > 0, y > 0\right\}, \tag{3}$$

where $f_{\mu r}(x, y) = x^t y - \mu \sum_{i=1}^{n} r_i \log(x_i y_i),\ \mu > 0$ be the barrier parameter and $r = (r_1, r_2, ..., r_n) \in \Re^n_+$ is a weighted vector introduced to ensure that the initial point $(x^0, y^0)$ verified $\delta(x^0y^0, \mu^0) = 0$ (proximity measure define bellow), if $r_i = 1, i = 1, ..., n$, then the weighted central path coincides with the classical one. Hence, this approach can be seen as a generalization of central path methods.

The problem (3) is a convex optimization problem and then its first order optimality conditions are:

$$\begin{cases} Mx + q = y, \\ xy = \mu r,\ x > 0, y > 0. \end{cases} \tag{4}$$

If the Assumptions 1 and 2 hold then for a fixed $\mu > 0$, the problem (3) and the system (4) have a unique solution [**?**] denoted as $(x(\mu), y(\mu))$, with $x(\mu) > 0$ and $y(\mu) > 0$. We call $(x(\mu), y(\mu))$, with $\mu > 0$, the$\mu$-centers of (4). The set of the $\mu$-centers defines the so-called the central path of (1).

In the next section, we introduce a method for tracing the central path based a new class of search directions.

## 3 A new search directions

Now, the basic idea behind this approach is to replace the non linear equation $\frac{xy}{\mu r} = e$ in (4) by an equivalent equation $\psi(\frac{xy}{\mu r}) = \psi(e)$, where $\psi$ is a real valued function on $[0, \infty)$ and differentiable on $[0, \infty)$ such that $\psi(t)$ and $\psi^{'}(t) > 0$, for all $t > 0$. Then the system (4) can be written as the following equivalent form:

$$\begin{cases} Mx + q = y, \ x > 0, y > 0 \\ \psi(\frac{xy}{\mu r}) = \psi(e). \end{cases} \tag{5}$$

Suppose that we have $(x, y) \in o\digamma$. Applying Newton's method for the system (5), we obtain a new class of search directions:

$$\begin{cases} M\triangle x = \triangle y, \\ \frac{y}{\mu r}\psi^{'}(\frac{xy}{\mu r})\triangle x + \frac{x}{\mu r}\psi^{'}(\frac{xy}{\mu r})\triangle y = \psi(e) - \psi(\frac{xy}{\mu r}). \end{cases} \tag{6}$$

Now, the following notations are useful for studying the complexity of the proposed algorithm.

The vectors

$$v = \sqrt{\frac{xy}{\mu r}}, \ d = \sqrt{xy^{-1}},$$

these notations lead to

$$\frac{d^{-1}x}{\sqrt{\mu r}} = \frac{dy}{\sqrt{\mu r}} = v.$$

Denote by

$$d_x = \frac{d^{-1}\triangle x}{\sqrt{\mu r}}, \ d_y = \frac{d\triangle y}{\sqrt{\mu r}}, \tag{7}$$

and hence, we have

$$\mu r v(d_x + d_y) = y\triangle x + x\triangle y, \tag{8}$$

and

$$d_x d_y = \frac{\triangle x \triangle y}{\mu r} \tag{9}$$

So using (7) and (8), the system (6) becomes

$$\begin{cases} \overline{M}d_x = d_y, \\ d_x + d_y = p_v, \end{cases}$$

where $\overline{M} = MDM$ with $D = diag(d)$
and

$$p_v = \frac{\psi(e) - \psi(v^2)}{v\psi'(v^2)}.$$

We shall consider the following function:

$$\psi(t) = \frac{1}{2}(t^2 - 1), with \psi'(t) = t for all t > 0.$$

Hence, the Newton directions in (6) is

$$\begin{cases} M\triangle x = \triangle y, \\ d_x + d_y = p_v = \frac{1}{2}(v^{-1} - v) \end{cases} \tag{10}$$

and we define for all vector $v$ the following proximity measure by

$$\delta(xy, \mu) = \frac{\|p_v\|_2}{2} = \left\|v^{-1} - v\right\|_2 = \left\| \left( \sqrt{\frac{xy}{\mu r}} \right)^{-1} - \sqrt{\frac{xy}{\mu r}} \right\|_2 .$$

Now, the generic short-step primal-dual algorithm to solve $LCP$ has the following form

### 3.1 Algorithm

**Begin algorithm**
**Input:**
an accuracy parameter $\varepsilon > 0$,
an update parameter $\theta$, $0 < \theta < 1$ (default $\theta = \frac{1}{2\sqrt{n}}$),
a strictly feasible point $(x^0, y^0), X = diag(x), Y = diag(y)$ .
$\sigma = \left\|X^0 Y^0 e\right\|\sqrt{n}$ , $r = X^0 Y^0 e\sigma . R = diag(r)$, $\mu^0 = \frac{(x^0)^t \ Ry^0}{n}$
$k = 0$
**While** $(n\mu^k) > \varepsilon$ **do**
$1^\circ$) Compute $(\triangle x, \triangle y)$,
$2^\circ$) Update $(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\triangle x^k, \triangle y^k)$
$3^\circ$) Set $\mu^{k+1} = (1 - \theta)\mu^k = (1 - \theta)\frac{x^k R y^k}{n}$ and $k = k + 1$.
**End While.**
**End algorithm.**

## 4 Complexity analysis

Let

$$p_v = d_x + d_y, \ q_v = d_x - d_y,$$

and, we have

$$d_x = \frac{1}{2}(p_v + q_v),\ d_y = \frac{1}{2}(p_v - q_v),$$

hence,

$$d_x d_y = \frac{1}{4}(p_v^2 - q_v^2) and \left\|q_v\right\|_2 \leq \left\|p_v\right\|_2 .$$

This last result follows directly from the equality

$$\|p_v\|_2^2 = \|q_v\|_2^2 + 4 d_x^t d_y,$$

since,

$$d_x^t d_y = d_x^t \overline{M} d_x \geq 0, because \overline{M}\ is\ positive\ semidefinite.$$

We have

$$\delta(v, \mu) \geq \|q_v\|_2$$

In the following lemma, we state a condition which ensures the feasibility of the full Newton step.

Let

$$x^+ = x + \alpha \triangle x and y^+ = y + \alpha \triangle y,$$

be the new iterate after a full Newton step.

**Lemma 1.** *Let $(x, y)$ is a strictly feasible iteration. If $e + d_x d_y > 0$ then $(x^+, y^+) = (x + \alpha\triangle x, y + \alpha\triangle y)$ is strictly feasible*

*Proof.* Let $0 < \alpha \leq 1$ is step lenght.

We define:

$$x(\alpha) = x + \alpha\triangle x,\ y(\alpha) = y + \alpha\triangle y,$$

we have

$$\begin{aligned} x(\alpha)y(\alpha) &= (x + \alpha\triangle x)(y + \alpha\triangle y) \\ &= xy + \alpha(x\triangle y + y\triangle x) + \alpha^2 \triangle x \triangle y \\ &= xy + \alpha(\mu r - xy) + \alpha^2 \triangle x \triangle y. \end{aligned}$$

We assume that $e + d_x d_y > 0$, we deduce that $\mu r + \triangle x \triangle y > 0$, which is equivalent to $\triangle x \triangle y > -\mu r$, by substitution we obtain

$$\begin{aligned} x(\alpha)y(\alpha) &> xy + \alpha(\mu r - xy) - \alpha^2 \mu r \\ &= (1 - \alpha)xy + (\alpha - \alpha^2)\mu r \\ &= (1 - \alpha)xy + \alpha(1 - \alpha)\mu r. \end{aligned}$$

Since

$xy > 0$ and $\mu r > 0$, it follows that $x(\alpha)y(\alpha) > 0$ for all $\alpha \in ]0, 1]$.Hence, none of the entries of $x(\alpha)$ and $y(\alpha)$ vanish for $2\alpha \in ]0, 1]$. Since$x^0$ and $y^0$ are positive, this implies that $x(\alpha) > 0$ and $y(\alpha) > 0$ for $\alpha \in ]0, 1]$. Hence, by continuity the vectors $x^1$ and $y^1$ must be positive which proves that $x^+$ and $y^+$ are strictly feasible.

Now for convenience,we may write

$$(v^+)^2 = \frac{x^+y^+}{\mu r} = e + d_x d_y.$$

**Lemma 2.** *If $\delta(xy, \mu) < 1$,Then $x^+ > 0$ and $y^+ > 0$.*

*Proof.* In the Lemma , we have $(x^+, y^+)$ are strictly feasile if $(e + d_x d_y) > 0$. So $(e + d_x d_y) > 0$ holds if $(1 + (d_x d_y)_i)$ for all $i \in \Re^n$.

we have

$$\begin{aligned}(1 + (d_x d_y)_i) &\geq (1 - |(d_x d_y)_i|, for all i \in \Re^n \\ &\geq (1 - \delta^2).\end{aligned}$$

Thus $(e + d_x d_y) > 0$ if $\delta(xy, \mu) < 1$.

In the next lemma we proved the local quadratic convergence for our algorithm

**Lemma 3.** *Let $\delta = \delta(xy, \mu) < 1$ then*

$$\delta(x^+y^+, \mu^+) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

*Proof.* letting $\alpha = 1$,

we have

$$\begin{aligned}4\delta_+^2 &= \left\|(v^+)^{-1} - v^+\right\|_2^2 \\ &= \left\|(v^+)^{-1}(e - (v^+)^2)\right\|_2^2,\end{aligned}$$

where $(v^+)^2 = (e - d_x d_y)$ and $(v^+)^{-1} = \frac{1}{\sqrt{(e+d_x d_y)}}$, then it follows that

$$\begin{aligned}4\delta_+^2 &= \left\| \frac{d_x d_s}{\sqrt{(e + d_x d_y)}} \right\|_2^2 \\ &= \left\| \frac{d_x d_s}{\sqrt{(e + d_x d_y)}} \right\|_2^2 \\ &\leq \frac{\|d_x d_y\|_2^2}{(1 - \|d_x d_y\|_\infty)}.\end{aligned}$$

We deduce that

$$4\delta_+^2 \leq \frac{2\delta_+^4}{(1-\delta_+^2)}.$$

This proves the lemma.

**Lemma 4.** *Let* $\delta(xy,\mu) < \frac{1}{\sqrt{2}}$ *and* $\mu^+ = (1-\theta)\mu,\ 0<\theta<1$. *Then*

$$\delta^2(x^+y^+,\mu^+) \leq (1-\theta)\delta_+^2 + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

*Furthermore, if* $\delta \leq \frac{1}{\sqrt{2}},\ \theta = \frac{1}{2\sqrt{n}}$ *and* $n \geq 2$, *then we have* $\delta(x^+y^+,\mu^+) \leq \frac{1}{\sqrt{2}}$.

*Proof.* Let $v^+ = \sqrt{\frac{x^+y^+}{\mu^+r}}$ and $\mu^+ = (1-\theta)\mu$, then

$$\begin{aligned}
4\delta^2(x^+y^+,\mu^+) &= \left\| \sqrt{\frac{\mu^+r}{x^+y^+}} - \sqrt{\frac{x^+y^+}{\mu^+r}} \right\|_2^2 \\
&= \left\| \sqrt{1-\theta}(v^+)^{-1} - \frac{1}{\sqrt{1-\theta}} v^+ \right\|_2^2 \\
&= \left\| \sqrt{1-\theta}\left((v^+)^{-1} - v^+\right) - \frac{\theta}{\sqrt{1-\theta}} v^+ \right\|_2^2 \\
&= (1-\theta)\left\|(v^+)^{-1} - v^+\right\|_2^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta\left((v^+)^{-1} - v^+\right)^t v^+ \\
&= (1-\theta)\left\|(v^+)^{-1} - v^+\right\|_2^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta(v^+)^{-t}v^+ + v^{+^t}v^+ \\
&= 4(1-\theta)\delta_+^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta\left\|v^+\right\|_2^2
\end{aligned}$$

since,
$(v^+)^{-t}v^+ = n$ and $(v^+)^t v^+ = \|v^+\|_2^2$, and we have $\delta_+^2 \leq \frac{\delta_+^4}{2(1-\delta_+^2)}$.Then

$$4\delta^2(x^+y^+,\mu^+) \leq 4(1-\theta)\frac{\delta^4}{2(1-\delta_+^2)} + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta\left\|v^+\right\|_2^2,$$

since,

$$x^+s^+ = \mu r(e + d_x d_y),$$

and if $\delta < \frac{1}{\sqrt{2}}$, it follows

$$\left\|v^+\right\|^2 = \frac{x^+y^+}{\mu r} = e + d_x d_y \leq 1 + n.$$

Consequently,

$$4\delta^2(x^+y^+,\mu^+) \leq 4(1-\theta)\frac{\delta^4}{2(1-\delta_+^2)} + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta(n+1),$$

and

$$\delta^2(x^+y^+,\mu^+) \leq (1-\theta)\delta_+^4 + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

The last statement the proof goes as follows. If $\delta < \frac{1}{\sqrt{2}}$ , then $\delta_+^2 = \frac{1}{4}$ and this yields the following upper bound for $\delta(x^+y^+,\mu^+)$ as:

$$\delta^2(x^+y^+,\mu^+) \leq \frac{(1-\theta)}{4} + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

Now, taking $\theta = \frac{1}{2\sqrt{n}}$ then $\theta^2 = \frac{1}{4n}$ if follows that

$$\delta^2(x^+y^+,\mu^+) \leq \frac{\frac{(n+1)}{4n}}{4(1-\theta)} + \frac{(1-\theta)}{4} + \frac{\theta}{2},$$

since $\frac{(n+1)}{4n} \leq \frac{3}{8}$ for all $n \geq 2$ then we have

$$\delta^2(x^+y^+,\mu^+) \leq \frac{3}{32(1-\theta)} + \frac{\theta+1}{4}.$$

Now for $n \geq 2$, we have $0 < \theta \leq \frac{1}{2\sqrt{2}}$ and since the function $f(\theta) = \frac{3}{32(1-\theta)} + \frac{\theta+1}{4}$ is continuous and monotonic increasing on $0 < \theta \leq \frac{1}{2\sqrt{2}}$, consequently,

$$f(\theta) \leq f(\frac{1}{2\sqrt{2}}) < \frac{1}{2}, forall 0 < \theta \leq \frac{1}{2\sqrt{2}}.$$

Hence,

$$\delta(x^+y^+,\mu^+) < \frac{1}{\sqrt{2}}.$$

The following theorem gives an upper bound for the total number of iteration for our algorithm.

**Theorem 1.** *Let $\varepsilon > 0$ be an accuracy parameter. If $\delta(x^0y^0,\mu^0) < \frac{1}{\sqrt{2}}$, then the algorithm converges to optimal solution with complexity analysis, namely $O(n^{2.5}L)$ iterations, where $L = \ln \frac{\mu^0}{\varepsilon}$.*

*Proof.* We have

$$\mu^k = (1-\theta)^k\mu^0,$$

thus,

$$(1-\theta)^k\mu^0 \leq \varepsilon.$$

Now taking logarithms of $(1-\theta)^k\mu^0 \leq \varepsilon$, we may write

$$\ln((1-\theta)^k\mu^0) \leq \ln\varepsilon$$

equivalent

$$k \ln(1-\theta) \leq \ln \varepsilon - \ln \mu^0,$$

using the fact that $\log(1-\theta) \leq \theta$, for $0 \leq \theta \leq 1$, then the above inequality holds if

$$k \geq \frac{1}{\theta} \left[ \ln \frac{\mu^0}{\varepsilon} \right].$$

Let $L = \ln \frac{\mu^0}{\varepsilon}$, then at most $k = 2\sqrt{n} \ln \frac{\mu^0}{\varepsilon} = O(\sqrt{n} \ln \frac{\mu^0}{\varepsilon}) = O(\sqrt{n}L)$ iterations in the algorithm, we can obtain $\varepsilon$-solution of (2). However, in every step, the complexity bound of computing the linear system is $O(n^2)$. Therefore, the total complexity bound of the algorithm is $O(n^{2.5}L)$.

## 5 Numerical implementation

In this section, we deal with the numerical implementation of this algorithm applied to some problems of monotone $LCPs$. Here we used $(x^0, y^0)$ to denote the feasible starting point of the algorithm, $\delta(x^0 y^0, \mu^0) < \frac{1}{\sqrt{2}}$, the proximity condition, **Iter** means the iterations number produced by the algorithm and $\mu^*$ denotes the value when the algorithm terminates. The implementation is manipulated in **DEV C++**. Our tolerance is $\varepsilon = 10^{-5}$. For the update parameter we have vary $0 < \theta < 1$. Finally we note that the linear system of Newton in (6) is solved thanks to the Gauss elimination procedure.

**Problem 1.**

$$M = \begin{pmatrix} 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ -2 & -1 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix}, q = \begin{pmatrix} -4 & -5 & 8 & 7 & 3 \end{pmatrix},$$

The feasible starting point is

$x^0 = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \end{pmatrix}, y^0 = \begin{pmatrix} 2 & 3 & 2 & 1 & 1 \end{pmatrix}$

$\delta(x^0 y^0, \mu^0) = 2.517539 > \frac{1}{\sqrt{2}}$, then the classical method is diverge.

The numerical results with this problem are summarized in the table below:

| **Results of the Algorithm** | | | | |
|---|---|---|---|---|
| $\delta(x^0 y^0, \mu^0) = 0.000000 < \frac{1}{\sqrt{2}}$ | | | | |
| $\theta$ | 0.15 | 0.20 | 0.60 | 0.95 |
| $\mu^*$ | 0.000009 | 0.000010 | 0.000008 | 0.000008 |
| **iter** | 87 | 65 | 24 | 19 |

**Problem 2.**

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3\ 0.8 & 0.32 & 1.128 & 0.0512 \\ 0 & 0 & 0 & 0 & 0 & 0\ 1 & 0.8 & 0.32 & 0.128 \\ 0 & 0 & 0 & 0 & 0 & 0\ 0 & 1 & 0.8 & 0.32 \\ 0 & 0 & 0 & 0 & 0 & 0\ 0 & 0 & 1 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0\ 0 & 0 & 0 & 0 \\ -0.8 & -1 & 0 & 0 & 0 & 0\ 0 & 0 & 0 & 0 \\ -0.32 & -0.8 & -1 & 0 & 0 & 0\ 0 & 0 & 0 & 0 \\ -1.128 & -0.32 & -0.8 & -1 & 0 & 0\ 0 & 0 & 0 & 0 \\ -0.0512 & -1.128 & -0.32 & -0.8 & -1 & 0\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$q = \begin{pmatrix} -0.0256 & -0.064 & -0.16 & 5.59 & -1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$,

The feasible starting point is

$x^0 = \begin{pmatrix} 0.18 & 0.18 & 0.18 & 0.18 & 0.25 & 3 & 4 & 5 & 6 & 9 \end{pmatrix}$,

$y^0 = \begin{pmatrix} 21.0032 & 11.008 & 12.52 & 12.8 & 8 & 0.46 & 0.676 & 0.6184 & 0.41536 & 0.336144 \end{pmatrix}$.

$\delta(x^0y^0, \mu^0) = 2.278802 > \frac{1}{\sqrt{2}}$, then the classical method is diverge.

The numerical results with this problem are summarized in the table below:

| **Results of the Algorithm** | | | | |
|---|---|---|---|---|
| $\delta(x^0y^0, \mu^0) = 0.000000 < \frac{1}{\sqrt{2}}$ | | | | |
| $\theta$ | 0.15 | 0.20 | 0.60 | 0.95 |
| $\mu^*$ | 0.000009 | 0.000010 | 0.000008 | 0.000005 |
| **iter** | 84 | 64 | 22 | 16 |

**Problem 3.** Let $M \in \Re^{n\times n}$ and $q \in \Re^n$are given by:

$$M = \begin{pmatrix} 1 & 2 & 2 & \dots & 2 \\ 0 & 1 & 2 & \dots & 2 \\ 0 & 0 & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & 2 \\ 0 & 0 & 0 & . . & 0\ 1 \end{pmatrix}, q = \begin{pmatrix} -1 & \dots & -1 \end{pmatrix},$$

Case 1: $n = 10$.

The feasible starting point is

$x^0 = \begin{pmatrix} 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 1.0009 \end{pmatrix}$,

$y^0 = \begin{pmatrix} 1.0171 & 1.0153 & 1.0135 & 1.0108 & 1.0099 & 1.0081 & 1.0063 & 1.0045 & 1.0027 & 0.0009 \end{pmatrix}$.

$\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{\sqrt{2}}$, then the classical method is converge.

The numerical results with this problem are summarized in the table below:

| **Results of the Algorithm** | | | | |
|---|---|---|---|---|
| $\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{\sqrt{2}}$ | | | | |
| $\theta$ | 0.15 | 0.20 | 0.60 | 0.95 |
| $\mu^*$ | 0.000009 | 0.000009 | 0.000007 | 0.000007 |
| **iter** | 31 | 23 | 9 | 7 |

Case 2: $n = 15$.

The feasible starting point is

$$x^0 = \begin{pmatrix} 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 0.0009 \\ 0.0009\ 0.0009\ 0.0009\ 0.0009\ 1.0009 \end{pmatrix},$$
$$y^0 = \begin{pmatrix} 1.0261\ 1:0243\ 1.0225\ 1.0198\ 1.0189\ 1.0171\ 1.0153\ 1.0135\ 1.0108\ 1.0099 \\ 1.0081\ 1.0063\ 1.0045\ 1.0027\ 0.0009 \end{pmatrix}.$$

$\delta(x^0y^0, \mu^0) = 0.059169 < \frac{1}{\sqrt{2}}$, then the classical method is converge.

The numerical results with this problem are summarized in the table below:

| **Results of the Algorithm** | | | | |
|---|---|---|---|---|
| $\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{\sqrt{2}}$ | | | | |
| $\theta$ | 0.15 | 0.20 | 0.60 | 0.95 |
| $\mu^*$ | 0.000097 | 0.000085 | 0.000060 | 0.000058 |
| **iter** | 15 | 12 | 5 | 4 |

## 6 Conclusion

In this paper, we have proposed a feasible short-step interior point algorithm for solving monotone linear complementarity problem. The algorithm deserves the best wellknown theoretical iteration bound $O(n^{2.5}L)$ when the starting point is $(x^0, y^0)$ is strictly feasible and verified proximity measure condition. This choice of initial point can be done by the technique of Djamel Benterki [**?**]. For the numerical tests we vary the parameter $\theta$, and we note that each problem addressed when the parameter $\theta$ crosses we get the good numerical behavior. Future research might extended the algorithm for other optimization problems.

**Acknowledgements**

The authors thank the referees for their careful reading and their precious comments. Their help is much appreciated.

## References

1. D. Benterki, A. Leulmi. An improving procedure of the interior projective method for linear programming. Applied Mathematics and Computation 199 (2008) 811 − 819
2. N.K. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proceedings of the 16th Annual ACM Symposium on Theory of Computing, 1984, *pp.* 302 − 311.
3. J. Ji, F.A. Potra, S. Huang, Predictor–corrector method for linear complementarity problems with polynomial complexity and superlinear convergence, J. Optim. Theory Appl. 85 (1) (1995) 187 − 199.
4. F.A. Potra, A superlinearly convergent predictor–corrector method for degenerate $LCP$ in a wide neighborhood of the central path with $O(\sqrt{n}L)$ iteration complexity, Math. Program. 100 (2) (2004) 317 − 337.
5. J. Peng, C. Roos, T. Terlaky, Self-regular functions and new search directions for linear and semidefinite optimization, Math. Program. 93 (1) (2002) 129 − 171.
6. J. Peng, C. Roos, T. Terlaky, Self-Regularity. A New Paradigm for Primal–Dual Interior-Point Algorithms, Princeton University Press, 2002.
7. S.J. Wright, Primal-dual Interior-Point Methods, SIAM, Philadelphia, USA, 1997.

# A Piecewise Quadratic Underestimation For Univariate Global Optimization

Aaid Djamel[1] and Noui Amel[2] and Ouanes Mohand[3]

[1] Department of Mathematics, University of Constantine,Algéria
[2] Département de Mathématiques Université de Khenchela,
[3] Département de Mathématiques université de Tizi-ouzou

**Abstract.** A new method for the convex underestimation of univariate functions nonconvex is presented in this article. The method is based on a piecewise function from the work of the authors in [12], which produces a set of convex piecewise underestimator. The resulting convex underestimators are very tight, the enormous advantages resides in the finest possible partitioning of the domain and also the problem of the lower bound uses local optimizers, since it has explicit solutions. The method was applied to a series of test functions presented previously in the literature and the results indicate that the method produces convex uenderestimators high quality in terms of tightening and especially in terms of execution times.

## 1 Introduction

Due to recent theoretical and algorithmic advances, global optimization has found an increased number of applications across many branches of engineering and science. For instance, complex problems, like the ones arising in refinery pooling (Meyer and Floudas 2006), azeotropic distillation (Maranas et al. 1996; Harding et al. 1997) and phase and chemical equilibrium (McDonald and Floudas 1994, 1995, 1997), have all been tackled by global optimization approaches. Furthermore,many interestingmathematical problems (e.g., enclosure of all solutions of systems of nonlinear equations (Maranas and Floudas 1995), parameter estimation in nonlinear algebraic models (Esposito and Floudas 1998), bilevel programming problems (Gümüş and Floudas 2001)) can be expressed with global optimization formulations, something that expands the applicability of the relevant results. The publications by Sherali and Adams (1999), Floudas (2000), Horst and Tuy (2003), Horst et al. (2000), Tawarmalani and Sahinidis (2002a) and Floudas and Pardalos (1995, 2003), as well as the recent reviewpapers by Floudas (2005) andFloudas et al. (2005), provide thorough insight on the current status of the field from both the theoretical and application perspective. In their effort to locate the global solution, deterministic global optimization algorithms, like the áBB (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman et al. 1998a,b), employ a branch and bound framework. During this process, convex underestimation techniques are used to formulate relaxed convex problems that can be solved to optimality with the use of local solvers, thus providing valid

lower bounds for the original problem. The tightness of the underestimators used is of fundamental importance for the computational performance of these algorithms, since a tighter relaxation can lead to faster fathoming and less nodes of the branch and bound tree to be visited (Floudas 2000). As a consequence, a lot of research effort has been focused on finding tight convex underestimators, particularly for functions of some special structure. From the pioneering work of McCormick (1976) and Al-Khayyal and Falk (1983) who introduced the convex and concave envelope of the bilinear term, up to more recent results on the trilinear envelope (Meyer and Floudas 2003, 2004), amultitude of underestimators has been proposed in the literature.These include results on univariate monomials of odd degree (Liberti and Pantelides 2003), multilinear functions (Ryoo and Sahinidis 2001), fractional (Maranas and Floudas 1995; Tawarmalani and Sahinidis 2001, 2002b) and trigonometric terms (Caratzoulas and Floudas 2005). In the case of arbitrary nonconvex functions that do not exhibit an exploitable mathematical structure, the $QBB$ underestimator (Le thi and Ouanes [12] ) can be used:

$$q(s) = L_h f(s) - \frac{1}{2} K h^2, \forall s \in S \tag{1}$$

to solve the problem of global optimization with simple constraints defined as follows:

$$(P) : \alpha = \min f(s),\ s \in S = [a, b]$$

we assume that $f$ is twice differentiable on $S$ on their second derivatives are bounded, ie there are positive numbers $K$

$|f^{'}(s)| \leq K$ for all s ∈ S. Such a $K$ can be defined in several ways in practice: is it possible to know a priori values, or they are estimated in a manner to course

the algorithm. In our approach, $K$ are assumed to be known.

Let $\{s_1, s_2, ..., s_m\}$ be a uniform discretization with mesh size $h$ of $S = [a, b]$ where $s_1 = a$ and $s_m = b$.Let$\{w_1, w_2, ..., w_m\}$ be a finite sequence of functions defined as ([3], [5])

$$\omega_i = \begin{cases} \dfrac{s - s_{i-1}}{s - s_{i-1}} & \text{if } \ s_{i-1} \leq \ s \leq s_i \\ \dfrac{s_{i+1} - s}{s_{i+1} - s_i} & \text{if } \ s_i \leq \ s \leq s_{i-1} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $s_0 = s_1$ and $s_{m+1} = s_m$. We have ([3], [5])

$$\begin{cases} \sum_{i=1}^{m} \omega_i(s) = 1, \forall s \in S \text{ and } \omega_i(s_j) = 0 \ if \ i \neq j \\ 1 \quad \text{otherwise} \end{cases}, \tag{3}$$

Let $L_h f$ be the piecewise linear interpolant to $f$ at points $s_1, ...s_m$([4],[5])

$$L_h f \ = \sum_{i=1}^{m} \omega_i(s) f(s_i). \tag{4}$$

The main advantage of this work is that the problem of lower bound has a one explicit optimal solution and does not require the use of local solvers.

Our contribution consists in detecting a better lower bound in a short period of time, while preserving the advantage of the explicit solution.

The method is presented in detail for the case of univariate functions, where it can be directly applied. Theoretical and algorithmic extensions of the method for application on multivariate functions have also been developed.

## 2 Tightness of proposed underestimation

Let $f(s)$ be a univariate function that needs to be underestimated in $S = [a, b]$.We select an integer $N > 1$ and partition the complete domain in $N$ segments of equal length.Thus, the $i$ the subdomain would be defined as $D_{i+1} = [s_i, s_{i+1}]$, where: $s_i = a + \left(\frac{b-a}{N}\right) i$ , $i = 0, ..., N-1$. For every subdomain $D_{i+1}, i = 0, ..., N-1$, we construct the corresponding $QBB$ underestimator:

$$\begin{cases} P_{i+1}(s) = K_{i+1} \dfrac{(s - s_{i+1})(s - s_i)}{2} + L_{i+1}(s), \\ \text{with } L_{i+1}(s) = \dfrac{s - s_{i+1}}{s_{i+1} - s_i} f(s_{i+1}) + \dfrac{s - s_{i+1}}{s_i - s_{i+1}} f(s_i) \ , \\ \textit{such as } K \geq K_{i+1} \geq |f''(s)| \end{cases} \tag{5}$$

where $K_{i+1}$is a upper bound of the second derivative that is valid for the entire subdomain $D_{i+1}$.Note that although an underestimator $P_{i+1}(s)$ can be defined outside its respective subdomain, its convexity is only guaranteed for $s \in [s_i, s_{i+1}]$.

That is to say, in each of our subdomain must determine the lower bound of quadratic underestimiator.

In order to detect the best lower bound we compare all lower bounds and maintaining the smallest as follows:

For all $s \in [s_i, s_{i+1}]$, $i = 0, ..., N-1$, the lower bounds are computed explicitly

$$s^*_{i+1} = \begin{cases} \mu = \dfrac{1}{2}(s_i + s_{i+1}) - \dfrac{1}{K_{i+1}}(f(s_{i+1}) - f(s_i)) \ if \ \mu \in [s_i, s_{i+1}] \\ \text{if } \mu \leq s_i \\ \text{if } \mu \geq s_{i+1} \end{cases}, \tag{6}$$

Now, we solve the problem:

$$s^* = \min_i P_{i+1}(s^*_{i+1}),\ i = 0, ..., N-1 \tag{7}$$

Even now, we divide the original domain into two subdomains can not be necessarily equal it depends on the position of $s^*$ it is called $\omega-$subdivision, every one in the two subdomains, we will take the same steps performed for the initial domain. This time, we get four intervals, then the selection criteria and other rules of eliminations will be applied, this process is repeated until an optimal solution is found or all intervals are explored in this case we keep the best solution found.

Practically the different stages of solving the problem are summarized in brunch and bound algorithm which will be explained in the next section.

**Theorem 1.** *We define $P(s)$, $s \in [a, b]$ to be the following piecewise function:*

$$\begin{cases} P(s) = P_{i+1}(s),\ if\ s_i \leq s \leq s_{i+1} \\ \qquad i = 0, ..., N-1 \end{cases}, \tag{8}$$

*this function is a piecewise convex valid underestimator of $f(s)$ for all $s \in [a, b]$.*

**Theorem 2.** *$P(s)$ is tighter than the underestimator $q(x)$ introduced in [12].*

### 2.1 Branch and bound algorithm

Denote by, $LB_k$, $UB_k$ and $s_k$ respectively the best lower bound, the best upper bound of $\alpha$ and the best solution to $(P)$ at iteration $k$.

**The Proposed Algorithm**

**Step 1 : Initialization**

a) $\varepsilon$ a given accuracy, an integer $N > 1$, $K_{i+1}$ constants.

b) Set $k := 0, T^0 = [a, b], M := \{T^0\}$,

c) Determine $s^*_0$ and $LB_0$ using (6)(7)

d)Set $UB_0 = \min\{f(a), f(b), f(s^*_{i+1})\},\ i = 0, ..., N-1$

**Step 2 : Iteration**

**While** $UB_k - LB_k > \varepsilon$ **do**

**1)** Let $T^k = [a_k, b_k] \in M$ be the interval such that $LB_k = LB(T^k)$.

**2)** Bisect $T^k$ into two intervals by $\omega$-subdivision procedure:

$$T^k_1 = [a_k, s^*_k] = [a^1_k, b^1_k], T^k_2 = [s^*_k, b_k] = [a^2_k, b^2_k].$$

**3) For j = 1, 2 do**

- Compute $s^*_{k,j}$ and $LB(T^k)$ using (6)(7).

- **If** $s^*_{k,j} \notin ]a^j_k, b^j_k[$ then update $LB(T^k_j) = UB(T^k_j) = \min\{f(a^j_k), f(b^j_k)\}$.

-To fit in to $M$ the intervals $T^k_j$ :

$$M \leftarrow \ M \cup \left\{T^k_j : UB_k - LB(T^k_j) \geq \varepsilon, j = 1, 2\right\} \setminus \{T^k\}$$

- Update $UB_k = \min\left\{UB_k, f(a^j_k), f(b^j_k), f(s^*_{i+1})\right\}, i = 0, ..., N-1.$

**4)** Update $LB_k = \min\ \{LB(T) : T \in M\}$

**5)** Delete from $M$ all interval $T$ such that $LB(T) > UB_k - \varepsilon$.

**6)** Set $k \leftarrow k + 1$.

**End while**

**7) STOP:** $s_k$ is an $\varepsilon-$optimal solution to $(P)$.

## 3 Computational results

Our programs are written in $C$. The Numerical tests are made with an accuracy $\varepsilon = 10^3$ : By various types of objective functions we compare the performance of the proposed method knowing that $N = 2$ with B & B reported in [12] which was compared with several standard methods, it turned out better for more details see [12].

Table 01:Description of the test functions are taken from [12] [3], $n$: function's IDS :the search interval, LM : the number of local minimizers, GM the number of global mimizers.Table 1 presents the results for various levels of partitioning. The original QBB method corresponds to no partitioning ($N = 1$)

**Table 1**

| $N$ | Function $f(x)$ | $S$ | $LM$ | $GM$ |
|---|---|---|---|---|
| 01 | $\sum_{k=1}^{5} -\cos[(k+1)x] + 4$ | [0.2,7.0] | 7 | 1 |
| 02 | $(3x - 1.4)\sin(18x) + 1.7$ | [0.2,7.0] | 21 | 1 |
| 03 | $x + \sin(5x)$ | [0.2,7.0] | 7 | 1 |
| 04 | $-x - \sin(3x) + 1.6$ | [0.2,7.0] | 4 | 1 |
| 05 | $x\sin(x) + \sin(\frac{10x}{3}) + \ln(x) - 0.84x$ | [2.7,7.5] | 3 | 1 |
| 06 | $-x + \sin(3x) + 1$ | [0.2,7.0] | 5 | 1 |
| 07 | $x\sin(x) + \sin(\frac{10x}{3}) + \ln(x) - 0.84x + 1.3$ | [0.2,7.0] | 4 | 1 |
| 08 | $\sin(x) + \sin(\frac{2x}{3})$ | [3.1,20.4 ] | 3 | 1 |
| 09 | $\sum_{k=0}^{5} k\cos[(k+1)x + k] + 12$ | [0.2,7.0] | 8 | 1 |
| 10 | $-\sum_{k=1}^{5} \sin[(k+1)x + k] + 3$ | [0.2,7] | 7 | 1 |
| 11 | $x^2 - \cos(18x)$ | [-5,5] | 29 | 1 |
| 12 | $(\frac{x^2}{20}) - \cos(x) + 2$ | [-20,20] | 7 | 1 |
| 13 | $2\cos(x) + \cos(2x) + 5$ | [0.2,7] | 3 | 2 |
| 14 | $\sin(x)\cos(x) - 1.5\sin^2(x) + 1.2$ | [0.2,7] | 3 | 2 |
| 15 | $\sin(x)$ | [0,20] | 4 | 3 |
| 16 | $-\sum_{k=1}^{5} \sin[(k+1)x + k]$ | [-10,10] | 20 | 3 |
| 17 | $x^4 - 12x^3 + 47x^2 - 60x - 20\exp(-x)$ | [-1,7] | 1 | 1 |
| 18 | $x^6 - 15x^4 + 27x^2 + 250$ | [-4,4] | 2 | 2 |
| 19 | $x^4 - 10x^3 + 35x^2 - 50x + 24$ | [-10,20] | 2 | 2 |
| 20 | $24x^2 142x^3 + 303x^2 - 276x + 3$ | [0,3] | 1 | 1 |
| 21 | $4x^2 - 4x^3 + x^4$ | [-5,5] | 2 | 2 |
| 22 | $1.75x^2 - 1.05x^4 + \frac{1}{6}x^6$ | [-5,5] | 3 | 1 |
| 23 | $x^6 - 15x^4 + 27x^2 + 250$ | [-5,5] | 3 | 2 |
| 24 | $x^4 - 3x^3 - 1.5x^2 + 10s$ | [-5,5] | 2 | 1 |
| 25 | $\begin{pmatrix} 89248 \times 10^{-6}x \ -218343 \times 10^{-2}x^2 \\ +0.998266x^3 - 1.6995x^4 + 0.2x^5 \end{pmatrix}$ | [0,10] | 2 | 1 |

table 02 ; show the computational results.GO $f^*$ : the value of global minimum (rounded to three decimal digits), $E\%$ : the ability to eliminate the unnecessary rectangles.

$$E\% = \frac{\text{the number of rectangles eliminated}}{\text{the total number of rectangles}} \times 100,$$

iter: number of iterations.

| $n$ | GO $f^*$ | | iter | | $_{E\ \%}$ | |
|---|---|---|---|---|---|---|
| | $N=1$ | $N=2$ | $N=1$ | $N=2$ | $N=1$ | $N=2$ |
| 1 | -1 | -0.983 | 18 | 09 | 5.4% | 10.53% |
| 2 | -17.583 | -17.583 | 65 | 33 | 3.82% | 4.75% |
| 3 | -0.077 | -0.077 | 10 | 08 | 9.52% | 17.65% |
| 4 | -6.263 | -6.263 | 04 | 03 | 0% | 14.29% |
| 5 | -4.601 | -4.601 | 07 | 05 | 6.67% | 9.09% |
| 6 | -6.263 | -6.263 | 05 | 04 | 0% | 22.22% |
| 7 | -3.101 | -3.101 | 11 | 07 | 13.04% | 26.67% |
| 8 | -1.906 | -1.905 | 09 | 04 | 5.26% | 22.22% |
| 9 | -0.871 | -0.871 | 18 | 13 | 2.70% | 3.70% |
| 10 | -11.508 | -11.508 | 18 | 12 | 2.70% | 12% |
| 11 | -1 | -1 | 39 | 36 | 0% | 9.59% |
| 12 | 1 | 1 | 15 | 15 | 0% | 12.90% |
| 13 | 3.5 | 3.5 | 12 | 08 | 0% | 0% |
| 14 | -0.451 | -0.451 | 10 | 08 | 10.63% | 11.76% |
| 15 | -1 | -1 | 14 | 09 | 10.34% | 5.26% |
| 16 | -3.373 | -3.373 | 72 | 27 | 4.83% | 3.64% |
| 17 | -32.781 | -32.778 | 23 | 07 | 4.25% | 6.67% |
| 18 | -7 | -7 | 51 | 31 | 0% | 14.30% |
| 19 | -1 | -1 | 228 | 161 | 0% | 4.02% |
| 20 | -89 | -89 | 36 | 25 | 0% | 1.96% |
| 21 | 0 | 0 | 115 | 38 | 0% | 1.30% |
| 22 | 0 | 0 | 291 | 84 | 0% | 4.14% |
| 23 | -7 | -7 | 55 | 32 | 0% | 10.77% |
| 24 | -7.5 | -7.495 | 43 | 8 | 0% | 5.88% |
| 25 | -443.672 | -443.672 | 37 | 21 | 1.33% | 9.30% |
| Total | | | 1206 | 608 | 80.49 | 244.61 |

### 3.1 comment

As it has been described in Sect. 2, the underestimator should become tighter with doubling of the number of subdomains used and all the results are indeed consistent with this. All runs were performed on a 3.20GHz Intel(R) Pentium(R) 4 processor with 1Gb of RAM. Computations were very fast, in the order of a few hundredths of a second

Certain criteria such as the number of evaluation functions are considered as factors that affect others more general, more reliable and useful to use.

That is why, in our comparison tables we are interested in execution time and the percentage removal of unnecessary intervals.

## 4 Conclusion

Our contribution to intervene to improve the quality of lower bounds which leads to a certain speed in the resolution confirmed by various numerical tests we conducted.

Our proposed approach also uses a hybrid subdivision:

the multi-regular subdivision with a partial exploration is used to detect a better lower bound.

-w-subdivision with total exploration and elimination of certain intervals in seeking global optimum.

This technique can be adapted to the more general case and without loss of benefits required.

## References


1. Basso, P.: Iterative method for localization of the global maximum. SIAM J. Num. Anal. 19, 781–792 (1982)
2. Breiman, L., Culter, A.: A derterministic alogorithm for global optimization. Math. Program. 58, 179–199 (1993)
3. Casado.L.G. , MartÍnez.J.A. ,GarcÍa .I. and Sergeyev.Ya.D: New Interval Analysis Support Functions Using Gradient Information in a Global Minimization Algorithm, Journal of Global Optimization 25: 345–362, 2003.
4. Ciarlet.P.G. , The Finite Element Method for Elliptic Problems Studies in Math.and its Appl., 1979.
5. De Boor.C, A Practical Guide to Splines, Applied Mathematical Sciences, Springer Verlag, 1978.
6. Falk J. E and Soland R. M., An algorithm for separable nonconvex programming problems, Management Science, 15 (1969), pp. 550-569.
7. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: advances and challenges. Comp. Chem. Engng. 29, 1185–1202 (2005)
8. Gergel, V.P., Sergeyev, Y.D.: Sequential and parallel algorithms for global minimizing functions with Lipschitzian dervatives, Comput. Math. Appl. 37, 163–179 (1999)
9. Hansen, P., Jaumard, B., Lu, S.-H.: Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison. Math. program. 55, 273–292 (1992a)
10. Hansen, P., Jaumard, B.: Lipschitz optimization. In: Horst, R., Pardalos, M.P. (eds.) Handbook of Global Optimization. Kluwer Academic Publishers, The Netherlands (1995)
11. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer-Verlag, Berlin (1996)
12. Le Thi Hoai An and Pham Dinh Tao, A Branch-and-Bound method via D.C. Optimization Algorithm and Ellipsoidal technique for Box Constrained Nonconvex Quadratic Programming Problems, Journal of Global Optimization, 13 (1998), pp. 171-206.

13. Le Thi.H.A, Ouanes.M. , Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint, RAIRO Operations Research 40(2006) 285-302.
14. MacLagan, D., Sturge, T., Baritompa, W.P.: Equivalent methods for global optimization. In: Floudas,C.A., Pardalos, P.M. (eds.) State of the Art in Global Optimization. Kluwer Academic Publishers, Dordrecht (1996)
15. Maranas, C.D., Floudas, C.A.: A global optimization approach for Lennard-Jones microclusters. J. Chem. Phys. 97, 7667–7677 (1992)
16. Mayne,D.Q., Polak, E.: Outer approximation algorithm for non-differentiable optimization problems, J. Optim. Theory Appl. 42, 19–30 (1984)
17. Pijavskii, S.A.: An algorithm for finding the absolute extremum of a function. USSR Comput. Math. Math. Phys. 12, 57–67 (1972)
18. Sergeyev, Y.D.: Global one-dimensional optimization using smooth auxiliary functions. Math. Program. 81, 127–146 (1998)
19. Thai Quynh Phong, Le Thi Hoai An and Pham Dinh Tao, On the global solution of linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method. RAIRO, Recherche Op´erationnelle, 30 (1) (1996), 31-49.

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

# A Combined methods for Multiple Objective Integer Linear Programming

Boualem BRAHMI[1], Zoubir RAMDANI[2], and Djamal CHAABANE[3]

USTHB, Faculty of Mathematics, Department of Operations Research, Bab-Ezzouar, BP32 El-Alia, 16311 Algiers, Algeria

[1]boualemmg@yahoo.fr

**Abstract.** In this paper, an algorithm for enumerating all non-dominated vectors of multiple objective integer linear programs is presented. Starting from an initial non-dominated vector, at each iteration, the procedure determines a new solution by solving a constrained weighted Tchebycheff program. Progressively more constraints are added to this program in order to reduce the admissible research set.

**Keywords:** Multiple objective integer programming; Tchebycheff norm; Branch and Bound.

## 1 Introduction

Multiple objective integer linear programming (MOILP) is very useful for many areas of application as any model that incorporates discrete phenomena requires the consideration of integer variables (such as, for modeling investment choices, production levels, fixed charges, logical conditions or disjunctive constraints).

Over the last decades, several methods have been developed to solve MOILP problems, some methods require the presence of human decision maker (DM) (interactive) and generate only a subset of nondominated vectors, and other methods consist in enumerating all nondominated vectors without intervention of DM. In general, the approaches can be classified as exact or heuristic and grouped according to the methodological concepts they use. Among others,the concepts employed in exact algorithms include branch-and-bound Ramesh, Karwan and Zionts [16]; Marcotte and Soland [11], dynamic programming Villarreal and Karwan [15], implicit enumeration Klein and Hannan [6]; Sylva and Crema [4], [7], reference directions Karaivanova et al [17], weighted norms Eswaran, Ravindran and Moskowitz [8]; Steuer and Choo [9]; Ted and al [5]; weighted sums with additional constraints Chalmet, Lemonidis and Elzinga [12]; Ferreira, Climaco and Paixão [19], and 0-1 programmation Bitran [13]. Heuristic approaches, such as simulated annealing, tabu search, and evolutionary algorithms, have been proposed for multiobjective integer programs with an underlying combinatorial structure Ehrgott and Gandibleux [20]. Several survey articles have already been published in this area. Teghem and Kunsch (1986b)

presented a survey of interactive methods for multiobjective integer and mixed-integer linear programming, a brief overview of MOILP approaches can be found in Alves and Clímaco [22].

The algorithm presented in this work based on a parameterized exploration of the outcome space and reduce the problem of finding the set of nondominated objective vectors to that of solving a sequence of single-objective mixed-integer programs. The main idea is to use the Weighted Tchebycheff Program (WTP) for identifying the nondominated objective vectors. As known, WTP program is a mixed-integer linear program (MILP) which can be examined using standard integer-linear programming techniques such as Branch-and-Bound. Thus, it may yield several optimal solutions which some can be nondominated or weakly nondominated by others. In order to avoid the delicate situation lies this norm and the weakly nondominated vectors, we propose to modify the program WTP by adding some constraints. This technique of additional constraints known in the literature as " Corner constraints" is developed by Klein Hannan [6], also used by Sylva Crema [4].

The organization of the work is as follows: Section 2 briefly reviews basic definitions, results and foundations of Tchebycheff norms. The algorithm is presented in Section 3 and a number of propositions are provided to support finiteness and convergence properties. Some considerations concerning the implementation of the algorithm and an illustrative example are also provided.

## 2 Basic results and Tchebycheff metrics

The MOILP problem under consideration has the following form:

$$(P) \quad Vmax\{Cx, \quad x \in D\} \tag{1}$$

Where $D = S \cap \mathbb{Z}$ with $S = \{x \in \mathbb{R}^n \mid Ax \leq b;\ x \geq 0\}$ is nonempty bounded set; $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $C = (c^i)_{i \in \{1,\cdots,p\}} \in \mathbb{Z}^{p \times n}$ and $p \geq 2$.

Unlike single-objective problems, the resolution of multiple criteria problems imposes a set of feasible solutions, using the property that no improvement on any criterion is possible without sacrificing on at least one other criterion. These solutions are called efficient solutions or nondominated solutions, which are defined as follows:

A feasible solution $\hat{x} \in D$ is said to be an *efficient solution* of MOILP if and only if, there is no feasible solution $x \in D$ such that $Cx \geq C\widehat{x}$ and $Cx \neq C\widehat{x}$ ($c^i x \geq c^i \hat{x}$ for all $i = 1, ..., p$ and $c^i x > c^i \hat{x}$ for at least one $i$). The point $\hat{z} = C\hat{x}$ is then called *nondominated vector*. Otherwise, $\hat{x}$ is not efficient and $\hat{z} = C\hat{x}$ is said to be dominated by $z = Cx$.

$\hat{x} \in D$ is called weakly efficient if there is no $x \in D$ such that $Cx > C\hat{x}$, i.e. $c^i x > c^i \hat{x}$ for all $i = 1, ..., p$. The point $\hat{z} = C\hat{x}$ is then called weakly nondominated objective vector.

Since the feasible region of $P$ is nonconvex, unsupported nondominated solutions may exist. A nondominated point $z \in Z$ is called unsupported if it

is dominated by a convex combination (which does not belong to Z) of other nondominated objective vector (belonging to Z).

$Z(P)$ will be used henceforth to denote, the set of all nondominated objective solutions of $(P)$, $E(P)$ denotes a subset of efficient solutions of $(P)$.

The ranges of the nondominated objective vectors in the outcome space provide valuable information about the problem MOILP considered if the objective functions are bounded over the feasible region. Upper bounds of the nondominated solutions set are available in the ideal objective vector $z^\star \in \mathbb{R}^p$. Its components $z_i^\star$ are obtained by maximizing each of the objective functions individually subject to the feasible region $D$. A vector strictly better than $z^\star$ can be called a utopian objective vector $z^{\star\star}$. In this work, we use the utopian and not the ideal objective values in order to avoid dividing by zero in all occasions. Since, the components of the matrix $C$ are assumed integer, then we can set $z^{\star\star} = z^\star + 1$.

The Tchebycheff theory, whose foundation originated from Bowman [3], has been successfully exploited within the scope of interactive algorithms for multiple objective optimization in Steuer and Choo [9] and since, the scalarization techniques based on Tchebycheff norms was intensively used to solve multiple objective programming problem involving discrete decisions. However, Bowman [3] proved that the Tchebycheff scalarization norms is appropriate for generating the nondominated objective vectors set, in particular those which are unsupported(see for exemple [2]).

We denote by $\Delta$ the weighting vectors space defined as

$$\beta \in \Delta = \left\{ \beta \in \mathbb{R}^p \,|\, 0 < \beta_i < 1, \sum_{i=1}^{p} \beta^i = 1 \right\}$$

Given a point $z \in Z$, the weighted Tchebycheff norm of $z$ in $\mathbb{R}^p$ according to $z^{\star\star}$ is defined as

$$\|z^{\star\star} - z\|^\beta = \max_{i=1,\ldots,p} \{\beta_i | z_i^{\star\star} - z_i|\}\,. \tag{2}$$

Here $\beta \in \Delta$ represents its weighted vector which can be calculated as follows

$$\beta_i = \frac{1}{z_i^{\star\star} - z_i} \left[ \sum_{i=1}^{p} \frac{1}{z_i^{\star\star} - z_i} \right]^{-1} \quad \forall\ 1 \leq i \leq p. \tag{3}$$

The aim for introducing this norm is to measure the distance between any $z$ and the utopian objective vector $z^{\star\star}$. Therefore, this technique consists in selecting the feasible objective vectors with minimum weigh distance from $z^{\star\star}$. In others words, for a given $\beta$, to reach this goal one has to solve the so-called *minimization of the norm* problem defined as follows

$$\min_{z \in Z} \{||z^{\star\star} - z||_\beta\} \tag{4}$$

Bowman [3] has proposed to solve an equivalent problem called *weighted Tchebycheff program* defined as follows

$$P(\beta)\begin{cases}\min \quad \omega \\ \omega \geq \beta_i(z_i^{\star\star} - z_i)\ , 1 \leq i \leq p; \\ z_i = c^i x; \\ x \in D; \\ \omega \geq 0.\end{cases} \tag{5}$$

Problem $P(\beta)$ is a mixed-integer linear program (MILP) which can be examined using standard integer-linear programming techniques such as Branch and Bound. However, $P(\beta)$ may yield several optimal solutions of which some can be nondominated or weakly nondominated by others. We have the following results.

**Theorem 1.** *[10] Let $Z$ be finite and*

$$M = \{z \in Z \mid (x, z, \omega) \text{ is a minimal solution of } P(\beta) \text{ for some } \beta \in \Delta\}$$

*Then there exists $\bar{z} \in M$ such that $\bar{z} \in Z(P)$.*

**Theorem 2.** *[3] $z = C\hat{x}, \hat{x} \in D$ is nondominated solution for MOILP only if it is a solution to $P(\beta)$ for some $\beta$.*

Eswaran et al [8], Ted et al [5] developed two algorithms based on solving $P(\beta)$ for enumerating all nondominated vectors of MOILP but solely with two objectives where the technique of comparison is used to eliminate the weakly nondominated solutions. For a problem having more than two objective functions this technique is not appropriate.

In this work, we propose to solve the weighted Tchebycheff program augmented by adding some constraints in order to avoid the trap related by the weighted norm and the weakly nondominated vectors. The main idea of our technique consists in moving from a nondominated solution to another adjacent solution by solving the modified Tchebycheff program according to the weighted vector which is obtained from some nondominated vectors. The technique of additional constraints, firstly developed by Klein and Hannan [6] and also used by Sylva and Crema [4], consists in reducing progressively the admissible space and eliminating the nondominated solution previously found.

**Proposition 1.** *[4] Let $\hat{x}$ be efficient solution to problem $(P)$ and $D_s = \{x \mid x \in \mathbb{Z}_+^n, Cx \leq C\hat{x}^s\}$. Let $\hat{x}^*$ be an efficient solution to the multiple objective integer problem "* $\max$*"$\{Cx, x \in D - D_s\}$. Then, $\hat{x}^*$ is an efficient solution to problem $(P)$.*

**Proposition 2.** *Let $\hat{z} = C\hat{x}$ be nondominated solution to problem $(P)$ and $D_s = \{x \mid x \in \mathbb{Z}_+^n, Cx \leq C\hat{x}^s\}$ and $\hat{\beta}$ his weight vector defined as in formula (3). If $\bar{z}$*

*is an optimal solution to problem $P(\hat{\beta})$ such that*

$$P(\hat{\beta}) \begin{cases} \min \quad \omega \\ \omega \geq \hat{\beta}_i \left(z_i^{\star\star} - z_i\right), 1 \leq i \leq p \\ z_i = c^i x, 1 \leq i \leq p \\ x \in D - D_s; \\ \omega \geq 0. \end{cases}$$

*Then $\bar{z}$ is nondominated objective solution to problem $P$.*

*Proof.* Let as suppose there exists an efficient solution $x' \in D$ such that $C\bar{x} \leq Cx'$ with at least one strict inequality. $x'$ cannot belong to $D_s$, $Cx'$ is not dominated by $\hat{z}$. However, for all $i, (z_i^{\star\star} - C^i x') \leq (z_i^{\star\star} - C^i \bar{x})$, since $\forall z \in Z, z < z^{\star\star}$ then for all $i \;\; \hat{\beta}_i > 0$, thus

$$\hat{\beta}_i(z_i^{\star\star} - C^i x') \leq \hat{\beta}_i(z_i^{\star\star} - C^i \bar{x}), \; \forall i$$

we must have

$$\max_{i=1,\ldots,p} \hat{\beta}_i(z_i^{\star\star} - C^i x') \leq \max_{i=1,\ldots,p} \hat{\beta}_i(z_i^{\star\star} - C^i \bar{x}) \tag{6}$$

The substitutions $z' = Cx'$ and $\bar{z} = C\bar{x}$ in each of the $p$ constraints of $P(\hat{\beta})$, we get

$$\bar{\omega} \geq \hat{\beta}_i(z_i^{\star\star} - C^i \bar{x}), \; \forall i$$
$$\omega' \geq \hat{\beta}_i(z_i^{\star\star} - C^i x'), \; \forall i$$

According to the definition of the weighted Tchebycheff norm and the inequality (6) , we should have

$$\omega' \leq \bar{\omega}$$

Two cases are to be discussed: If $\omega' < \bar{\omega}$ then the optimality of $\bar{z}$ is not preserved. Otherwise, contradiction with non efficiency of $\bar{x}$.

## 3 Algorithm

The algorithm we proposed here is proved to enumerate all nondominated objective vectors for the problem MOILP. After having calculated the ideal objective vector $z^\star$, we can set the utopian objective value as $z^{\star\star} = z^\star + 1$. The procedure starts with an initial nondominated solution $\hat{z}^0$ which can be calculated by solving the parametric problem defined as $P(\lambda) \equiv \max\{\lambda Cx, x \in D\}$ for an arbitrary $\lambda \in \Delta$. Consider $\beta^0 = \lambda$ as an initial weighted vector for the iterative procedure, at each iteration $k$, the weighted Tchebycheff program $P(\beta^{(k-1)})$ is solved in the reduced space $D - D_s$ such that $D_s = \{x, x \in \mathbb{Z}^n, Cx \leq C\hat{x}^s, s = 1, ..., k-1\}$. If $P(\beta^{(k-1)})$ is feasible then, according to proposition (2) a new efficient/nondominated solution $(\hat{x}^k, \hat{z}^k)$ is obtained in the neighborhood of the

last nondominated solution found. The current associated weighted vector $\beta^k$ can be calculated as in (3) and considered for the next iteration. Otherwise, the process ends with all nondominated objective vectors.

Mathematically, the technique of additional constraints can be formulated as

$$D - \bigcup_{s=1}^{k} D_s = \left\{ \begin{array}{ll} c^i x \geq (c^i \hat{x}^s + 1) y_i^s + M_i(1 - y_i^s), & \text{i=1,2,...,p; s=1,2,...,k ;} \\ \sum_{i=1}^{p} y_i^s \geq 1; & \\ y_i^s \in \{0,1\} & \text{i=1,2,...,p; s=1,2,...,k;} \\ x \in D & \end{array} \right\}$$

where $M_i$ is a lower bound for any feasible value of the *ith* objective function such that, if $c_j^i \geq 0, j = 1, ..., n$ $M_i = \min\{c^i x \mid x \in D\}$, otherwise $M_i = 0$. The variables $y_i^s$, $i = 1, ..., p$ associated to $\hat{x}^s$ and additional constraints are added in order to impose an improvement on at least objective function. Note that when $y_i^s = 0$, the associated constraint is redundant and when $y_i^s = 1$, an strict improvement is forced in the *ith* objective function evaluated in $\hat{x}^s$.

**Proposition 3.** *[4] Let $\hat{x}^1, \hat{x}^2, ..., \hat{x}^k$ be efficient solutions to problem MOILP and $D_s = \{x \mid x \in \mathbb{Z}_+^n, Cx \leq C\hat{x}^s\}$. Let $\hat{x}^*$ be an efficient solution to the multiple objective integer problem $(P_k) \equiv$ " max "$\{Cx, x \in D - \bigcup_{s=1}^{k} D_s\}$ Then, $\hat{x}^*$ is an efficient solution to problem MOILP. Furthermore, if $(P_k)$ is unfeasible then $\{C\hat{x}^s\}_{s=1}^k$ is the entire set of nondominated objective vectors for MOILP.*

**Proposition 4.** *Let $\hat{x}^1, \hat{x}^2, ..., \hat{x}^k$ be efficient solutions to problem $(P)$ and $D_s = \{x \mid x \in \mathbb{Z}_+^n, Cx \leq C\hat{x}^s\}$. Let $\hat{\beta}^k$ the weighted vector of $\hat{z} = C\hat{x}^k$, if $P(\hat{\beta}^k)$ such that*

$$P(\beta^\star) \left\{ \begin{array}{l} Min \quad \omega \\ \\ \omega \geq \beta_i^\star (z_i^{\star\star} - c^i x), 1 \leq i \leq p \\ z_i = c^i x 1 \leq i \leq p \\ x \in D - \cup_{s=1}^{k} D_s \\ \omega \geq 0 \end{array} \right.$$

*is unfeasible then $\{C\hat{x}\}_{s=1}^k$ is the entire set of nondominated objective vectors for MOILP.*

*Proof.* According to Proposition (2), for any weighted vector $\hat{\beta}$ of nondominated objective vector, the weighted Tchebycheff program $P(\hat{\beta})$ admits one optimal solution wish is nondominated. If for some $\hat{\beta}^k$, $P(\hat{\beta}^k)$ is unfeasible then $D \subseteq \cup_{s=1}^{k} D_s$, for any $x \in D$ there exists $x^s \in \cup_{s=1}^{k} D_s$ such that $Cx \leq Cx^s$, we must have that $Cx = Cx^s$ and $Cx \in \{Cx^s\}_{s=1}^k$ or $Cx \leq Cx^s$ with at least one strict inequality (and $Cx$ is a dominated vector).

A Constrained Weighted Tchebycheff Program for MOILP

**Input**

$\downarrow A_{(m\times n)}$: matrix of constraints;
$\downarrow b_{(m\times 1)}$: RHS vector;
$\downarrow C_{(p\times n)}$: matrix of criteria;

**Output**

$\uparrow Z(p)$: The entire of nondominated vector solution.
$\uparrow E(p)$: Subset of efficient solutions.

**Initialization**

- Let $z^{\star\star} = z_i^{\star} + 1$ the utopian vector such that $z_i^{\star} = \max\{c^i x, x \in D, i = 1, ..., p\}$ and for the lower bounds where $\forall i = 1, ..., p$ $M_i = \min\{c^i x \mid x \in D\}$,if $c_j^i \geq 0$,$j = 1, ..., n$ else set $M_i = 0$.
- Let $(\hat{x}^1, \hat{z}^1)$ the efficient/nondominated objective vector solution of $\max\{\sum_{i=1}^{p} \frac{1}{2} c^i x, x \in D, i = 1, ..., p\}$.
- $k = 1$, Compute $\hat{\beta}^1$ of $\hat{z}^1$ defined as in (3).
- Let $\beta^{\star} = \hat{\beta}^1$, $E_k = \{\hat{x}^1\}$ and $Z(P)_k = \{\hat{z}^1\}$.

***Repeat***

- Solve $P(\beta^{\star})$ in $D - \bigcup_{s=1}^{k} D_s$ where $D_s = \{x \mid x \in \mathbb{Z}_+^n, Cx \leq C\widehat{x}^s\}$ and $\widehat{x}^s \in E_k$.
  - If $P(\beta^{\star})$ is unfeasible, Stop;
  - Otherwise, Let be $(\hat{x}^k, \hat{z}^k)$ its optimal solution compute $\hat{\beta}^k$ of $\hat{z}^k$
- Set $E_{k+1} = E_k \bigcup \{\hat{x}^k\}$, $Z(P)_{k+1} = Z(P)_k \bigcup \{\hat{z}^k\}$,$k = k+1$ and $\beta^{\star} = \widehat{\beta}^k$.

***Until*** $P(\beta^{\star})$ is unfeasible

## 4 Numerical example

Let us consider the MOILP problem

$$\begin{cases} \text{" max"} & x_1 + x_2 \\ & x_1 - x_2 \\ \text{S.C} & 3x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0 \text{ and integer.} \end{cases} \tag{7}$$

This example contains 5 efficient solutions/nondominatined objective vectors where $(4; -4)$ is unsupported (see Fig.1). The ideal vector is $z^{\star} = (5; 1)$ and $z^{\star\star} = (6; 2)$, the lower bounds of the corresponding objective functions are $M_1 = 0$,$M_2 = 5$. To obtain an initial nondominated solution, we can take the optimal solution of parametric problem $P(\frac{1}{2}; \frac{1}{2}) \equiv \max\{\frac{1}{2}c^1 x + \frac{1}{2}c^2 x, 3x_1 + x_2 \leq 5, x_1, x_2 \geq 0 \text{ and integer}\}$ which is $(\hat{x}^0, \hat{z}^0) = ((1; 2), (3; -1))$ and $\hat{\beta} = (\frac{1}{2}; \frac{1}{2})$.

Let $E_0 = \{\hat{x}^0\}$, $Z(P)_0 = \{\hat{z}^0\}$ and $\beta^\star = \hat{\beta} = (\frac{1}{2}; \frac{1}{2})$
**Iteration 1**: Solve $P(\beta^\star)$ in the reduced space by adding the additional constraints in order to eliminate the point $(\hat{x}^0, \hat{z}^0)$

$$P(\beta^\star)\begin{cases} Min \quad \omega \\ \\ \omega \geq \frac{1}{2}(6 - x_1 - x_2) \\ \omega \geq \frac{1}{2}(2 - x_1 + x_2) \\ \omega \geq 0 \\ D - D_1 \begin{cases} 3x_1 + x_2 \leq 5 \\ x_1, x_2 \quad \geq 0 \\ x_1 + x_2 \quad \geq (3+1)y_1^1 \\ x_1 - x_2 \quad \geq (-1+1)y_2^1 - 5(1-y_2^1) \\ y_1^1 + y_2^1 \quad \geq 1 \\ y_1^1, y_2^1 \quad \in \{0,1\} \end{cases} \end{cases}$$

$(\hat{x}^1, \hat{z}^1) = \{(1;1), (2;0)\}$ and $y_1^1 = 0, y_2^1 = 1$ is obtained as an optimal solution,

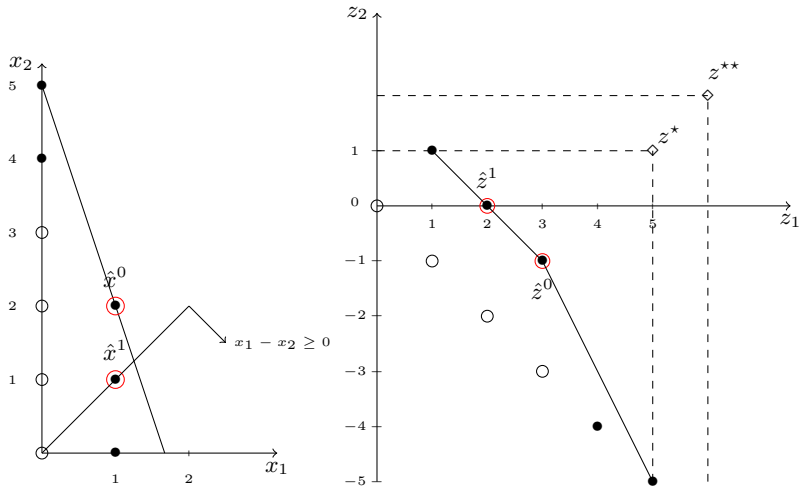



**Fig. 1.** Iteration 1.

the relative constraint to $y_1^1$ is redundant, the current space of new solution is only constrained by $x_1 - x_2 \geq 0$ (See Fig. 1). We compute $\hat{\beta}^1$ of $\hat{z}^1$ as defined as in formula (3), we must have $\hat{\beta}^1 = (\frac{1}{3}, \frac{2}{3})$. We let $E_1 = E_0 \bigcup \hat{x}^1$ $Z(P)_1 = Z(P)_0 \bigcup \hat{z}^1$ and $\beta^\star$=$(\frac{1}{3}, \frac{2}{3})$
**Iteration 2** In this iteration the problem $P(\beta^\star)$ is solved in the space that

reduced by two solutions(Non-dominated vectors) previously found

$$P(\beta^\star)\begin{cases} Min \quad \omega \\ \omega \geq \frac{1}{3}(2 - x_1 - x_2) \\ \omega \geq \frac{2}{3}(0 - x_1 + x_2) \\ \omega \geq 0 \\ D - D_2 \begin{cases} x, y_j^1 & \in D - D_1, j = 1, 2 \\ x_1 + x_2 & \geq (2+1)y_1^2 \\ x_1 - x_2 & \geq (0+1)y_2^2 - 5(1 - y_2^2) \\ y_1^2 + y_2^2 & \geq 1 \\ y_j^i & \in \{0, 1\}, i, j = 1, 2 \end{cases} \end{cases}$$

As showed in (Fig.2), $(\hat{x}^2, \hat{z}^2) = \{(1;0), (1;1)\}$ is a new efficient/nondominated

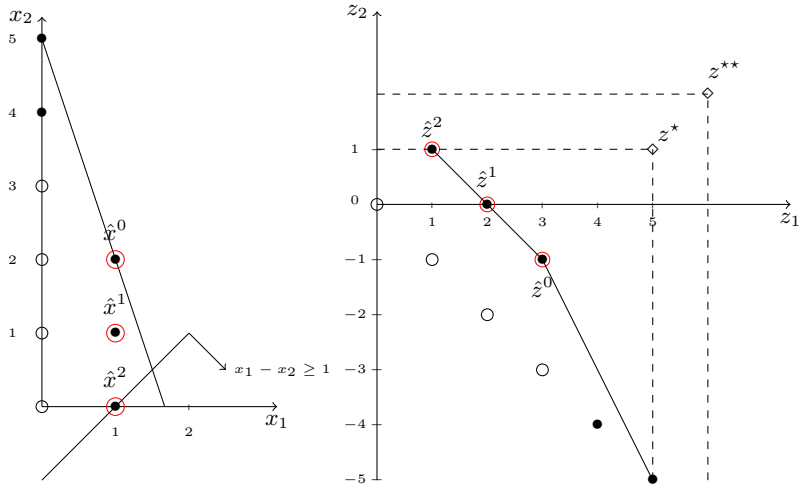



**Fig. 2.** Iteration 2.

solution with secondary variables $y_1^1 = 0, y_2^1 = 1, y_1^2 = 0, y_2^2 = 1$, the constraints associated to $y_1^1, y_2^1, y_1^2$ are redundant; according to the formula (3), the weighted vector $\hat{\beta}^2$ is $(\frac{1}{6}, \frac{5}{6})$. Set $E_2 = E_1 \bigcup \hat{x}^2$ $Z(P)_2 = Z(P)_1 \bigcup \hat{z}^2$ and $\beta^\star$=$(\frac{1}{6}, \frac{5}{6})$.
**Iteration 3** The following step adds constraints that delete the efficient points

previously found

$$P(\beta^\star)\begin{cases} Min \quad \omega \\ \\ \omega \geq \frac{1}{6}(1-x_1-x_2) \\ \omega \geq \frac{5}{6}(1-x_1+x_2) \\ \omega \geq 0 \\ D-D_3 \begin{cases} x, y_j^i & \in D-D_2, i=1,2, j=1,2 \\ x_1+x_2 & \geq (1+1)y_1^3 \\ x_1-x_2 & \geq (1+1)y_2^3-5(1-y_2^3) \\ y_1^3+y_2^3 & \geq 1 \\ y_1^3, y_2^3 & \in \{0,1\} \end{cases} \end{cases}$$

The optimal solution to the last problem is $(\hat{x}^3, \hat{z}^3) = \{(0;4),(4;-4)\}$ and

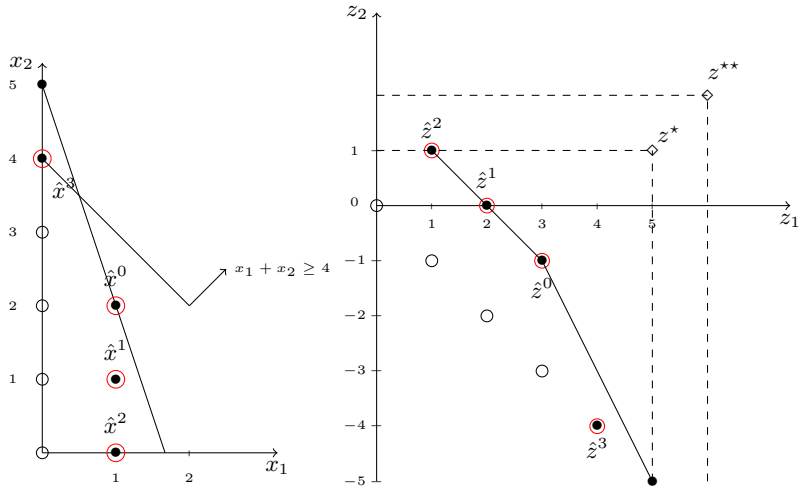



**Fig. 3.** Iteration 3.

$y_1^1 = 1, y_2^1 = 0, y_1^2 = 1, y_2^2 = 0$ $y_1^3 = 1, y_2^3 = 0$. The additional constraints to $y_2^1, y_1^2, y_1^3, y_2^3, y_2^2$ are redundant. This solution is unsupported and gotten relatively to the constraint $x_1 + x_2 \geq 4$ according to the variable $y_1^1$, see figure (Fig.3), the current weighted vector is $\hat{\beta}^3 = (\frac{3}{4}, \frac{1}{4})$. Let $E_3 = E_2 \bigcup \hat{x}^3$ $Z(P)_3 = Z(P)_2 \bigcup \hat{z}^3$ and $\beta^\star$=$(\frac{3}{4}, \frac{1}{4})$.

**Iteration 4** Now, problem $P(\beta^\star)$ is defined as:

$$P(\beta^\star)\begin{cases} Min \quad \omega \\ \\ \omega \geq \frac{3}{4}(4 - x_1 - x_2) \\ \omega \geq \frac{2}{7}(-4 - x_1 + x_2) \\ \omega \geq 0 \\ D - D_4 \begin{cases} x, y_j^i \quad \in D - D_3, i = 1,2,3; j = 1,2 \\ x_1 + x_2 \geq (4+1)y_1^4 \\ x_1 - x_2 \geq (-4+1)y_2^4 - 5(1 - y_2^4) \\ y_1^4 + y_2^4 \geq 1 \\ y_1^4, y_2^4 \quad \in \{0,1\} \end{cases} \end{cases}$$

$(\hat{x}^4, \hat{z}^4) = \{(0;5),(5;-5)\}$, $y_1^1 = 1, y_2^1 = 0$,$y_1^2 = 1, y_2^2 = 0$, $y_1^3 = 1, y_2^3 = 0$ and

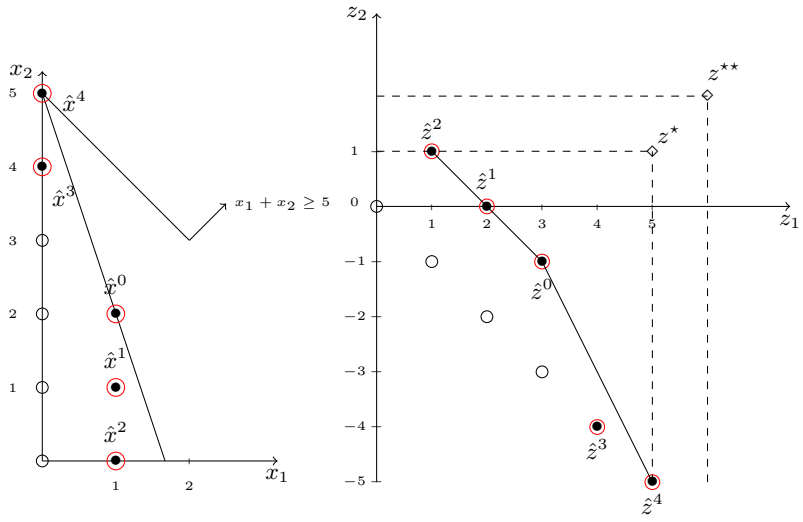



**Fig. 4.** Iteration 4.

$y_1^4 = 1, y_2^4 = 0$ is the optimal solution to the current Tchebycheff program. This optimal solution is obtained from the only constraint $x_1 + x_2 \geq 5$ imposed by the first objective function relatively to the additional variable $y_1^4$, see (Fig.4); for this iteration, we obtain $\hat{\beta}^4 = (\frac{7}{8}, \frac{1}{8})$. Set $E_4 = E_3 \bigcup \hat{x}^4$ $Z(P)_4 = Z(P)_3 \bigcup \hat{z}^4$ and $\beta^\star$=$(\frac{7}{8}, \frac{1}{8})$.

**Iteration 5** The next Tchebycheff $P(\beta^\star)$ to be solved is

$$P(\beta^\star)\begin{cases} Min \quad \omega \\ \\ \omega \geq \frac{7}{8}(5 - x_1 - x_2) \\ \omega \geq \frac{1}{8}(-5 - x_1 + x_2) \\ \omega \geq 0 \\ \qquad D - D_5 \begin{cases} x, y_j^i & \in D - D_4, i = 1,2,3,4; j = 1,2 \\ x_1 - 2x_2 & \geq (0+1)y_1^5 \\ -x_1 + 3x_2 & \geq (1+1)y_2^5 - 5(1 - y_2^5) \\ y_1^5 + y_2^5 & \geq \\ y_1^5, y_2^5 & \in \{0,1\} \end{cases} \end{cases}$$

The current Tchebycheff problem $P(\beta^\star)$ is unfeasible, then the process is complete and we have the complete set of nondominated objective vectors, and the subset of the corresponding efficient solutions
$E(P) = \{(1;2),(1;1),(1;0),(0;4),(0;5)\}$,$Z(P) = \{(3;-1),(2;0),(1;1),(4;-4),(5;-5)\}$.

## 5 Conclusion

We have described an algorithm for multiple objective integer linear programs based on weighted Tchebycheff norm. The algorithm improves on the similar method of sylva and Crema(2004) by providing a guarantee that all nondominated objective vectors are identified.

The main idea is to combine the weighted Tchebycheff program with the cuts idea of sylva and Crema(2004), This technique increases the complexity of the Tchebycheff program but avoids the trap lies the weakly nondominated vectors and the used norm and prevents the recourse to the generalized norm and its theoretical aspects.

The procedure possesses the advantage of the displacement of a solution to another neighbor solution, it can be very useful in the interactive procedures, specially at early stages even though we don't match information on the DMs preferences. For future research, we suggest an updated survey for a comparison between several methods.

## References


1. M. Luque, F. Ruiz, and R.E. Steuer. Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operations Research*, 78:557–564, 2010.
2. D. Chaabane, B. Brahmi, and Z. Ramdani. The augmented weigted Tchebytchev Norm for optimizing a linear function over an integer effiient set of multicriteria linear program. *intl trans in Ope. Res. DOI:10.1111/j.1475-3995.2012.00851.x*, 19(2012),531–545.
3. V. J. Bowman. On the Relationship of the Tchebytcheff Norm and the Efficient Frontier of Multiple-Criteria Objectives. *In Thiriez H. & Zionts S. (Eds)*,76–85,1976.

4. J. Sylva, and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Reserach*, 158: 46–55, 2004.
5. T. K. Ralphs, M. J. Saltzman, and M. M. Wiecek. An improved algorithm for solving biobjective integer programs. *Ann Oper Res*, 157: 43–70, 2007.
6. D. Klein and E. Hannan. An Algorithm for Multiple Objective Integer Linear Programming Problem. *European Journal of Operational Research*, 9: 378–385,1982.
7. J. Sylva and A. Crema. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Reserach*, 180: 1011–1027, 2007.
8. P. K. Eswaran, A. Ravindran, and H. Moskowitz. Algorithms for Nonlinear Integer Bicriterion Problems. *Journal of Optimization Theory and Applications*, 2:63, 2000.
9. R. E. Steuer and E.E. Choo. An Interactive Weighted Tchebycheff Procedure For Multiple Objective Programming. *Mathematical Programming*, 26: 326–344, 1983.
10. R. E. Steuer *Multiple Criteria Optimization : Theory, Computation and Applications.* John Wiley & Sons, New-York, 1986.
11. O. Marcotte, and R.M. Soland. An Interactive Branch-and-Bound Algorithm for Multiple Criteria Optimization. *Management Science*, 32: 1231–1240, 1986.
12. L. G. Chalmet, L. Lemondis, and D. J. Elzinga An Algorithm for the Bi-Criterion Integer Programming Problem. *European Journal of Operational Research*, 25: 292–300, 1981.
13. G.W. Bitran. Linear multiple objective programs with zero-one variables. *Mathematical Programming*, 13: 121–139, 1977.
14. E. L. Ulungu and J. Teghem. Multi-objective combinatorialoptimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3: 83–104, 1994.
15. B. Villarreal and M. H. Karwan. Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical Programming*, 21: 204–223, 1981.
16. R. Ramesh, M. H. Karwan, and S. Zionts. Preference Structure Representation Using Convex Cones in Multicriteria Integer Programming. *Management Science*, 35: 1092–1105, 1989.
17. J. Karaivanova, P. Korhonen, S. Narula, J. Wallenius, and V. Vassilev. A Reference Direction Approach to Multiple Objective Integer Linear Programming. *European Journal of Operational Research*, 81: 176–187, 1995.
18. B. Schandl, K. Klamroth, and M. M. Wiecek. Norm-Based Approximation in Bicriteria Programming. *Computational Optimization and Applications*,20: 23–42, 2001.
19. C. Ferreira, J. Climaco, and J. Paixão. The Location-Covering Problem: A Bicriterion Interactive Approach. *Investigación Operativa*, 4: 119–139, 1994.
20. M. Ehrgott, and X. Gandibleux. Multiobjective Combinatorial Optimization-Theory, Methodology and Applications. *In M. Ehrgott and X. Gandibleux (eds.), Multiple Criteria Optimization-State of the Art Annotated Bibliographic Surveys, Kluwer Academic Publishers, Boston*, 369–444, 2002.
21. J. Teghem, and P. L. Kunsch. A survey of techniques for finding efficient solutions to multi-objective integer linear programming. *Asia-Pacific Journal of Operational Research*, 3: 95–108, 1986.
22. M. J. Alves, and J. Clímaco. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180: 99–115, 2007.

# Processus métier, services web et artefacts

# Une Approche pour le Matching des Graphes de Processus Métiers à Base d'Opérateurs Logiques

Farid Kacimi[1], Abdelkamel Tari[1], and Hamamache Kheddouci[2]

[1] Université Abderrahmane Mira, Route Targa Ouzemour, Béjaïa, Algérie, 06000
kacimifarid@gmail.com
[2] Université Claude Bernard Lyon1, 43 Bd du 11 Novembre 1918, F-69622 Villeurbanne Cedex, France

**Abstract.** Avec la prolifération des collections de modèles de processus métiers, un système rapide et efficace pour trouver des modèles de processus parmi des centaines ou des milliers de modèles candidats est nécessaire. La recherche de similarité ou le matching est l'une des opérations requise pour le management de ces collections, elle consiste à comparer les modèles de processus et à sélectionner celui qui correspond le mieux à un modèle requête. Les approches de matching de modèles de processus à base de graphes sont très efficaces en termes de précision pour le matching et le classement de processus. Cependant, la complexité des algorithmes de matching de graphes croît exponentiellement avec la taille des graphes. Dans cet article, nous proposons une approche de matching basée sur des opérateurs logiques pour la comparaison et l'évaluation de la similarité des modèles de processus. Pour réduire la complexité de matching, nous proposons de décomposer les graphes de processus en chemins, puis de calculer leur similarité sur la base de la similarité des représentations binaires de leurs chemins. D'autre part, l'implémentation et l'expérimentation de l'approche proposée sont en cours de réalisation.

**Keywords:** Matching de graphes, Matching de processus métiers, Processus métiers, Services web

## 1 Introduction

Durant ces dernières années l'environnement des entreprises a rapidement évolué, rendant ainsi leurs systèmes plus complexes, de plus avec l'utilisation des nouvelles technologies de nouveaux besoins et de nouvelles exigences sont apparues (réutilisabilité, interopérabilité, etc.). L'une des solutions pour faire face à ces nouvelles exigences est l'architecture orientée services (SOA), elle repose principalement sur le concept de services qui sont des composants autonomes, réutilisables et indépendants des plateformes et des langages de programmations.

De plus en plus d'organisations se tournent vers l'utilisation des services web afin d'externaliser et de mettre en œuvre leurs processus métiers qui constituent un des atouts majeurs de ces organisation, permettant ainsi de créer des applications complexes. Par conséquent, il est fréquent pour une organisation d'avoir

des collections de centaines, voire de milliers de modèles de processus métiers. Un exemple d'une telle collection, est celle de la compagnie d'assurance Suncorp, qui contient plus de 6000 modèles de processus métiers [1]. La performance des organisations sur les marchés est étroitement liée à l'optimisation, la flexibilité et la capacité de mise à jour de leur processus métiers [2]. Ainsi, les collections de modèles de processus métiers offrent de nouvelles opportunités, mais aussi de nouveaux challenges. Par ailleurs, découvrir et comprendre les similarités entre les différents processus métiers peut être utile aux organisations pour de nombreuses raisons, notamment une meilleure gestion et maintenance de tous ces processus. Par conséquent, des techniques sont requises pour comparer rapidement des modèles de processus métiers dans une telle collection.

De nombreuses approches proposent de modéliser les modèles de processus métiers sous forme de graphes [3, 4, 5, 6], ce choix est motivé par le fait que la transformation des processus en graphes est simple (les nœuds représentent les activités et les arcs les relations entre eux). Par conséquent, le problème de comparaison entre les modèles de processus métiers est ramené à un problème de comparaison de graphes. Afin de mesurer la similarité entre les graphes comparés, la distance d'édition entre ces graphes est calculée, elle est basée sur des opérations d'éditions qui sont appliquées aux graphes [7]. Ainsi, la similarité entre deux graphes est définie comme le nombre minimum d'opérations d'éditions nécessaires pour transformer un graphe en un autre. Plusieurs approches récentes pour le matching de processus ont montrés l'efficacité des approches à base de graphes pour le matching de processus [3, 4, 8, 9]. Cependant, les algorithmes de matching de graphes (par exemple, la détection d'isomorphisme de graphes avec tolérance d'erreurs), souffrent d'une grande complexité de calcul.

Dans cet article, nous proposons une approche qui décompose les graphes de modèles de processus en chemins représentés sous forme binaires, puis nous déterminons la similarité structurelle entre les chemins en utilisant des opérateurs logiques. En plus de cette similarité structurelle une similarité syntaxique permet de comparer les noms d'activités. L'approche proposée décompose les graphes en chemins puis compare chaque chemin dans son ensemble, plutôt que de comparer les nœuds un par un en utilisant des opérateurs logiques, ce qui permet de réduire le temps du matching.

Le reste de cet article est organisé comme suit : La section 2, présente un état de l'art sur les approches existantes. Dans la section 3, nous commençons par la définition de quelques concepts utilisées dans cet article, puis nous présentons notre approche, ainsi que la mesure de similarité utilisée pour comparer les modèles de processus. La section 4, présente un scénario d'exécution pour expliquer notre approche. Enfin, la section 5, présente nos conclusions et travaux futurs.

## 2 Etat de l'Art

Les premières approches proposées pour le matching de services web sont basées sur une recherche par mots clés. Cependant, ces approches basées sur la syntaxe uniquement sont insuffisantes pour la plupart des applications. La tendance des

approches actuelles est d'exploiter plus d'informations sur la sémantique et le comportement des services. En effet, dans [10, 11], le matching entre les services est basé sur le matching sémantique de leurs entrées/sorties appartenant à un ensemble d'ontologies. Ensuite, plusieurs extensions de cette approche ont été proposées, en considérant aussi le matching des pré-conditions [12] ou les préférences de l'utilisateur [13]. Les auteurs dans [14, 15, 16, 17], ont signalé la nécessité de prendre en compte dans le processus de matching du comportement décrit par le modèle de processus. Dans [15], afin d'améliorer la précision de la découverte de services web, le modèle de processus est utilisé pour capturer le comportement du service.

Plusieurs auteurs ont proposé de représenter les modèles de processus comme des graphes. Les auteurs dans [4, 5, 6, 18] ont proposé une adaptation de l'algorithme de matching de graphes (isomorphisme de sous-graphes avec tolérance d'erreurs) pour le matching sémantique et structurelle de processus. Ainsi, dans [5] pour améliorer l'efficacité en terme de temps d'exécution de l'algorithme de matching de graphes, les auteurs ont proposé une technique de résumé qui permet de réduire la taille des graphes, en transformant les processus organisés en blocs délimités par des nœuds connecteurs, en nœuds abstraits qui résument ces blocs. Dans [3], les auteurs ont proposé trois algorithmes pour mesurer la similarité des graphes de processus basés sur la distance d'édition de graphe. Le premier explore un seul mapping à la fois et ajoute la paire de nœuds la plus similaire. Les deux autres algorithmes sont des variantes de l'algorithme de matching de graphes A*, mais intègre une fonction de réduction qui réduit l'espace de recherche lorsque le nombre de mappings est supérieur à un seuil donné. Cependant, les techniques proposées pour réduire la complexité des algorithmes de matching de modèles de processus à base de graphes se concentrent sur la réduction de l'espace de recherche en définissant des fonctions de réductions. Par conséquent, leur efficacité dépend du choix des paramètres de matching [5].

Deux approches récentes [2, 19] ont été proposées, elles proposent une décomposition des graphes de modèles de processus à comparer. Dans [2], les auteurs proposent de représenter chaque processus métier, par un graphe orienté, et de le décomposer en ses séquences d'exécutions possibles. Ensuite, la distance entre deux graphes est calculée par un algorithme de calcul de distance entre séquence d'exécutions possibles des deux processus métiers qui est basé sur une distance d'édition de chaines de caractères. Cependant, cette décomposition engendre un grand nombre de séquences à comparer. Dans [19], les auteurs proposent de représenter les processus à comparer en arbre orientés et étiquetés, et de les décomposer en chemins. Pour chaque chemin, les informations concernant les actions atomiques et les nœuds de contrôle sont représentées au sein d'une structure de données appelées : signature de chemin. Par la suite, utiliser le noyau de graphe pour calculer la similarité des arbres comparés en se basant sur l'ensemble des signatures de chemin de chaque arbre. L'inconvénient de cette approche est quelle ne tient pas compte de l'ordre d'apparition des différents nœuds de contrôles.

Dans cet article, nous proposons une méthode pour l'approximation de la distance entre des modèles de processus basée sur l'algèbre booléenne. Notre approche est basée sur une décomposition des modèles de processus représentés comme des graphes en un ensemble de chemins codés en binaires.

## 3 Approche Proposée

Dans cette section, nous introduisons quelques notions utilisées pour définir un modèle de processus, puis la description de notre approche pour le matching de modèles de processus.

### 3.1 Modèles de Processus

Un modèle de processus consiste en une collection d'activités liées qui sont combinées par des structures de contrôles (séquence, parallèle, etc.). Dans cet article, un modèle de processus est défini comme un graphe orienté et attribué $G = (V, A)$, où $V$ représente un ensemble de nœuds et $A$ un ensemble d'arcs définissant les relations entre les nœuds (Voir Fig. 1). Deux types de nœuds peuvent être distingués : les nœuds d'activités qui représentent les tâches atomiques, et les nœuds connecteurs qui représentent les contraintes du flux de contrôle. Les nœuds connecteurs représentent des opérateurs `split` et `join` de types `XOR` ou `AND`. Un connecteur `XOR-Split` représente un choix entre une ou plusieurs alternatives regroupées par un `XOR-Join`, tandis qu'un connecteur `AND-Split` représente une exécution concurrente synchronisée par un `AND-Join` [2].

### 3.2 Mesure de Similarité de Chaines de Caractères

Pour comparer deux activités, nous utilisons une mesure de similarité syntaxique sur leurs noms. Cette mesure de similarité a été introduite dans [3], elle est basée sur la distance de Levenshtein aussi appelée distance d'édition [20]. La distance d'édition entre deux chaines de caractères est définie comme le nombre minimum d'opérations d'éditions (insertions, suppressions, et substitutions) requises pour transformer une chaine de caractères en une autre.

Etant donné deux chaines de caractères $S_1$ et $S_2$, leur similarité est :

$$sim(S_1, S_2) = 1 - \frac{ed(S_1,S_2)}{\max(|S_1|,|S_2|)}$$

Où $ed(S_1, S_2)$ correspond à la distance d'édition entre $S_1$ et $S_2$.

Par exemple, la distance d'édition entre «Représenter» et «Représentation» est égale à 5, substituer '$e$' par '$a$', '$r$' par '$t$' et insérer '$ion$'. Ainsi, la similarité entre les deux chaines de caractères est : $1 - 5/14 = 0.64$.

### 3.3 Décomposition en Chemins

Le problème majeur des solutions proposées pour le matching de graphes, est la complexité de calcul. Pour résoudre ce problème nous proposons de décomposer le graphe de processus en chemins. Plusieurs solutions ont proposé de décomposer les graphes de processus en petites sous-structures [19, 21], l'avantage majeur de la décomposition est de réduire le temps du matching. Dans ce qui suit, nous donnons les détails de notre décomposition des graphes de modèles de processus en chemins.

Un graphe de modèle de processus représente toutes les exécutions possibles du modèle de processus. Afin de faciliter la comparaison et la rendre plus rapide et efficace, nous avons choisi de décomposer le graphe en un ensemble de chemins à partir du nœud `START` représentant le début du graphe, jusqu'au nœud `END` représentant la fin du graphe. Ce choix est motivé par le fait que, chaque chemin extrait du graphe représente une exécution possible du modèle de processus. La Fig. 1, montre un exemple de décomposition d'un graphe de processus en un ensemble de chemins.

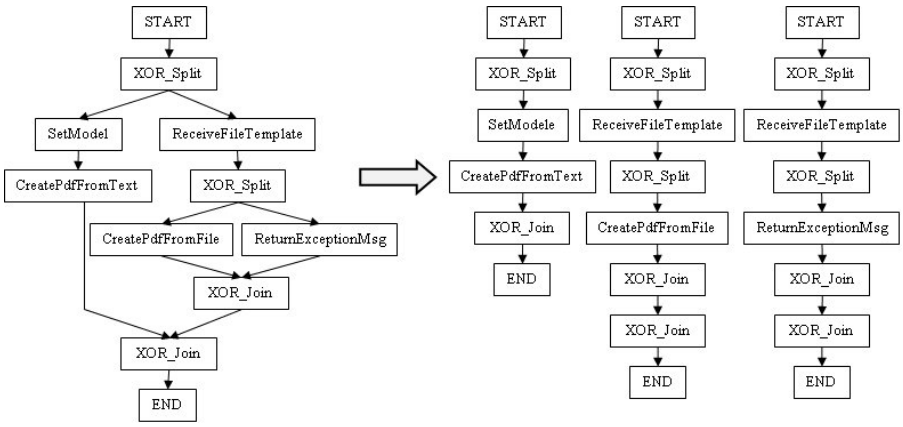



**Fig. 1.** Décomposition du graphe de modèle de processus en chemins

### 3.4 Représentation Binaire des Chemins

Un chemin du graphe est constitué d'une suite de nœuds, chaque nœud peut être soit un nœud d'activité, ou bien un nœud connecteur de type `XOR-Split`, `XOR-Join`, `AND-Split`, ou `AND-Join`. Nous proposons de résumer les informations caractérisant un chemin, à savoir les différents nœuds et l'ordre d'exécution de ces nœuds, dans une représentation binaire. Pour cela, chaque nœud est représenté sur trois bits. Ainsi, les nœuds d'activités, et les nœuds `XOR-Split`, `XOR-Join`, `AND-Split`, et `AND-Join`, sont représentés respectivement par 111, 001, 010, 011

et 100. Un exemple de représentation binaire d'un chemin est donné dans la Fig. 2.

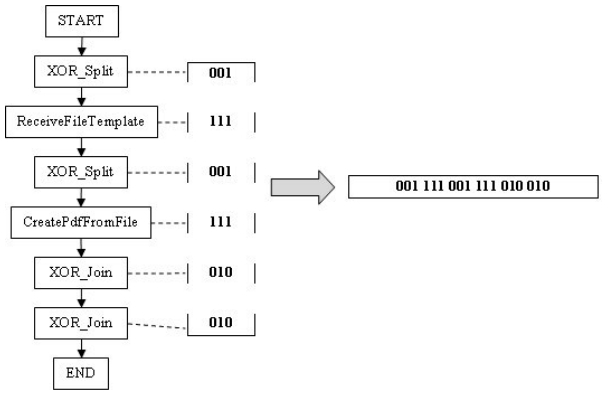



**Fig. 2.** Représentation binaire d'un chemin

### 3.5 Comparaison de Deux Chemins

Etant donné deux chemins de graphes à comparer $c_1$ et $c_2$, ainsi que leurs représentations binaires respectives : $b_1 = 001\ 011\ 111\ 100\ 010$ et
$b_2 = 001\ 001\ 111\ 010\ 010$. L'application de l'opérateur d'équivalence logique (noté $\Leftrightarrow$ ou `XNOR`) entre deux valeurs binaires retourne 1 si les deux valeurs sont similaires et 0 autrement. Nous proposons d'utiliser l'opérateur `XNOR` sur les deux représentations binaires correspondantes aux chemins. Le résultat de l'application de l'opérateur logique est une chaine binaire $R$. Afin de mesurer la similarité entre $b_1$ et $b_2$, dans la chaine en résultat, trois 1 successifs sont remplacés par un 1, sinon par un 0 s'ils sont différents. En résultat 1 indique une similarité et 0 une dissimilarité.

Ainsi pour les deux représentations binaires précédentes :
$b_1 = 001\ 011\ 111\ 100\ 010$
$b_2 = 001\ 001\ 111\ 010\ 010$
$R = b_1$ `XNOR` $b_2 = 111\ 101\ 111\ 001\ 111$
$F = 1\ 0\ 1\ 0\ 1$

### 3.6 Algorithme

Pour comparer deux graphes de modèles de processus $A$ et $B$, d'abord les graphes à comparer sont décomposés en chemins représentés en binaire. Puis la similarité entre les chemins est calculée en comparant chaque représentation binaire de $A$ à chaque représentation binaire de $B$, ainsi que les activités contenues

dans les chemins. Comparer deux représentations binaires consiste à appliquer l'opérateur logique `XNOR` sur les deux représentations puis de calculer leur similarité $sim_{binaire}$ en fonction du résultat de l'opération (Voir Algorithme 1). La similarité entre deux activités $sim_{activite}$ est déterminée en utilisant la mesure de similarité syntaxique détaillée dans la Section 3.2.

**Algorithm 1** Algorithme de Matching

**Input:** Deux graphes de modèles de processus $A$ et $B$
**Output:** Degré de similarité entre $A$ et $B$

Décomposer $A$ et $B$ en chemins représentés en binaire, $b_{Ai}$ et $b_{Bj}$ respectivement
**pour** chaque chemin $i$ du graphe $A$ **faire**
  **pour** chaque chemin $j$ du graphe $B$ **faire**
    */* Calculer la similarité entre les représentations binaires */*
    `binaire` $= b_{Ai}$ `XNOR` $b_{Bj}$
    **pour** $k$ allant de 1 à `taille(binaire)` pas 3 **faire**
      **si** (`binaire`$[k]$ == `binaire`$[k+1]$ == `binaire`$[k+2]$ == 1) **alors**
        `resultat` ← `resultat` + "1" */* Ajoute 1 à la chaine resultat */*
      **sinon**
        `resultat` ← `resultat` + "0" */* Ajoute 0 à la chaine resultat */*
      **fin si**
    **fin pour**
    Calculer `nbre` = nombre de 1 dans resultat
    $\texttt{sim}_{binaire} = \texttt{nbre}/\texttt{taille}(\texttt{resulat})$
    Calculer $\texttt{sim}_{activite}$ = la similarité entre les noms des nœuds d'activités
    $\texttt{sim}(i,j) = (\texttt{sim}_{binaire} + \texttt{sim}_{activite})/2$
  **fin pour**
**fin pour**
*/* Calculer la similarité globale des graphes */*

$$\texttt{sim}_{globale} = \frac{\sum_{i=1,j=1}^{n,m} \max sim(i,j)}{\max(n,m)}$$, avec $n$ et $m$ le nombre de chemins de $A$ et $B$

## 4 Scénario d'Exécution

La Fig. 3 illustre un exemple de matching de deux graphes de processus $A$ et $B$ en utilisant l'approche proposée. La décomposition retourne 3 représentations binaires pour chaque graphe, ainsi qu'une matrice de similarité entre les chemins des graphes. L'algorithme recherche pour chaque chemin de $A$, le chemin de $B$ le plus similaire. Pour cela, nous utilisons des opérateurs logiques sur les représentations binaires des chemins et une mesure de similarité de chaine de caractères (syntaxique) sur les activités.

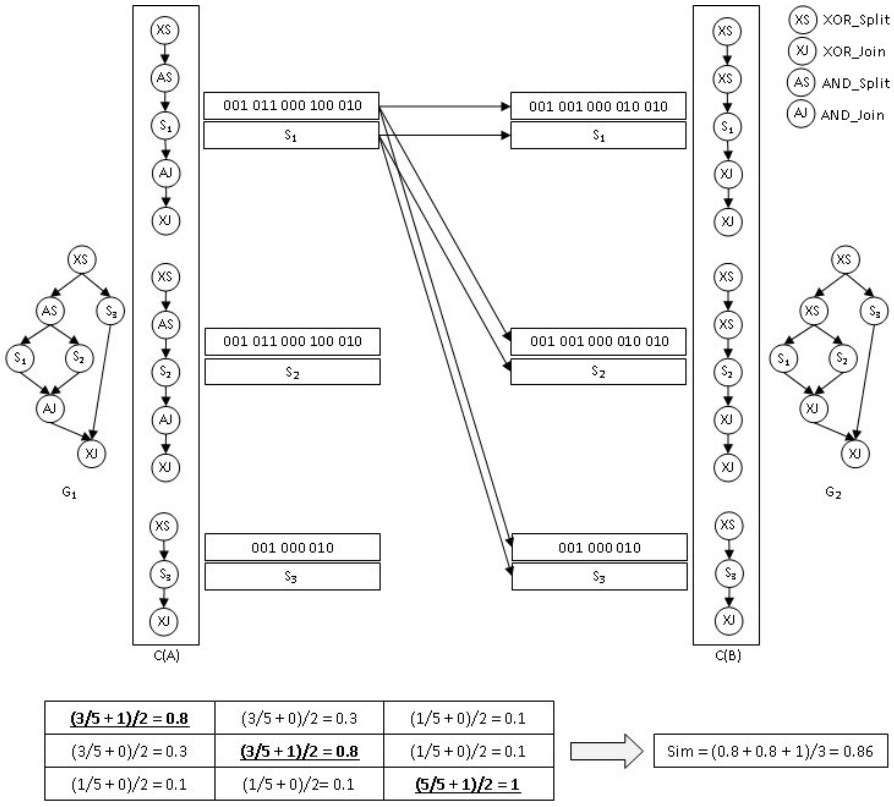



**Fig. 3.** Exemple de matching

Etant donné deux chemins $C_1(A)$ du graphe $A$ et $C_2(B)$ du graphe $B$, leurs représentations binaires respectives 001011111100010 et 001001111010010, et les activités $S_1$ et $S_2$ contenues dans chaque chemins respectivement. Pour calculer la similarité entre les deux chemins, d'abord l'opérateur logique `XNOR` est appliqué sur les représentations binaires, on obtient 111101111001111. En remplaçant trois 1 successifs par un seul, sinon par un 0, on obtient le résultat suivant : 10101. Ainsi, la similarité des représentations binaires précédentes $sim_{binaire}$ est égale à $3/5 = 0.6$. Puis, on utilise la mesure de similarité syntaxique détaillée dans la Section 3.2 sur les activités des chemins. Dans notre exemple, nous avons considéré que la similarité est égale 1 si les deux activités sont similaires sinon elle est égale à 0, dans notre cas $sim_{activite}$ est égale à 0. Par conséquent, la similarité globale des chemins comparés est égale $(0.6 + 0)/2 = 0.3$.

## 5 Conclusion

Dans cet article, nous avons proposé une solution pour le matching de modèles de processus à base d'opérateurs logiques appliqués sur les représentations binaires des décompositions en chemins des graphes. L'approche proposée utilise des opérateurs logiques pour comparer les représentations binaires des chemins, qui sont manipulés comme un ensemble plutôt que nœuds par nœuds. L'avantage majeur de la décomposition en représentation binaire et de l'utilisation des opérateurs logiques est de réduire la complexité en termes de temps du matching.

Comme travaux futures, nous envisageons d'expérimenter notre approche sur un ensemble de modèles de processus, et d'améliorer les performances de notre approche, en considérant d'autres mesures de similarités entre les activités.

## References


[1] La Rosa, M., Dumas, M. , Uba, R. , Dijkman, R.: Business process model merging: An approach to business process consolidation. In: ACM Transactions on Software Engineering and Methodology (TOSEM), Vol 22. No. 2 (2012)

[2] Belhoul, Y., Haddad, M., Duchêne, E., Kheddouci, H.: String Comparators Based Algorithms for Process Model Matchmaking. In: Proc. of the 9th IEEE International Conference on Service Computing (SCC'12), pp. 649-656, Honolulu, Hawaii, USA (2012)

[3] Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)

[4] Gater, A., Grigori, D., Bouzeghoub, M.: OWL-S process model matchmaking. In: IEEE International Conference on Web Services (ICWS'10), pp. 640–641, Miami, Florida, USA, July 5-10 (2010)

[5] Gater, A., Grigori, D., Haddad, M., Bouzeghoub, M., Kheddouci, H.: A summary-based approach for enhancing process model matchmaking. In: IEEE International Conference on Service Oriented Computing & Applications (SOCA 2011), page 1-8, Irvine, USA (2011)

[6] Grigori, D., Corrales, J. C., Bouzeghoub, M., Gater, A.: Ranking bpel processes for service discovery. In: IEEE Transactions on Service Computing, 3(3), 178–192 (2010)

[7] Bunke, H.: Recent developments in graph matching. In: International Conference on Pattern Recognition (ICPR'00), vol. 2, pp. 117–124, Barcelona (2000)

[8] Dijkman, R. M., Dumas, M., van Dongen, B., Käärik, R., Mendling, J. Similarity of business process models: Metrics and evaluation. In: Inf. Syst. 36(2), 498–516 (2011)

[9] García-Bañuelos, L., Dijkman, R., Dumas, M., Krik, R.: Aligning Business Process Models. In: Proc. of the 13th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 45–53, Auckland, New Zealand (2009)

[10] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic Matching of Web Services Capabilities. In: Proc. of the 1st International Semantic Web Conference (ISWC'02), pp. 333–347, Sardinia, Italy, (2002)

[11] Benatallah B., Hacid M., Rey C., Toumani F.: Semantic Reasoning for Web Services Discovery. In: Proc. of WWW Workshop on E-Services and the Semantic Web (2003)

[12] Bellur, U., Vadodaria, H., Gupta, A.: Greedy Algorithms. In: Bednorz, W. (ed.) Semantic Matchmaking Algorithms. InTech, Croatia (2008)

[13] Beck, M., Freitag, B.: Semantic matchmaking using ranked instance retrieval. In: SMR 2006: 1st International Workshop on Semantic Matchmaking and Resource Retrieval, Colocated with VLDB (2006)

[14] Bansal, S., Vidal, J. M., : Matchmaking of web services based on the DAML-S service model. In: Proc. of Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 926–927, New York (2003)

[15] Bernstein, A., Klein, M.: Towards High-Precision Service Retrieval. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 84–101. Springer, Heidelberg (2002)

[16] Zdravkovic, J., Johanesson, P.: Cooperation of Processes through Message Level Agreement. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 564–579. Springer, Heidelberg (2004)

[17] Wombacher, A., Mahleko, B., Fankhauser, P., Neuhold, E.: Matchmaking for Business Processes based on Choreographies. In: Proc. of IEEE International Conference on e-Technology, e-Commerce and e-Service, Taipei, Taiwan (2004)
[18] Gater, A., Grigori, D., Bouzeghoub, M.: Matching and similarity evaluation of OWLS-S process models. In: BDA, 25emes journées Bases de Données Avancées, Namur, Belgique (2009)
[19] Lagraa, S., Seba, H., Khennoufa, R., Kheddouci, H.: Matchmaking OWL-S processes: an approach based on path signatures. In: Proc. of The International Conference on Management of Emergent Digital EcoSystems (MEDES 2011), San Francisco, USA (2011)
[20] Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet Physics-Doklady 10, vol. 10(8), pp. 707– 710 (1966)
[21] Seba, H., Lagraa, S., Kheddouci, H.: Web Service Matchmaking by Subgraph Matching. J. Filipe and J. Cordeiro (Eds.): WEBIST 2011, Lecture Notes in computer Science: LNBIP 101, pp. 43–56, 2012. Springer-Verlag Berlin Heidelberg (2012)

# QoS-Aware Web Service Selection Based On Bees Algorithm

Hadjila Fethallah[1], Chikh Mohammed Amine[1], Merzoug mohammed[1]

[1] Université de Abou Bekr Belkaid Temcen
{f_hadjila, mea_chikh,merzoug.mohammed}@mail.univ-tlemcen.dz

**Abstract.** QoS driven selection approaches for Web Services Composition are used to choose the best solution among candidate services which have the same functions but different QoS properties. In this paper, we propose a bee algorithm, in order to select a near optimal composition, this approach uses several operators to explore the space search, such as the local search, the social exchange, the random insertion…. Experimental results show that the bee algorithm is more effective than the other local search techniques such as the tabu search.

**Key words:** Web Service Selection , Bees Algorithm, Combinatory Optimization, Meta-heuristics, Quality Of service.

## 1 Introduction

The Service-oriented architecture (SOA) is a paradigm that aims to build distributed applications over the internet; SOA applications have several properties, they are loosely coupled, platform independent, language independent …etc.

Web services is a promising implementation of the SOA paradigm, they provide a new model of the web in which distributed programs exchange dynamic information on demand. The tools and technology for building and deploying web services are readily available.

Automatic service discovery and composition have received much attention because upon these technologies, the automatic business-to-business or enterprise level application integration become possible. Several standards have been created by the W3C consortium support the web services: SOAP, UDDI, WSDL[9], and BPEL[16].

A composite service is assembled by several tasks to accomplish a mission. In internet there are maybe many available web services with various QoS (Quality of Service) providing the same functionality specific to a task. So a selection needs to be made. During the composition, there are demands for QoS constraints to be met and QoS criterions to optimize.

Therefore web service composition has to search for an optimal set of services to construct a composite service and result in a best QoS, under user's QoS constraint and basic functionality claim. Our objective is to propose an algorithm for constructing such compositions. We notice that, this problem is NP Hard, in fact it is

an instance of the Multi-Choice Multidimensional Knapsack problem (MMKP)[18], which is well known in the domain of combinatory optimization.

The figure1 shows the different elements that constitutes the problem, we have a user request that can be satisfied by an abstract workflow F, this later consists of n service's classes, the client has to choose one service from each class, so that the set of QOs criteria is optimized, moreover he must check the satisfiability of the global constraints (for instance the execution time of the composition must be lesser than 1sec).

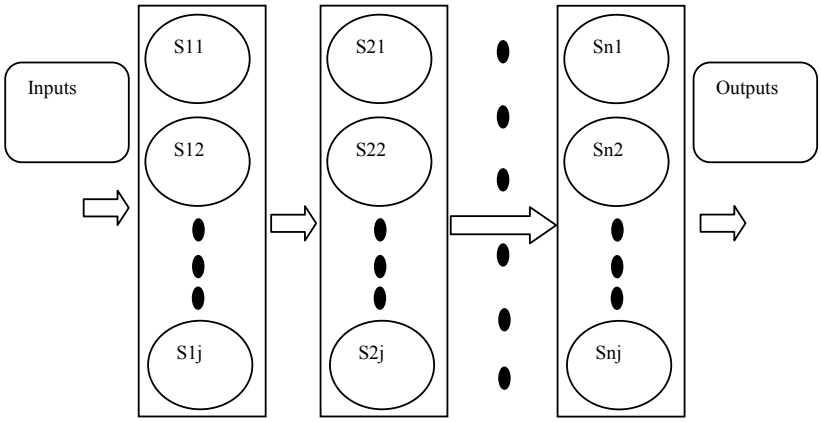



**Fig. 1.** The Qos Aware Selection of services

To solve this issue, we propose a mono objective optimization technique based on bees algorithm [17].

The Bees Algorithm is inspired by the foraging behavior of honey bees. Honey bees collect nectar from vast areas around their hive (more than 10 kilometers). Bee Colonies have been observed to send bees to collect nectar from flower patches relative to the amount of food available at each patch. Bees communicate with each other at the hive via a waggle dance that informs other bees in the hive as to the direction, distance, and quality rating of food sources.

We notice that this meta-heuristic is able to handle a large space of possible solutions (NP Hard problems).

The reminder of the paper is organized as follows: the section 2 presents a survey on the selection problem, the third section formalize the problem, in the fourth section we introduce the developed approach, the fifth section shows the obtained results and finally we present in the sixth section our conclusions and, we give the directions for future work.

## 2 State of the Art

A lot of efforts have been devoted to the QoS service selection, as depicted in figure 2, we distinguish three major classes [4]: the multi-objective optimization and the mono-objective optimization and the hybrid optimization (mono and multi objective). According to [22] we can adopt several database techniques to tackle the multi-objective selection, (or the skyline based optimization) for instance we can use the divide and conquer algorithm, the bitmap algorithm, the index based algorithm (B tree, hash table), and the nearest neighbor algorithm.
Furthermore there are several works which takes into account the user preferences to select the top k dominant skylines [19,20] some of them uses the fuzzy set theory to model the preferences and the dominance relationship, the others uses the pareto-dominance concept to rank the web services.

The mono-objective class involves several approaches, [1,3,5, 6, 7,8,1, 11, 22,24,25,26]. In [14] the authors propose an extensible QoS computation model that supports open and fair management of QOS data. The problem of QOS-based composition is not addressed by this work.

The mono-objective category can use a global selection model [5, 6,15,25, 26] or a local selection model [13, 7] or a hybrid selection model [3].

The global selection model can get the optimal solution, but it has an exponential complexity, however the local model has only a linear complexity but cannot handle the global constraints (there are only, local constraints).

The third category is a compromise of the two approaches, it has a reduced complexity in comparison with the global approach, and it can also handle the global constraints.

The global optimization adopted in[11], uses a genetic algorithm to select a near-optimal composition, The obtained results are very satisfactory but they are less efficient in terms of optimality and execution time, in comparison with the swarm particle optimization[12].

In [25, 26] the authors focus on dynamic and quality-driven selection of services. they adopt linear programming techniques to find the best service components for the composition.

Similar to this approach Ardagna et, [5, 6] extends the linear programming model to include local constraints. Linear programming methods are very effective when the size of the problem is small. Nevertheless these methods suffer from weak scalability due to the exponential time complexity of the applied search algorithms [14].

In [24] the authors use two algorithms that to get near-to-optimal solution. The authors propose two models for the QoS-based service composition problem:
1) a combinatorial model and 2) a graph model.
A heuristic algorithm is introduced for each model. The time complexity of the heuristic algorithm for the combinatorial model (WS HEU) is polynomial, whereas the complexity of the heuristic algorithm for the graph model (MCSP-K) is exponential.

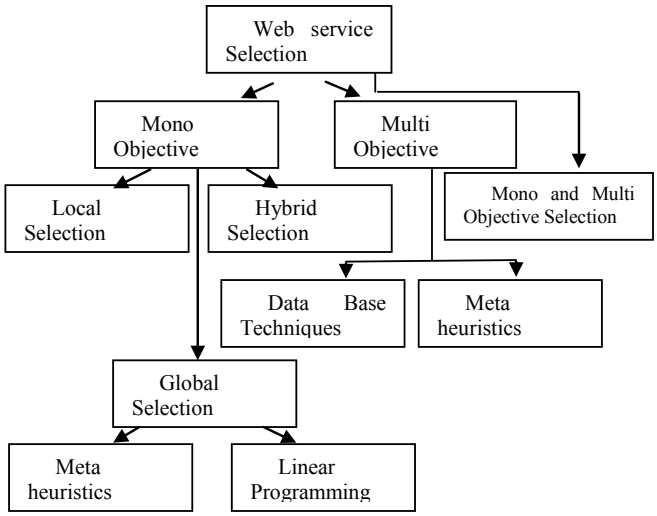



**Fig. 2.** The selection approaches

The hybrid optimization (mono and multi optimization) involves two steps, the first one applies a multiobjective selection in order to get the set of skylines (in each class), the second step refines the obtained set in case to retain a reduced set , this later contains only feasible solutions [4].

Our contribution (Bees algorithm) belongs to the mono-objective class, and more specifically it belongs to the global optimization category. Our choice is motivated by the presence of a set of moving operators (permutations, social exchange, random insertion) that allows the inspection of promising regions of the space search.

The presence of global constraints is handled by the introduction of penalty functions. If a global constraint is violated by a solution then we introduce a penalty quantity on the objective function.

## 3 The Problem Statement

### 3.1 The QOS Modeling

The user's request, is composed of n abstract classes of web services. Each service class Sj ∈ S (e.g. car purchase services) is used to describe a set of functionally-equivalent web services.

In this paper we assume that the information about service classes is managed by a set of service facilitators[13]. Web services can join and leave service classes at any time by means of a subscription mechanism. It is worth noting to distinguish between

an abstraction composition (also called solution template) noted (S1,S2,…Sn) and concrete composition c (which composed of real instances) noted (Ins1,Ins2….Insn)

This later can be obtained by binding each abstract service class Si to a concrete web service Insj , such that Insj ∈ Si. We use c to denote a concrete composite service.

We suppose that we have R quantitative QOS values for a service, (we don't use qualitative values) [25]. These criteria can include generic QOS attributes such as cost, response time, availability, etc, or domain-specific properties, such as bandwidth for multimedia applications.

We use the vector Q = {Q1(s), . . . , QR(s)} to represent the QOS attributes of a service s, where the function Qi(s) determines the value of the i-th quality attribute of s. The QOS values can be either collected from service providers directly (e.g. price), recorded from previous execution monitoring (e.g. response time) or from social networks or feedbacks (e.g. reputation). The set of QOS attributes involves two subsets: positive and negative QOS attributes.

The values of positive attributes need to be maximized (e.g. reliability, availability…), however the value of negative attributes need to be minimized (e.g. price, response time). To homogenize these criteria, we convert the negative attributes into positive attributes by multiplying their values by -1.

The QOS value of a composite service depends on the QOS values of its components as well as the composition model used (e.g. sequential, parallel, conditional and/or loops). In this paper, we consider only the sequential model. The Other models can be treated by using other Techniques [8].

The QOS vector for a composite service c is defined as QOS'(c) = {Q'1(c), . . Q'R(c)}. Q'i(c) represents the value of the i-th QOS attribute of c and can be aggregated from the QOS values of its component services. For this purpose we use three types of functions (addition, multiplication, Min,Max), inspired from [22] (see the table 1):

**Table 1.** The Qos aggregation functions

| QOS Criterion | Aggregation function |
|---|---|
| Response Time | Q1' (C) =$\sum_{j=1}^{n}$ Q1(sj) |
| Reputation | Q2' (C) = 1/n * $\sum_{j=1}^{n}$ Q2(sj) |
| Price | Q3' (C) = $\sum_{j=1}^{n}$ Q3(sj) |
| Reliability | Q4' (C) = $\prod_{j=1}^{n}$ Q4(sj) |
| Availability | Q5' (C) = $\prod_{j=1}^{n}$ Q5(sj) |

### 3.2 The Global constraints

The Global QOS constraints may be expressed in terms of upper and/or lower bounds for the aggregated values of the different QOS criteria. As mentioned in the section A, we consider only positive QOS criteria. Therefore we have only lower bound constraints. Let CONS= {cons1, ..$cons_m$…, $cons_R$ }, $0 \leq m \leq R$, be a vector of global

constraints (CONS is a vector of real values). Let c a concrete composition, in which a concrete web service is associated for each stage. We say that c is feasible iff :

QOS' (c) ≥ CONS , This means that all the global constraints are satisfied.

### 3.3 The Objective Function

In order to rank the service compositions, we need an objective function that associates a real value to the QOS vector of a composition c.

In this work we adopt a mono-objective approach (ie a single objective function), this later is designed as a Weighted sum of the different QOS values [23]. The score computation involves a scaling process of the QOS attributes' values, to allow a uniform measurement of the multi-dimensional service qualities. The scaling step is then followed by a weighting process which models the user priorities. The scaling process of the QOS values gives a score comprised between 0 and 1.

Formally, the minimum and the maximum aggregated values of the k-th QoS attribute of c are computed as follows:

$Qmin'(k) = \sum_{j=1}^{n} Qmin(j,k)$ …… (1)

$Qmax'(k) = \sum_{j=1}^{n} Qmax(j,k)$

$Qmin(j, k) = \min_{\forall sji \in Sj} Qk(sji)$ ……..(2)

$Qmax(j, k) = \max_{\forall sji \in Sj} Qk(sji)$

Where Qmin(j, k) is the minimum value (e.g. minimum price) and Qmax(j, k) is the maximum value (e.g. maximum price) contained in the service class Sj. The utility of a component web service s ∈ Sj is computed as follows:

$U(s) = \sum_{k=1}^{R} wk * (Qk(s) - Qmin(j, k))/(Qmax(j, k) - Qmin(j, k))$ ……...(3)

And the overall utility of a composite service c is computed as follows

$U'(c) = \sum_{k=1}^{R} wk * ( (Q'k(c) - Qmin'(k))/(Qmax'(k) - Qmin'(k))$…… (4)

with wk ∈ R+ and $\sum_{k=1}^{R}(wk) = 1$ are the weights (importance) of Q'k .

A concrete composition c is optimal if it is feasible and if it has the maximum value of the function U'. We notice that the selection of the optimal concrete composition is NP hard (we must enumerate an exponential number of cases). The selection problem addressed here is formalized as follows:

Let CA = {S1, . . . , Sn} be a client request (or an abstract composition) And Cons a vector of R global QOS constraints {cons1,.. , consR}. We must find a concrete composition c={s1, . . . , sn} by binding each Sj to a concrete service sj' ∈ Sj such that:

1. U'(c) is maximized, and

2. $Q'_k(c) \geq Cons(k), \forall Cons(k) \in CONS$

The objective function of the optimization algorithm 'f(x)', combines the function U'(x) and a penalty function p(x).

p(x) decreases the utility score 'f(x)'of the solution that violates the global constraints. Several penalty functions are proposed in the literature [2,21], (we have static, dynamic, adaptive.. functions), for the sake of simplicity we have chosen a static function, because the two others does not give a significant improvement (they increase only the execution time):

$P(c) = -\sum_{k=1}^{R} (D_k)^2(c)$ Where

$$D_k(c) = \begin{cases} 0 \text{ if } Q'_k(c) \geq Cons(k) \\ |Q'_k(c) - Cons(k)| \text{ otherwise} \end{cases}$$

This formula means that a rigorous penalty is applied when we have a solution that violates a constraint. Finally the objective function is defined as follows:

$f(x) = U'(x) + p(x)$

## 4 The Proposed Approach

The objective of the bees algorithm [17] is to locate and explore good sites within a problem search space. Scouts are sent out to randomly sample the problem space and locate good sites. The good sites are exploited via the application of a local search, where a small number of good sites are explored more than the others. Good sites are continually exploited, although many scouts are sent out each iteration always in search of additional good sites.

In this study, we suppose that the bee's position is a vector that contains 10 elements (or classes), each element denotes a service identifier. The bees algorithm and its explanation are given below:

1-Initialize the population, P, of Bees such that the position xi( t ) of each bee $Pi \in P$ is random within the hyperspace,( with t = 0). Bees_number=|P|.

2. while (t<=Tmax) do

begin

a. Evaluate the performance f(xi(t)) of each bee, using its current position xi(t).

b. bee_best= get_best_solution(P)

c. next_generation= ∅

d. sites_best=selectBestSites(P,sites_number)

e. for each $Sitei \in Sites\_best$ do

if i < Elite_Sites_number then

neighborhood=create_NeighborhoodBee(Sitei, patch_size,e_bee)

next_generation= next_generation ∪ get_best(neighborhood)

else

neighborhood=create_NeighborhoodBee(Sitei, patch_size,o_bee)

next_generation= next_generation ∪ get_best(neighborhood)

end

f. Remaining_Bees_number ← (Bees_number- Sites_number);

g. for j =1 to RemainingBeesnum do

NextGeneration ←NextGeneration ∪ CreateRandomBee();

end

h. P ← NextGeneration;

i. Move to the next iteration: t = t + l

end_while

3. Return bee_best

The main loop involves several steps:

In steps a and b, we sort the population according to the fitness, and we memorize the best position 'bee_best'.

The step c initializes the next generation with an empty set.

In step d we retain the bees that have the highest scores in terms of fitness.

In step e we decompose the precedent set into two subsets: the elite_sites and the others, the elite_sites have the highest scores, and the others have the weakest scores.

For each element of elite_sites, we generate e_bee neighbors and we insert the best of them in the next generation.

For each element of others, we generate o_bee neighbors and we insert the best of them in the next generation.

In steps f and g, we generate the remaining bees randomly ie (remaining _bees=|P|-|sites_best|)

The step h updates the population

In step I we increment the iteration number and we execute the loop until the exhaustion of the iterations.

## 5 The Experimentation

The effectiveness of our approach has been tested on a benchmark inspired from [22] we have 10 classes of services, and each class contains 100 instances, the total number

of candidate solutions= $100^{10}$. The number of QOS attributes is fixed to 5. The QOS value of each attribute is generated by a uniform random process which respects the bounds specified in the following table.

All the criteria have the same priority ($w_k$=0.2).

**Table 2.** The Selection Data Set

| QOS Criterion | Class1 | …. | Class10 |
|---|---|---|---|
| Response Time | 0-300(s) | …. | 0-300(s) |
| Reputation | 0-5 | …. | 0-5 |
| Price | 0-30($) | …. | 0-30($) |
| Reliability | 0.5-1.0 | …. | 0.5-1.0 |
| Availability | 0.7-1.0 | …. | 0.7-1.0 |

The bees algorithm parameters are changed to get the best results (in terms of optimality), after several experimentations we have chosen this configuration (that gives the best result):

1- Bees_number= 50.

2- Sites_number=5.

3- Elite_sites_number= 2

4- patch_size = 3

5- e_bees = 7

6- o_bees = 2

3-The maximum number of iterations: Tmax$\in$ [100, 1000],

We notice that the response time will not be reasonable, if we exceed 1000 iterations.

We define the "optimality rate", as the ratio between the fitness (the objective function) of the current solution and the fitness of the optimal solution.

The optimal solution's fitness for this base, is equal to 0.78, therefore the optimality rate, of a solution 'a' is f(a)/0.78.

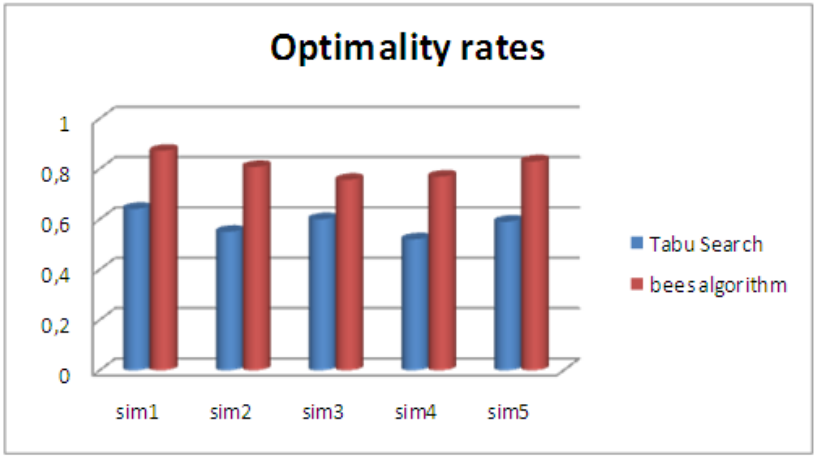



**Fig. 3.** The optimality rates

Several simulations have been made according to the mentioned configuration; the results are resumed as follows:

The figure 3 shows 05 simulations of the bee algorithm, the optimality rate, is close to 87% . This result confirms the efficiency of the proposed approach.

We notice also that the tabu search is less efficient than the bee algorithm in terms of optimality [10]. This fact can be explained as follows:

-First of all, the tabu search moves only one point according the current neighborhood, however the bee algorithm changes several points according to several neighborhoods.

-Besides, the random scouts enable the avoidance of local optimums, and consequently, they can explore promising regions in the hyperspace.

- The permutation is the unique moving operator in the tabu search, however the bees algorithm can access each position which belongs to the patch size.

## 6 Conclusion

In this paper, we have investigated the problem of QOS service selection,. Our solution is based on the bee optimization algorithm. The proposed algorithm handles efficiently a large space of solutions, in order to find a near optimal composition that satisfies the QOS requirements and the end to end user's constraints. The performance can reach more than 87% of optimality.

For future work, we are considering alternative implementations of the framework to improve the performance, for instance we can merge local search whith other moving opearators, such as crossover, hypermutation, reproduction.

## References


1. M. M. Akbar, E. G. Manning, G. C. Shoja, and S. Khan. Heuristic solutions for the multiple-choice multi-dimension knapsack problem. In Proceedings of the International Conference on Computational Science-Part II, pages 659–668, London, UK, 2001. Springer-Verlag.
2. J. Adeli, H. and Cheng, N.T. Augmented lagrangian genetic algorithm for structural optimization, Journal of Aerospace Engineering, 7, 104-118, 1994.
3. E.Alrifai , T. Risse Combining Global Optimization with Local election for Efficient QoS-aware Service Composition In WWW09, April 20–24, 2009, Madrid, Spain.
4. E.Alrifai, T. Risse Selecting Skyline Services for QoS-based Web Service Composition In Proceedings of the WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
5 D. Ardagna and B. Pernici. Global and local qos constraints guarantee in web service selection. In Proceedings of the IEEE International Conference on Web Services, pages 805–806, Washington, DC, USA, 2005. IEEE Computer Society.
6. D. Ardagna and B. Pernici. Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering, 33(6):369–384, 2007. Dustdar, S. and Schreiner, W. 'A survey on web services composition', Int. J. Web and Grid Services, Vol. 1, No. 1, pp.1–30. (2005).
7. B. Benatallah, Q. Z. Sheng, A. H. H. Ngu, and M. Dumas. Declarative composition and peer-to-peer provisioning of dynamic web services. In Proceedings of the International Conference on Data Engineering, pages 297–308, Washington, DC, USA, 2002. IEEE Computer Society.
8. J. Cardoso, J. Miller, A. Sheth, and J. Arnold. Quality of service for workflows and web service processes. Journal of Web Semantics, 1:281–308, 2004.
9. F.Curbera, F.Duftler, R. Khalaf, W.Nagy, N. Mukhi, and S.Weerawarana .Unraveling . the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. IEEE Internet Computing, 6(2). (2002).
10. F. Hadjila, Chikh A QoS-aware Service Selection Based on Tabu search In Proceedings of JEESI'12 Alger, Algeria 2012.
11. F. Hadjila, Chikh A, M. Dali Yahiya QoS-aware Service Selection Based on Genetic Algorithm In Proceedings of CIIA'11 Saida Algeria 2011.
12. F. Hadjila, Chikh A, M. Merzoug, Z Kameche QoS-aware Service Selection Based on swarm particle optimization In Proceedings of IEEE ICITES'12 Sousse Tunisia 2012
13. F. Li, F. Yang, K. Shuang, and S. Su. Q-peer: A decentralized qos registry architecture for web services. In Proceedings of the International Conference on Services Computing, pages 145–156, 2007
14. I. Maros. Computational Techniques of the Simplex Method. Springer, 2003.
15. G. L. Nemhauser and L. A. Wolsey. Integer and Combinatorial Optimization. Wiley-Interscience, New York, NY, USA, 1988.
16. OASIS. Web services business process execution language, April 2007. http://docs.oasis-open.org/ wsbpel/2.0/wsbpel-v2.0.pdf.
17. D. T. Pham, Ghanbarzadeh A., Koc E., Otri S., Rahim S., and M.Zaidi. The bees algorithm - a novel tool for complex optimisation problems. In Proceedings of IPROMS 2006 Conference, pages 454–461, 2006
18. D. Pisinger. Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen, Dept. of Computer Science, February 1995.
19. D. Skoutas, D. Sacharidis, A. Simitsis, V. Kantere, and T. K. Sellis. Top- dominant web services under multi-criteria matching. In EDBT, pages 898–909, 2009.
20. D. Skoutas, D. Sacharidis, A. Simitsis, and T. K. Sellis. Ranking and clustering web services using multicriteria dominance relationships. IEEE T. Services Computing, 3(3):163–177, 2010.

21.O Yeniay penalty function methods for constrained optimization with genetic algorithms journal of Mathematical and Computational Applications, Vol. 10, No. 1, pp. 45-56, 2005.
22.Q Yu, A Bouguettaya. Foundations for Efficient Web Service Selection Springer Science+Business Media, 2010. ISBN 978-1-4419-0313-6.
23.K. . P. Yoon and C.-L. Hwang. Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences). Sage Publications, 1995
24.T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end qos constraints. ACM Transactions on the Web, 1(1), 2007.
25.L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality driven web services composition. In Proceedings of the International World Wide Web Conference, pages 411–421, 2003.
26.L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. IEEE Transactions on Software Engineering, 30(5):311–327, 2004.

# Une approche évolutionnaire pour l'exploration collaborative optimisée d'un environnement inconnu

Amine BENDAHMANE, Abderrahmane BENDAHMANE, Abdelkader BENYETTOU

Laboratoire SIMPA, Département d'informatique, USTOMB
BP 1505 EL M'Naouer 31000 Oran, Algérie

*Résumé.* Ce travail s'inscrit dans le cadre de la robotique collective. L'objectif est d'utiliser plusieurs robots en collaboration pour la couverture totale d'un environnement inconnu. Pour cela, chaque robot explore l'environnement indépendamment des autres et le cartographie par la méthode des grilles d'occupation. Un algorithme génétique est utilisé pour générer un chemin de déplacement optimisé permettant de maximiser l'espace visité tout en évitant les obstacles et les collisions avec les autres robots. Les résultats obtenus lors des tests en simulation ont validé l'efficacité de l'approche utilisée.

*Mots clés :* collaboration multi-robots, exploration, grilles d'occupation, algorithme génétique, optimisation de chemin.

## 1 Introduction :

La robotique collective se base sur l'utilisation de plusieurs robots pour effectuer une tâche commune (explorer une région, ramasser des objets... etc.) [1]. Cette collectivité implique la collaboration ou la coopération des robots entre eux, et offre des avantages considérables. Parmi ces avantages on peut citer le gain de temps, le gain de performances et le gain de robustesse [2][3].
Cependant, un tel système présente aussi de sérieux inconvénients, dont principalement: le risque d'interférences [4] et la difficulté de coordonner et synchroniser les robots entre eux. Ceci suppose de mettre en œuvre des protocoles de communication complexes [1].
Toutefois, certains chercheurs préfèrent éviter ce problème. Dans ce cas chaque robot agit individuellement et traite les autres robots comme des objets qui font partie de l'environnement. Il y aura donc émergence d'un comportement collaboratif à partir d'un collectif d'agents réactifs. Ce comportement émergent est souvent appelé intelligence en essaim (*swarm inteligence* [5]) en analogie aux comportements sociaux de certains insectes comme les colonies de fourmis.
Notre travail s'inscrit dans le cadre de l'exploration collective d'une zone inconnue. Cette tâche est considérée comme l'une des plus anciennes applications de la robotique mobile. Sa problématique essaie de répondre à la

question: comment construire une carte de l'environnement tout en utilisant cette carte pour planifier le déplacement du robot? [6]
En d'autres termes, le robot se déplacera dans une zone inconnue en utilisant une carte partielle du périmètre exploré, cette carte sera construite à fur et à mesure que le robot se déplace dans l'environnement. Cette problématique réunie les tâches de cartographie, de planification et de navigation.
La navigation regroupe l'ensemble des stratégies utilisées par le robot pour se déplacer dans l'environnement tout en évitant les obstacles [7]. La planification consiste à générer un chemin optimal permettant au robot de se déplacer d'un point A à un point B en un minimum de temps [8]. Quant à la cartographie, son but est de permettre à un robot de créer un modèle interne de l'environnement à partir de ses capteurs [9], ce modèle lui permettra par la suite de s'auto-localiser à partir de ses observations.
Ces trois tâches sont tellement liées ensemble qu'il est difficile voire impossible de les traiter séparément.
Les chercheurs dans ce domaine ont donné une grande importance à ces trois problématiques (qui forment aujourd'hui un axe de recherche à part entière). Les progrès réalisés dans ce secteur ont permis la création de robots pour l'exploration spatiale, l'exploration sous-marine, ainsi que l'exploration terrestre dans les zones dangereuses. Mais les recherches dans ce domaine ne se sont pas limitées seulement aux grandes expéditions scientifiques. Les aspirateurs autonomes et les tondeuses automatiques sont deux exemples de robots exploreurs bon marché destinés au large public.
Le but de notre travail est de faire collaborer un groupe de robots autonomes pour cartographier un environnement inconnu tout en maximisant la surface totale balayée (couvrir toutes les régions) et en minimisant l'énergie consommée (éviter de repasser par la même zone plusieurs fois).
Dans les sections 2 et 3 nous allons présenter plus en détail les problématiques de cartographie et d'exploration collective tout en citant quelques travaux connus dans ce domaine. Les objectifs de notre étude seront exposés dans la section 4, tandis que les sections 5 et 6 détailleront le fonctionnement des méthodes utilisés dans notre approche. Les résultats des tests seront discutés dans la 7ème section avant de conclure le travail dans la dernière section tout en citant quelques pistes pour de futurs travaux.

## 2 Cartographie d'un environnement inconnu:

La cartographie est généralement considérée comme une fonction nécessaire dont un robot autonome devra être doté [9]. Elle lui permet grâce aux observations acquises par ses capteurs de représenter l'espace physique qui l'entoure sous forme de modèle numérique exploitable par son processeur.

Ce modèle est essentiel lors de la tâche de localisation ainsi que la tâche de planification de trajectoire.
On distingue deux grandes catégories de modèles [9]: D'une part, il y a les cartes métriques où sont représentées les propriétés géométriques de l'environnement, elles sont faciles à construire mais nécessitent la connaissance de la position exacte du robot.
D'autre part, on trouve les cartes topologiques sous forme de graphes (dont les nœuds représentent les lieux d'intérêt et les arcs représentent les chemins possibles entre ces lieux). Ces cartes sont plus difficiles à construire et à mettre à jour, mais sont plus adaptés à la planification de trajectoire.
La méthode des *grilles d'occupations* [10] appartient à la première catégorie. Dans cette méthode, l'environnement est modélisé sous forme d'une grille à deux dimensions, dont chaque cellule contient l'évidence que la zone associée soit occupée ou non par un objet. Un de ces avantages est qu'elle peut combiner des informations recueillies par plusieurs types de capteurs [11].
Cette méthode fut utilisée par les auteurs de [2] et [12] pour planifier le déplacement de plusieurs robots lors d'une tâche d'exploration, elle fut aussi utilisé par ceux de [11] pour créer deux cartes: une carte à court-terme (locale) utilisée pour la localisation, et l'autre à long-terme (globale) utilisée pour la navigation.
Le problème avec cette approche est qu'il est nécessaire de savoir la position exacte du robot lors de la construction de la carte. Ce qui pose problème quand on ne connait pas la position de départ du robot dans l'environnement. Un nouveau concept est donc apparu dans le milieu des années 80 sous le nom de SLAM: *Simultaneous Localisation And Mapping* (localisation et cartographie simultanées).
Le problème du SLAM consiste à localiser le robot à partir d'une position de départ inconnue dans un environnement inconnu et en même temps créer la carte de cet environnement [13]. L'une des premières méthodes qui ont permis de satisfaire à ce genre de problématiques a été proposée par *Smith* et *Cheesman* [14] [15] sous le nom d'algorithme de cartographie stochastique. Cette méthode utilise le filtre de *Kalman* qui permet d'obtenir une estimation robuste de la position du robot par rapport à des amers virtuels [13].
Un autre avantage de cette technique est qu'elle offre une cartographie topologique (efficace pour la planification de trajectoires). Cependant, elle nécessite d'effectuer plusieurs manipulations matricielles, et requiert donc plus de temps de calcul que la méthode des grilles d'occupation.

## 3 Exploration multi-robots :

Le problème d'exploration d'un environnement connu ou inconnu peut être vu comme un problème de maximisation de la surface visitée par le robot.

Une couverture efficace maximise la zone balayée par le robot tout en minimisant le temps nécessaire pour ce balayage.
Une des méthodes les plus utilisées dans ce genre de problématiques est la *Boustrophedon Decomposition* [16].
Cette méthode consiste à décomposer l'environnement en plusieurs cellules puis à couvrir chaque cellule en utilisant un déplacement en zigzag. Ceci marche bien quand l'environnement est déjà cartographié mais présente quelques difficultés dans les environnements inconnus. Cette méthode a été adaptée pour plusieurs robots en collaboration [17], [18].
Récemment, les auteurs de [19] ont utilisé une cartographie à base de grilles d'occupation en modélisant l'environnement sous forme d'un graphe de recherche, un chemin de couverture est alors généré à partir de ce graphe en utilisant un algorithme D* modifié. Cependant, cet algorithme n'a pas été étendu pour l'utilisation de plusieurs robots.
Un travail plus ancien basé sur les grilles d'occupation [12] à montré que les robots se déplaçant en coopération permettaient d'explorer l'environnement plus rapidement par rapport aux robots se déplaçant indépendamment les uns des autres.
En ce qui concerne les travaux dans le domaine de la robotique en essaim, une heuristique bio-inspirée à été utilisée par *Wanger* et ses collègues [20], dans cette approche les robots (imitant le comportement des fourmis) déposent des phéromones virtuelles dans une cellule pour la marquer comme étant visitée.
Une autre approche en essaim a été décrite dans [21], elle ne requiert aucune cartographie ni aucune communication, chaque robot se déplace tout simplement dans une direction différente des autres, ceci est plus efficace qu'un déplacement aléatoire puisque les robots ont tendance à se disperser dans l'environnement.
Par contre, les robots utilisés par *Dasgupta* et *Cheng* [22] préfèrent se regrouper de temps en temps pour échanger des informations. Les résultats expérimentaux ont montré que cette technique améliore la qualité du balayage de 5 à 20% comparé à la méthode précédente.

## 4 Notre travail :

Le but de notre travail est la couverture totale d'un environnement inconnu par un groupe de robots mobiles autonomes. Pour cela, chaque robot devra explorer l'environnement indépendamment des autres et le cartographier par la méthode des grilles d'occupation. Une carte globale sera construite simultanément par la fusion des cartes locales produites par les robots.
Utilisant ses capteurs, et connaissant sa position dans la carte globale, le robot utilisera un algorithme évolutionnaire pour la génération d'un chemin optimisé qui maximise l'espace exploré tout en évitant les obstacles et les collisions

avec les autres robots. Le résultat sera l'émergement d'un comportement collaboratif pour l'exploration de l'environnement en utilisant les informations recueillies par l'ensemble des robots.
L'objectif final reste de visiter toutes les places vides de l'environnement et éviter de repasser par les mêmes zones plusieurs fois, minimisant ainsi le temps nécessaire pour le balayage.

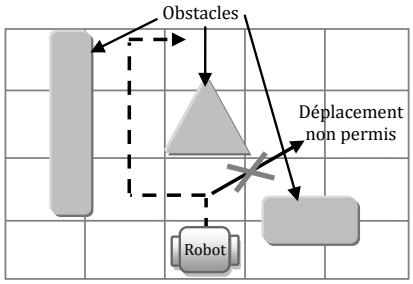



Fig. 1. Décomposition de l'environnement en cellules de taille fixe.

## 5 Modélisation:

Créer un modèle de l'environnement est essentiel lors de l'exploration d'une zone inconnue. Cela permet au robot de déterminer s'il a déjà visité une région ou pas encore. Cependant, cette tâche requiert de devoir représenter le milieu où évolue le robot de manière très précise.
Dans la méthode des grilles d'occupation l'environnement est décomposé en cellules de taille fixe (*voir Figure 1*). Le robot utilise ses capteurs pour localiser les obstacles (objets, murs… etc) et les représente en interne dans une grille.
Chaque cellule de cette grille contient l'évidence que la région correspondante dans l'environnement soit vide ou non. Cette évidence (valeur comprise entre -1 et 1) est bien entendu affectée par la quantité de bruits présents dans les mesures des capteurs, et permet de calculer la probabilité d'occupation de la zone correspondante.
Les robots utilisés dans notre approche sont dotés de capteurs laser. Ce type de capteurs est largement utilisé dans la robotique mobile (surtout dans les environnements d'intérieur) à cause de leur coût réduit et la facilité de traitement des informations qu'ils offrent. Ils permettent de calculer la distance entre le robot et l'obstacle; cependant, ils n'offrent aucune information sur la nature de l'objet détecté. Par conséquent, quand un robot détecte un autre robot, il le considère comme étant un obstacle qu'il faut éviter. Au final, ce comportement n'est pas un inconvénient, car il permet d'éviter les collisions entre les robots pendant leurs déplacements.

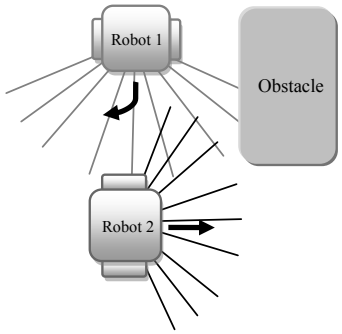



Fig. 2. Processus d'évitement d'obstacles (navigation)

Quand le robot détecte un objet à $d$ mètres de sa position actuelle ($x_r$, $y_r$) sur l'angle $\theta$, il calcule la position de cet obstacle par rapport à lui et modifie la grille. Ce calcul se fait en utilisant une formule géométrique très simple.

$$\begin{cases} x_{obs} = x_r + d\cos\theta \\ y_{obs} = y_r + d\sin\theta \end{cases}$$

Une fois les mises à jour de la grille effectuées, le robot les envoie à l'agent central qui procède à la fusion des cartes. Durant ce processus, les probabilités d'occupation sont recalculées selon la grille locale de chaque robot. Une carte globale est donc construite à la fin de cette opération. La *Figure 3* résume le schéma général de l'approche utilisée.

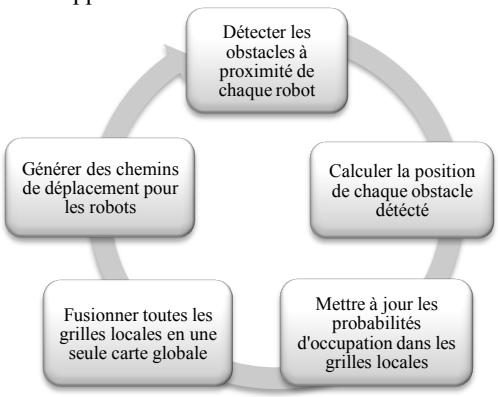



Fig. 3. Schéma général de la méthode des grilles d'occupation dans un environnement multi-robot

Etant donné que le chemin de déplacement du robot est mis à jour à chaque itération, la méthode gagne en robustesse, parce que chaque robot génère un nouveau chemin de déplacement selon les changements effectués sur la carte. En d'autres termes, l'algorithme marche bien dans le cas d'environnements dynamiques où les obstacles peuvent changer de position fréquemment (objets déplacés, portes ouvertes/fermées, personnes se déplaçant dans l'environnement...etc).
Notons aussi que si la vitesse du robot est trop grande, il ne pourra pas s'arrêter instantanément lorsqu'il perçoit un obstacle (à cause de l'effet de glissement entre le sol et les pneus), ce qui conduit généralement à des collisions. Si, par contre, la vitesse du robot est trop petite, son déplacement sera lent et le temps d'exploration sera grand.
La solution est de trouver un compromis entre la vitesse de déplacement et un bon évitement d'obstacles, ce qui ne peut être déterminé qu'expérimentalement en effectuant plusieurs tests.

## 6 Planification de chemin et prise de décision :

Planifier le chemin optimal qui couvre toutes les cellules de la grille revient à résoudre le problème de voyageur de commerce (*Traveling Salesman Problem*); par conséquent, rechercher la solution exacte est à bannir si on veut résoudre des problèmes de taille réaliste. Pour cette raison, des méthodes approchées sont généralement utilisées dans ce genre de situations. Dans notre travail nous allons utiliser un algorithme évolutionnaire pour la planification du chemin.
Les recherches sur les algorithmes évolutionnaires ont commencé avec les travaux de Holland [23] puis ceux de Goldberg [24] sur les algorithmes génétiques (AGs). Ce sont des méta-heuristiques d'optimisation globales, bio-inspirées, basées sur des populations. Une population est constituée d'un certain nombre d'individus, eux-mêmes constitués d'un ensemble de gènes.
Les algorithmes évolutionnaires ont rapidement été utilisés en robotique à cause de leurs capacités d'adaptation et de généralisation, le programmeur n'a plus besoin de déterminer l'action à effectuer dans chaque situation puisque le comportement du robot s'adapte à son environnement automatiquement pendant l'évolution. La *Figure 4* résume les étapes générales d'un algorithme génétique.

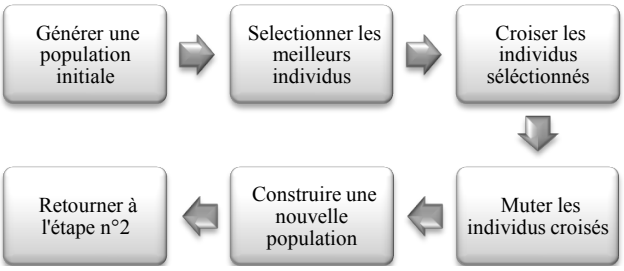



Fig. 4. Schéma global d'un algorithme génétique

Dans notre travail, un individu représente un chemin possible que le robot peut prendre. Il parait donc logique que chaque gène représente une direction à prendre puisqu'un chemin n'est par définition qu'un ensemble de directions à suivre.
La qualité de cet individu est donnée par le calcul d'une fonction fitness. L'objectif est de visiter le maximum de cellules vides de la grille en évitant de repasser par les mêmes cellules plusieurs fois. Mathématiquement, cela revient à choisir le chemin qui contient le maximum de cellules ayant la valeur 0 dans la grille (cellules vides non visitées) et minimiser le nombre de cellules ayant la valeur 1 (cellules déjà visitées). La fonction d'évaluation peut donc prendre la forme suivante :

$$fitness(chemin) = \sum_{i=1}^{n} grille\ (chemin_X(i), chemin_Y(i))$$

où : $n$ = longueur du chemin.
$chemin_X(i)$ et $chemin_Y(i)$ représentent les coordonnées de la $i^{ème}$ cellule appartenant au chemin.

En sachant que la grille à été modélisée de la façon suivante :

$$grille(x,y) = \begin{cases} 0 & si\ espace\ non\ explor\acute{e} \\ 1 & \text{si } espace\ d\acute{e}ja\ explor\acute{e} \end{cases}$$

En résumé, la fonction fitness affecte à chaque chemin une pénalité. Le but de l'algorithme évolutionnaire est de minimiser cette pénalité à travers les itérations. Notons que le nombre d'itérations doit rester petit pour pouvoir prendre la décision en temps réel. Dans le cas contraire le calcul du chemin prendra trop de temps, ce qui aura pour effet de ralentir l'exploration.

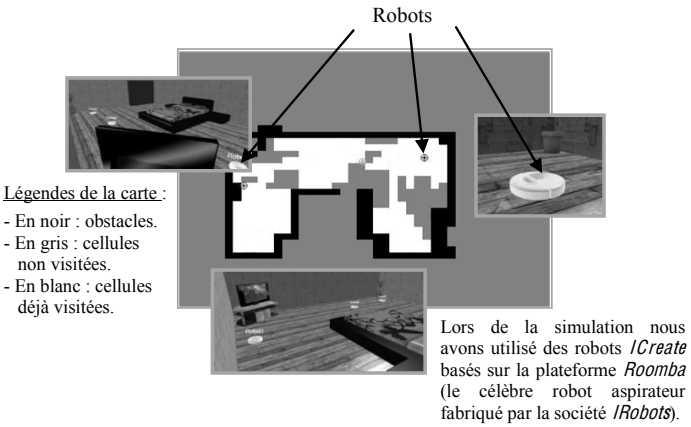



Fig. 5 : Captures d'écran lors des tests en simulation

## 7 Simulation et résultats :

Pour évaluer notre approche on a procédé à une série de tests en utilisant le simulateur *Microsoft Robotics Developer Studio*.
Ce simulateur offre une panoplie d'outils très utiles pour le développement et le débogage d'applications robotiques. Il offre aussi la possibilité d'exécuter ces applications dans un environnement de simulation 3D, et ajoute du bruit artificiel dans les mesures des capteurs de telle sorte que le test soit le plus proche possible des conditions réelles.
Utiliser un simulateur avant la mise en œuvre réelle est très bénéfique. En effet, les simulateurs permettent d'essayer rapidement différentes positions pour les capteurs, mettre en place des scénarios complexes et aussi éviter d'endommager le matériel pendant les essais.
La première série de tests à été effectuée dans un environnement de simulation représentant une pièce de 50m² en utilisant un *AG* pour la prise de décision avec les paramètres suivants: 70 générations, 70 individus et 30 gènes. Les résultats obtenus (présentés dans la *Figure 6*) ont clairement montré qu'utiliser plusieurs robots en collaboration permettait d'accélérer le balayage de la zone, ceci par le simple fait que les robots partagent leur travail pour aller plus vite.

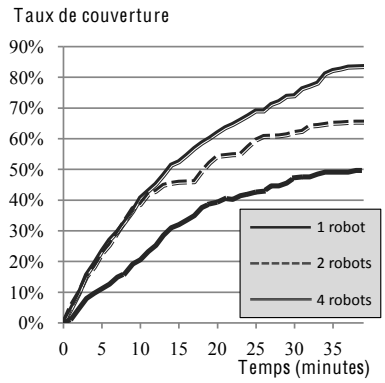



Fig. 6. Première série de tests

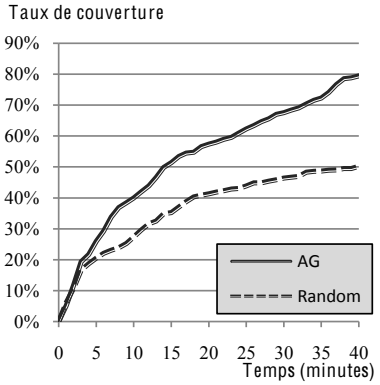



Fig. 7. Deuxième série de tests

Paramètre de l'AG :
- type de sélection: roulette stochastique.
- type de croisement: single point.
- probabilité de croisement : 0.7

La deuxième série de tests avait pour objectif de mesurer la contribution de l'algorithme évolutionnaire dans la qualité de la prise de décision. La *Figure 7* montre que l'utilisation des *AGs* donne des taux de couverture nettement supérieurs à une approche aléatoire.
L'explication de cette amélioration est simple: L'algorithme évolutionnaire utilisé dans notre approche avait pour rôle d'optimiser la génération de chemins de déplacement pour les robots. En utilisant une fonction fitness favorisant la couverture des cellules non encore visitées, il a poussé les robots à se diriger vers ce type de cellules et du coup maximiser le taux de couverture. Cependant, le calcul d'un nouveau chemin devant se faire en temps réel, il a été nécessaire de réduire le nombre de générations ainsi que la taille de la population pour que la prise la décision soit plus rapide. Les solutions générées n'étaient donc pas optimales mais assez bonnes pour permettre de diminuer le temps nécessaire au balayage total de la zone.

## 8 Conclusion :

Nous avons abordé dans ce travail une des plus actives problématiques de la robotique mobile: Comment un groupe de robots autonomes peuvent collaborer entre eux pour explorer une zone inconnue en maximisant la surface balayée et en minimisant le retour vers les endroits déjà visités? Ce type de problèmes peut trouver son utilité dans les robots aspirateurs ou des robots anti-mines par exemple.

La phase de tests a permis de valider notre modélisation, de montrer l'intérêt de la collaboration et de mesurer la contribution apportée par l'algorithme évolutionnaire dans la prise de décision.
Pour améliorer cette approche, plusieurs pistes potentielles peuvent être étudiées: par exemple étendre la collaboration à la prise de décision (prendre en compte les intensions des autres robots) ou encore utiliser un algorithme d'apprentissage pour améliorer les comportements des robots.
Lors des tests en simulation l'approche a atteint des taux de couverture relativement élevés, mais son implémentation sur des robots réels peut poser quelques problèmes techniques, surtout au niveau du bruit dans les mesures des capteurs ainsi que le type d'algorithme de localisation utilisé.

## 9 Références :


[1] Y.U. Cao, A.S. Fukunaga & A.B. Kahng «Cooperative Mobile Robotics: Antecedents and Directions », *Autonomous Robots 4*, 7–27, 1997.

[2] D. Guzzoni, A. Cheyyer, L. Julia & K. Konolige «Many robots make work short» *AI Magazine* vol. 18 n°1, pp. 55–64, 1997.

[3] G. Dudek, M. Jenkin, E. Milios & D. Wilkes «A taxonomy for multiagent robotics». *J. Autonomous Robots* vol. 3 n°4, pp. 375–397, 1996.

[4] D. Goldberg & M.J. Matari'c «Interference as a tool for designing and evaluating multi-robot controllers» *In Proceeding, AAAI-97*, pp. 637–642, 1997.

[5] S. Hackwood & G. Beni «Self-organizing sensors by deterministic annealing», *IEEE/RSJ IROS*, 1177–1183, 1991.

[6] S. Thrun «Robotic Mapping: A Survey» *Carnegie Mellon University*, 2002.

[7] E. Garcia & P. Gonzalez de Santos «Mobile Robot Navigation with Complete Coverage of Unstructured Environments» J. *Robotics and Autonomous Systems* vol. 46, pp. 195-204, 2004.

[8] K.H. Sedighi, K.Ashenayi, T.W. Manikas, R.L. Wainwright & H.M. Tai «Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm» *in Congress of Evoutionary Compututation* vol. 2, pp. 1338-1345, 2004.

[9] S.Thrun «Robotic Mapping: A Survey» *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufmann, 2002.

[10] H. Moravec & A. Elfes «High resolution maps from wide angle sonar» In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pp. 116-121, 1985.

[11] A.C. Schultz, W. Adams & B. Yamauchi «Integrating Exploration, Localization, Navigation and Planning with a Common Representation» *Journal of Autonomous Robots* vol. 6, pp. 293–308, 1999.

[12] B. Yamauchi «Frontier-Based Exploration Using Multiple Robots» *Proceedings of the Second International Conference on Autonomous Agents*, ACM Press, Minneapolis, 1998.

[13] H. Durrant-Whyte & T. Bailey «Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms» *Robotics and Automation Magazine*, 2006.

[14] R.C. Smith & P. Cheeseman «On the Representation and Estimation of Spatial Uncertainty» *The International Journal of Robotics Research* vol. 5 n°4, pp. 56–68. 1986.

[15] R.C. Smith, M. Self & P. Cheeseman «Estimating Uncertain Spatial Relationships in Robotics» *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*. pp. 435–461, 1986.

[16] H. Choset & P. Pignon «Coverage path planning: the boustrophedon decomposition» *In Proceedings of the International Conference on Field and Service Robotics*, Australia, Dec 1997.

[17] D. Latimer, S. Srinivasa, V. Lee-Shue, S. Sonne, H. Choset & A. Hurst «Towards Sensor Based Coverage with Robot Teams» *In Procee-dings of the IEEE International Conference of Robotics & Automation*, Washington, May 2002.

[18] I. Rekleitisy, V. Lee-Shue, A. Peng New & H. Choset «Limited Communication, Multi-Robot Team Based Coverage» *In Proceedings of the IEEE International Conference on Robotics & Automation, New Orleans*, April 2004.

[19] M. Đakulovic & I. Petrovic «Complete Coverage D* Algorithm for Path Planning of a Floor-Cleaning Mobile Robot» *Preprints of the 18th IFAC World Congress,* Milano, Italy, 2011.

[20] I. Wagner, M. Lindenbaum & A. Bruckstein, «Distributed covering by ant-robots using evaporating traces» *IEEE Transactions On Robotics & Automation,* Vol. 15, No. 5, Oct 1999.

[21] M. Batalin & G. Sukhatme «Spreading Out: A Local Approach to Multi-robot Coverage», *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems,* pp. 373-382, Japan, June 2002.

[22] R. Dasgupta & K. Cheng «Distributed Coverage of Unknown Environments using Multi-robot Swarms with Memory and Commu-nication Constraints» *UNO CS Technical Report (cst-2009-1), University of Nebraska*, 2009.

[23] J.Holland «Adaptation In Natural And Artificial Systems» *Univ of Michigan Press*, 1975.

[24] D. Goldberg «Genetic Algorithms in Search, Optimization and Machine Learning», *Addison-Wesley*, Redwood City, CA, 1989.

# Optimisation dans les réseaux

# Optimizing Bandwidth Usage in Multi-Radio and Multi-Channel Wireless Mesh Networks with Power Control*

Amel Faiza Tandjaoui and Mejdi Kaddour

Laboratory of Computer Science and
Information Technologies of Oran
University of Oran Es-Senia
BP. 1524 El M'Naouer, ORAN

**Abstract.** Routers in wireless mesh networks are generally equipped with several radios. Moreover, they can transmit on different channels, which allows them to transfer, simultaneously, more concurrent traffic through several links by taking advantage of the spectral diversity. Furthermore, in wireless mesh networks, routers have to cover only limited size areas, which permits an important spatial reuse. We propose, in this paper, to take all these elements into account, through a mixed integer linear program. The main aim of the model is to provide a useful tool to reduce, as much as possible, the total width of the frequency band allocated to the whole wireless mesh network, while satisfying users demands in terms of throughput. This is done by considering an adaptive transmission power. We show how several parameters, such as the number of radio interfaces on the gateways are correlated with the needed bandwidth. Hence, our model can be used to estimate the range of frequencies needed to carry a given set of traffic demands.

**Keywords:** Wireless mesh networks, mixed integer linear programming, power adaptation, channel assignment

## 1 Introduction

Wireless mesh networks (WMNs) are an attractive technology for providing last mile access to users, as they are easy to deploy, dynamic, auto-configuring and robust. A WMN is composed of two types of nodes: mesh routers and mesh clients. Mesh clients can have different forms (smart phones, laptops, domestic appliances, etc.), and are generally mobile. At the opposite, mesh routers are often fixed. Mesh routers form the backbone of the network, and have the task to forward and aggregate traffic to/from mesh clients in a multi-hop manner. Moreover, in WMNs some of the mesh routers can have gateway/bridge functionalities, which allow them to be connected to other networks.

In multi-radio and multi-channel WMNs, mesh routers are equipped with several radio interfaces, and can transmit more traffic, simultaneously, through different radio channels. This can really enhance the network performance, if it is managed efficiently. Our main concern in this paper is to propose a mathematical optimization model that captures the most important advantages and constraints that characterize these networks, in order to use optimally one of the most critical

* This work is supported by TASSILI research program 11MDU839 (France, Algeria).

resources in wireless networks: the radio bandwidth. This is done by considering an adaptive transmission power, that aims at avoiding the use of too high powers which causes too much interferences, and which naturally results in wasting the bandwidth.

Our model is valid for any number of gateways, and it can be used to know the right number to use. Also, no particular MAC layer is considered in this model, which allows it to be general, so it can be applied to large number of WMNs.

The remainder of the paper is organized as follows. Related work is presented in Sect. 2. The problem formulation is explained in Sect. 3. Our model can be found in Sect. 4. Some numerical results can be found in Sect. 5. And finally, Sect. 6 concludes the paper.

## 2 Related Work

The interesting features of WMNs make this technology a really attractive one compared to classical wireless networks. They naturally bring, also, new issues, and thus, new research areas.

Proposals for these networks first intended to revisit and adapt traditional protocols, especially, those of 802.11 [1] or ad hoc wireless networks. It has, after that, been shown that specific protocol need to be conceived, due to the differences that characterize WMNs [2].

Many works have been done in the field of the optimization of WMNs. In [3], [4], and [5], the problem of WMNs planning is addressed. In [3], authors handle gateways placement by dividing the network into disjoint clusters, where each cluster comprises one router that will have the role of a gateway. The problem is first formulated as an integer linear programming by minimizing the number of clusters, and taking care of quality of service (QoS) constraints, but it is shown that it is NP-hard. A near-optimal approach, which recursively computes minimum weighted dominating sets, is then proposed.

In [4], a mixed integer linear model, whose objective is to minimize the installation cost, while covering all the mesh clients, is proposed to address the planning problem. In [5], authors considered a multi-objective approach, where the two objectives are the total deployment cost and the network throughput. Authors proposed a population-based meta-heuristic algorithm to solve the problem.

Authors of [6] and [7] addressed the routing problem in WMN by proposing an ant colony optimization based on multicasting and anycasting algorithms, respectively.

The problem of channel assignment in multi-radio and multi-channel WMNs is addressed in [8], [9] and [10]. In [8] and [9], authors formulate this issue by joining it to a routing problem, with the objective of optimizing the overall throughput of the network. [9] evaluates the influence of different genetic operators. In [10], authors develop a heuristic based on the graph coloring principal to assign channels to mesh routers radios, by relying on a system of priority for mesh routers and minimizing maximum interference in the network while maintaining network connectivity.

To the best of our knowledge none of the previous works in the literature has addressed the issue of the frequency bandwidth optimization by considering a power control approach. In this paper, we provide a tool to optimally deploy multi-radio and multi-channel WMNs with the minimum channel bandwidth with respect to throughput conditions and taking into account adaptive power transmission, in contrast to other works like [11], where authors considered a fixed power, our model is targeting for WMNs with multiple gateways. Also, no particular MAC layer is considered, which makes the model applicable to any access scheme.

## 3 Problem Formulation

This section presents the structure of the WMN to be considered in our modeling and the different notations that will be used in the rest of the paper.

We first consider a radio bandwidth divided into $C$ equal width ($w$ MHz), equal capacity ($\theta$ bits/sec), and non-overlapping radio channels. A channel is said to be active if is assigned to at least one link. This is denoted by $z_j$, i.e. $z_j$ is 1 if channel $j$ is active, and 0 otherwise. The optimization model aims to activate the smallest possible number of channels.

In our model, the network is composed of two kinds of nodes: simple mesh routers and gateways. Hence, we represent the network by a directional graph $G = (V \cup V', E)$, where $V$ is the set of the $N$ mesh routers, $V'$ is the set of the $N'$ gateways, and $E$ is the set of activable links between pairs of distinct nodes in $V \cup V'$ (activable links are defined later). The distance between any couple of nodes $u$ and $v$ from $V \cup V'$ is denoted by $d_{u,v}$.

Every mesh router and gateway possesses $n$ and $n'$ radio interfaces, respectively. Every mesh router generates a flow, which is aggregated from its clients. This flow has to be directed toward the outside of the network through one or several gateways. The rate of the flow generated by a mesh router $u$ is denoted by $r_u$.

Data in the WMN have to be routed from mesh routers, to the gateways, through different paths, in a multi-hop manner. Every link $l = (u, v)$, between two nodes $u$ and $v$ in $V \cup V'$, belonging to such a path, is an active link. The quantity of flow carried by this link is denoted by $f_l$.

Our WMN is a multi-channel one, hence, when transmitting data packets, the radio interface of the transmitting end in a link $l$, and the corresponding receiver interface have to choose the channel to use from the $C$ available channels. This channel is denoted by $c_l$.

In this paper, we seek to exploit the adaptability of the transmission power, by avoiding to use high power where it's not necessary, hence, reducing interferences. The transmission power used in link $l$ is denoted by $p_l$.

In order to estimate the strength of the signal perceived by the receiver of a link $l = (u, v)$, we use the notion of signal to interference plus noise ratio (SINR). It is denoted by ($\text{SINR}_l$) and is calculated by taking the ratio of received signal strength from $u$ at $v$ to total noise at $v$; the total noise includes aggregate received signal strength at $v$ from all interfering transmitters on the same channel as well as the ambient noise:

$$\text{SINR}_l = \frac{p_l.d_{u,v}^{-\alpha}}{\eta_0.w + \sum_{l'=(u',v')\in E, l'\neq l, c_{l'}=c_l} p_{l'}.d_{u',v}^{-\alpha}} \tag{1}$$

where $\alpha$ is the path loss exponent and $\eta_0$ is the power spectral density of the thermal noise, and are considered as fixed parameters. Interference between non overlapping channels is neglected.

The value of the SINR has to be equal or greater than a threshold $\beta$, so that the receiver can decode the signal correctly.

### 3.1 Activable Links

The notion of signal to noise ratio (SNR) is the same as SINR, but it considers there is only a transmitter, a receiver, and no interferer. It is given for a link $l = (u, v)$ by:

$$\text{SNR}_l = \frac{p_l.d_{u,v}^{-\alpha}}{\eta_0.w} \quad . \tag{2}$$

Is is used it to determinate the set of activable links, by considering only the links that have a SNR superior to a certain threshold $\beta$.

$\eta_0$, $w$ and $\beta$ are fixed values. Hence, a link between nodes $u$ and $v$ is activable if the distance between the two is less or equal to the value $D_{\max}$:

$$D_{\max} = \sqrt[\alpha]{\frac{P_{\max}}{\eta_0 w \beta}} \tag{3}$$

where $P_{\max}$ is the maximum transmission power. Links having this feature form the set $E$.

## 4 The Optimization Model

In this section, we present in detail our mixed integer linear optimization model (MILP).

### 4.1 Decision variables

**Links Attributs:** Links properties are the first decision variables of the model. Those properties are the ones described in Sect. 3, to which is added another dimension that will be necessary to linearize the model.

Links properties for each activable link $l = (u, v)$, then, will be:

- $c_{j,l}$ : A binary decision variable that takes the value 1 if $l$ is active and assigned to channel $j$, and takes the value 0 otherwise.
- $p_{j,l}$ : A continuous non negative decision variable that takes the value of the power used by the transmitter of link $l$ if this one is active and assigned to channel $j$. It takes the value 0 otherwise. This variable is comprised between 0 and $P_{\max}$.
- $f_{j,l}$ : A continuous non negative decision variable representing the quantity of flow that link $l$ is carrying when this one is active and assigned to channel $j$. This variable is between 0 and $\theta$.
- $A_{j,l}$ : A binary decision variable that will take the value 0 if $\text{SINR}_{j,l} < \beta$, and the value 1 otherwise

**Channel Attributs:** The only decision variable relative to a channel $j$ is the binary decision variable which represents its state, $z_j$.

### 4.2 The Mixed Integer Linear Program

The MILP is given below :

$$\text{minimize} \sum_{j=1}^{C} z_j \tag{4}$$

Subject to:

*Transmission power constraints:*

$$\forall l \in E, \forall 1 \leq j \leq C : \quad p_{j,l}.\frac{d_{u,v}^{-\alpha}}{\eta_0.w} \geq \beta - (1 - c_{j,l})L_1 \tag{5}$$

$$\forall l \in E, \forall 1 \leq j \leq C : \quad p_{j,l} \leq c_{j,l} L_2 \tag{6}$$

*Constraint on the capacity of the interfering links:*

$$\forall l \in E, \forall 1 \leq j \leq C : \quad p_{j,l}.d_{u,v}^{-\alpha} < \beta.\eta_0.w + \beta. \sum_{l'=(u',v') \in E, l' \neq l} p_{j,l'}.d_{u',v}^{-\alpha} + A_{j,l} L_3 \tag{7}$$

$$\forall l \in E, \forall 1 \leq j \leq C : \quad \sum_{l' \in E} f_{j,l'} \leq \theta + (1 - c_{j,l} + A_{j,l}) L_4 \tag{8}$$

$$\forall l \in E, \forall 1 \leq j \leq C : \quad \beta.\eta_0.w + \beta. \sum_{l'=(u',v') \in E, l' \neq l} p_{j,l'}.d_{u',v}^{-\alpha} \leq p_{j,l}.d_{u,v}^{-\alpha} + (1 - A_{j,l}) L_5 \tag{9}$$

*Constraint on link capacity:*

$$\forall l \in E : \quad \sum_{j=1}^{C} f_{j,l} \leq \theta \tag{10}$$

*Constraints on flow conservation:*

$$\forall u \in V : \quad r_u + \sum_{l=(v,u) \in E} \sum_{j=1}^{C} f_{j,l} = \sum_{l=(u,v) \in E} \sum_{j=1}^{C} f_{j,l} \tag{11}$$

$$\sum_{l=(u,v) \in E | v \in V'} \sum_{j=1}^{C} f_{j,l} = \sum_{u \in V} r_u \tag{12}$$

*Constraint on channel variables:*

$$\forall l \in E : \quad \sum_{j=1}^{C} c_{j,l} \leq 1 \tag{13}$$

*Constraints on nodes degree:*

$$\forall u \in V : \quad \sum_{l \in E | l=(u,v) \vee l=(v,u)} \sum_{j=1}^{C} c_{j,l} \leq n \tag{14}$$

$$\forall u \in V' : \quad \sum_{l \in E | l=(v,u)} \sum_{j=1}^{C} c_{j,l} \leq n' \tag{15}$$

*Channels activity:*

$$\forall 1 \leq j \leq C : \quad \sum_{l \in E} c_{j,l} \leq z_i L_6 \tag{16}$$

*Other constraints:*

$$\forall l \in E, \forall 1 \leq j \leq C : \quad f_{j,l} \leq c_{j,l} L_7 \tag{17}$$

$$\forall l \in E, \forall 1 \leq j \leq C : \quad f_{j,l} + (1 - c_{j,l}) > 0 \tag{18}$$

where $L_i$, with $i \in \{1, 2, 3, 4, 5, 6, 7\}$ are big constants.

Recall that our objective is to minimize the total bandwidth used by the WMN. This is done, in the MILP by minimizing the number of active channels in (4).

The contraints of the model can be divided into several categories. Constraints (5) and (6) ensures that if a link is active i.e. $c_{j,l} \geq 0$ for some $j$ $(0 \leq j \leq C)$, then the transmission power $(p_{j,l})$ would produce an SNR has to be greater or equal to threshold $\beta$, otherwise, $p_{j,l}$ should be 0.

Interfering links have to share the same channel. We consider a set of links to be interfering on each other, if they are assigned to the same channel, and at least, one of them, has an associated SINR that is below the threshold $\beta$. So, the sum of the flow quantities carried by these interfering links must be below the channel capacity $\theta$. This is expressed by (7), (8) and (9).

Constraint (10) makes sure the flows passing through a link do not exceed the channel capacity, while flow conservation constraints at level of mesh routers and gateways are represented by (11) and (12), respectively.

A link $l$ is in active state if and only if it is carrying some flow ((17) and (18)). Constraint (13) indicates that a link can be active only on a certain channel, while nodes degree constraints ((14) and (15)) ensure that the number of active links going in or out of a node are limited to the number of interfaces available at that node.

Finally, a channel is considered to be active if it is assigned to at least one link (16).

## 5 Numerical Results

In order to show how the model interact with system parameters of a WMN, a series of numerical results obtained by solving the proposed model are presented in this section.

The model had been implemented in Java and solved using IBM ILOG CPLEX Concert Technology [12].

In this section, as in [13] we consider a path loss exponent $\alpha = 2$, a power spectral density of thermal noise $\eta_0 = 10^{-6}$ Watt/MHz, and a SINR threshold $\beta = 1.3$.

We first show how does the number of routers in a network impacts on the number of channels needed. To this end, we consider an area of dimension of $40\,000\,\mathrm{m}^2$, working on a 200 MHz radio bandwidth, divided to 10 channels of 20 MHz each. All channels are considered to have the same and constant capacity $\theta$ that can be retrieved through Shannon's formula, by fixing the SINR to $\beta$, which gives:

$$\theta = w. \log_2(1 + \beta) \tag{19}$$

The maximum power is chosen to be 2 Watts.

The network is constituted of varying number of routers ($N$ from 1 to 15) and of $N' = 2$ gateways. Mesh routers possess 2 interfaces while gateways possess 4. The nodes (routers and gateways) are distributed uniformly on the considered area. Each router generates a traffic with a rate equal to 500 kbits/sec. We perform a series of 10 successive runs for each value of $N$. Figure 1 displays the results for the average number of needed channels for each value of $N$.

As it can be expected, the number of used channels increases with the number of routers. When $N$ varies between 1 and 3 routers, only one channel needs to be activated. This number grows, then, progressively with the number of routers in the network, and exceeds 6 channels when the number

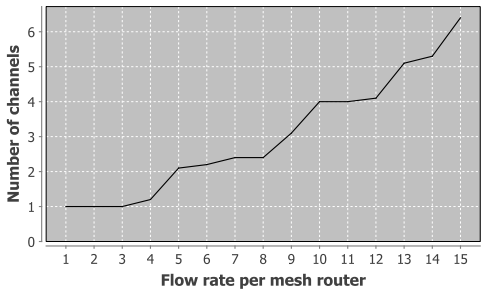



**Fig. 1.** Number of needed channels vs. number of routers.

of routers is 15. This shows that as the size of a network grows up, more bandwidth needs to be available. At the opposite, when a network has a small number of nodes, then, there is no need to allocate a too large bandwidth.

In Fig. 2, the impact of increasing the flow rate per mesh router, from 100 kbits/sec to 500 kbits/sec, is shown. The black line is for a network with $N = 10$ mesh routers and $N' = 2$ gateways, while the doted line shows results for the same network to which 5 other mesh routers are added (hence, $N = 15$). We can see that as the traffic grows, more bandwidth is needed. Also, the case of $N = 15$ generates naturally more channels than the case of $N = 10$ mesh routers.

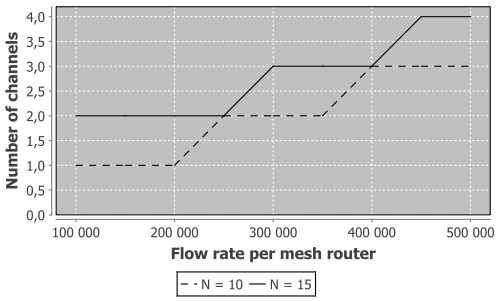



**Fig. 2.** Number of needed channels vs. flow rate per mesh router.

Note that our model stands for any charge of traffic, at the contrast to other works that consider only saturated traffic. Also, traffic does not have to be the same for all mesh routers. It is, hence, easy to determine the right number of channels needed with any traffic schema.

Figure 3 shows the evolution of the processing time with the evolution of the network size, when $N$ goes from 1 to 10 mesh routers. The time measured is the time that was needed to solve the MILP to optimality using CPLEX. The graph shows clearly that it increases exponentially with the network size. Hence, for futur work, we plan to use more efficient optimization techniques to solve the model.

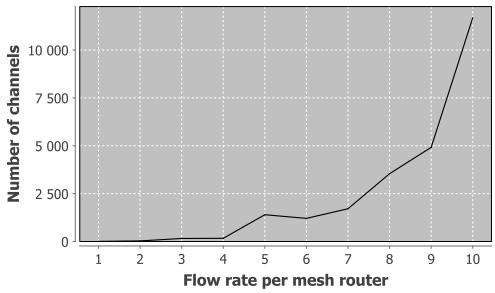



**Fig. 3.** Processing time vs. number of routers.

## 6 Conclusion

We presented, through this paper, a mathematical optimization model for multi-channel and multi-radio wireless mesh networks, by considering the most important advantages and constraints that characterize these systems. A special care has been made on the adaptability of the transmission power. The objective of our work was to optimize the use of a critical resource in wireless networks which is the radio bandwidth. In contrast of other works, our model can stand for a network with a multi-gateway scenario. It is also generic, and can stand for several kinds of MAC layers.

Our proposed model can easily be used to know the bandwidth of radio needed with any applied configuration. We show, through some numerical results how the variation of different parameters of a WMN interact with the radio bandwidth needed to satisfy users demands in terms of throughput. Hence, it is shown that the bandwidth assigned to a WMN has to take into account the network size and the clients traffic rate. It is, useless to reserve a too wide band when a network contains a reduced number of mesh routers or when it has to transport a low traffic, because this does not enhance the WMN performances, but rather increases its deployment costs.

We also show that the resolution execution time of the current form of the proposed model increases exponentially with the network size, we thus plan to implement our model by more adapted methods so that it become more scalable.

Also, until now, we have considered the case of non overlapping channels. In order to minimize the bandwidth even more, we plan to consider, in futur, the case of WMNs working on overlapping channels.

## References

1. http://www.ieee802.org/11/ : (IEEE 802.11 Working Group)
2. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: A survey. Computer Networks and ISDN Systems **47** (2005) 445–487
3. Aoun, B., Boutaba, R., Youssef, I., Kenward, G.: Gateway placement optimization in wireless mesh networks with qos constraints. IEEE Journal on Selected Areas in Communications **24** (2006) 2127–2136
4. Amaldi, E., Capone, A., Cesana, M., Malucelli, F.: Optimization models for the radio planning of wireless mesh networks. In: Proceedings of the 6th International Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. Networking (2007) 287–298

5. Benyamina, D., Hafid, A., Gendreau, M.: Wireless mesh network planning: A multi-objective optimization approach. In: Proceedings of the 5th International Conference on Broadband Communications, Networks and Systems. BROADNETS (2008) 602–609
6. Pan, D., Xue, Y., Zhan, L.: A multicast wireless mesh network network routing algorithm with ant colony optimization. In: Proceedings of the Conference on Wavelet Analysis and Pattern Recognition. ICWAPR (2008) 744–748
7. Ling, S., Jie, C., Xue-jun, Y.: Multi-path anycast routing based on ant colony optimization in multi-gateway wmn. In: Proceedings of the 5th International Conference on Computer Science and Education. ICCSE (2010) 1694–1698
8. Alicherry, M., Bhatia, R., Li, L.: Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking. MobiCom (2005) 58–72
9. Pries, R., Staehle, D., Stoykova, M., Staehle, B., Tran-Gia, P.: A genetic approach for wireless mesh network planning and optimization. In: Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. IWCMC (2009) 1422–1427
10. Marina, M.K., Das, S.R., Subramanian, A.P.: A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. Computer Networks **54** (2010) 241–256
11. Uddin, M., Alazemi, H., Assi, C.: Joint routing, scheduling and variable-width channel allocation for multi-hop wmns. In: Proceedings of the IEEE International Conference on Communications. ICC (2010) 1–6
12. http://www-01.ibm.com/software/integration/optimization/cplex optimizer/ : (CPLEX Homepage)
13. Uddin, M.F.: Design Methods for Optimal Resource Allocation in Wireless Networks. PhD thesis, Concordia University (2011)

# Modélisation de la Sécurité d'un Réseau Ad hoc sous la Contrainte d'Energie par une Approche à Deux Etapes: Clusterisation - Jeu Evolutionnaire

Karima Adel-Aïssanou, Mohammed Said Radjef,
Myria Bouhaddi, and Sara Berri

Laboratoire LAMOS, Département de Recherche Opérationnelle,
Faculté des Sciences Exactes, Université de Béjaia, Algérie
ak_adel@yahoo.fr,radjefms@yahoo.fr
myria.bouhaddi@gmail.com,berri.sara2012@gmail.com

**Abstract.** Les réseaux ad hoc, composés d'un ensemble de nœuds mobiles, sont soumis à une multitude de challenges, en particulier le problème de ressources limitées, comme l'énergie et leur vulnérabilité en termes de sécurité. En effet, les nœuds d'un réseau ad hoc doivent contrecarrer diverses attaques et actions malveillantes. Ainsi, chaque mobile est confronté à un dilemme: coopérer pour assurer la sécurité en dépensant de l'énergie ou ne pas coopérer à la sécurité, ce qui lui permet de préserver son énergie, mais rendant plus vulnérable la sécurité du réseau. Dans ce travail, nous développons une approche qui prend en compte les deux objectifs antagonistes: contribuer à la sécurité du réseau tout en réduisant la consommation de l'énergie. L'approche repose sur une alternance de deux étapes: Clusterisation-Jeu évolutionnaire.
L'étape de clusterisation se fait par un algorithme qui prend en considération la contrainte d'énergie pour l'élection des cluster-heads. Les interactions entre ces derniers dans leur contribution à la sécurité du réseau, sont modélisées sous forme de jeu évolutionnaire qui constitue la seconde étape de l'approche.

**Keywords:** réseaux ad hoc, sécurité, clusterisation, IDS (Intrusion Detection System), jeux évolutionnaires, réplicateur dynamique, ESS, convergence, simulation.

## 1 Introduction

Au cours de ces dernières années, le besoin à plus de mobilité et à pouvoir partager ou échanger de l'information à tout moment a fait naître une nouvelle technologie de réseaux mobiles sans fil, appelés réseaux ad hoc. Ces réseaux sont constitués d'un ensemble de nœuds mobiles qui se déplacent et communiquent de manière autonome par une transmission sans fil, appelée ondes radio, qui ne suppose pas d'infrastructure préexistante.

Les réseaux ad hoc, comme tout autre type de réseaux, peuvent être des cibles de maintes attaques qui peuvent causer des dommages et ainsi dégrader

leurs performances. La mise en œuvre de certains mécanismes de sécurité développés pour les réseaux filaires est délicate, voire impossible dans les réseaux ad hoc. En raison de leur caractère spontané, ces derniers ne peuvent bénéficier des mécanismes de sécurité s'appuyant sur l'infrastructure, comme un pare feu ou un serveur d'authentification. Toutes ces contraintes concourent à rendre la sécurité des réseaux ad hoc difficile et complexe à appréhender. On distingue dans la littérature différentes approches de modélisation et de résolution de ces différentes problématiques, parmi lesquelles nous trouvons la théorie des jeux [8–10].

Dans la plupart des travaux, il est supposé que tous les nœuds sont des nœuds de contrôle c'est-à-dire que les IDS [1] de tous les nœuds sont activés, ce qui n'est pas très judicieux en terme de consommation d'énergie qui est un facteur déterminant dans la durée de fonctionnement du réseau. Afin de traiter la problématique de la sécurité d'un réseau ad hoc tout en aménageant la consommation de l'énergie de tous les nœuds du réseau, nous apportons dans ce papier une approche fondamentalement coopérative. Le principe consiste à activer les IDS sur un nombre restreint de nœuds du réseau, tout en prenant en considération que certains peuvent ne pas accomplir leur tâche de contrôle en n'activant pas leur IDS. Afin de voir l'évolution de l'état du réseau qui émerge des attitudes des nœuds sélectionnés pour assurer le contrôle du trafic dans le réseau, nous allons élaborer un modèle basé sur la théorie des jeux évolutionnaires en se référant au modèle proposé dans [7]. Le choix s'est porté sur cette classe de jeux en raison de la faible rationalité des nœuds. De plus, cette catégorie de jeux permet de modéliser la dynamique de l'évolution d'une population d'individus en interaction en se basant sur deux concepts fondamentaux: les stratégies évolutionnairement stables (ESS) et le réplicateur dynamique.

Le reste de ce papier est organisé comme suit. Dans la section 2, nous présenterons notre modèle qui permet de répondre aux besoins des réseaux ad hoc en termes d'énergie et de sécurité. Nous commencerons par donner une présentation détaillée du modèle qui débute par une étape de clusterisation du réseau à travers un algorithme que nous présenterons. Par la suite, nous appliquerons la théorie des jeux évolutionnaires pour modéliser le comportement stratégique d'un ensemble restreint de nœuds et nous illustrons les résultats de l'évolution de la population. Nous présenterons les résultats de la simulation et nous évaluerons le modèle à base de ces résultats dans la section 3.

## 2 Modèle

Nous considérons la problématique de la sécurité d'un réseau ad hoc composé d'un certain nombre de nœuds. L'approche que nous développons dans ce papier est basée sur l'alternance de deux étapes:

[1] IDS: processus de surveillance des événements se trouvant dans un réseau, il permet de détecter en temps réel et de façon continue des tentatives d'intrusion.

- La clusterisation, dont le but est de répartir les nœuds dans des clusters et d'élire pour chacun d'eux un cluster-head;
- La modélisation des interactions entre les cluster-heads par un jeu, dont les stratégies consistent à contribuer ou non à la sécurité du réseau en activant ou désactivant leur IDS.

### 2.1 Etape de clusterisation

La clusterisation consiste à partitionner le réseau en groupes d'entités, appelés clusters. Chaque cluster sera par la suite identifié par son cluster-head, qui agira comme un coordinateur local dans son cluster. Le choix du cluster-head se fait sur la base d'une métrique bien définie.
Un cluster-head prend en charge le contrôle de tout le trafic destiné aux membres de son cluster. Autrement dit, lorsqu'un paquet est destiné à l'un de ses membres, il l'intercepte, le teste et le retransmet à son destinataire dans le cas où le paquet ne représente aucune menace.

Plusieurs algorithmes de clusterisation ont été proposés dans la littérature et ils se distinguent par le critère de sélection des cluster-heads. Dans cet axe, certains algorithmes ont choisi des critères simples, comme l'identifiant (ID) [1], le degré de connectivité [5]. D'autres approches ont adopté des sélections plus élaborées en s'appuyant sur une combinaison de critères comme (WCA) [4].

**2.1.1 Algorithme de clusterisation** Gerla et Tsai ont proposé dans [5] un algorithme de clusterisation, appelé High-Connectivity Clustering (HCC), qui se base sur l'élection comme cluster-head, le nœud dont le degré est le plus élevé. Le degré d'un nœud se calcule en fonction de sa distance par rapport aux autres. Les différentes phases de cet algorithme sont les suivantes:

1. *Chaque nœud diffuse son (ID) aux nœuds qui se trouvent à sa portée de transmission (ses voisins) et le nœud avec un nombre maximum de voisins, c'est-à-dire avec un degré maximal, est choisi comme cluster-head et ses voisins deviennent membres de ce cluster et ne peuvent plus participer au processus électoral.*
2. *Le processus continue jusqu'à ce qu'il n'y ait plus de nœuds à affilier aux clusters.*

L'avantage de cet algorithme est qu'il génère un nombre réduit de cluster-heads, mais il omet la contrainte d'énergie. Pour cela, nous avons ajouté une phase préliminaire qui prendra en compte le niveau d'énergie des nœuds. Le nouvel algorithme de clusterisation est décrit ci-dessous.

**Algorithme**

**(0). Initialisation:** l'étape d'initialisation consiste à:

1. Affecter pour chaque nœud une position de départ et lui fixer une charge d'énergie initiale,
2. Fixer le niveau d'énergie requis pour qu'un nœud soit prioritaire dans le processus d'élection des cluster-heads;

**(1). Calcul des degrés:** dans cette étape, nous calculons le degré de chaque nœud qui correspond au nombre de ses voisins à un saut;

**(2). Test d'énergie:** cette étape consiste à sélectionner un ensemble S, qui contiendra les nœuds vérifiant le seuil fixé dans l'étape d'initialisation;

**(3). Désignation des cluster-heads:** en arrivant à cette étape, l'ensemble S peut être vide ou contenant au moins un élément:

**Si** $S \neq \emptyset$:

1. le principe de l'algorithme HCC sera appliqué. On sélectionne le nœud dont le degré est le plus élevé dans l'ensemble S, que nous noterons CH,
2. S=S\{CH};

**Sinon**, si $S = \emptyset$ dans ce cas, nous appliquerons l'algorithme HCC sur l'ensemble des nœuds restants;

**(4). Rattachement aux cluster-heads:** une fois que le cluster-head est élu, tous ses voisins à un saut le rejoignent et ainsi ils formeront un cluster.

Les étapes 3 et 4 seront répétées jusqu'à ce que tous les nœuds soient affiliés aux clusters que ce soit en tant que cluster-head ou bien membre.

Comme les nœuds d'un réseau ad hoc sont dynamiques et à énergie limitée, il serait indispensable de mettre à jour la procédure de clusterisation. Ainsi, chaque nœud est potentiellement capable d'être élu cluster-head. Il est donc nécessaire d'installer des IDS sur tous les nœuds du réseau et de les garder en veille jusqu'au moment où ils seront élus cluster-heads.

### 2.2 Etape de modélisation sous forme d'un jeu

Dans cette seconde étape, nous proposons un modèle de jeu décrivant le comportement stratégique des cluster-heads sous forme de jeu évolutionnaire. Dans [7], les auteurs ont considéré que la population est constituée de tous les nœuds du réseau, alors que dans notre approche nous nous limitons qu'aux cluster-heads. D'autre part, à la différence de [7] dans la définition des utilités des joueurs, nous considérons les dommages et les pertes causés dans le cas où le réseau n'est pas sécurisé.
Une fois que le réseau est partitionné en clusters et que les cluster-heads sont élus, nous considérons le jeu entre les cluster-heads. Chaque cluster-head dispose de deux stratégies pures:

1. Protéger (P), qui se traduit par l'activation de son IDS;
2. Ne pas protéger (NP) en désactivant son IDS.

Un cluster-head qui contribue à la sécurité, donc opte pour la stratégie (P) supporte un coût égal à $c$ qui correspond à la dépense énergétique lorsqu'il assure la sécurité de son cluster et obtient une récompense $r$ lorsque tout le réseau est sécurisé, c'est à dire tous les cluster-heads choisissent la stratégie (P). Lorsqu'au moins un cluster-head choisit de ne pas activer son IDS, le réseau devient non sécurisé, car il est supposé être constamment exposé aux attaques. Ce fait engendrera des pertes de données et des dommages que nous noterons $l$.

Les cluster-heads choisissent simultanément leurs stratégies. Ainsi, le jeu entre deux cluster-heads peut être représenté sous forme stratégique, avec les gains, associés à chaque couple de stratégies, représentés dans la matrice suivante:

$$A = \begin{matrix} \\ P \\ NP \end{matrix} \begin{matrix} \begin{matrix} P & \quad NP \end{matrix} \\ \begin{pmatrix} (r-c, r-c) & (-c-l, -l) \\ (-l, -c-l) & (-l, -l) \end{pmatrix} \end{matrix}, \tag{1}$$

où: $r$, $c$ et $l > 0$.
Nous supposons que:

$$r \ > \ c \ \ et \ \ l \ > \ c, \tag{2}$$

car sinon les cluster-heads ne seront pas incités à protéger le réseau.
Ce jeu présente deux équilibres de Nash stricts (P,P) et (NP, NP) et donc le jeu admet deux ESS [2].
Il existe un processus de sélection spécifiant comment une population est associée avec différentes stratégies pures dans un jeu qui évolue dans le temps, appelé le réplicateur dynamique qui est donné par le système d'équations différentielles suivant [6] :

$$\dot{p_i} = p_i[(Ap)_i \ - \ p^T Ap], \qquad i = 1, 2. \tag{3}$$

Avec la matrice des gains (1), le système (3) prend la forme suivante:

$$\dot{p_1} = p_1(1-p_1)[p_1(l+r) - c] \ \ , \ \ \dot{p_2} = -\dot{p_1},$$

où $p_1$ est la proportion de la population jouant la stratégie (P) et $p_2$ la proportion de la population jouant la stratégie (NP).

### 2.3 Résultats de l'implémentation du réplicateur dynamique

Pour étudier l'évolution de la population, nous avons implémenté le réplicateur dynamique sous Matlab avec les paramètres du jeu $r$=3, $c$=1 et $l$=2. Nous avons fixé ces paramètres en respectant les conditions déjà établies dans l'équation (2). Nous avons obtenu la figure suivante pour des taux initiaux de la population optant pour la stratégie (P), variant entre 0 et 1.

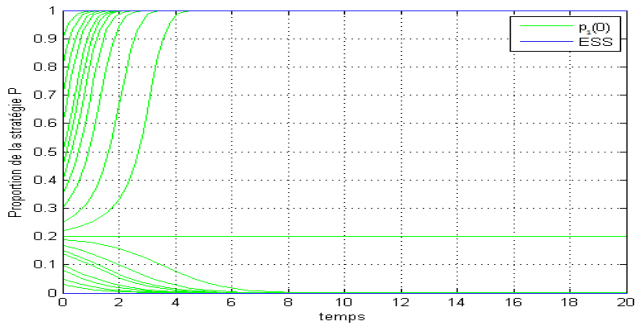



**Fig. 1.** Convergence du réplicateur dynamique.

Nous remarquons à partir de cette figure que:

- Avec un taux initial de la population choisissant la stratégie (P) inférieur strictement à $0.2=\frac{c}{r+l}$, le nombre de cluster-heads qui participent à la sécurité décroît jusqu'à ce qu'ils finissent tous par ne pas participer.
- Avec un taux initial de la population choisissant la stratégie (P) supérieur strictement à 0.2, le nombre de cluster-heads qui participent à la sécurité croît jusqu'à ce qu'ils finissent tous par participer.
- Avec un taux initial de la population choisissant la stratégie (P) égal à 0.2, le nombre de cluster-heads qui participent à la sécurité reste inchangé.

## 3 Simulation

Dans cette section, nous évaluons les performances de notre approche à deux étapes: Clusterisation-Jeu évolutionnaire en la simulant sous Matlab.

### 3.1 Paramètres de simulation

Le réseau ad hoc implémenté est défini par sa couverture[2] et le nombre de nœuds qu'il contient. Le réseau choisi est un réseau 1000 m X 1000 m comprenant N=20 nœuds. Chaque nœud dispose d'une énergie initiale de 100 Joule qui diminue avec une quantité égale à 10 J [3] dans le cas de l'activation de l'IDS et avec une quantité choisie d'une manière aléatoire dans l'intervalle [5 J, 15 J] lors d'une réception ou d'une transmission de paquets. Le modèle de mobilité choisi est Random Walk avec une vitesse $v \in [0, 15$ m/s] et une direction de mouvement $\theta \in [0, 2\pi]$. Le rayon de transmission de chaque nœud est de 250 m.
La période de simulation s'étale sur 150 secondes. Les nœuds changent de position chaque 15 secondes avec des directions et des vitesses différentes. Le seuil

[2] Taille en coordonnées X et Y.

requis pour qu'un nœud devienne prioritaire à être élu cluster-head est fixé à 25 J. Les paramètres de simulation sont récapitulés dans le tableau suivant:

**Table 1.** Paramètres de la simulation.

| | |
|---|---|
| (X, Y) | 1000 m X 1000 m |
| N | 20 |
| R | 250 m |
| $[v_{min}, v_{max}]$ | [0, 15] |
| Modèle de Mobilité | Random Walk |
| Einitial | 100 J |
| Seuil | 25 J |
| T | 150 secondes |
| h | 15 secondes |

### 3.2 Résultats de la simulation

La simulation qui dure 150 secondes, a été faite conformément aux paramètres mentionnés dans le tableau 1 et la représentation graphique des dernières positions des nœuds est montrée dans la figure suivante:

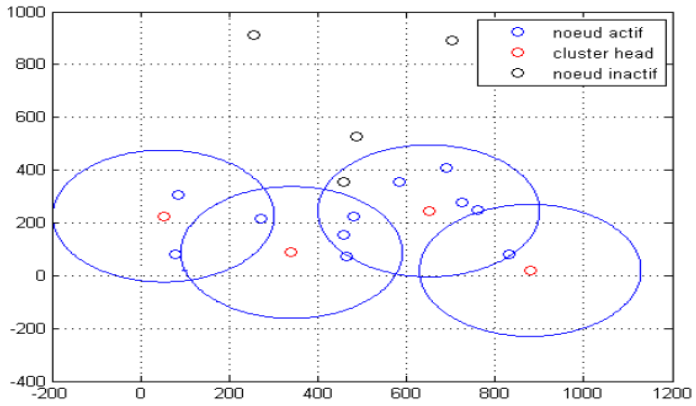



**Fig. 2.** Positions des nœuds.

Les nœuds inactifs[3] peuvent être considérés comme des nœuds isolés[4], malgré qu'ils se trouvent géographiquement dans un cluster ou plus, du fait qu'ils

[3] Un nœud inactif est un nœud dont l'énergie est égale à 0.

[4] Un nœud isolé est un nœud dont l'ensemble des voisins à un saut est vide.

apparaissent invisibles aux autres nœuds et ne peuvent maintenir la connectivité du réseau.

### 3.3 Evaluation de la clusterisation

L'objectif de la clusterisation est de mieux gérer la consommation de l'énergie des nœuds tout en maintenant la sécurité du réseau et ceci en n'activant que les IDS des cluster-heads. Nous avons fait une étude comparative entre l'approche que nous avons développée dans ce papier (étape avec clusterisation) et une approche n'utilisant pas cette étape (étape sans clusterisation). La comparaison de ces deux approches est basée sur les critères suivants: le nombre moyen de nœuds inactifs et le nombre moyen de nœuds isolés.
Pour les mêmes paramètres de simulation, nous représenterons le nombre de nœuds actifs en fonction du temps pour les deux modèles, avec clusterisation et sans clusterisation.

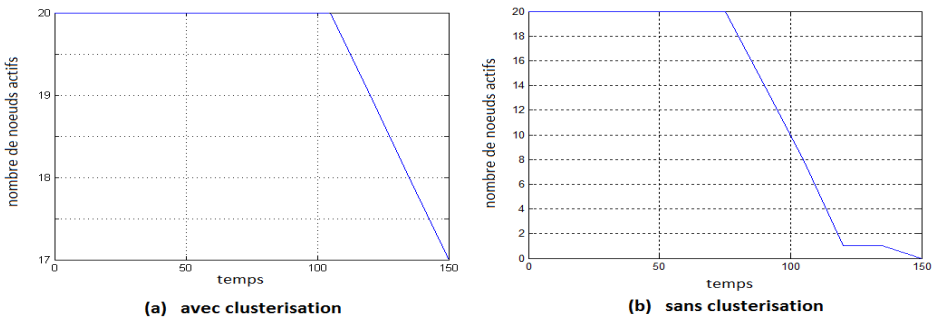



**Fig. 3.** Nombre de nœuds actifs en fonction du temps.

Nous remarquons que l'instant où le premier nœud devient inactif pour le modèle avec clusterisation correspond à l'instant 120, quant au second modèle cet instant correspond à 80 à partir duquel le nombre de nœuds actifs dans le modèle sans clusterisation décroît rapidement jusqu'à devenir 0 à la fin de la simulation. En revanche, le nombre de nœuds actifs dans le modèle avec clusterisation décroît lentement et à la fin de la simulation nous ne trouvons que trois nœuds inactifs. En passant à un nombre plus important de nœuds avec différents temps de simulation, nous obtenons les résultats suivants:

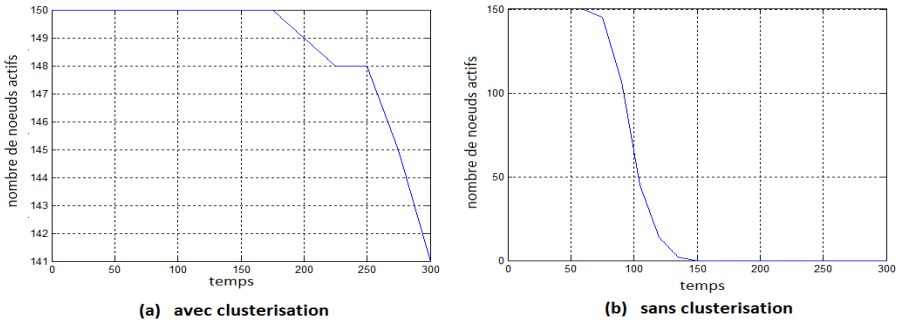



**Fig. 4.** Nombre de nœuds actifs en fonction du temps, **N=150, T=300**.

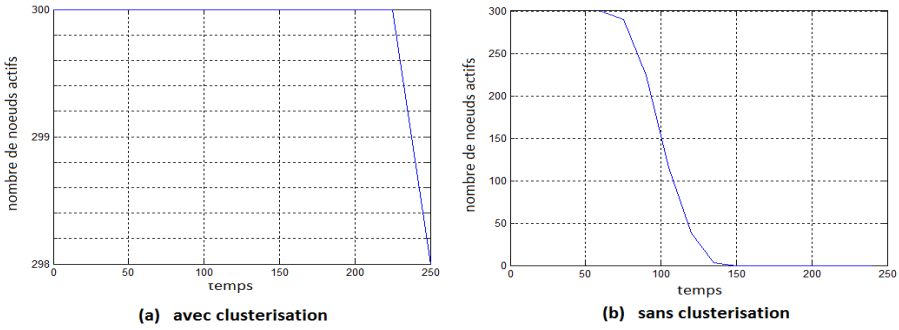



**Fig. 5.** Nombre de nœuds actifs en fonction du temps, **N=300, T=250**.

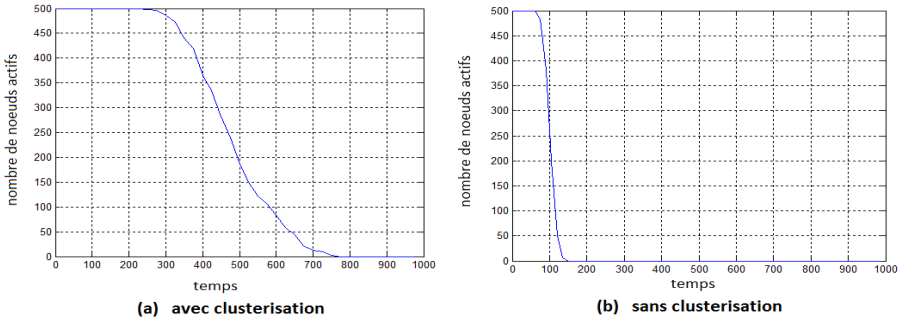



**Fig. 6.** Nombre de nœuds actifs en fonction du temps, **N=500, T=1000**.

D'après les figures 4, 5 et 6, nous remarquons que le passage à l'échelle nous donne d'aussi bons résultats que dans le cas où le nombre de nœuds n'est que de 20. Ceci nous montre davantage l'efficacité de la clusterisation dans la conservation de l'énergie, ce qui permet aux nœuds de demeurer actifs plus longtemps et donc de participer plus au mécanisme de sécurité du réseau.

En somme, les résultats concernant le nombre de nœuds actifs donnés par le modèle avec clusterisation sont nettement meilleurs que ceux obtenus par le modèle sans clusterisation. Cette différence montre bien comment la clusterisation et l'activation des IDS uniquement sur les cluster-heads influent sur le nombre de nœuds actifs dans un réseau.
Quant au nombre moyen de nœuds isolés et la probabilité moyenne de connectivité du réseau durant la simulation, ils sont donnés dans le tableau suivant:

**Table 2.** Connectivité du réseau.

| | | Avec clusterisation | Sans clusterisation |
|---|---|---|---|
| N=20, T=150 | Nombre moyen de nœuds isolés | 9 | 20 |
| | Probabilité moyenne de connectivité | 0.4 | 0 |
| N=300, T=250 | Nombre moyen de nœuds isolés | 2 | 300 |
| | Probabilité moyenne de connectivité | 0.99 | 0 |

Nous constatons à partir de ce tableau qu'un réseau partitionné en clusters est plus connecté qu'un réseau non clusterisé. En effet, dans un réseau clusterisé, l'énergie des nœuds diminue lentement comparé à un réseau non clusterisé en raison de l'activation des IDS que sur un ensemble restreint de nœuds. De ce fait, le nombre de nœuds inactifs dans un réseau clusterisé s'accroît lentement. Ainsi, le réseau demeure plus connecté pour longtemps.

### 3.4 Impact du taux initial de participation à la sécurité sur le taux de détection des attaques

Pendant la simulation, une génération d'attaques a été faite et afin de voir l'impact du taux initial de la population participant à la sécurité sur le taux de détection de ces attaques, nous avons implémenté le réplicateur dynamique dans le simulateur. Ainsi, en disposant de l'ensemble des cluster-heads dont la stratégie choisie est (P) et de l'ensemble des nœuds attaqués, nous pourrions obtenir le taux de détection à chaque fois qu'il y ait génération d'attaques. En fixant les paramètres du jeu: r=3, l=2 et c=1, l'évolution du taux de détection en fonction du taux initial de participation est illustrée dans la figure ci-après:

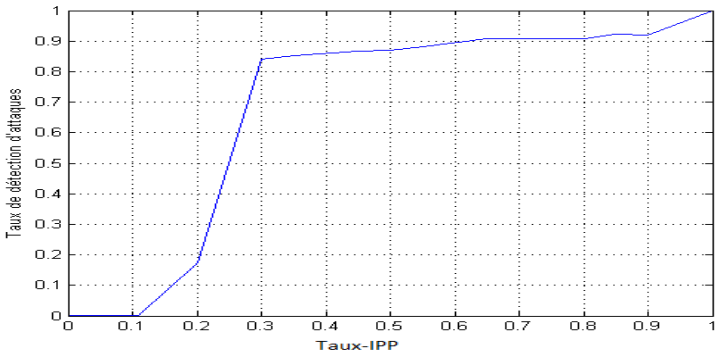



**Fig. 7.** Impact du taux-IPP sur le taux de détection des attaques.

Nous remarquons que pour des taux initiaux inférieurs à 0.2 le taux de détection n'est pas très important, ceci est dû au fait que pour les mêmes paramètres du jeu, r=3, l=2 et c=1, les cluster-heads finissent par ne pas participer et donc toutes les attaques générées à partir de cet instant ne seront pas détectées. Dans le cas, où la proportion initiale de cluster-heads choisissant (P) dépasse 0.2, le taux de détection associé est élevé car de tels états finissent toujours par atteindre la stratégie évolutionnairement stable $p_1^*$=1. Ainsi, à partir de cet instant, toutes les attaques générées seront détectées, car tous les cluster-heads choisissent de participer et c'est ce qui augmente le taux de détection total pour ces états.

## 4 Conclusion

Dans notre travail, nous avons traité le problème de la sécurité dans les réseaux ad hoc par une approche en deux étapes: clusterisation et modélisation par un jeu évolutionnaire. Nous avons optimisé les ressources en énergie des nœuds par le biais d'une clusterisation et mis en avant l'aspect évolutif du réseau pour pouvoir suivre l'interaction existant entre les cluster-heads du réseau dans leur participation au processus de sécurité.

Nous avons mis en évidence la convergence du réplicateur dynamique, qui décrit l'évolution de la proportion des cluster-heads jouant la stratégie (P), vers l'ESS. Les deux résultats importants obtenus sont que le taux de détection est une fonction croissante du taux initial de la population optant pour (P) et que la technique de clusterisation est efficace pour réduire le nombre de nœuds inactifs, et donc le nombre de nœuds isolés, ce qui rend le réseau plus connecté tout en maintenant sa sécurité.

## References


1. Baker, D.J., Ephremides, A., Wieselthier, J.E.: A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling. In Proceedings of the IEEE. **75** (1987) 56–73
2. Samuelson, L.: Evolutionary Games and Equilibrium Selection. Massachusetts: The MIT Press. Cambridge. (1997)
3. Bhattacharya, P., Debbabi, M., Mohammed, N., Otrok, H., Wang, L.: Mechanism Design-Based Secure Leader Election Model for Intrusion Detection in MANET. IEEE Transaction on Dependable and Secure Computing. **8** (2009) 89–103
4. Chatterjee, M., Das, S.K., Turgut,D.: WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Cluster Computing. **5** (2002) 193–204
5. Gerla, M., Tzu-Chieh Tsai, J.: Multicluster Mobile, Multimedia Radio Network. ACM/Baltzer Journal of Wireless Networks. **1** (1995) 225–265
6. Jonker, L.B., Taylor, P.D.: Evolutionary Stable Strategies and Game Dynamics. Mathematical Biosciences. Elsevier. **40** (1978) 145–156
7. Kamhoua, C. A., Makki, K., Pissinou, N.: Game Theoretic Modeling and Evolution of Trust in Autonomous Multi-Hop Networks Application to Network Security and Privacy. IEEE Communications Society. (2011) 1–6
8. Park, J. M., Patcha, A.: A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks. International Journal of Network Security. **2** (2006) 131–137
9. Sun, H., Wei, H.: Using Bayesian Game Model for Intrusion Detection in Wireless Ad Hoc Networks. Int. J. Communications. Network and System Sciences. **3** (2010) 602–607
10. Wei Lye, K., Wing, J-M.: Game Strategies in Network Security. Int J Inf Secur. **4** (2005) 71–86

# A Multi-Objective Tabu Search Method for the Dial-A-Ride Problem

Ali Lemouari and Oualid Guemri

Computer Science Department, University of Jijel,
BP 98, 18000, JIJEL, ALGERIA,
{lemouari_ali@yahoo.fr,wal_guemri@hotmail.com}

**Abstract.** The dial-a-ride problem (DARP), is a variant of the pickup and delivery problem (PDP), consists of designing vehicle routes of n customers transportation requests. The problem arises in many transportation applications, like door-to-door transportation services for elderly and disabled people or in services for patients. This paper consider a static multivehicle DARP, which the objective is to minimize a combined costs of total travel distance, total duration, passengers waiting time, the excess ride time of customers, and the early arrival time while respecting maximum route duration limit, the maximum costumer ride time limit, the capacity and the time window constraint. We propose a new heuristic combined to the tabu search method. Our experimentation report best results for Cordeau Benchmark test problem, compared to reported results.

**Keywords:** dial-a-ride problem, tabu search, precedence heuristic, variable neighborhod search, genetic algorithm

## 1 Introduction

The Dial-a-Ride Problem (DARP) is a variant of the Pickup and Delivery Problem (PDP), frequently arising in door-to-door transportation services for elderly and disabled people or in services for patients. In recent years, dial-a-ride services have been steadily increasing in response to popular demand (Cordeau et al. (2007)). A DARP consists of n customers who want to be transported from an origin to a destination. Requests can be classified as outbound (say from home to the hospital) or inbound (from hospital back to the home). The objectives applied range from the maximization of the number of patients served to the minimization of user waiting time or routing costs. The constraints considered are usually tailored to the specific problem situation.

Let us give some examples: Aldaihani et al. (2003) solve a mixed DARP, i.e. transportation on demand, and a fixed route transit system by means of a tabu search heuristic. Customer inconvenience in terms of the total ride time of all the passengers and the total distance traveled by vehicles are minimized. Cordeau and Laporte (Cordeau et al. (2003)) describe a tabu search heuristic. In Their algorithm solutions recently visited are put on the tabu list and are therefore

forbidden for a number of iterations. Infeasible solutions may be explored in the search, the time window, ride time, vehicle capacity, and route duration constraints are relaxed and their violation are penalized in the objective function. Jorgensen et al. (2007) propose a genetic algorithm (GA) to solve the DARP, minimizing a weighted sum of customer transportation time, excess user ride time with respect to direct ride time, customer waiting time, time window violations, excess user ride time with respect to maximum ride time, and excess work time. The variable neighborhood search (VNS) heuristic has been applied to the DARP by Mladenovic et al. (1997). An earlier version of this heuristic combined with path relinking (Glover et al. (1997)) has successfully been applied in the context of the multi-objective DARP (Parragh et al. (2010)).

Additional information about existing solution methods and algorithms can be found in (Cordeau et al. (2007)), and (Paquette at al. (2009)), also for an overview of the different quality of service criteria that have been applied in dial-a-ride systems. A problem definition together with a benchmark data set has been introduced by Cordeau et al. (2003), they consider time windows, a maximum user ride time limit, and a maximum route duration limit, while minimizing total routing costs. Moreover results for the proposed data set obtained by the tabu search (TS) heuristic are reported.

In this paper, we present a tabu search method combined with a proposed precedence heuristic to the multi-objective DARP. An intra-optimization and solution-repair procedures are introduced and combined with the TS method. Our experimentation carry out best results for Cordeau and Laporte Benchmark test problem, compared to the presented results in literature, more particularly GA and VNS algorithms.

## 2 The Dial-A-Ride Problem

The DARP is defined on a complete graph $G(V, A)$, where $V = \{0, 1, 2, .., 2n\}$ is the vertex set and $A = \{(i, j)\}, i, j \in V, i \neq j$ is the arc set. For each arc a non-negative travel cost $c_{ij}$ and a non-negative travel time $t_{ij}$ are considered. Vertex 0 represents a depot at which is based a fleet of $m$ vehicles, and the remaining $2n$ vertices represent origins and destinations for the transportation requests. Each vertex pair $(i, i + n)$ represents a request for transportation from origin $i$ to destination $i + n$. With each vertex $i$ are associated a load $q_i$, and a load when leaving vertex noted $y_i$, a nonnegative service duration $d_i$ and a time window $[e_i, l_i]$, where $e_i$ and $l_i$ are nonnegative. The load is equal to 1 for vertices $(1, .., n)$ and $-1$ for vertices $(n + 1, .., 2n)$.

Let $T$ denote the end of the planning horizon, $L$ denote the maximum ride time of a user. We denote by $A_i$ the arrival time of a vehicle at vertex $i$, by $B_i = max\,(e_i, A_i)$ the beginning of service at the vertex, by $D_i = B_i + d_i$ the departure time from vertex $i$. The time window constraint at vertex $i$ is violated if $B_i > l_i$. Arrival before $e_i$ is, however, allowed and the vehicle then incurs a waiting time $W_i = B_i - A_i$. The ride time associated with request $i$ is computed as $L_i = B_{(i+n)} - D_i$.

The DARP consists of designing $m$ vehicle routes on graph $G$ such that **(1)** every route starts and ends at the depot; **(2)** The load of a vehicle never exceeds its capacity $Q$; **(3)** For each request, $i$ and $i+n$ are serviced by the same vehicle and $i+n$ is visited after $i$ (pairing constraint); **(4)** the service at vertex $i$ begins in the interval $[e_i, l_i]$ and every vehicle leaves the depot and returns to the depot in the interval $[e_0, l_0]$; **(5)** The ride time of any customer does not exceed a limit $L$; **(6)** The total duration never exceeds a preset bound $T_k$; **(7)** The total routing cost of all vehicles is minimized.

## 3 The Most Related Work

### 3.1 Tabu Search Approach

Cordeau et al. (2003) describe a tabu search heuristic. The heuristic use a single objective function, described as $f(s) = c(s) + \alpha q(s) + \beta d(s) + \gamma w(s) + \tau t(s)$ , where $\alpha, \beta, \gamma, \tau$ are self-adjusting positive parameters, $c(s)$ is the routing cost, $q(s)$ the load violation, $d(s)$ the route duration violation, $w(s)$ the time window violation, and $t(s)$ the ride-time violation. The search tries to minimize the routing cost and the violations simultaneously The penalties for the violations are adjusted dynamically through the search. At every iteration, if a constraint is being violated in the current solution, the penalty for that constraint is multiplied by a factor $(1 + \delta)$ otherwise the penalty is multiplied by the inverse of the same factor where $\delta > 0$. If a penalty reaches a fixed upper bound, then it is reset to 1.

### 3.2 Genetic and Variable Neighborhood Search Approaches

Jorgensen et al. (2007) uses a genetic algorithm approach to solve DARP, the authors adopt the chromosome representation used in Parera et al. (2003). In the chromosome representation both the allocation of customers to vehicles and the order of the customers on the routes are encoded. A two level binary chromosome representation are used. The routing is solved using an extended version of the modified space-time nearest neighbor heuristic developed by (Baugh et al. (1998)).

The VNS method used by (Parragh et al. (2010)), start with an initial solution $s_0$, then in every iteration a random solution $s'$ is generated in $N_k(s)$, where $N_k(s)$ denote a $k$-neighborhood of solution $s$. A local search step is applied to $s'$ yields $s''$. $s''$ Replace $s$, if $s''$ is better than $s$ the search continues. If $s''$ is worse, the next neighborhood used in the subsequent iteration $k = k + 1$. A maximum number of neighborhoods $k_{max}$ has to be defined. Whenever $k_{max}$ is attained, the search continues with the first neighborhood $k = 1$. This is repeated until some stopping criterion is met. Like in Cordeau et al. (2003), infeasible solutions are allowed during the search, with a determined penalty. Also, and in order to reduce the search space, authors use a local search step based on the graph pruning and time window tightening techniques (Cordeau (2006)).

## 4 Proposed Heuristic

We generate a tightening time window for all strengthened time window, the purpose is to find adequate too tight time interval (Parragh et al. (2010)). Let $i$ a vertex with $[e_{initial}, l_{initial}]$ as initial time window. if $i$ the origin of outbound request; the time window is modified by $e_i = max(e_{initial}, e_{(n+i)} - L - d_i)$, and $l_i = min(l_{initial}, l_{(n+i)} - t_{(i,i+n)} - d_i)$. In case of an inbound request the artificial time window is set to $e_i = max(e_{initial}, e_{(i-n)} + d_{(i-n)} + t_{(i,i-n)})$, and $l_i = min(l_{initial}, l_{(i-n)} + d_{(i-n)} + L)$, where $e_{initial}$ is the initial time for the planning horizon and $l_{initial}$ is the end of the planning horizon. For our case $e_{initial}$ is set to 0 and $l_{initial}$ is set to $24 * 60 = 1440$. The change in the time window, thus defined leads us to define a precedence relation between vertices graph.

$$e_i + d_i + t_{ij} > l_i. \tag{1}$$

Through the relation 1, we can define a list of precedence for each vertex. If $j \in pred(i)$ the service at vertex $i$ begins before the service at vertex $j$, and $\forall k \in s, \forall i, j \in k, i \notin pred(j)$, where $k$ is a given route and $s$ a solution. Two vertices $i$ and $j$ are called parallel and do not imply the precedence constraint checking for an insert operation, if and only if $\forall k \in s, \forall i, j \in k, (i||j) \Leftrightarrow (i \notin pred(j)$ and $j \notin (pred(i))$ Otherwise, the vertices having tight time windows have fewer opportunities, to insert it into a solution. The best strategy is to place these vertices first, This strategy shares many similarities with the memory heuristic described by Lacomme et al. (2008), where requests with a maximum number of rejections will be promoted first in the next insertion. The first step in Algorithm (1), reorganizes routes in order to preserve the precedence relation between vertices.

**Algorithm 1** Repair-Optimisation

1: **For** $i \leftarrow 1, |k| - 1$ /*$|k|$ is the size route */
2: **if** $(v_i \in pred(v_{i+1})$ and $(v_{i+1} \notin pred(v_i))$
   $swap(v_i, v_{i+1})$; $i \leftarrow max(1, i-2)$ /* $v_i$ is the vertex in position $i$ */
   /* Route Repair */
3: **For** $i \leftarrow 1, |k| - 3$ /*$|k|$ is the size route */
4: **if** $(t_{(v_i,v_{i+2})} + t_{(v_{i+1},v_{i+3})}) < (t_{(v_i,v_{i+1})} + t_{(v_{i+2},v_{i+3})})$
   and $(v_{i+1} \in pred(v_{i+2}))$ and $(v_{i+2} \notin pred(v_{i+1}))$ **then**
      $swap(v_{i+1}, v_{i+2})$; /* $v_i$ is the vertex in position $i$ */
   /* Route intra-Optimisation */

The Algorithm (1) is done in a polynomial time. The exploration of the search space is performed by handling parallel vertices. Then any solution is corrected and the search will be limited in a space where all the solutions satisfy the precedence relation. The second step in Algorithm, allows improvement in

the quality of routes. This is the $2-opt$ heuristic enriched by the precedence property between route vertices.

## 5 Solution Methodology

The solution methodologies used to solve the static DARP, depart from a preprocessing step, started before the optimization procedure. We apply a time window tightening techniques, as described in section 4. The second preprocessing step consists, to define a list of precedence vertex for each node that satisfy the relations (1) . To generate the first incumbent solution $s$, we construct the initial solution completely at random like in (Cordeau et al. (2003)). We apply the tabu search ($TS$) heuristic to solve the DARP problem. The $TS$ method departs from an initial solution $s_0$, then in each $TS$-iteration a best solution $s'$ is selected from the current neighborhood $N_k(s)$. The repair and intra-optimization procedure (Algorithm 1) are applied for each solution in neighborhood $N_k(s)$. To avoid cycling, solutions recently visited are declared forbidden, or tabu, for a number of iterations.

Several neighborhood structures have been used in the literature. Let $i$ be a desired solution, the neighborhood structures frequently used, are obtained through three types of movements. **(i)** Moving a request in the same route. **(ii)** Displacement the request between two routes. **(iii)** Switching between two requests belonging to different routes. In this paper, we opt for the easy neighborhood structure. Let $s$ be a given solution, the set of neighbors denoted $N(s)$, is obtained by removing a request $(i, i+n)$ from a route $k$, and then reinsertion into another route $k'$. The insertion of pickup vertex $i$ and the delivery vertex $i+n$ in route $k'$, are performed so as to minimize the total increase in $f(s)$. Keeping the two vertices $i$ and $i+n$ closely connected in the route, allow consequently optimal results in ride time. Moreover, the complexity to check all constraints will be reduced, particularly load and time window constraint.

The evaluation of a solution is based on the procedure $Forward\ Slack\ Time$, developed in Cordeau et al. (2003). This procedure can delay as much as possible early service to reduce the duration of the tour and travel time. The route evaluation is based on the forward time slack $F_i$, defined for the first time by Savelsbergh (1992).

$$F_i = min_{i \leq j \leq q}( \sum_{i<p\leq j} w_p + (min\{l_j - B_j, L - P_j\})^+) \tag{2}$$

Where $P_j$, is the ride time of the user and $L$ the maximum user ride time. Cordeau observe that delaying the departure time from the depot by $\sum_{0<p<q} w_p$ does not affect the arrival time $A_q$ at the end of the route whereas delaying the departure by more would simply increase $A_q$ by as much. Cordeau conclude, the minimal route duration that does not increase constraint violations is given by $A_q - (e_0 + min\left\{F_0, \sum_{0<p<q} w_p\right\})$. The following Algorithm (2) used to evaluate the route duration.

**Algorithm 2** Forward time slack

1: Set $D_0 \leftarrow e_0$
2: **For each** vertex $i$ Compute $A_i$, $B_i \leftarrow max\{e_i, A_i\}$, $W_i$, $D_i \leftarrow B_i + A_i$
3: **if** $B_i > L_i$ or $y_i > Q$ Goto step 8 /* Parragh et al. 2010 */
4: Compute $D_0 = e_0 + min\left\{F_0, \sum_{0<p<q} w_p\right\}$
5: **For each** vertex $i$ Update $A_i, w_i, B_i, D_i$
6: **For each** request assigned to the route **Do** Compute $L_i$
**if all** $L_i \leq L$ Goto step 8 /* Parragh et al. 2010 */
7: **For** every vertex origin of a request $j$ **Do**
Compute $F_j$
set $B_j = B_j + min\left\{F_j, \sum_{0<p<q} w_p\right\}$;$D_j = B_j + d_j$
**For each** vertex $i$ Update $A_i, w_i, B_i, D_i$ that comes after $j$
**For each** request whose destination after $j$ **Do** Update $L_i$
**if all** $L_i \leq L$ of requests whose destinations lie after $j$ Goto step 8
/* Parragh et al. 2010 */
8: Compute changes in violations of vehicle load, duration,time window and ride time constraints.

Let now $c(s)$ denote the total routing costs of all vehicles, which is the sum of the costs $c_{ij}$ , $r(s)$ is the excess ride time computed as $r(s) = \sum_{i=1}^{n}(B_{i+n} - D_i - t_{(i,i+n)})$, $l(s)$ is the total waiting time with passengers aboard computed as $l(s) = \sum_{i=1}^{n}(w_i(y_i - q_i))$, $g(s)$ is the total route duration computed as $g(s) = \sum_{k=1}^{n}(B_{2n+1}^{k} - B_0^k)$ and $e(s)$ the sum over early arrivals computed as $e(s) = \sum_{i=1}^{n}(e_i - A_i)$. The load violation $q(s)$, duration violation $d(s)$, time window violation $w(s)$, and ride time violation $t(s)$ are penalized in the evaluation function. All violation computed as follow $q(s) = \sum_{i=1}^{n}(y_i - Q)^+$, where $x^+ = max\{0, x\}$, $d(s) = \sum_{i=1}^{n}(B_{2n+1}^{k} - B_0^k - T_k)^+$, $w(s) = \sum_{i=0}^{2n}(B_i - l_i)^+$ , and $t(s) = \sum_{i=1}^{n}(L_i - L)^+$.

## 6 Computational Results

### 6.1 Data Instance

All instances of DARP concerned by our tests, given by Cordeau et al. (2003), available at: `http://neumann.hec.ca/chairedistributique/data/darp`. Half of the requests are outbound and half are inbound. They are divided into classes $(a)$ and $(b)$, the difference being that class $(a)$ instances have tighter time windows. In the instances, $m$ denotes the number of vehicles and $n$ is the number of requests. These instances are composed of 24 and 144 requests. In each instance each request has a time window associated with either its origin or its destination. Service time duration equal to 10 is imposed for loading and unloading. Each request is associated with a unit load. The maximum duration of a tour is set to 480, the maximum transport time for request is 90 and the capacity of the vehicle is equal to 6.

Jorgensen et al. (2007) and Parragh et al. (2010) minimize a weighted combination of total routing costs. The objective function thus applied by Jorgensen is,

$$f(s) = w_1c(s)+w_2r(s)+w_3l(s)+w_4g(s)+w_5(w(s)+e(s))+w_6t(s)+w_7d(s) \quad (3)$$

Early arrivals e(s) penalized in the same way as late arrivals w(s). The authors set the weights to $w_1 = 8, w_2 = 3, w_3 = 1$, and $w_4 = 1$, and $w_5 = w_6 = w_7 = n$. We adapted our evaluation function to the same function proposed by Parragh et al. (2010). The function objective is the following:

$$f'(s) = w_1c(s)+w_2r(s)+w_3l(s)+w_4g(s)+w_5e(s)+ \alpha q(s)+\beta d(s)+\gamma w(s)+\tau t(s) \quad (4)$$

Table 1 gives the results generated by the proposed MOTS[1]. We provide values for the objective function $f^{'}(s)$, total travel distance $c(s)$, total route duration $g(s)$, total passenger waiting time $l(s)$, total excess ride time with respect to direct ride time $r(s)$, the sum over early arrivals $e(s)$, all other violations are 0 in our case, $q(s) = d(s) = w(s) = t(s) = 0$. The GA proposed by (Jorgensen et al. (2007))only provides solution values for 13 instances. Therefore, we also restrict our computations to these instances. In order to yield shorter computation times than in our experiments, the maximum iteration is limited to $10 * 10^4$ iterations. and we also provide the best values over five runs. Left columns in Table 2 gives the GA results obtained.

Table 1: MOTS Results ($10^5$ iterations. Average values over 5 run)

| Instance | Travel dis. | Total duration | Pass. wait. | Excess ride time | Early arrival | Total costs |
|---|---|---|---|---|---|---|
| | **c(s)** | **g(s)** | **l(s)** | **r(s)** | **e(s)** | $f^{'}(s)$ |
| R1a | 265,42 | 1092,57 | 0.0 | 21,77 | 0.0 | 3281,21 |
| R2a | 478,06 | 1833,3 | 0.0 | 88,35 | 0.0 | 5922,84 |
| R3a | 861,08 | 2434,84 | 0.0 | 293.75 | 0.0 | 10204.73 |
| R5a | 1134,49 | 3736,15 | 4,02 | 210,02 | 0.0 | 13446,16 |
| R9a | 1022,27 | 3235,61 | 2,35 | 1368,38 | 0.0 | 15521,24 |
| R10a | 1218,87 | 3392,05 | 0.0 | 1988,67 | 0.0 | 19109,05 |
| R1b | 258,44 | 775,18 | 0.0 | 22,84 | 0.0 | 2911,42 |
| R2b | 435,61 | 1397,42 | 0.0 | 21,04 | 0.0 | 4945,43 |
| R5b | 1033,96 | 3532,45 | 0.0 | 160,62 | 0.0 | 12287,05 |
| R6b | 1374,6 | 4316,39 | 0.0 | 222,78 | 0.0 | 15981,56 |
| R7b | 378,36 | 1197,89 | 0.0 | 44,55 | 0.0 | 4358,46 |
| R9b | 1142,97 | 3366,51 | 0.0 | 214,05 | 0.0 | 13152,39 |
| R10b | 1588,32 | 4496,45 | 0.0 | 810,3 | 0.0 | 19633,97 |
| **Avg.** | **860.96** | **2677.45** | **0.49** | **420,55** | **0.0** | **10827,24** |

**Avg.**: Average, **Dev.**: Deviation, **dis.**: distance, **Pass.**: Passanger, **Wait.**: waiting

[1] All tests are done with Java Eclipse, on a Intel(R) i3 CPU, 2.4 GHz, 2Go RAM.

### 6.2 Comparison to Genetic Algorithm

The GA proposed by (Jorgensen et al. (2007))only provides solution values for 13 instances We summarize directly the values provided by Table 1 and GA results in the following Table 2. The percentage deviations for the objective function values $f^{'}(s)$, are given in the columns headed with %. $Ar(s)$ : is the average ride time, computed as $Ar(s) = \sum_{i=1}^{n} L_i$. $w(s)$ : the vehicle waiting time, computed as $w(s) = \sum_{i=1}^{2n} w_i$.

Table 2: MOTS vs. GA

| Instance | GA by Jorgensen et al. (2007) | | | | | | Proposed MOTS | | | | | | Dev. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **c(s)** | **g(s)** | **l(s)** | **Ar(s)** | **w(s)** | $f(s)$ | **c(s)** | **g(s)** | **l(s)** | **r(s)** | **e(s)** | $f^{'}(s)$ | |
| R1a | 309 | 1041 | 29 | 19.86 | 252 | 4696 | 265,42 | 1092,57 | 0.0 | 21,77 | 0.0 | 3281,21 | -30 % |
| R2a | 539 | 1969 | 81 | 28.47 | 470 | 19426 | 478,06 | 1833,3 | 0.0 | 88,35 | 0.0 | 5922,84 | -70% |
| R3a | 1047 | 2779 | 144 | 42.79 | 292 | 65306 | 861,08 | 2434,84 | 0.0 | 293.75 | 0.0 | 10204.73 | -84% |
| R5a | 1350 | 4250 | 286 | 42.79 | 500 | 213420 | 1134,49 | 3736,15 | 4,02 | 210,02 | 0.0 | 13446,16 | -94% |
| R9a | 1343 | 3579 | 132 | 57.88 | 94 | 333283 | 1022,27 | 3235,61 | 2,35 | 1368,38 | 0.0 | 15521,24 | -95% |
| R10a | 1811 | 5006 | 401 | 58.42 | 315 | 740890 | 1218,87 | 3392,05 | 0.0 | 1988,67 | 0.0 | 19109,05 | -97% |
| R1b | 284 | 907 | 5 | 26.24 | 143 | 4762 | 258,44 | 775,18 | 0.0 | 22,84 | 0.0 | 2911,42 | -39% |
| R2b | 561 | 1719 | 53 | 25.30 | 198 | 13580 | 435,61 | 1397,42 | 0.0 | 21,04 | 0.0 | 4945,43 | -64% |
| R5b | 1344 | 4296 | 221 | 38.46 | 552 | 98111 | 1033,96 | 3532,45 | 0.0 | 160,62 | 0.0 | 12287,05 | -87% |
| R6b | 1799 | 5309 | 361 | 42.59 | 630 | 185169 | 1374,6 | 4316,39 | 0.0 | 222,78 | 0.0 | 15981,56 | -91% |
| R7b | 478 | 1299 | 27 | 27.50 | 102 | 9169 | 378,36 | 1197,89 | 0.0 | 44,55 | 0.0 | 4358,46 | -52% |
| R9b | 1372 | 3679 | 166 | 49.65 | 147 | 167709 | 1142,97 | 3366,51 | 0.0 | 214,05 | 0.0 | 13152,39 | -92% |
| R10b | 1740 | 4733 | 202 | 55.34 | 113 | 474758 | 1588,32 | 4496,45 | 0.0 | 810,3 | 0.0 | 19633,97 | -96% |
| **Avg.** | 1075 | 3122 | 162 | 40 | 293 | **179252** | 860.96 | 2677.45 | 0.49 | 420,55 | 0.0 | **10827,24** | -76% |

A negative percentage deviation indicates that the corresponding average value obtained by means of the MOTS is better than the according value computed by the GA. The proposed MOTS yields better results for all instances. Comparing the different components of the objective function at an individual level, the results obtained by the MOTS are better in all cases. Comparing a weighted combination of the three terms in the objective function where individual values are available for both algorithms $w_1c(s) + w_3l(s) + w_4g(s)$, the MOTS outperforms the GA by, on average, 18%.

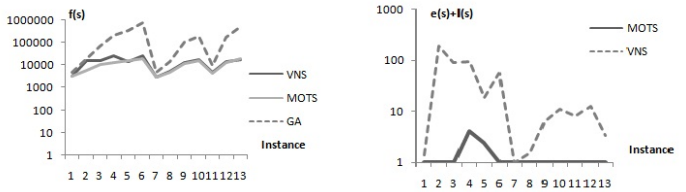
### 6.3 Comparison to Variable Neighborhood Search

The VNS proposed by Parragh et al. (2010) provide solution values also for 13 instances. In order to give meaning to the comparison between the VNS method and results obtained by our approach, the same evaluation function is applied (see equation 4), We also restrict our computations to the 13 instances and we also provide the best values over five runs. Left results in Table 3 gives the VNS

results. We report the total cost $f^{'}(s)$ from Table 1 in the last column. The parameters are also fixed at $w_1 = 8, w_2 = 3, w_3 = 1, w_4 = 1, w_5 = n$.

Table 3: MOTS vs. VNS

| Instance | | VNS by Parragh et al. (2010) | | | | | Proposed MOTS | $\Delta f^{'}(s)$ | Dev. |
|---|---|---|---|---|---|---|---|---|---|
| | **c(s)** | **g(s)** | **l(s)** | **r(s)** | **e(s)** | $f^{'}(s)$ | $f^{'}(s)$ | | |
| R1a | 273.70 | 863.65 | 0.00 | 49.23 | 1.40 | 3234,54 | 3281,24 | 46,70 | 1% |
| R2a | 431.15 | 1774.19 | 0.30 | 113.52 | 189.08 | 14640,09 | 5922,83 | -8717,26 | -60% |
| R3a | 777.92 | 2425.92 | 0.42 | 321.99 | 88.24 | 15968,95 | 10204,73 | -5764,22 | -36% |
| R5a | 1023.77 | 3712.18 | 0.18 | 246.75 | 93.41 | 23851,97 | 13446,15 | -10405,82 | -44% |
| R9a | 1045.34 | 3232.18 | 12.93 | 529.10 | 5.66 | 13806,41 | 15521,26 | 1714,85 | 12% |
| R10a | 1389.73 | 4499.45 | 4.02 | 594.58 | 52.86 | 25016,89 | 19109,02 | 85,69 | -24% |
| R1b | 238.51 | 737.27 | 0.00 | 60.06 | 0.00 | 2825,53 | 2911,22 | -5907,87 | 3% |
| R2b | 431.18 | 1403.74 | 0.00 | 25.30 | 1.54 | 5003 | 4945,42 | -57,58 | -1% |
| R5b | 951.74 | 3417.62 | 0.29 | 197.15 | 6.14 | 12360,08 | 12285,99 | -74,09 | -1% |
| R6b | 1241.22 | 4182.99 | 0.00 | 275.67 | 10.83 | 16499,28 | 15981,53 | -517,75 | -3% |
| R7b | 359.19 | 1123.56 | 0.00 | 103.33 | 8.18 | 4601,55 | 4358,42 | -243,13 | -5% |
| R9b | 977.59 | 3211.71 | 0.02 | 337.99 | 12.65 | 13412,62 | 13152,42 | -260,20 | -2% |
| R10b | 1310.00 | 4218.37 | 1.73 | 493.34 | 1.67 | 16420,6 | 19633,91 | 3213,31 | 20% |
| **Avg.** | 803.93 | 2677.14 | 1.53 | 257.54 | 36.28 | **12895.50** | **10827.24** | -2068,26 | -11% |




(a) total cost (b) early arrival and passenger waiting

Fig. 1: MOTS vs. VNS and GA on a logarithmic scale.

We are thus able to directly compare the values summarized in Table 3, the percentage deviations for the objective function values are given in the columns headed with %, the difference for the costs values are given in the columns headed with $\Delta f(s)$. The proposed MOTS[2] outperforms the VNS by, on average, 11%. The MOTS improved with 69% of instances the VNS method. We observe

[2] Runtime instances, which is not detailed here. it is around 2 minutes for instances with 24 requests, to 60 minutes for large problem size.

that MOTS, gives better results for instances of the class (a), for which the time windows are tight, comparable to other instances Figure 1(a). The Figure 1(b) show clearly the difference on early arrival and the passanger waiting time between the proposed MOTS and the VNS method.

## 7 Conclusion

In this paper we have proposed a multi objective tabu search method combined with a simple heuristic, that we call precedence heuristic, for the static multi-vehicle DARP. The algorithm outperforms proposed results in GA algorithm and VNS method, with an objective that minimizes a weighted combination of total routing costs.
The method promotes customer comfort to the detriment of transport operator costs. Provides results for passengers waiting time, and early arrival time near zero. those is possible only with routes, for which the capacity when leaving a vertex is closer to the load associated to the vertex. Finally we are looking in perspective to other proposed heuristics like memory heuristic and evolutionary local search heuristic.

## References


Aldaihani, M., Dessouky, M. M.: Hybrid scheduling methods for paratransit operations. Computers Industrial Engineering **45** (2003) 75-96.

Baugh,J. W., Krishna,G., Kakivaya,R., Stone,J. R.: Intractability of the dial-a ride problem and a multiobjective solution using simulated annealing. Eng. Optim. **30** (1998) 91123.

Cordeau, J. F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B. **37** (2003) 579-594.

Cordeau, J. F.: A Branch-and-cut algorithm for the dial-a-ride problem. Operations Research **54** (2006) 573-586.

Cordeau, J. F., Laporte, G.: The dial-a-ride problem: Models and algorithms. A.O.R. **153** (2007) 29-46.

Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston (1997).

Jorgensen, R. M., Larsen, J., Bergvinsdottir, K. B.: Solving the dial-a-ride problem using genetic algorithms. J Oper Res Soc. **58** (2007) 1321-1331.

Lacomme P., Quilliot A., Zhao X.: A heuristic and split procedure for dial-a-ride problems. EU/MEeting Troyes, France, October 23-24 2008.

Mladenovic, N., Hansen, P.: Variable neighborhood search. Comput Oper Res. **24** (1997) 1097-1100.

Paquette, J., Cordeau, J. F., Laporte, G.: Quality of service in dial-a-ride operations. C. I.E. **56** (2009) 1721-1734.

Parera, S. N., Doerner, K. F., Gandibleux, X., Hartl, R. F.: A heuristic two-phase solution method for the multi-objective dial-a-ride problem. Networks. doi:10.1002/net.20335 (2009).

Parragh, S. N., Doerner, K. F., Hartl, R. F.: Variable neighborhood search for the dial-a-ride problem. Computers Operations Research **37 6** (2010) 1129 - 1138.

Savelsbergh, M.W.P. : The vehicle routing problem with time windows: Minimizing route duration. ORSA Journal on Computing **4** (1992) 146-154.

# Clustering des données

# Clustering multi-niveaux de graphes : hiérarchique et topologique

Nhat-Quang Doan and Amine Chaibi and Hanane Azzag and Mustapha Lebbah

LIPN-UMR 7030
Universite Paris 13 - CNRS
99, av. J-B Clement - F-93430 Villetaneuse
{firstname.secondname}@lipn.univ-paris13.fr

**Abstract.** Nous présentons dans cet article une approche nommée GrSOtree pour la classification de données structurées en graphes. Cette méthode a l'avantage de proposer une décomposition du graphe dans un nouvel espace de représentation en fournissant une organisation multi-niveaux : topologique et hiérarchique afin de faciliter l'interprétation des relations intrinsèques présentent dans le graphe. Nous évaluerons les capacités et les performances de notre approche sur des graphes de difficultés variables. Des résultats numériques et visuels seront présentés et discutés.

## 1 Introduction

Actuellement nous rencontrons de plus en plus de données structurées sous forme de graphes. La fouille de graphes est devenue une problématique de recherche intéressante et un défi réel en matière de fouille de données. Les méthodes de visualisation et de classification permettent d'aider à la compréhension des données structurées et particulièrement celles présentées sous forme de graphe. Le cas des grands graphes représente une problématique à part entière dans le domaine de l'apprentissage.

Le but de ce travail est de présenter un algorithme de classification qui permet de visualiser et de décomposer le graphe en plusieurs sous-arbres organisés sur une carte 2D. De manière plus générale, un graphe non orienté est formé par un ensemble d'arêtes qui représentent des connections entre des paires de sommets (ou noeuds). La fouille de graphes consiste à extraire l'information utile se trouvant dans la structure formée par les arrêtes et les sommets. Actuellement les algorithmes de classification standards effectuent cette tâche de manière simpliste sans prendre en compte la topologie qui existe entre les sommets.

Dans ce travail, nous souhaitons fournir une décomposition multi-niveaux du graphe d'origine : hiérarchique et topologique. Les modèles topologiques sont souvent utilisés pour la visualisation et la classification non supervisée. Des extensions et des reformulations du modèle SOM ont été décrites dans [1] [2] [3]. Ces approches sont différentes les unes des autres, mais partagent la même idée

de représenter de grands ensembles de données par une relation géométrique projetée sur une carte topologique 2D. Habituellement, nous disposons d'algorithme offrant une classification topologique ou encore uniquement hiérarchique [4]. Dans la littérature il existe peu d'algorithmes offrant la possibilité d'avoir en une seule passe une classification multi-niveaux. Les seuls travaux similaire sont ceux de SOM hiérarchique de [5] où les auteurs présentent une organisation multi-niveaux sur des référents de la carte en proposant ainsi plusieurs niveaux de cartes.

Certaines méthodes hiérarchiques basées sur les carte SOM ont également été proposées par exemples TS-SOM [6], GH-SOM [7], TreeSOM [8] et SOM-AT [9]. Notre approche Gr-SOTree génère non seulement une carte topologique mais aussi simultanément plusieurs arbres hiérarchiques pendant l'étape d'apprentissage. Un noeud dans la structure d'arbre représente une donnée du graphe.

Nous avons précédemment proposé une version de ce travail mais qui traite des données traditionnelles (individu/variable) [10]. Nous avons adapté notre modèle aux données graphes en proposant ainsi un nouvel espace de représentation du graphe : topologique et hiérarchique. Nous avons introduit de nouvelles notions liées à la structure en graphes des données (référent 'leader', fonction de coût, fonction d'affectation en groupe).

## 2 Clustering de graphes : le modèle Gr-SOTree

Les graphes sont des objets combinatoires décrits par : le degré, la connectivité, le chemin et le poids. Ainsi, les méthodes de clustering ne peuvent pas s'appliquer directement sur les graphes. Nous avons ainsi chercher à étudier comment les méthodes vectorielles peuvent être appliquées aux données de type graphes. L'idée commune est d'utiliser une transformation de graphe dans un nouvel espace où la similarité peut être calculée. Des approches utilisant les modèles des cartes auto-organisées comme : dissimilarité SOM (D-SOM) [11] et Kernel SOM [12] ont été proposés pour s'adapter à ce type de données en utilisant des fonctions noyau. Une autre alternative serait d'utiliser la matrice Laplacienne.La combinaison d'un vecteur ou de plusieurs vecteurs propres $L$ est suffisante pour calculer la similarité ou la distance géométrique entre les nœuds [13], [14].

L'idée principale de notre modèle est de reconstruire le graphe d'origine $G(V, E)$ en le décomposant de manière hiérarchique en plusieurs sous-arbres auto-organisés, formant ainsi une forêt d'arbres projetée sur une carte 2D. En suivant la théorie spectrale des graphes, nous utilisons, en partie, le laplacien et ses vecteurs propres. Ainsi, nous considérons les premiers $\lambda$ vecteurs propres $e_1, .., e_\lambda$ $(\forall i = 1..\lambda, e_i \in \Re^n)$ pour former une nouvelle matrice des données $X \in \Re^{n \times \lambda}$. Une ligne de $\mathbf{x}_i \in X$ illustre un nœud $v_i \in V$ du graphe. Ainsi, nous pouvons décrire un graphe $G$ dans l'espace continu $X$ où la similitude entre deux sommets $sim(v_i, v_j)$ est équivalente à la distance entre leurs vecteurs respectifs $\mathbf{x}_i$ et $\mathbf{x}_j$. L'utilisation de $\lambda$ vecteurs propres $(1 \leq \lambda \leq n)$ pour représenter tous les nœuds du graphe dans $\Re^{n \times \lambda}$, permet de réduire la dimension.

En utilisant les modèles topologiques, un graphe $G$ représenté par $X$ peut être regroupé et visualisé dans une grille régulière en 2D ou 1D (de taille $K$) en utilisant le processus d'auto-organisation. La grille $\mathcal{C}$ représente une forêt d'arbres organisés sur la grille en $K$ sous-graphes représentés en arbre. Chaque cellule $c$ de la grille est appelée par la suite «support (root)» d'un arbre noté $tree_c$ et chaque nœud de l'arbre représente un nœud $v_i \in V$. Les arbres sont construits en utilisant les principes d'un algorithme de classification non supervisé hiérarchique [15]. Chaque nœud $v_i$ sera connecté au plus proche voisin $v_j$ de la même manière que $\mathbf{x}_i$ est le vecteur le plus proche de $\mathbf{x}_j$. Pour chaque paire d'arbres $tree_c$ et $tree_r$ sur la carte, la distance $\delta(c, r)$ est définie comme la longueur de la plus courte chaîne reliant les deux arbres. L'influence mutuelle entre deux sous-arbres $tree_c$ et $tree_r$ de racine $c$ et $r$ sera donc définie, de la même manière que les cartes topologiques, par la fonction $\mathcal{K}^T(\delta(c, r))$ où $T$ représente la taille du voisinage (la température).

Principalement, les algorithmes de clustering ont en commun de considérer le barycentre de chaque cluster comme le prototype qui doit être mis à jour pour chaque itération. Toutefois, le centroïde n'est pas l'élément le plus important dans un graphe. Dans notre approche, au lieu d'utiliser le centroïde, nous avons opté pour l'utilisation du meneur "leader" qui est considéré comme le sommet représentant de tous les sommets d'un graphe ou d'un sous-graphe, [16]. Dans la première version de ce travail, nous avons considéré un "leader" comme le nœud qui a le plus grand degré. Ainsi, nous associons à chaque $tree_c$ un prototype noté $leader(c)$ dont l'expression est définie comme suit:

$$leader(c) = \max_{\forall i \in tree_c} (deg(v_i)) \tag{1}$$

Le choix d'un prototype représentatif permet facilement d'adapter notre algorithme à d'autres types de données. Ainsi, la fonction objectif de l'auto-organisation des arbres s'écrit comme suit:

$$\mathcal{R}(\phi, \mathcal{L}) = \sum_{c=1}^{k} \sum_{r=1}^{k} \sum_{i \in tree_c} \mathcal{K}(\delta(\phi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{x}_{leader(r)}||^2 \tag{2}$$

où $\mathcal{L} = \cup_{r=1}^{k} leader(r)$ and $\phi$ est la fonction d'affectation d'un groupe de nœuds à la fois.

$$\phi(childNode(v_i)) = \arg\min_{r} \sum_{c=1..k} \mathcal{K}^T(\delta(r, c)) \left\|\mathbf{x}_i - \mathbf{x}_{leader(c)}\right\|^2 \tag{3}$$

où $childNode(v_i)$ est l'ensemble des nœuds contenant $v_i$ et tous les nœuds récursivement connectés à $v_i$. La figure 1 montre un exemple d'affectation simultanée d'un ensemble de nœuds $\{e, f, g\} \subset V$ formant un arbre.

La minimisation de la fonction de coût $\mathcal{R}$ est un problème d'optimisation combinatoire. En pratique, on se contente d'une solution sous-optimale. Nous

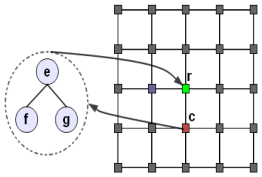



**Fig. 1.** Exemple d'une affectation d'un arbre de nœuds de la cellule $c$ à la cellule $r$. $\{e, f, g\} \subset V$, $childNode(e) = \{e, f, g\}$

**Require:** $v_i \in V$ ($\mathbf{x}_i \in X$), Carte $\mathcal{C}(c \in \mathcal{C})$
**Ensure:** Forêt d'arbre
1: **repeat**
2: $c \leftarrow \phi(childNode(v_i))$.
3: **if** $v_i$ est connecté **then**
4: $tree_c \leftarrow$ constructTree($v_i$, $tree_c$)
5: **else**
6: **if** $c! = c_{old}$ **then**
7: **if** $v_i$ n'est pas déconnecté **then**
8: déconnecter $v_i$ et ses fils de l'arbre $tree_{c_{old}}$
9: **end if**
10: $childNode(v_i) \leftarrow$ reçoit tous les nœuds de $v_i$
11: $tree_c \leftarrow$ constructTree($childNode(v_i)$,$tree_c$)
12: Mise à jour des meneurs "Leaders" ($r \in \mathcal{C}$ )
13: **end if**
14: **end if**
15: **until** Itération Finale

Alg 1: Gr-SOTree

proposons ici de minimiser la fonction de coût de la même manière que la version des nuées dynamiques, mais en utilisant les caractéristiques statistiques fournies par les arbres (associés à chaque cellule) afin d'accélérer la convergence de l'algorithme.

**Algorithme Gr-SOTree : self-Organizing Tree** Notre algorithme principal est donné en pseudo-code par l'algorithme 1. La fonction *constructTree* est présentée par l'algorithme 2. Notons par $v_{pos}$ le nœud ou le support où le nœud $v_i$ est connecté dans l'arbre qui est associé à une cellule, et $v^+$ le nœud connecté à $v_{pos}$ le plus similaire à $v_i$. Si un nœud $v_i$ a été déconnecté de l'arbre, ce nœud et ses nœuds-fils vont être reconnectés selon la ligne 11 de l'algorithme 2. La fonction *constructTree* est appelée pour chaque nœud de l'ensemble $childNode(v_i)$.

**Require:** $v_i \in V$, $tree_j$
**Ensure:** *Arbres*
**if** pas de nœuds ou uniquement un seul nœud connecté à $v_{pos}$ **then**
  $tree_j \leftarrow$ connecter $v_i$ à $v_{pos}$
**end if**
**if** Deux nœud connectés à $v_{pos}$ et c'est la première fois **then**
  $tree_j \leftarrow$ déconnecter le plus similaire à $v_i$ de $v_{pos}$ (et récursivement tous les nœuds-fils $v_i$)
  $tree_j \leftarrow$ connecter $v_i$ à $v_{pos}$
**else**
  $T_{Dissim}(v_{pos}) \leftarrow min(sim(v_k, v_j))$ où $v_j, v_k$ sont les fils de $v_{pos}$
  **if** $sim(v_i, v^+) < T_{Dissim}(v_{pos})$ **then**
    $tree_j \leftarrow$ connecter $v_i$ à $v_{pos}$
  **else**
    déplacer $v_i$ à $v^+$
  **end if**
**end if**

ALG 2: constructTree

## 3 Validation numérique et visuelle

Pour évaluer la qualité de l'approche proposée. Nous avons utilisés 3 bases de données supervisées disponibles sur `http://www-personal.umich.edu/~mejn/netdata/`. Ces bases représentent des graphes non orientés et non pondérés.

**Table 1.** Bases de données

| Bases | Noeuds | Arcs | Classes |
|---|---|---|---|
| Adjective and Noun | 112 | 425 | 2 |
| Football Teams | 115 | 616 | 10 |
| Political blogs | 1490 | 19090 | 2 |

### 3.1 Mesure de qualité

Soit $G(V, E)$ un graphe partitionné en $k$ clusters $tree_1, .., tree_c, .., tree_k$. Le nombre de noeuds dans $tree_c$ est noté par $N_c$; $M_c$ est le nombre d'arcs dans $tree_c$, $M_c = \{\{u, v\}; u, v \in tree_c\}$; et $B_c$ est le nombre d'arcs à la frontière de $tree_c$, $B_c = \{\{u, v\}; u \in tree_c, v \notin tree_c\}$. Pour mesurer la qualité d'un clustering de graphe, nous utilisons 3 critères d'evaluation [17] :

- Conductance: $C = \frac{B_c}{2M_c + B_c}$ mesure la proportion des arêtes qui pointent en dehors du cluster $tree_c$.
- Densité: $V = \frac{M_c}{N_c(N_c-1)/2}$ est la densité des arêtes internes au sein du cluster $tree_c$.

– Modularité [18], [19], [20]:

$Q = \frac{1}{2m} \sum_{c=1}^{k} \sum_{i \in tree_c} \sum_{j \in tree_c} \left( w_{ij} - \frac{deg(i)deg(j)}{2m} \right)$

où $m = \frac{1}{2} \sum_{i=1}^{n} deg(i)$.

L'idée est de prendre le nombre d'arêtes relevant des groupes moins le nombre prévu dans un réseau équivalent avec des arêtes placées au hasard.

On juge qu'un cluster est de bonne qualité s'il a plus de liens internes que de liens externes avec les autres clusters. Ainsi une bonne classification doit retourner une bonne valeur de la modularité et de la densité, à contrario une faible valeur de la conductance.

### 3.2 Résultats numériques

La table 3.2 présente les mesures moyennes de la qualité et leurs écarts-types obtenus à partir de 10 expériences. Les bases étant supervisées, nous avons pu également calculer la proportion de données bien classées appelée pureté. Tout d'abord, on remarque que la qualité de la classification dépend fortement du choix de $\lambda$, mais dans ce travail nous ne cherchons pas pour l'instant à optimiser la valeur de $\lambda$ et son influence sur les mesures de qualité. Dans cette section expérimentale, nous nous limitons à effectuer certaines expériences afin d'évaluer et de comparer notre méthode avec d'autres approches: $K$-means, SOM et MST (Minimum Spanning Tree). La méthode MST [21] construit des graphes de voisinages à partir d'un ensemble de données (noeuds) en connectant toutes les pairs de noeuds qui satisfont une certaine fonction de coût basée sur la distance

Dans le tableau 3.2, les résultats de la base "Adjective and noun" montre que les mesures de qualité pour $K$-means, SOM et Gr-SOTree sont pour la plupart égaux sauf pour la modularité. Le meilleur résultat est obtenu par Gr-SOTree lorsque $\lambda = 11$ avec une densité de 0.30. Cependant, les mauvais résultats pour MST sont assez décevants. Concernant la base "Football", notre méthode est meilleur avec une densité de 0.792 pour $\lambda = 11$. Par contre, la différence dans les valeurs de conductance n'est pas significative, et les valeurs de notre modularité donnent des résultats comparables à ceux obtenues par $K$-means et SOM. Contrairement au premier exemple, l'algorithme MST semble meilleur, mais encore loin derrière. La meilleure situation pour Gr-SOTree concerne la base «Political blogs» où la méthode proposée domine toutes les mesures pour $\lambda = 39$, sauf pour la pureté.

| Method | $\lambda$ | Conductance $\searrow$ | Density $\nearrow$ | Purity $\nearrow$ | Modularity $\nearrow$ |
|---|---|---|---|---|---|
| Adjective and noun | | | | | |
| K-means | 5 | 0.734 ± 0.012 | 0.242 ± 0.044 | 0.580 ± 0.023 | 0.139 ± 0.012 |
| | 11 | 0.684 ± 0.015 | 0.286 ± 0.048 | 0.577 ± 0.019 | 0.167 ± 0.027 |
| SOM | 5 | 0.743 ± 0.024 | 0.175 ± 0.015 | 0.574 ± 0.028 | 0.160 ± 0.014 |
| | 11 | 0.691 ± 0.022 | 0.214 ± 0.025 | 0.576 ± 0.022 | 0.199 ± 0.017 |
| MST | 5 | 0.894 | 0.014 | 0.580 | 0.028 |
| | 11 | 0.897 | 0.008 | 0.571 | 0.013 |
| Gr-SOTree | 5 | 0.784 ± 0.014 | 0.208 ± 0.038 | 0.565 ± 0.004 | 0.125 ±0.015 |
| | 11 | 0.736 ± 0.029 | 0.300 ± 0.086 | 0.560 ± 0.016 | 0.126 ± 0.042 |
| Football Teams | | | | | |
| K-means | 5 | 0.573 ± 0.024 | 0.717 ± 0.038 | 0.849 ± 0.045 | 0.455 ± 0.033 |
| | 11 | 0.541 ± 0.036 | 0.770 ± 0.057 | 0.863 ± 0.056 | 0.480 ± 0.045 |
| SOM | 5 | 0.568 ± 0.085 | 0.529 ± 0.076 | 0.692 ± 0.037 | 0.505 ± 0.018 |
| | 11 | 0.406 ± 0.081 | 0.645 ± 0.078 | 0.735 ± 0.055 | 0.544 ± 0.020 |
| MST | 5 | 0.610 | 0.440 | 0.800 | 0.563 |
| | 11 | 0.478 | 0.706 | 0.965 | 0.589 |
| Gr-SOTree | 5 | 0.564 ± 0.012 | 0.715 ± 0.060 | 0.880 ± 0.016 | 0.464 ± 0.025 |
| | 11 | 0.532 ± 0.039 | 0.792 ± 0.057 | 0.878 ± 0.058 | 0.479 ± 0.037 |
| Political Blogs | | | | | |
| K-means | 5 | 0.801 ± 0.039 | 0.054 ± 0.013 | 0.877 ± 0.008 | 0.114 ± 0.015 |
| | 39 | 0.813 ± 0.028 | 0.083 ± 0.030 | 0.840 ± 0.011 | 0.180 ± 0.024 |
| SOM | 5 | 0.854 ± 0.016 | 0.081 ± 0.021 | 0.861 ± 0.003 | 0.116 ± 0.022 |
| | 39 | 0.845 ± 0.024 | 0.062 ± 0.017 | 0.827 ± 0.016 | 0.160 ±0.017 |
| MST | 5 | 0.961 | 0.007 | 0.53 | 0 |
| | 11 | 0.928 | 0.003 | 0.512 | 0 |
| Gr-SOTree | 5 | 0.884 ± 0.027 | 0.094 ± 0.009 | 0.854 ± 0.013 | 0.117 ± 0.025 |
| | 39 | 0.785 ± 0.015 | 0.227 ± 0.071 | 0.767 ± 0.028 | 0.197 ± 0.035 |

### 3.3 Visualisation des graphes

L'objectif de cette partie expérimentale est de montrer que notre méthode fournit des informations supplémentaires. Dans notre approche nous proposons une décomposition du graphe d'origine en fournissant une organisation multi-niveaux : hiérarchique et topologique. Ces structures permettent de simplifier l'exploitation du graphe en offrant une visualisation conviviale. Nous avons utilisé Tulip ,[22], comme plateforme de visualisation.

- **Visualisation du graphe original**:
  Le graphe original s'affiche avec une couleur unique pour l'ensemble des nœuds dont les étiquettes sont disponibles. La taille du nœud varie selon le degré.
- **Visualisation multi-niveaux : topologique et hiérarchique**:
  Cette visualisation représente un nouvelle espace de reconstruction qui produit un graphe hyperbolique et interactif où le premier niveau d'organisation

est la vue de la carte topologique qui affiche à chaque cellule un arbre. Nous pouvons également voir la hiérarchie des $k$ arbres créés par Gr-SOTree. Un arbre est clairement identifié par sa structure et par une couleur différente. Les meneurs "leaders" apparaissent également dans chaque arbre. Par conséquent, Il devient plus facile d'extraire des informations de cette nouvelle arborescence extraite du graphe d'origine. D'autre part, en raison des relations hiérarchiques, cette visualisation a permis d'éliminer les nœuds isolés ou les groupes isolés. Un exemple est présenté dans les figures 2(b) et 3(b).

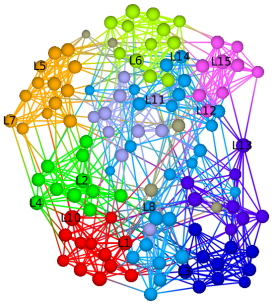
– **Visualisation des groupes**:
Chaque cluster est représenté par un "leader" et une couleur unique. L'étiquette d'un cluster (ou une cellule) est déterminée par la règle du vote majoritaire. Par conséquent, nous pouvons dessiner le graphe et permettre aux utilisateurs de comparer avec le graphe d'origine. Par exemple, nous remarquons que les "leaders" sont regroupés en deux grands groupes dans la figure 3(c). Chaque cluster peut être représenté par plusieurs "leaders" que nous pouvons considérer par la suite comme des points d'intérêts. Il convient de noter que les nœuds qui ont le plus haut degré ne sont pas toujours choisis comme "leader" parce que ces nœuds ont de nombreux liens avec d'autres groupes et que ce type de nœud ne satisfait pas l'équation 1.

– **Visualisation topologique**:
Pour la visualisation topologique, nous présentons ici deux notions: les liens forts (en rose) et les liens faibles (en gris). Comme nous l'avons mentionné plus haut, l'algorithme SOM utilise une fonction de voisinage pour déterminer les voisins d'une cellule et par conséquent d'un arbre. Un lien faible est créé entre deux cellules voisines. Un lien fort est crée s'il existe une arête dans graphe original qui relie un couple de "leaders" situés dans deux cellules voisines. Les figures 2(e) et 3(e) représentent un exemple de visualisation des liens forts et faibles. La taille d'une cellule est proportionnelle aux nombre de noeuds affectés à cette cellule.

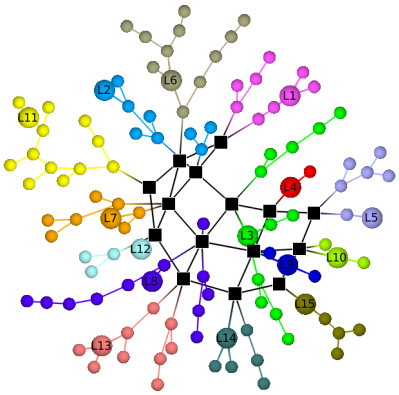
## 4 Conclusions et perspectives

Dans ces travaux nous avons présenté une nouvelle approche pour le clustering de graphes. Cette nouvelle méthode basée sur l'auto-organisation fournit un nouvel espace de représentation : la carte 2D et les arbres permettant une meilleure représentation d'un graphe. Nous avons introduit deux nouvelles notions importante : la notion de leader en tant que prototype et l'utilisation de la structure d'arbre pour définir une fonction d'affectation d'un ensemble de noeuds. Notre modèle offre un nouvel espace de visualisation proposant une representation du graphe plus riche en information.

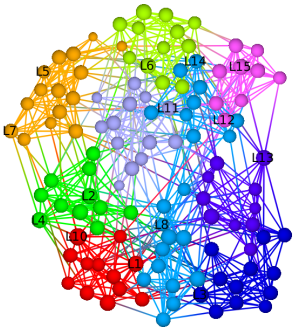
Comme travail futur, nous souhaitons étudier une approche de graphe dynamique et évolutif. En construisant ainsi des sous-parties du graphe de manière



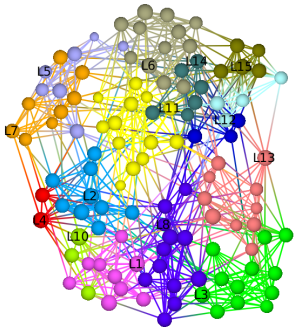
(a) Graphe original: chaque couleur indique la classe réelle.



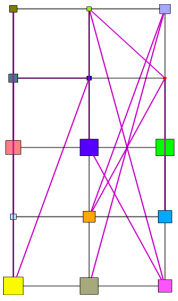
(b) Visualisation multi-niveaux : topologique et hiérarchique. Les "leaders" sont indiqués avec des nœuds de taille plus grande et avec le symbole $L$.



(c) Visualisation des groupes: chaque couleur indique la classe obtenue après vote majoritaire.

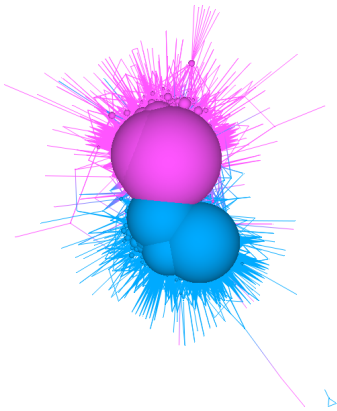


(d) Visualisation des groupes: Le graphe est partitionné en $5 \times 3$ groupes.

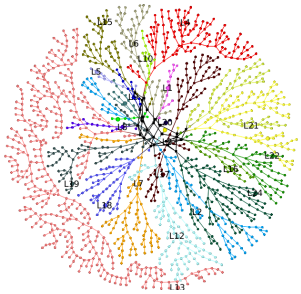


(e) Visualisation topologique: Carte $3 \times 3$ avec les liens forts et faibles. La taille des cellules est proportionnelle aux nombres de noeuds. Chaque couleur indique la classe trouvée après vote majoritaire.

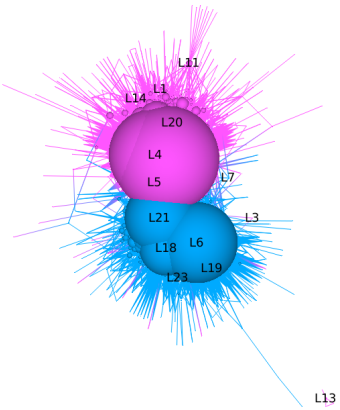
**Fig. 2.** Graph visualization of "Football Teams"



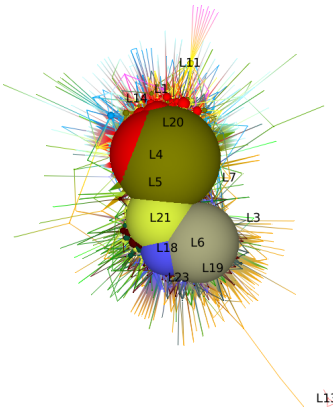
(a) Graphe original: chaque couleur indique la classe réelle.



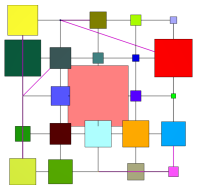
(b) Visualisation multi-niveaux : topologique et hiérarchique. Chaque arbre représente une cellule. Les 'leaders' sont indiqués avec des nœuds de taille plus grande et avec le symbole $L$.




(c) Visualisation des groupes: chaque couleur indique la classe obtenue après vote majoritaire.




(d) Visualisation des groupes: Le graphe est partitionné en $5 \times 5$ groupes.



(e) Visualisation topologique: Carte $3 \times 3$ avec les liens forts et faibles. La taille des cellules est proportionnelle aux nombres de noeuds. Chaque couleur indique la classe trouvée après vote majoritaire.

**Fig. 3.** Graph visualization of "Political Blogs"

incrémentale, nous souhaitons, montrer comment des sous-graphes peuvent évoluer au cours du temps. Comme autre perspective nous soulignons l'importance de se concentrer sur le nouveau concept de points d'intérêt qui caractérisent les points "leader" d'un cluster.

## References


1. C. M. Bishop, M. Svensén, and C. K. I.Williams. Gtm: The generative topographic mapping. *Neural Comput*, 10(1):215–234, 1998.
2. Nathalie Villa-Vialaneix Fabrice Rossi. Optimizing an organized modularity measure for topographic graph clustering: A deterministic annealing approach. *Neurocomputing*, 73(7-9), (2010).
3. Fabrice Rossi. Barbara Hammer, Alexander Hasenfuß. Median topographic maps for biomedical data sets. *CoRR*, page 0909.0638, 2009.
4. E. Diday and J.C. Simon. Cluster analysis. In K.S. Fu, editor, *Digital Pattern Recognition*, pages 47–94. Springer-Verlag, Berlin, 1976.
5. Michael Dittenbach, Andreas Rauber, and Dieter Merkl. Recent advances with the growing hierarchical self-organizing map. 2001.
6. Pasi Koikkalainen and Ismo Horppu. Handling missing data with the tree-structured self-organizing map. In *IJCNN*, pages 2289–2294, 2007.
7. Michael Dittenbach, Dieter Merkl, and Andreas Rauber. The growing hierarchical self-organizing map. pages 15–19. IEEE Computer Society, 2000.
8. Elena V. Samsonova, Joost N. Kok, and Ad P. Ijzerman. Treesom: Cluster analysis in the self-organizing map. neural networks. *American Economic Review*, 82:1162–1176, 2006.
9. Markus Peura. The self-organizing map of attribute trees, 1999.
10. Anonymes. Topological hierarchical tree using artificial ants. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ICONIP'10, pages 652–659, Berlin, Heidelberg, 2010. Springer-Verlag.
11. Teuvo Kohonen and Panu Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15(8-9):945–952, 2002.
12. Donald Macdonald and Colin Fyfe. The kernel self-organising map. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 317–320, 2000.
13. Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, February 1997.
14. Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.
15. Hanene Azzag, Christiane Guinot, and Gilles Venturini. Data and text mining with hierarchical clustering ants. In *Swarm Intelligence in Data Mining*, pages 153–189. 2006.
16. Angel Stanoev, Daniel Smilkov, and Ljupco Kocarev. Identifying communities by influence dynamics in social networks. *CoRR*, abs/1104.5247, 2011.
17. Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 631–640, New York, NY, USA, 2010. ACM.
18. M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006.

19. Aaron Clauset, M. E. J. Newman, , and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.
20. M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3):036104+, 2006.
21. Oleksandr Grygorash, Yan Zhou, and Zach Jorgensen. Minimum spanning tree based clustering algorithms. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '06, pages 73–81, Washington, DC, USA, 2006. IEEE Computer Society.
22. D. Auber. Tulip : A huge graph visualisation framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Softwares*, Mathematics and Visualization, pages 105–126. Springer-Verlag, 2003.

# Optimisation par Essaim de Particules Quantiques et ses Nouvelles Hybridations pour le Clustering des Données

**Imen Boulnemour [(1)]- Chafika Ramdane [(1)]- Amina Laib[(1)]**

[(1)]Département d'informatique, Université du 20 août 1955 de Skikda
BP 24, Skikda-21000, Algérie
{boulnemourimen, ramdanechafika, amina_laib}@yahoo.fr

**Résumé.** Les algorithmes de clustering des données flous et possibilistes peuvent donner des résultats satisfaisant. Cependant leurs principaux inconvénients sont la forte dépendance de leurs résultats à leur configuration initiale et leur convergence prématurée vers des optimums locaux, pas toujours satisfaisant. Pour résoudre ces problèmes, des métaheuristiques telles que l'algorithme d'optimisation par essaim de particules quantiques QPSO, ont été utilisé. Ce dernier est connu pour avoir des résultats meilleurs que ceux des algorithmes classiques de clustering et ceux grâce à sa grande capacité de recherche et sa convergence globale. Afin de faire l'équilibre entre la capacité de recherche globale et locale de l'algorithme QPSO, nous proposons deux algorithmes hybrides, le premier combinant l'algorithme de clustering floue FCM à l'algorithme QPSO et le deuxième combinant l'algorithme de clustering possibiliste PCA à l'algorithme QPSO. Les résultats expérimentaux et les comparaisons effectuées montrent que nos algorithmes hybrides sont meilleurs que l'algorithme classique QPSO en terme de percions des résultats de clustering.

**Mots clés**: clustering par essaim de particules quantiques, clustering flou, clustering possibiliste, optimisation.

## 1 Introduction

Le clustering aussi appelé classification non supervisée est le processus qui permet d'identifier des groupes homogènes au sein d'un ensemble de données multidimensionnelles. Le clustering est fait de telle manière que les données appartenant au même groupe soient les plus similaires possibles les uns des autres, au sens d'un certain critère de similarité et que les données appartenant à des groupes différents soient les plus dissimilaires possibles [1].

Le clustering est une technique importante dans l'analyse exploratoire des données, la reconnaissance des formes, la fouille des données, la segmentation d'images et beaucoup d'autres domaines.

Généralement, les méthodes de clustering se répartissent en deux classes : les méthodes hiérarchiques et les méthodes par partition. Les algorithmes de clustering par partitionnement génèrent une seule partition, avec un certain nombre déterminé ou estimé de clusters. K-means est l'algorithme le plus connu de cette catégorie. D'un autre côté, le clustering peut être exact ou basé sur des degrés d'appartenance. Le clustering exact, restreint l'appartenance de chaque point du jeu de données à un seul cluster. Tandis que dans le clustering basé sur des degrés d'appartenance, chaque

point du jeu de données peut appartenir à tous les clusters avec des degrés d'appartenance différents, qui sont décrits par une fonction variant du clustering flou au clustering possibiliste [1].

Les algorithmes classiques de clustering sont connus pour leur convergence rapide vers des optimums locaux. Pour palier ce problème, le clustering a été reformulé comme un problème d'optimisation et plusieurs métaheuristiques lui ont été appliquées, tels que les algorithmes génétiques (GA) [2], l'algorithme de colonies de Fourmies(ACO) [3,4] et l'algorithme d'optimisation par essaim de particules (PSO) [5].

La méthode PSO est née de l'intelligence en essaim. Elle a été influencée par les travaux de Reynold [6] et Heppener [7] qui cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale, formant un ''V''. Cette technique est souvent décrite comme une sorte d'algorithme évolutionnaire, avec une population d'agents appelés particules, dans laquelle, à chaque pas de temps, les « meilleurs » (selon un critère prédéfini) sont plus ou moins imités par les autres et coopèrent pour la réussite d'un travail commun. La méthode PSO est favorablement comparable aux algorithmes évolutionnaires [8] car elle utilise moins de paramètres de control et un nombre d'itérations et d'exécution nettement inférieur à celui de ces derniers.

L'algorithme d'optimisation par essaim de particules quantiques(QPSO) est une hybridation entre l'algorithme PSO et des concepts quantiques issus de la mécanique quantique. Il a été proposé par [9,10,11] et a prouvé son efficacité dans plusieurs problèmes. Sun et al [12] ont été les premiers à l'appliquer au problème du clustering. Leurs résultats expérimentaux indiquent que QPSO est plus performant que PSO et K-means, sur plusieurs jeux de données. Ceci est du au fait que, QPSO est un algorithme d'optimisation à convergence globale, tandis que PSO ne l'est pas selon les critères utilisés par Van de Bergh [13].

Cependant l'algorithme QPSO donne parfois des résultats instables, Ceci est du a sa nature quantique.

Dans ce papier nous contribuons à équilibrer et à améliorer la précision des résultats de clustering de l'algorithme QPSO en le combinant une fois avec l'algorithme FCM et une autre avec l'algorithme PCA. Le reste de ce papier est organisé comme suit : Les sections 2 et 3 présentent une brève description des algorithmes FCM et PCA respectivement. L'algorithme QPSO pour le clustering est présenté dans la section 4. Les algorithmes hybrides proposés sont présentés dans la section 5. La section 6 présente les résultats expérimentaux sur cinq jeux de données et le papier est conclu dans la section 7.

## 2 Algorithme de clustering flou FCM

L'algorithme FCM (Fuzzy Clustering Means) est une méthode de clustering flou, inventée par Bezdek [5], qui permet de regrouper les points de données en *C* clusters.

Dans cet algorithme, on n'affecte pas les points de données aux clusters, mais on calcule une matrice d'appartenance dont les lignes sont les points et les colonnes sont les clusters. Soit cette matrice $M$, l'élément $M_{ij}$ donne la probabilité d'appartenance du point $o_j$ au cluster $c_i$ ayant son centroïde $g_i$. La matrice d'appartenance est définie par l'équation (1). L'algorithme FCM est itératif, il utilise les conditions nécessaires

pour minimiser la fonction objective, définie par l'équation (2), où *C* est le nombre de clusters, N le nombre d'objets et $m$ est un paramètre appelé coefficient de flou, il contrôle la quantité de flou dans la partition.

$$M_{ij} = \left( \sum_{c=1}^{C} \left( \frac{d(o_j, g_i)}{d(o_j, g_c)} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (1)$$

$$f_{FCM} = \sum_{i=1}^{C} \sum_{j=1}^{N} M_{ij}^{m} d(o_j, g_i)^2, m > 1 \quad (2)$$

L'algorithme FCM est décrit par les étapes suivantes :

1. Fixer arbitrairement la matrice d'appartenance et les centroïdes. La matrice représente une partition floue des données et doit vérifier les conditions suivantes :

$$\forall i \in \{1..C\}, \forall j \in \{1..N\} \begin{cases} M_{ij} \in [0,1] \\ 0 < \sum_{j=1}^{N} M_n < N \end{cases}, \sum_{i=1}^{C} M_{ij} = 1$$

2. Calculer les nouveaux centroïdes avec l'équation (3)

$$g_i = \frac{\sum_{i=1}^{N} (M_{ij})^m o_j}{\sum_{i=1}^{N} (M_{ij})^m} \quad (3)$$

3. Réajuster la matrice d'appartenance suivant la position des centroïdes en utilisant l'équation (1).

4. Si l'algorithme ne converge pas, retoure à l'étape 2.

Plusieurs conditions d'arrêt peuvent être utilisées, comme terminer l'algorithme lorsque le changement relatif dans les valeurs des centroïdes devient petit ou lorsque la fonction objective devient stable.

## 3 Algorithme de clustering possibiliste PCA

Yang et al ont proposé l'algorithme de clustering possibiliste PCA [15], pour palier le problème de la dépendance des appartenances des point aux clusters de l'algorithme FCM, due à la contrainte $\sum_{i=1}^{C} M_{ij} = 1$. Ainsi, le nouvel ensemble de contraintes qui a été défini est le suivant :

$$\forall i \in \{1..C\}, \forall j \in \{1..N\} \quad M_{ij \in [0,1]}$$

$$\forall j \in \{1..N\} \quad \begin{cases} 0 < \sum_{j=1}^{N} M_{ij} < N \\ \max_i M_{ij} > 0 \end{cases}$$

Les étapes de traitement de l'algorithme PCA sont les mêmes que celles de l'algorithme FCM, mais ses équations de calculs sont différentes. La fonction à minimiser $f_{PCA}$ est décrite par l'équation (4). Le degré d'appartenance $M_{ij}$ est décrit par une fonction exponentielle robuste aux bruits, elle est donnée par l'équation (5).

σ est l'écart type du jeu de données, il mesure le degré de séparation des données qui donne une idée de sa distribution.

$$f_{PCA} = \sum_{i=1}^{C}\sum_{j=1}^{N} M_{ij}^{m} d\left(o_j, g_i\right)^2 + \frac{\sigma}{m^2\sqrt{C}} \sum_{i=1}^{C}\sum_{j=1}^{N} M_{ij}^{m} \log M_{ij}^{m} - M_{ij}^{m} \tag{4}$$

$$M_{ij} = \exp\left(-\frac{m\sqrt{C}d\left(o_j, g_i\right)^2}{\sigma}\right) \quad , \quad g_i = \frac{\sum_{i=1}^{N}\left(M_{ij}\right)^m o_j}{\sum_{i=1}^{N}\left(M_{ij}\right)^m}$$
$$\sigma = \sum_{j=1}^{N} d(o_j, \bar{o})^2 \Big/ N \quad , \quad \bar{o} = \sum_{j=1}^{n} o_j \Big/ n \tag{5}$$

## 4 Algorithme d'optimisation par essaim de particules quantiques QPSO

L'optimisation par essaim de particules PSO, est une technique basée sur une population de recherche aléatoire. Elle a été introduite par Kennedy et Eberhart en 1995. Son domaine de prédilection depuis se jour est l'optimisation numérique hétérogène continue-discrète fortement non linéaire. À ce titre, elle est utilisé un peut partout dans le monde. Sa rapidité de convergence en fait aussi un outil privilégié en optimisation dynamique [16].

La méthode PSO s'inspire du comportement social des animaux évoluant en essaim, tels que les nuées d'oiseaux ou les essaims d'abeilles qui survolent l'espace de recherche pour exploiter les meilleures sources de nourriture (meilleures positions). Les membres (particules) de la population (essaim) changent de positions dans l'espace de recherche en s'appuyant sur leur propre expérience, pour trouver la meilleure position locale et sur celles de leurs voisines, pour trouver la meilleure position globale. Cette dernière constitue une solution optimale au problème posé.

Sun et al [9,10,11] ont introduit la théorie quantique dans le PSO et ont proposé l'algorithme d'optimisation par essaim de particules quantique QPSO. Ce dernier a non seulement moins de paramètres de contrôle que PSO [17] mais en plus, il est plus efficace et peut garantir théoriquement de trouver la solution optimale dans l'espace de recherche [18].

Avant ce papier, Il y a eu d'autres propositions d'algorithmes de clustering à base de QPSO pour remédier aux insuffisances des algorithmes classiques de clustering [12,19, 20, 21].

Nous définissons la notation adoptée dans ce document: la position de la i$^{ème}$ particule d'un essaim de taille *M*, est représenté par le vecteur *D* -dimensionnelle : $X_i = (X_{i1}, X_{i2}, ..., X_{iD})$. La particule se déplace avec l'équation (7) :

$$X_{id} = p_{lid} \pm \alpha.|mbest - X_{id}|.\ln(1/u) \quad (6)$$

$$p_{lid} = \varphi.P_{lid} + (1-\varphi).P_{gd} \quad (7)$$

Où *mbest* est la moyenne des meilleures positions locales des particules, calculée par l'équation (8), $P_{lid}$ est la meilleure position locale (personnelle) de la particule et $P_{gd}$ est la meilleure position globale, $p_{lid}$ est un point stochastique entre $P_{lid}$ et $P_{gd}$. $\varphi$ et $u$ sont deux nombres aléatoires distribués uniformément dans [0,1], $\propto$ est un paramètre de QPSO appelé coefficient de contraction expansion.

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_{lid} = \frac{1}{M}\sum_{i=1}^{M} P_{li1},..,\frac{1}{M}\sum_{i=1}^{M} P_{li2},..,\frac{1}{M}\sum_{i=1}^{M} P_{liD} \quad (8)$$

## 5 Algorithmes d'optimisation par essaim de particules quantiques hybrides floue et possibiliste pour le clustering

Dans QPSO pour le clustering, chaque particule représente un vecteur composé de *C* centroïdes. En d'autres termes, la particule $part_j$ est construite comme suit :

$$part_j = (g_{j1},..,g_{ji},..g_{jc}) \quad (9)$$

Où $g_{ji}$ désigne le i$^{ème}$ centroïde de la j$^{éme}$ particule. Chaque cluster possède un centroïde et chaque itération présente une solution qui donne un vecteur de centroïdes. Nous déterminons la position du vecteur $part_j$ pour chaque particule, la mettant à jour, puis nous modifiant la position des centroïdes basé sur les particules. Par conséquent, un essaim représente un nombre de solutions possibles pour le clustering.

La fonction objective d'une particule, appelée «Fitness» permet de mesurer la qualité d'une particule. Elle est définie par l'équation(10), où $C_{ij}$ est le nombre de points de données dans le cluster i de la particule j. La particule qui a la meilleure position globale est celle qui a la plus petite Fitness et la particule qui a la meilleure position locale est celle dont la Fitness actuelle est inférieure à la Fitness précédente.

$$f = \sum_{i=1}^{C}[\sum_{\forall o_{j\in C_j}} d(o_j, g_{ji})/|C_{ij}|] \ /C \quad (10)$$

Dans cette section, nous proposons d'améliorer les résultats de clustering de l'algorithme QPSO, en faisant l'équilibre entre sa convergence globale et locale. Dans cette optique, nous établissons deux nouvelles hybridations. La première est faite entre l'algorithme QPSO et l'algorithme FCM. Elle donne l'algorithme QPSO-F et la deuxième est faite entre l'algorithme QPSO et l'algorithme PCA. Elle donne l'algorithme QPSO-P.

QPSO-F est constitué principalement de deux étapes. La première est celle de l'initialisation de l'essaim. Une particule de l'essaim initial est générée par FCM et le reste de l'essaim est initialisé aléatoirement. La deuxième étape consiste à appliquer QPSO à l'essaim. De la même manière, QPSO-P passe aussi par deux étapes.

Premièrement, l'algorithme PCA est appliqué au jeu de données pour générer une particule initiale et les autres particules sont initialisées aléatoirement. Deuxièmement, QPSO est appliqué à l'essaim. Les algorithmes QPSO-F et QPSO-P sont décrits comme suit :

1. Initialiser les paramètre de QPSO et de FCM, notamment la taille de la population $M$ , $\propto$ et m.
2. Créer un essaim avec $M$ particules ($part_j, P_{lid}, p_{lid}, P_{gd}$, sont des matrices (N*C) et mbest est un vecteur D-dimensionnel).
3. Initialiser $part_j$ , $P_{lid}$ pour chaque particule et $P_{gd}$ pour l'essaim.
4. **Pour** t = 1 à MAXITER **faire**
   A. Exécuter soit l'algorithme FCM ou PCA, selon le choix des algorithmes QPSO-F et QPSO-P;
   B. Initialiser la première particule, soit avec les résultats de FCM, s'il s'agit de l'algorithme QPSO-F ou de de PCA, s'il s'agit de QPSO-P ;
   C. Calculer mbest de l'essaim avec l'équation (8);
   **(1)Pour** chaque particule *i* **faire**
   **(2) Pour** chaque vecteur de données $o_j$ **faire**
   (I) Calculez la distance euclidienne $d(o_j, g_{ji})$ de $o_j$ à tous les centroïdes $g_{ji}$;
   (II). Ajouter $o_j$ au cluster $C_{ij}$ tel que $d(o_j, g_{ji}) = min_{\forall j=1,2,\ldots,c}\{d(o_j, g_{ji})\}$;
   (III). Calculer la fitness des particules en utilisant l'équation (10)
   **Fin pour ;**
   **Fin pour ;**
   **(3)** Mettre à jour la meilleure position globale et les meilleures positions locales ;
   **(4)** Mettre à jour les centroides en utilisant les équations (6) et (7).
   **Fin pour ;**

Où MAXITER est le nombre maximum d'itérations.

## 6 Expériences

### 6.1 Jeux de données

Quatre jeux de données réels issus de l'UCI [21] et un jeu de données synthétique ont été utilisés.

(1). Dataset1 est un jeu de données synthétique bidimensionnel qui présente 600 points de données avec 4 classes de forme sphérique. La distribution normale permettant de le générer est la suivante :

*N([10,0],[2,2]), N([0,10],[2,2]), N([10,10],[2,2]), N([10,10],[2,2]).*

(2). Iris est un jeu de données sur la fleur Iris, qui contient 150 points de données, 3 classes et 4 attributs.

(3). Wisconsin breast cancer est un jeu de données qui contient 699 cas cliniques relatifs au cancer du sein répartis sur 2 groupes avec 9 attributs. L'objectif est de classer chaque cas en tumeurs bénignes ou malignes.

(4). Thyroid est un jeu de données qui contient des informations sur 215 glandes thyroïdiennes, chacune décrite par 5 attributs. L'objectif est de les classer en 2 groupes, l'hypothyroïdisme ou l'hyperthyroïdisme.
(5). Wine est un jeu de données qui contient 178 sortes de vin, décrite par 13 attributs. L'objectif est de les classer en 3 groupes.

### 6.2 Evaluation

Deux mesures d'évaluation ont été utilisées. La première est une mesure de la fonction objective [voir Equation(10)] basée sur la minimisation de la distance entre les points et leurs centroïdes. C'est une mesure de qualité interne qui ne prend pas en considération le partitionnement exacte des données, mais qui évalue l'optimisation globale de l'algorithme. Elle devrait être minimisée. La deuxième mesure est la mesure externe F-mesure. Elle compare la qualité du clustering trouvé $C=(C_1,C_2,..., C_c)$ aux classes connues et correctes $R = (R_1, R_2,..., R_{c'})$ pour un jeu de données. La F-mesure de $C$ par rapport à $R$ est défini par l'équation (11). Elle prend ses valeurs dans [0,1] et devrait être maximisée. Chaque cluster $C_j$ de C contient $N_j$ points de données et chaque cluster $R_i$ de R contient $N_i$ points de données, $N_{ij}$ est le nombre de points du cluster $R_i$ dans le cluster $C_j$ et $N$ est le nombre total de points du jeu de données.

$$Fmes(R_i, C_j) = \frac{(b^2+1).prec(R_i, C_j).rep(R_i, C_j)}{b^2.prec(R_i, C_j) + rep(R_i, C_j)}, \qquad (11)$$

$$prec(R_i, C_j) = \frac{N_{ij}}{N_j}, \qquad prec(R_i, C_j) = \frac{N_{ij}}{N_j}$$

### 6.3 Résultats

Afin de voir l'apport de nos algorithmes hybrides, nous les avons comparés avec l'algorithme QPSO. La Table1 montre les valeurs de la médiane et de l'interquartile de la F-mesure et la Table2 montre les valeurs de la médiane et de l'interquartile de la Fitness, obtenues à travers 50 exécutions de chaque algorithme, sur cinq jeux de données. L'interquartile indique l'intervalle de valeurs auxquelles les algorithmes convergent. La Figure 1 visualise la distribution des résultats présentés dans les Tables 1 et 2. Pour chaque algorithme, le nombre de générations est fixé à 100 et la taille de l'essaim de particules est fixée à 10. La valeur α varie linéairement de 1,0 à 0,5 au cours des générations. Le coefficient m de l'algorithme FCM et PCA est fixé à 2.

La Table1 et la Figure1 montrent qu'en terme de F-mesure et pour le jeu de données Dataset1, l'algorithme QPSO-F est meilleur que l'algorithme QPSO-P et QPSO et l'algorithme QPSO-P est meilleur que QPSO. Pour le jeu de données Iris, l'algorithme QPSO-P optimise la F-mesure mieux que les autres algorithmes et QPSO-F est meilleur que QPSO. Pour le jeu de données Wine, QPSO-F est meilleur que les autres algorithmes et QPSO-P est légèrement meilleure QPSO. Pour le jeu de données thyroid , l'algorithme QPSO-F a obtenu le plus grand score de la F-mesure,

tandis que les autres algorithmes ont obtenues le même score. Le jeu de données Wisconsin a obtenu les mêmes résultats que le jeu de données Wine.

La Table2 et la Figure1, montrent qu'en termes de fitness et pour le jeu de données Dataset1, QPSO-F a obtenue le meilleur résultat, suivie de près par QPSO-P et QPSO. Pour le jeu de données Iris, QPSO-P a le meilleur score de Fitness, suivi par QPSO-F et QPSO. Pour le jeu de données Wine, QPSO a le pire score de fitness et QPSO-F est meilleure que QPSO-P. Pour le jeu de données Thyroid, QPSO-F optimise la fonction objective mieux que les autres algorithmes mais QPSO-P optimise mieux que QPSO. Pour le jeu de données Wisconsin, QPSO-P a obtenue le meilleur score de fitness, suivi respectivement par QPSO-F et QPSO.

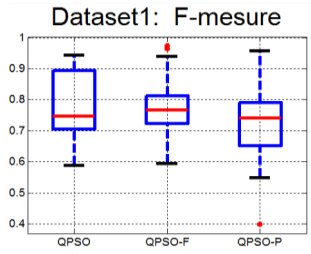
En résumé l'algorithme QPSO-F est en moyenne meilleur que l'algorithme QPSO-P en termes de F-mesure et de fitness et les deux algorithmes hybrides sont meilleurs que l'algorithme QPSO.

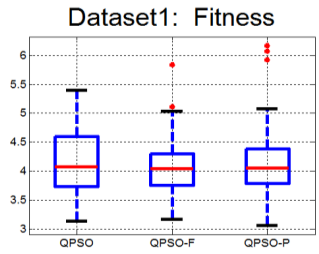
**Table 1.** Les valeurs de la médiane et de l'interquartile de la F-mesure, obtenues par (QPSO, QPSO-F, QPSO-P).

| Jeu de données | QPSO | QPSO-F | QPSO-P |
|---|---|---|---|
| **Dataset1** | 0.7238(0.0894) | 0.7544(0.0806) | 0.7410(0.1376) |
| **Iris** | 0.7187 (0.1216) | 0.7313 (0.1270) | 0.7497 (0.1189) |
| **Wine** | 0.6803 (0.1019) | 0.6996 (0.0593) | 06833 (0.0803) |
| **Thyroid** | 0.6246(0.0907) | 0.6620(0.1258) | 0.6273(0.1031) |
| **Wixconsin** | 0.9154(0.2472) | 0.9249(0.0907) | 0.9154(0.2282) |

**Table 2.** Les valeurs de la médiane et de l'interquartile de la fonction objective, obtenues par (QPSO, QPSO-F, QPSO-P).

| Jeu de données | QPSO | QPSO-F | QPSO-P |
|---|---|---|---|
| **Dataset1** | 4.1937(0.6222) | 4.0532(0.6040) | 4.0553(0.6047) |
| **Iris** | 0.9661(0.1866) | 0.9582(0.3085) | 0.9278(0.4478) |
| **Wine** | 128.7642 (27.9446) | 121.099 (15.0964) | 121.831(40.2099) |
| **Thyroid** | 11.6599(2.7967) | 11.0936(2.0610) | 11.3551(2.2801) |
| **Wixconsin** | 7.3069(1.8089) | 7.2569 (1.3725) | 7.1677 (1.8770) |



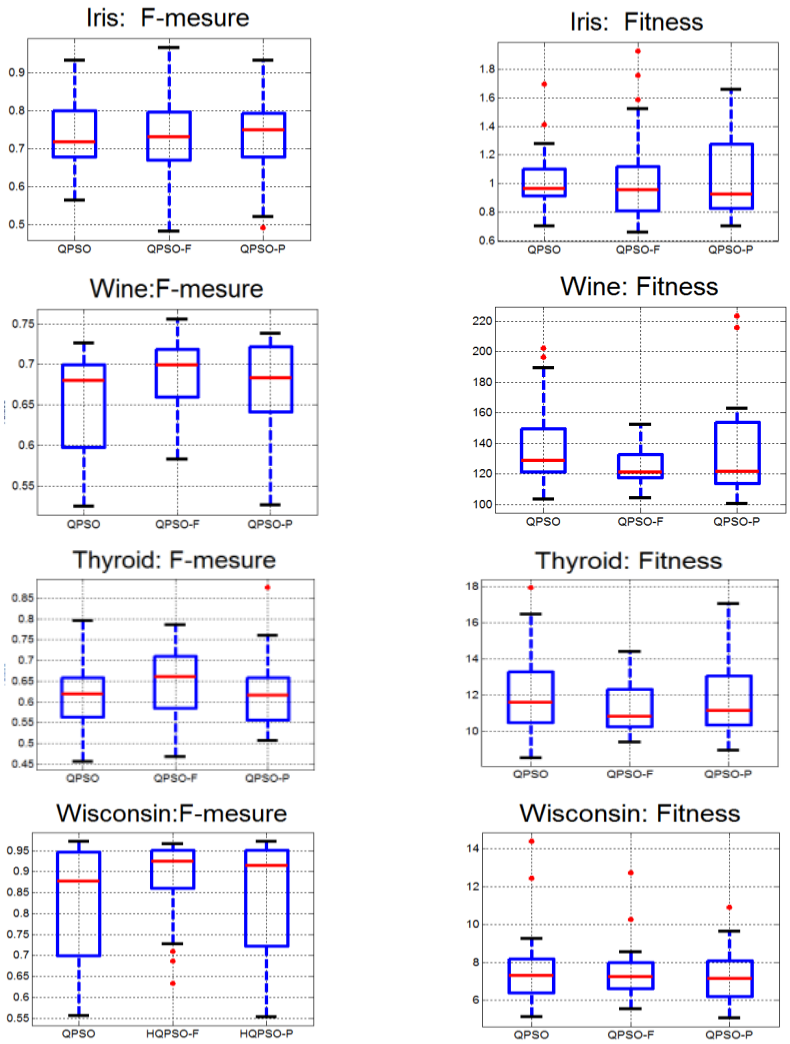



**Figure1.** Boxplots donnant la F-mesure et la Fitness réalisées au cours de 50 éxcutions des algorithmes QPSO, QPSO-F et QPSO-P sur 5 jeux de données.

## 7 Conclusion

L'algorithme QPSO souffre parfois de l'instabilité de ses résultats. Ceci est du à sa nature quantique. Pour palier ce problème, deux hybridations lui ont été apportées, une avec l'algorithme de clustering flou FCM et une autre avec l'algorithme de

clustering possibiliste PCA. Les algorithmes hybrides proposés prennent avantage des qualités de recherche locale des algorithmes FCM et PCA et de la qualité de recherche globale de l'algorithme QPSO. Les comparaisons effectuées et les résultats obtenus ont montré que nos deux hybridations améliorent la qualité des partitions trouvée, mieux que QPSO.

Notre futur travail consiste à appliquer le clustering hybride flou possibiliste à l'algorithme QPSO, sur des séries temporelles de grands volumes et plus particulièrement sur des tracés ECG, dans le but de détecter automatiquement des anomalies cardiaques.

## Références


1. Xu, R. and Wunsch, D. 'Survey of clustering algorithms', Journal of IEEE Transactions on Neural Networks, Vol. 16, No. 3, pp.645–678, 2005.
2. Maulik,U and Bandyopadhyay,S. 'Genetic algorithm-based clustering technique', Journal of the Pattern recognition, Vol.33, No. 9, pp.1455–1465, 2000.
3. Dorigo, M. 'Optimization, Learning and Natural Algorithms', PhD thesis, Department of Electronics, Italy, 1992.
4. Dorigo, M., Maniezzo, V and Colorni, A. 'The Ant System : Optimization by a Colony of Cooperating Agent', IEEE Transactions on Systems, Man, and Cybernetics –Part B, vol. 26, NO. 1, pp. 29–41, 1996.
5. Kennedy,J. and Eberhart,R. 'Particle Swarm Optimization', In Proceedings of IEEE International Conference on Neural Network, vol. 4, pp. 1942–1948, 1995.
6. Reynolds, C. W. 'Flocks, herds, and schools: A distributed behavior al model', in the proceeding of Computer Graphics (Siggraph), 21(4), pp.25-34, July 1987.
7. Heppner,F and Grenander, H. 'A stochastic non linear model for coordinated bird flocks', In The Ubiquity of Chaos, American Association for the Advancement of Science, pp.233-238, 1990.
8. Vesterstrom,J., Thomsen, R, 'A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems', proceeding of the IEEE Congress on evolutionary computation Vol. 3, pp.1980-1987. 2004.
9. Sun, J., Feng, B., Xu,W.-B, 'Particle Swarm Optimization with Particles Having Quantum Behavior'. Proceedings of Congress on Evolutionary Computation, Piscataway, NJ 325-331. 2004.
10. Sun, J., Xu, W.-B., Feng, B. 'A Global Search Strategy of Quantum-behaved Particle Swarm Optimization'. Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, Singapore, pp, 111-116. 2004.
11. Sun, J., Xu, W.-B., Feng, B. 'Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level'. Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ 3049-3054. 2005.
12. Sun, J., Xu, W. and Ye, B. (2006) 'Quantum-behaved particle swarm optimization clustering algorithm', in the Proceeding of the Advanced Data Mining and Applications, 14–16 August, Vol. 4093, pp.340–347, LNCS, Xi'an, China.
13. Van den Bergh, F. 'An Analysis of Particle Swarm Optimizers', Ph.D. Thesis, University of Pretoria, 2001.
14. Bezdek,J.C. 'Pattern Recognition with Fuzzy Objective Function Algorithms', PlenumPress, NewYork, 1981.
15. Yang, M.S. and Wu, K.L. 'Unsupervised possibilistic clustering', Journal of the Pattern Recognition, Vol. 39, No. 1, pp.5–21, 2006.
16. Clerc, M. 'L'optimisation par essaims particulaires' , France, ed. Hermes, 2005.
17. Abraham, A., Das, S. and Roy, S. 'Swarm intelligence algorithms for data clustering', Proceedings of Soft Computing for Knowledge Discovery and Data Mining, pp.279–313, Springer Verlag, Germany, 2008.
18. Xi, M., Sun, J., Xu, WB, 'An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position', Journal of the Applied Mathematics and Computation, Vol.205, No.2, pp.751–759, 2008.
19. Lu, K., Fang, K., Xie., G, 'A hybrid quantum-behaved particle swarm optimization algorithm for

clustering analysis', Proceedings of Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Vol.1, pp.21-25, 2008.
20. Chen,L., Wu,X., Gao,C, 'Semi-supervised Fuzzy Clustering Algorithm Based on QPSO', Journal of Information & Computational Science, Vol. 9, No.1, pp.93-101, 2012.
21. Wang, H., Yang, S., Xu, WB., Sun, J, 'Scalability of hybrid fuzzy c-means algorithm based on quantum-behaved PSO', Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 261-265, 2007.
22. Blake, C.L. and Merz, C.J, 'UCI repository of machine learning databases', 1998, available at: http://archive.ics.uci.edu/ml/datasets.html

# Classification with Support Vector Machines, New Quadratic Programming Algorithm

Ahmed Chikhaoui[1], Aek Mokhtari[2],

[1]IbnKhaldoune University Tiaret Algeria, [2]University Laghouat Algeria
ah_chikhaoui@yahoo.fr, mok_aek@yahoo.fr

**Abstract.** Support vector machines (SVM) are excellent tools for classification and regression. They seek the optimal separating hyperplane and maximal margin. The modeling results often lead to solving a quadratic programming problem. In this paper, we present a simple method to determine the hyperplane H that separates two classes of examples so that the distance between these two classes is maximal. This method is based on the geometric interpretation of the norm of a linear mapping. The result model of our algorithm modeling is a maximization of a concave quadratic program. This quadratic program is resolved by projection method. Example illustrates the method.

**Keywords:** Support vector machines, separating hyperplane, maximizing concave function, cosine, projection method.

## 1 Introduction

Learning to rank is an important problem in web page ranking information retrieval and other applications. Support Vector Machines (SVMs) are a powerful machine learning technique. Vapnik [7] showed how training a support vector machine for the pattern recognition problem leads to quadratic optimization problem (QP). The size of the optimization problem depends on the number of training examples. With 10000 training examples and more it becomes impossible to keep matrix data in memory. $SVM^{Light}$ uses the decomposition idea of Osuna and al. ([7]) and decompose the problem into a series of smaller tasks. This decomposition splits the initial problem in an inactive and en active part. These algorithms may need a long training time. To tackle this problem, T. Joachims [5], uses a method for selecting the working set, successive "shrinking" of the optimization problem and incremental updates of the gradient (Joachims [6]). Burges form AT&T [1], has even developed a QP solver specifically for training SVM.

In this paper we introduce new support vector machines method in order to define a decision surface separating two opposing classes of a training set of vectors. This method associates a distance parameter with each vector of the SVM's training set. The distance parameter is calculating as the shortest of distances from each vector

of one class to the opposite class. The method determines initial separating hyperplane and its maximum margin, where the margin is defined as the shortest distances of the hyperplane from the closest points of the two classes. The optimal vectors to preselect as potential support vectors are those closest to the decision hyperplane. The vectors with the smallest distance are then selected as pivots.
To determine the optimal hyperplane, we will use the well-known result:
if $f$ is a linear map from $R^n$ into $R$ defined by $f(x) = \langle a, x \rangle,\ a \in R^n$.
Then $\|a\| = d(0, H)$ where H is the hyperplane defined by $H = \{x \in R^n : \langle a, x \rangle = 1\}$.
The optimal hyperplane will be a boundary point of the set of feasible solutions which can be an extreme point.

## 2 Partition of examples $\widetilde{X}_+$ and $\widetilde{X}_-$

Suppose that separating hyperplane with maximum margin be written as $ax + b = 0$.

### 2.1 Formulation of the optimization problem

The inequalities $ax + b \geq 1$ and $ax + b \leq 1$ become $\frac{a}{2} x_+ + \frac{b}{2} \geq \frac{1}{2}$ and
$\frac{a}{2} x_- + \frac{b}{2} \leq -\frac{1}{2}$, and the hyperplane is $\frac{a}{2} x + \frac{b}{2} = \frac{1}{2}$; i.e. $ax + b = 0$.
As the couple $(a, b)$ is set to a multiplicative coefficient, the separating problem becomes then $\begin{cases} \inf \ \|a\|^2 \\ ax_+ + b \geq \frac{1}{2} \\ ax_- + b \leq -\frac{1}{2} \end{cases}$

Suppose that $\bar{x}_+$ is support vector, $a\bar{x}_+ + b = \frac{1}{2} \Rightarrow b = \frac{1}{2} - a\bar{x}_+$.

Then $ax_+ + b \geq \frac{1}{2} \Leftrightarrow ax_+ + \frac{1}{2} - a\bar{x}_+ \geq \frac{1}{2} \Leftrightarrow a(x_+ - \bar{x}_+) \geq 0$

$\Leftrightarrow a(\bar{x}_+ - x_+) \leq 0$ and $ax_- + b \leq -\frac{1}{2} \Leftrightarrow ax_- + \frac{1}{2} - a\bar{x}_+ \leq -\frac{1}{2} \Leftrightarrow a(x_- - \bar{x}_+) \leq -1$.

Then $\begin{cases} \inf \|a\|^2 \\ a(\bar{x}_+ - x_+) \leq 0 \\ a(x_- - \bar{x}_+) \leq -1 \end{cases} = \begin{cases} -Max\{-\|a\|^2\} \\ a(\bar{x}_+ - x_+) \leq 0 \\ a(x_- - \bar{x}_+) \leq -1 \end{cases}$

and consequently, the separating problem

becomes $(P)=\begin{cases} Max\left\{-\|a\|^2\right\} \\ \Omega=\begin{cases} a(\bar{x}_+ - x_+)\le 0, & x_+\in X_+ \\ a(x_- - \bar{x}_+)\le -1, & x_-\in X_- \end{cases} \end{cases}$ , $f(a)=\sum_{i=1}^{n}-a_i^2=-\|a\|^2$ is

concave, defined on closed bounded convex of $R^n$, then the local maximum is global, but $\frac{\partial f}{\partial a_i}(a)=0$ for all $i$, $\Rightarrow 2a_i=0,\quad a_i=a_i^*=0$.

The critical point $a^*=0\in R^n$ is not feasible solution, then the solution of the problem is the projection of $a^*=0$ on $\Omega$. This is a particular case of general optimization problem of concave quadratic programming,

where $\alpha_i=0,\beta_i=1,a^*=-\frac{\alpha_i}{2\beta_i}=0$.

This problem of maximizing concave quadratic function under linear constraints has solved by Chikhaoui and all. [3]. It is noted that $P_{a(\bar{x}_+ - x_+)}(0)=0$. This was made possible through the form $\begin{cases} \frac{a}{2}x_+ + \frac{b}{2}\ge \frac{1}{2} \\ \frac{a}{2}x_- + \frac{b}{2}\le -\frac{1}{2} \end{cases}$, this minimize the computing time.

**Increase in a margin**.

Let (H) the separating hyperplane of wide margin of equation $ax+b=0$.

We know that for all $x_+\in X_+,\ x_-\in X_-$, we have $\begin{cases} \frac{a}{2}x_+ + \frac{b}{2}\ge \frac{1}{2} \\ \frac{a}{2}x_- + \frac{b}{2}\le -\frac{1}{2} \end{cases}$

$$\Rightarrow \begin{cases} ax_+ + b \ge \frac{1}{2} \\ -ax_- - b \ge +\frac{1}{2} \end{cases} \Rightarrow (ax_+ + b)+(-ax_- - b)\ge \frac{1}{2}+\frac{1}{2};$$

from where $a(x_+ - x_-)\ge 1,\ \forall x_+\in X_+,\ \forall x_-\in X_-$.

By the inequality of Cauchy Schwartz, $\frac{1}{\|a\|}\le \|x_+ - x_-\|,\quad \forall x_+\in X_+, \forall x_-\in X_-$,

by passing to the lower bound, we obtain $\frac{1}{\|a\|}\le \inf_{x_+\in X_+,\, x_-\in X_-}\|x_+ - x_-\|$.

Whence important proposal:

Let $\widetilde{H}_+ = \{x \in R^n : ax - a\bar{x}_+ = 0\}, \widetilde{H}_- = \{x \in R^n : ax - a\bar{x}_- = 0\}.$

**Proposal:**

The width of the strip is increased by the constant $K = \inf_{x_+ \in X_+, x_- \in X_-} \|x_+ - x_-\|$, and this is best constant.

*Proof:* Indeed, suppose $x_+, x_-$ are two support vectors, i.e. $x_+ \in \widetilde{H}_+, x_- \in \widetilde{H}_-$;

$d(x_+, \widetilde{H}_+) = \frac{ax_+ + b}{\|a\|} = \frac{1}{2} \cdot \frac{1}{\|a\|}$, $d(x_-, \widetilde{H}_-) = \frac{|ax_- + b|}{\|a\|} = \frac{1}{2} \cdot \frac{1}{\|a\|}$

$\Rightarrow d(x_+, \widetilde{H}_+) + d(x_-, \widetilde{H}_-) = \frac{1}{\|a\|}$, and $\frac{1}{\|a\|} \le \inf_{x_+ \in X_+, x_- \in X_-} \|x_+ - x_-\| = K$.

This is the best ever because in cases where $\tilde{X}_+ = \{x_+\}$ and $\tilde{X}_- = \{x_-\}$, then $\frac{1}{\|a\|} = \inf_{x_+ \in X_+, x_- \in X_-} \|x_+ - x_-\| = K$; this completes the proof.

We see that the margin width does not exceed $\|x_+ - x_-\|$.

This leads us to consider the separating hyperplane with the widest possible margin $\widetilde{H}$.

### 2.2 Formulation of the optimization problem

Partition of $X_+$ and $X_-$

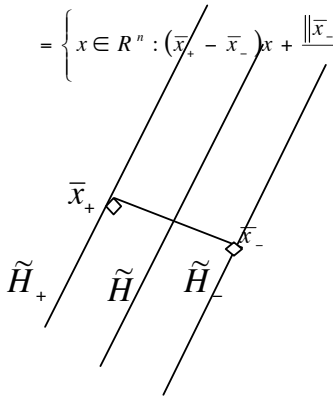
Let $\inf_{x_+ \in X_+, x_- \in X_-} \|x_+ - x_-\| = \|\bar{x}_+ - \bar{x}_-\|$, $\tilde{a} = \bar{x}_+ - \bar{x}_-$,

$\widetilde{H}_+ = \{x \in R^n : \tilde{a}x - \tilde{a}\bar{x}_+ = 0\}, \widetilde{H}_- = \{x \in R^n : \tilde{a}x - \tilde{a}\bar{x}_- = 0\},$

$$\widetilde{H} = \left\{ x \in R^n : \tilde{a}x - \left( \frac{\tilde{a}\bar{x}_+ + \tilde{a}\bar{x}_-}{2} \right) = 0 \right\}$$

$$= \left\{ x \in R^n : (\bar{x}_+ - \bar{x}_-)x - \frac{(\bar{x}_+ - \bar{x}_-)(\bar{x}_+ + \bar{x}_-)}{2} = 0 \right\}$$

$$= \left\{ x \in R^n : (\bar{x}_+ - \bar{x}_-)x + \frac{\|\bar{x}_-\|^2 - \|\bar{x}_+\|^2}{2} = 0 \right\}$$



The existence of optimal separating hyperplane $H$, and construction of $\tilde{H}$ define a partition of $X_+$ and a partition of $X_-$:

$$\tilde{X}_+ = \left\{ x \in X_+ : ax - a\bar{x}_+ < \frac{1}{2} \right\}, \qquad X_+ = \tilde{X}_+ \cup \left( X_+ / \tilde{X}_+ \right)$$

$$\tilde{X}_- = \left\{ x \in X_- : ax - a\bar{x}_- > -\frac{1}{2} \right\}, \qquad X_- = \tilde{X}_- \cup \left( X_- / \tilde{X}_- \right)$$

If the hyperplanes $H$ and $\tilde{H}$ separate the sets $X_+ / \tilde{X}_+$ and $X_- / \tilde{X}_-$ and $\tilde{H}$ is optimal. $\tilde{X}_+ = \tilde{X}_- = \phi$ then $H = \tilde{H}$. Stop.

## 2 Finding optimal separating hyperplane $H$:

### case $H \neq \tilde{H}$

The maximum margin separating $\tilde{X}_+$ and $\tilde{X}_-$ is greater than or equal to the maximum margin separating $X_+$ and $X_-$ because the optimal separating hyperplane $H$ separates $\tilde{X}_+$ and $\tilde{X}_-$.

Suppose that the maximum margin between $\tilde{X}_+$ and $\tilde{X}_-$ is strictly greater than that between $X_+$ and $X_-$, then this separating hyperplane is between $H$ and $\tilde{H}$ and hence it separates $X_+ / \tilde{X}_+$ and $X_- / \tilde{X}_-$. So this separating hyperplane separates $X_+$ and $X_-$ with a wider margin than strictly greater than that to $H$. Contradiction, because $H$ is optimal.

**Proposition:**

The optimal separating hyperplane of sets $\tilde{X}_+$ and $\tilde{X}_-$ is optimal separating hyperplane for sets $X_+$ and $X_-$.

*Proof:* Denote by $H^*$ optimal separating hyperplane of $\tilde{X}_+$ and $\tilde{X}_-$ whose normal is $a^*$. There positive $\lambda_1!$ and $\lambda_2!$ such that $a^* = \lambda_1 a + \lambda_2 \tilde{a}$.

In fact, $\begin{cases} \langle a, a^* \rangle = \|a\|.\|a^*\| \cos\alpha \\ \langle \tilde{a}, a^* \rangle = \|\tilde{a}\|.\|a^*\| \cos\beta \end{cases}$ $\langle a, \lambda_1 a + \lambda_2 \tilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha$

$\Rightarrow \lambda_1 \langle a, a \rangle + \lambda_2 \langle a, \tilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha \quad \Rightarrow \lambda_1 \|a\|^2 + \lambda_2 \langle a, \tilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha.$

As well

$\langle \tilde{a}, \lambda_1 a + \lambda_2 \tilde{a} \rangle = \|\tilde{a}\|.\|a^*\| \cos\beta \Rightarrow \lambda_1 \langle \tilde{a}, a \rangle + \lambda_2 \langle \tilde{a}, \tilde{a} \rangle = \|\tilde{a}\|.\|a^*\| \cos\beta$

$\Rightarrow \lambda_1 \langle \tilde{a}, a \rangle + \lambda_2 \|\tilde{a}\|^2 = \|\tilde{a}\| \|a^*\| \cos\beta.$

$$\begin{cases} \lambda_1 \|a\|^2 + \lambda_2 \langle a, \tilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha \\ \lambda_1 \langle \tilde{a}, a \rangle + \lambda_2 \|\tilde{a}\|^2 = \|\tilde{a}\|.\|a^*\| \cos\beta \end{cases}$$

Then

$$\Delta = \begin{vmatrix} \|a\|^2 & \langle a, \tilde{a} \rangle \\ \langle \tilde{a}, a \rangle & \|\tilde{a}\|^2 \end{vmatrix} = \|a\|^2 . \|\tilde{a}\|^2 - \langle a, \tilde{a} \rangle \langle \tilde{a}, a \rangle = \|a\|^2 \|\tilde{a}\|^2 - \|a\|^2 \|\tilde{a}\|^2 \cos^2\theta$$

Where $\theta$ is the angle formed between hyperplanes $H$ and $\tilde{H}$, as $H \neq \tilde{H}, \theta \neq 0$; then $\cos\theta \neq 1$, and $\Delta = \|a\|^2 \|\tilde{a}\|^2 (1 - \cos^2\theta)$. $\Delta > 0$

The system has a unique solution $\lambda_1$ and $\lambda_2$.

Then suppose that the margin of $H^*$ is strictly greater than that of $H$, as $H$ and $\tilde{H}$ separate $X_+ / \tilde{X}_+$ and $X_- / \tilde{X}_-$. i.e. $a$ and $\tilde{a}$ are solution of problem

$$\begin{cases} \underset{\Omega'}{Max} \left\{ \frac{1}{2} \|a\|^2 \right\} \\ \Omega' = \begin{cases} a(-x_+ + \bar{x}_+) \leq 0 \\ a(x_- - \bar{x}_+) \leq -1 \\ \bar{x}_+ \in X_+ / \tilde{X}_+ \\ \bar{x}_- \in X_- / \tilde{X}_- \end{cases} \end{cases}$$

$\Omega'$ is bounded below convex, then $a^* \in \Omega'$.

The separating hyperplane $H^*$ separates then $X_+$ and $X_-$ whose margin is strictly greater than that of $H$. Contradiction, $H$ is optimal by hypothesis.

**Consequence:**

To separate $X_+$ and $X_-$, just separate the sample $\tilde{X}_+$ and $\tilde{X}_-$. We then have a smaller number of constraints.

The maximum margin separating $\widetilde{X}_+$ and $\widetilde{X}_-$ is greater than or equal to the maximum margin separating $X_+$ and $X_-$ because the optimal separating hyperplane $H$ separates $\widetilde{X}_+$ and $\widetilde{X}_-$.

Suppose that the maximum margin between $\widetilde{X}_+$ and $\widetilde{X}_-$ is strictly greater than that between $X_+$ and $X_-$, then this separating hyperplane is between $H$ and $\widetilde{H}$ and hence it separates $X_+ / \widetilde{X}_+$ and $X_- / \widetilde{X}_-$. So this separating hyperplane separates $X_+$ and $X_-$ with a wider margin than strictly greater than that to $H$. Contradiction, because $H$ is optimal.

**Proposition:**

The optimal separating hyperplane of sets $\widetilde{X}_+$ and $\widetilde{X}_-$ is optimal separating hyperplane for sets $X_+$ and $X_-$.

Proof: Denote by $H^*$ optimal separating hyperplane of $\widetilde{X}_+$ and $\widetilde{X}_-$ whose normal is $a^*$. There positive $\lambda_1!$ and $\lambda_2!$ such that $a^* = \lambda_1 a + \lambda_2 \widetilde{a}$.

In fact, $\begin{cases} \langle a, a^* \rangle = \|a\|.\|a^*\| \cos\alpha \\ \langle \widetilde{a}, a^* \rangle = \|\widetilde{a}\|.\|a^*\| \cos\beta \end{cases}$ $\langle a, \lambda_1 a + \lambda_2 \widetilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha$

$\Rightarrow \lambda_1 \langle a, a \rangle + \lambda_2 \langle a, \widetilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha \quad \Rightarrow \lambda_1 \|a\|^2 + \lambda_2 \langle a, \widetilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha.$

As well

$\langle \widetilde{a}, \lambda_1 a + \lambda_2 \widetilde{a} \rangle = \|\widetilde{a}\|.\|a^*\| \cos\beta \qquad \Rightarrow \lambda_1 \langle \widetilde{a}, a \rangle + \lambda_2 \langle \widetilde{a}, \widetilde{a} \rangle = \|\widetilde{a}\|.\|a^*\| \cos\beta$

$\Rightarrow \lambda_1 \langle \widetilde{a}, a \rangle + \lambda_2 \|\widetilde{a}\|^2 = \|\widetilde{a}\| \|a^*\| \cos\beta.$ Then $\begin{cases} \lambda_1 \|a\|^2 + \lambda_2 \langle a, \widetilde{a} \rangle = \|a\|.\|a^*\| \cos\alpha \\ \lambda_1 \langle \widetilde{a}, a \rangle + \lambda_2 \|\widetilde{a}\|^2 = \|\widetilde{a}\|.\|a^*\| \cos\beta \end{cases}$

$$\Delta = \begin{vmatrix} \|a\|^2 & \langle a, \widetilde{a} \rangle \\ \langle \widetilde{a}, a \rangle & \|\widetilde{a}\|^2 \end{vmatrix} = \|a\|^2 . \|\widetilde{a}\|^2 - \langle a, \widetilde{a} \rangle \langle \widetilde{a}, a \rangle = \|a\|^2 \|\widetilde{a}\|^2 - \|a\|^2 \|\widetilde{a}\|^2 \cos^2\theta$$

Where $\theta$ is the angle formed between hyperplanes $H$ and $\widetilde{H}$, as $H \neq \widetilde{H}, \theta \neq 0$;

then $\cos\theta \neq 1$, and $\Delta = \|a\|^2 \|\tilde{a}\|^2 \left(1 - \cos^2\theta\right)$. $\Delta > 0$

The system has a unique solution $\lambda_1$ and $\lambda_2$.

Then suppose that the margin of $H^*$ is strictly greater than that of $H$, as $H$ and $\tilde{H}$ separate $X_+ / \tilde{X}_+$ and $X_- / \tilde{X}_-$. i.e. $a$ and $\tilde{a}$ are solution of problem

$$\begin{cases} \underset{\Omega'}{Max} \left\{ -\|a\|^2 \right\} \\ \Omega' = \begin{cases} a\left(-x_+ + \bar{x}_+\right) \leq 0 \\ a\left(x_- - \bar{x}_+\right) \leq -1 \\ \bar{x}_+ \in X_+ / \tilde{X}_+ \\ \bar{x}_- \in X_- / \tilde{X}_- \end{cases} \end{cases} \qquad \Omega' \text{ is bounded below convex, then } a^* \in \Omega'.$$

The separating hyperplane $H^*$ separates then $X_+$ and $X_-$ whose margin is strictly greater than that of $H$. Contradiction, $H$ is optimal by hypothesis.

**Consequence:**

To separate $X_+$ and $X_-$, just separate the sample $\tilde{X}_+$ and $\tilde{X}_-$.
We then have a smaller number of constraints.

**Example 1.** $X_+ = \left\{ (1,0), \left(2, \frac{3}{2}\right), (5,1), \right\}$ $X_- = \{(0,1), (-2,1), (-2,2), \}$

here, $\inf\limits_{\substack{x_+ \in X_+ \\ x_- \in X_-}} \left(\|x_+ - x_-\|\right) = \|(1,0) - (0,1)\| = \sqrt{2}$; $x_+ = (1,0)$, $x_- = (0,1)$

$\tilde{a} = \left(x_+ - x_-\right) = (1,-1)$ $\tilde{H}_+ : (1,-1)x - (1,-1)(0,0) = 0$

$\tilde{H}_+ : x_1 - x_2 - 1 = 0$ $\tilde{H}_- : x_1 - x_2 + 1 = 0$ $\tilde{H} : x_1 - x_2 = 0$.

$\left(2, \frac{3}{2}\right) \in \tilde{X}_+$ car $2 - \frac{3}{2} - 1 = -\frac{1}{2} < \frac{1}{2}$, $(5,1) \notin \tilde{X}_+$ car $5 - 1 - 1 = 3 > \frac{1}{2}$

$\Rightarrow \tilde{X}_+ = \left\{ (1,0), \left(2, \frac{3}{2}\right) \right\}$

$(-2,1) \notin \tilde{X}_-$ because $-2 - 1 + 1 = -2 < -\frac{1}{2}$,

$(-2,2) \notin \tilde{X}_-$ because $-2 - 2 + 1 = 3 < -\frac{1}{2}$ $\Rightarrow \tilde{X}_- = \{(0,1)\}$.

The constraint set $\Omega$, of problem becomes $\begin{cases} a\left((1,0)-\left(2,\frac{3}{2}\right)\right) \le 0 \\ a\left((0,1)-(1,0)\right) \le -1 \end{cases}$ the solution

is $\begin{cases} 2a_{11}+3a_{12}=0 \\ a_{11}+a_{12}+1=0 \end{cases} \Rightarrow \begin{cases} a_{11}=\frac{3}{5} \\ a_{12}=-\frac{2}{5} \end{cases}$. $(H)$ has the equation

$\left(\frac{3}{5},-\frac{2}{5}\right)(x_1,x_2)+\left(\frac{1}{2}-\left(\frac{3}{5},-\frac{2}{5}\right)\right)(1,0)=0$, $H:\ \frac{3}{5}x_1-\frac{2}{5}x_2-\frac{1}{10}=0$.

## 4 Projection Method ([2])

Consider the problem quadratic result of our modeling:

$$(P') = \begin{cases} \underset{\Omega}{Max}\left\{-\|a\|^2\right\} \\ \Omega = \left\{ a \in R^n,\ \begin{pmatrix} a(-x_+ + \bar{x}_+) \le 0 \\ a(x_- - \bar{x}_+) \le -1 \end{pmatrix},\ x_+ \in \widetilde{X}_+,\ x_- \in \widetilde{X}_- \right\} \end{cases}$$

Since the function $f(x)=\sum_{i=1}^{n}-a_i^2=-\|a\|^2$ is concave defined on a closed convex of $R^n$, then the local maximum of $f$ is global. But $\frac{\partial f}{\partial a_i}(a)=0 \Leftrightarrow a_i=0\ \ i=1,2,...,n$

Critical point $\left(a_i^*\right)_i=0\ for\ all\ i=1,2,...,n$ is not feasible solution of problem $(P')$.

Then the solution of problem $(P')$ is the projection of point $0\in R^n$ on $\Omega$.

This is a particular case of more general problem of quadratic optimization:

$f(a)=\sum_{i=1}^{n}\alpha_i a_i+\sum_{i=1}^{n}\beta_i a_i^2$ *with* $\alpha_i \in R^n, \beta_i<0$ *for all* $i$ ,under linear constraints.

In classification with SVM we have $a^*=\left(-\frac{\alpha_i}{2\beta_i}\right)_i$ with $\alpha_i=0,\ \forall i\ \beta_i=-1$. For more details see [3]. We recall that if a concave function $f$ defined on closed convex and that the critical point does not belong to convex, then the maximum of $f$ is reached on a boundary point of closed convex. See [3].

The projection of point $0\in R^n$ on the hyperplane $a(x_- - \bar{x}_+) = -1$ is given by

$$P_{a(x_- - \bar{x}_+)=-1}(0)=0-\frac{1}{\left\|(x_- - \bar{x}_+)\right\|^2}(x_- - \bar{x}_+).$$

**Example 2** $X_+ = \{(1,3),(1.5,4),(2,3),(3,3.5),(3,4)\}$

$X_- = \{(1,1.5),(1.5,1),(2,1),(2,2),(2.5,1.5)\}$.

$\inf_{\substack{x_+ \in X_+ \\ x_- \in X_-}} (\|x_+ - x_-\|) = \|(2,3) - (2,2)\| = 1$; $x_+ = (2,3)$, $x_- = (2,2)$

$\tilde{a} = (x_+ - x_-) = (2,3) - (2,2) = (0,1)$

$\tilde{H}_+ : (0,1)(x_1, x_2) + \left(\frac{1}{2} - (0,1)(2,3)\right) = \frac{1}{2} \quad \Rightarrow \tilde{H}_+ : x_2 - 3 = 0$

$\tilde{H}_- : (0,1)(x_1, x_2) + \left(-\frac{1}{2} - (0,1)(2,2)\right) = -\frac{1}{2} \quad \Rightarrow \tilde{H}_- : x_2 - 2 = 0$

$\tilde{H} : (0,1)(x_1, x_2) + \left(\frac{-3-2}{2}\right) = 0 \quad \Rightarrow \tilde{H} : x_2 - \frac{5}{2} = 0$

Construction of $\tilde{X}_+$ and $\tilde{X}_-$

$(1,3) \in \tilde{X}_+$ because $3 - 3 \le 0$ $\quad (1.5,4) \notin \tilde{X}_+$ because $4 - 3 \not\le 0$

$(2,3) \in \tilde{X}_+$ because $3 - 3 \le 0$ $\quad (3,3.5) \notin \tilde{X}_+$ because $3.5 - 3 \not\le 0$

$(3,4) \notin \tilde{X}_+$ because $4 - 3 \not\le 0$ $\quad (1,1.5) \notin \tilde{X}_-$ because $1.5 - 2 \not\le 0$

$(1.5,4) \notin \tilde{X}_+$ because $4 - 3 \not\le 0$ $\quad (2,3) \in \tilde{X}_+$ because $3 - 3 \le 0$

$(3,3.5) \notin \tilde{X}_+$ because $3.5 - 3 \not\le 0$ $\quad (3,4) \notin \tilde{X}_+$ because $4 - 3 \not\le 0$

Then $\tilde{X}_+ = \{(2,3),(1,3)\}$ $\quad \tilde{X}_- = \{(2,2)\}$

Constraints are: $\begin{cases} a((2,2) - (2,3)) \le -1 & \Leftrightarrow \quad a_2 \ge 1 \\ a((2,1) - (2,3)) \le -1 & \Leftrightarrow \quad a_2 \ge \frac{1}{2} \end{cases}$

$P_{a_2 = 1}(0) = 0 - \frac{-1}{1}(0,1) = (0,1) \quad \Rightarrow \|P_{a_2=1}(0)\| = 1$

$P_{a_2 = \frac{1}{2}}(0) = 0 - \frac{-\frac{1}{2}}{1}(0,1) = \left(0, \frac{1}{2}\right) \quad \Rightarrow \left\|P_{a_2 = \frac{1}{2}}(0)\right\| = \frac{1}{4}$

The solution is $a = (0,1)$, because $a = \left(0, \frac{1}{2}\right)$ is not feasible solution, and the optimal separating hyperplane is:

$H : (0,1)(x_1, x_2) + \left(\frac{1}{2} - (0,1)(2,3)\right) = \frac{1}{2} \quad \Rightarrow x_2 - \frac{5}{2} = 0$.

**Remark:** The feasible solution set of separating hyperplanes is the half-space $a_2 \ge 1$ and the projection of $0$ on this half-space is $(0,1)$.The set of feasible solutions do here no extreme point. It is interesting to study the nature of the set of separating hyperplanes.

**Example 3:** $X_+ = \left\{\left(2,2,\frac{1}{2}\right),(2,3,2),(3,3,1)\right\}$ $\quad X_- = \left\{\left(1,0,\frac{1}{2}\right),(1,-1,3)\right\}$

$$\inf_{\substack{x_+\in X_+\\ x_-\in X_-}} \|x_+ - x_-\| = \|(2,2,0.5)-(1,0,0.5)\| = \sqrt{5}$$

$$x_+ = (2,2,0.5),\ \ x_- = (1,0,0.5)\ \ \tilde{a} = (x_+ - x_-) = (1,2,0)$$

$$\tilde{H}\ :\ (1,2,0)(x_1, x_2, x_3) - \frac{3}{2} = 0 \qquad \tilde{X}_+ = \left\{\left(2,2,\frac{1}{2}\right)\right\}\ \ \tilde{X}_- = \left\{\left(1,0,\frac{1}{2}\right)\right\}.$$

Here, inside of band is empty. So $(\tilde{H}) = (H)$.

**Conclusion**

In this paper, we gave a geometric interpretation of the hyperplane that separates two classes linearly separable. In fact, the search algorithm to the optimum is nothing other than a particular case of general optimization problem:

$$\begin{cases} \sum_{i=1}^{n} \alpha_i x_i + \beta_i x_i^2, \quad with\ \ \alpha_i = 0,\ \beta_i = -1. \\ Ax \le b \end{cases}$$

The nature of solution (extreme point or not) provides to better track the support vectors.

## References


1. Burges C. J.C. A Tutorial on Support Vector machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2): 121-167 (1998).
2. Chikhaoui A, Djebbar B., Bellabacci A, Mokhtari A., Optimization of a quadratic function under its Canonical form *Asian Journal of Applied Sciences* 2(6):499-510 (2009)
3. Chikhaoui A, Djebbar B., and Mekki R., New Method for Finding an Optimal Solution to Quadratic Programming Problem, *Journal of Applied Sciences* 10(15):1627-1631-2010. ISSN 1812-5654 (2010).
4. Chikhaoui A, Bellabacci A, Mokhtari A., New Algorithm for maximizing of linearly constrained convex quadratic programming. IRECOS Vol 7, ISSN 1828-6003 (2012).
5. Joachims T., Making Large-Scale SVM Learning practical, Advances in kernel methods-support learning. Page 169-184. MIT Press (2003).
6. Joachims, T., Granka, L., Pan, Bing, Hembrooke, H.,Radlinski, F., and Gay, G. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Transactions on Information Systems (TOIS), 25(2), April (2007).
7. Osuna, E., Freund, R. and Girosi, F. An improved training algorithm for support vector machines. In Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing, Eds. J. Principe, L. Giles, N. Morgan (1997).
8. Vapnik, V.N., Statistical learning theory. John Wiley & Sons (1998).
9. N.M., Deris S. and Chin K.K., A comparaison of support vector machines training. Journal Teknologi, Vol 39, pp 45-56 (2003).
10. Wilson E., pages 276 – 285, Amelia Island, FL, (1997).

# Programmation logique et théorie des jeux

# Local and global symmetry in answer set programming

Belaïd Benhamou

Aix-Marseille Université
Laboratoire des Sciences de l'Information et des Systèmes (LSIS)
Domaine universitaire de Saint Jérôme
Avenue Escadrille Normandie Niemen 13397 MARSEILLE Cedex 20
belaid.benhamou@univ-amu.fr

**Abstract.** Many research works had been done in order to define a semantics for logic programs. The well know is the stable model semantics which selects for each program one of its canonical models. The stable models of a logic program are in a certain sens the minimal Herbrand models of its reducts. On the other hand, the notion of symmetry elimination had been widely used in constraint programming and shown to be useful to increase the efficiency of the associated solvers. However symmetry in non monotonic reasoning still not well studied in general. For instance Answer Set Programming (ASP) is a very known framework but only few recent works on symmetry breaking are known. Ignoring symmetry breaking in the answer set systems could make them doing redundant work and lose on their efficiency. Here we study the notion of local and global symmetry in the framework of answer set programming. We show how local symmetries of a logic program can be detected dynamically by means of the automorphisms of its graph representation. We also give some properties that allow to eliminate theses symmetries in SAT-based answer set solvers and show how to integrate this symmetry elimination in these methods in order to enhance their efficiency.

**Keywords:** symmetry, logic programming, stable model semantics, answer set programming, non-monotonic reasoning.

## Introduction

The work we propose here to investigate the notion of symmetry in Answer Set Programming (ASP). The (ASP) framework can be considered as a sub-framework of the default logic [38]. One of the main questions in ASP, is to define a semantics to logic programs. A logic program $\pi$ is a set of first order (formulas) rules of the form $r : concl(r) \leftarrow prem(r)$, where $prem(r)$ is the set of premises of the rule given as a conjunction of literals that could contain negations and negations as failure. The right part $concl(r)$ is the conclusion of the rule $r$ which is generally, a single atom, or in some cases a disjunction of atoms for logic programs with disjunctions. Some researchers considered $prem(r)$ as the $body$ of the rule $r$ and $concl(r)$ as its $head$ ($r : head(r) \leftarrow body(r)$). Each logic program $\pi$ is translated into its equivalent ground logic program $ground(\pi)$ by replacing each rule containing variables by all its ground instances, so that each literal in $ground(\pi)$ is ground. This technique is used to eliminate the variables even when the program contains function symbols and its Herbrand universe is infinite. Among the influential semantics that are given for these logic programs with negation and negation as failure are the completion semantics [17] and the

stable model for the answer set semantics [26]. It is well know that each answer set for a logic program is a model of its completion, but the converse, generally, is not true. Fages in his paper [22] showed that both semantics are equivalent for free loops logic programs that are called tight programs. A generalization of Fage's results to logic programs with eventual nested expressions in the bodies of their rules was given in [21]. On the other hand Fangzhen Lin and Yutin Zhao proposed in [32] to add what they called *loop formulas* to the completion of a logic program and showed that the set of models of the extended completion is identical to the program's answer sets even when the program is not tight.

On the other hand, symmetry is by definition a multidisciplinary concept. It appears in many fields ranging from mathematics to artificial intelligence, chemistry and physics. It reveals different forms and uses, even inside the same field. In general, it returns to a transformation, which leaves invariant (does not modify its fundamental structure and/or its properties) an object (a figure, a molecule, a physical system, a formula or a constraints network...). For instance, rotating a chessboard up to 180 degrees gives a board that is indistinguishable from the original one. Symmetry is a fundamental property that can be used to study these various objects, to finely analyze these complex systems or to reduce the computational complexity when dealing with combinatorial problems.

As far as we know, the principle of symmetry has been first introduced by Krishnamurthy [30] to improve resolution in propositional logic. Symmetries for Boolean constraints are studied in depth in [5, 6]. The authors showed how to detect them and proved that their exploitation is a real improvement for several automated deduction algorithms efficiency. Since that, many research works on symmetry appeared. For instance, the static approach used by James Crawford et al. in [28] for propositional logic theories consists in adding constraints expressing global symmetry of the problem. This technique has been improved in [1] and extended to 0-1 Integer Logic Programming in [2]. The notion of interchangeability in Constraint Satisfaction Problems (CSPs) is introduced in [23] and symmetry for CSPs is studied earlier in [37, 4].

Within the framework of the Artificial intelligence, an important paradigm is to take into account incomplete information (uncertain information, revisable information...). Contrary to the mode of reasoning formalized by a conventional or a classical logic, a result deducible from information (from a knowledge, or from beliefs) is not true but only probable in the sense that it can be invalidated further, and can be revised when adding new information.

To manage the problem of exceptions, several logical approaches in Artificial intelligence had been introduced. Many non-monotonic formalisms were presented since about thirty years. But, the notion of symmetry within this framework was not well studied. The principle of symmetry had been extended recently in [9, 10, 13] to non-monotonic reasoning. Symmetry had been defined and studied for three known non-monotonic logics: the preferential logic [15, 16, 14, 29], the X-logic [39] and the default logic [39]. More recently, global symmetry had been studied for the Answer Set Programming framework [19, 20]. In the same spirit as what it is done in [28, 1, 2] for the satisfiability problem, the authors showed how to break the global symmetry statically in a pre-processing phase by adding symmetry breaking predicates to the considered logic program. In this work, we investigate dynamic local symmetry detection and elimination and global symmetry exploitation in the framework of answer set programming. Local symmetry is the symmetry that we can discover at each node of the search tree

during search. Global symmetry is the particular local symmetry corresponding to the root of the search tree (the symmetry of the initial problem). Almost all of the known works on symmetry are on global symmetry. Only few works on local symmetry [5–7, 11] are known in the literature. Local symmetry breaking remains a big challenge. As far as we know, local symmetry is not studied yet in ASP.

The rest of the paper is structured as follows: in Section 2, we give some necessary background on answer set programming and permutations. We study the notion of symmetry for answer set programming in Section 3. In Section 4 we show how local symmetry can be detected by means of graph automorphism. We show how both global and local symmetry can be eliminated in Section 5. Section 6 shows how local symmetry elimination is implemented in a SAT-based answer set programming Method. Section 7 investigtes the first implementation and experiments. We give a conclusion in Section 8.

## Background

We summarize in this section some background on both the answer set programming framework and permutation theory.

### Answer set programming

A ground general logic program $\pi$ is a set of rules of the form $r : L_0 \leftarrow L_1, L_2, \ldots, L_m, notL_{m+1}, \ldots, notL_n$, $(0 \leq m < n)$ where $L_i$ $(0 \leq i \leq n)$ are atoms, and $not$ is the symbol expressing negation as failure. The positive body of $r$ is denoted by $body^+(r) = \{L_1, L_2, \ldots, L_m\}$, and the negative body by $body^-(r) = \{L_{m+1}, \ldots, L_n\}$. The word *general* expresses the fact that the rules are more general than Horn clauses, since they contain negations as failure. The sub-rule $r^+ : L_0 \leftarrow L_1, L_2, \ldots, L_m$ expresses the positive projection of the rule $r$. Intuitively the rule $r$ means "*If we can prove all of* $\{L_1, L_2, \ldots, L_m\}$ *and we can not prove all of* $\{L_{m+1}, \ldots, L_n\}$*, then we deduce* $L_0$". Given a set of atoms $A$, we say that a rule $r$ is applicable (active) in $A$ if $body^+(r) \subseteq A$ and $body^-(r) \cap A = \emptyset$. The reduct of the program $\pi$ with respect to a given set $A$ of atoms is the positive program $\pi^A$ where we delete each rule containing an expression $notL_i$ in its negative body such that $L_i \in A$ and where we delete the other expressions $notL_i$ in the bodies of the other rules. More precisely, $\pi^A = \{r^+ / r \in \pi, body^-(r) \cap A = \emptyset\}$. The most known semantics for general logic programs is the one of stable models defined in [26] which could be seen as an improvement of the negation as failure of Prolog. A set of atoms $A$ is a stable model (an answer set) of $\pi$ if and only if $A$ is identical to the minimal Herbrand model of $\pi^A$ which is called its canonical model (denoted by $CM(\pi^A)$). That is, if only if $A = CM(\pi^A)$. The stable model semantics is based on the world closed assumption, an atom that is not in the stable model $A$ is considered to be false.

An extended logic program is a set of rules as the ones given for general programs which could contain classical negation. The atoms $L_i$ could appear in both positive and negative parity. In other words, the atoms $L_i$ become literals. A logic program is said to be disjunctive when at least one of its rules contains a disjunction of literals in its head part.

**Permutations**

Let $\Omega = \{1, 2, \ldots, N\}$ for some integer $N$, where each integer might represent a propositional variable. A permutation of $\Omega$ is a bijective mapping $\sigma$ from $\Omega$ to $\Omega$ that is usually represented as a product of cycles of permutations. We denote by $Perm(\Omega)$ the set of all permutations of $\Omega$ and $\circ$ the composition of the permutation of $Perm(\Omega)$. The pair $(Perm(\Omega), \circ)$ forms the permutation group of $\Omega$. That is, $\circ$ is closed and associative, the inverse of a permutation is a permutation and the identity permutation is a neutral element. A pair $(T, \circ)$ forms a sub-group of $(S, \circ)$ iff $T$ is a subset of $S$ and forms a group under the operation $\circ$.

The orbit $\omega^{Perm(\Omega)}$ of an element $\omega$ of $\Omega$ on which the group $Perm(\Omega)$ acts is $\omega^{Perm(\Omega)}$=$\{\omega^\sigma : \omega^\sigma = \sigma(\omega), \sigma \in Perm(\Omega)\}$.

A generating set of the group $Perm(\Omega)$ is a subset $Gen$ of $Perm(\Omega)$ such that each element of $Perm(\Omega)$ can be written as a composition of elements of $Gen$. We write $Perm(\Omega)$=$< Gen >$. An element of $Gen$ is called a generator. The orbit of $\omega \in \Omega$ can be computed by using only the set of generators $Gen$.

## Symmetry in logic programs

Since Krishnamurthy's [30] symmetry definition and the one given in [5, 6] in propositional logic, several other definitions are given in the CP community.

We will define in the following both semantic and syntactic symmetries in answer set programming and show their relationship. In the sequel $\pi$ could be the logic program or its completion, the symmetry definitions and properties remains valuable.

**Definition 1.** *(semantic symmetry) Let $\pi$ be a logic program and $L_\pi$ its complete [1] set of literals. A semantic symmetry of $\pi$ is a permutation $\sigma$ defined on $L_\pi$ such that $\pi$ and $\sigma(\pi)$ have the same answer sets.*

In other words a semantic symmetry of a formula is a literal permutation that conserves the set of answer sets of the logic program $\pi$. We adapt in the following the definition of syntactic symmetry given in [5, 6] for satisfiability to logic programs.

**Definition 2.** *(syntactic symmetry) Let $\pi$ be a logic program and $L_\pi$ its complete set of literals. A syntactic symmetry of $\pi$ is a permutation $\sigma$ defined on $L_\pi$ such that the following conditions hold:*

1. $\forall \ell \in L_\pi, \sigma(\neg\ell) = \neg\sigma(\ell)$,
2. $\forall \ell \in L_\pi, \sigma(not\ell) = not\sigma(\ell)$,
3. $\sigma(\pi) = \pi$

In other words, a syntactical symmetry of a logic program is a literal permutation that leaves the logic program invariant. If we denote by $Perm(L_\pi)$ the group of permutations of $L_\pi$ and by $Sym(L_\pi) \subset Perm(L_\pi)$ the subset of permutations of $L_\pi$ that are the syntactic symmetries of $\pi$, then $Sym(L_\pi)$ is trivially a sub-group of $Perm(L_\pi)$.

**Theorem 1.** *Each syntactical symmetry of a logic program $\pi$ is a semantic symmetry of $\pi$.*

[1] The set of literals containing each literal of $\pi$ and its negation as failure

*Proof.* It is trivial to see that a syntactic symmetry of a logic program $\pi$ is always a semantic symmetry of $\pi$. Indeed, if $\sigma$ is a syntactic symmetry of $\pi$, then $\sigma(\pi) = \pi$, thus it results that $\pi$ and $\sigma(\pi)$ have the same set of answer sets.

*Example 1.* consider the logic program $\pi = \{d \leftarrow; c \leftarrow; b \leftarrow c, nota; a \leftarrow d, notb\}$ and the permutation $\sigma$=$(a, b)(c, d)(nota, notb)$ defined on the complete set $L_\pi$ of literals occurring in $\pi$. We can see that $\sigma$ is a syntactic symmetry of $\pi$ ($\sigma(\pi)$=$\pi$).

In the sequel we deal only with syntactic symmetry, we say only symmetry to designate syntactic symmetry.

**Definition 3.** *Two literals $\ell$ and $\ell'$ of a logic $\pi$ are symmetrical if there exists a symmetry $\sigma$ of $\pi$ such that $\sigma(\ell) = \ell'$.*

**Definition 4.** *Let $\pi$ be a logic program, the orbit of a literal $\ell \in L_\pi$ on which the group of symmetries $Sym(L_\pi)$ acts is $\ell^{Sym(L_\pi)}$=$\{\sigma(\ell) : \sigma \in Sym(L_\pi)\}$*

*Remark 1.* All the literals in the orbit of a literal $\ell$ are symmetrical two by two.

*Example 2.* In Example 1, the orbit of the literal $a$ is $a^{Sym(L_\pi)}$= $\{a, b\}$, the orbit of the literal $c$ is $c^{Sym(L_\pi)}$= $\{c, d\}$ and the one of the literal $nota$ is $nota^{Sym(L_\pi)}$= $\{nota, notb\}$ All the literals of a same orbit are all symmetrical.

If *I* is an answer of $\pi$ and $\sigma$ a syntactic symmetry, we can get another answer set of $\pi$ by applying $\sigma$ on the literals which appear in *I*. Formally we get the following property.

**Proposition 1.** *I is an answer set of $\pi$ iff $\sigma(I)$ is an answer set of $\pi$ .*

*Proof.* Suppose that $I$ is an answer set of $\pi$, then $I$ is a minimal Herbrand model of the reduct $\pi^I$. It follows that $\sigma(I)$ is a minimal model of $\sigma(\pi)^{\sigma(I)}$. We can then deduce that $\sigma(I)$ is a minimal model of $\pi^{\sigma(I)}$ since $\pi$ is invariant under $\sigma$. We conclude that $\sigma(I)$ is an answer set of $\pi$. The converse can be shown by considering the converse permutation of $\sigma$.

For instance, in Example 1 there are two symmetrical answer sets for the logic program $\pi$. The fist one is $I = \{d, c, a\}$ and the second is $\sigma(I) = \{d, c, b\}$. These are what we call symmetrical answer sets of $\pi$. A symmetry $\sigma$ transforms each answer set into an answer set and each no-good (not an answer set) into a no-good.

**Theorem 2.** *Let $\ell$ and $\ell'$ be two literals of $\pi$ that are in the same orbit with respect to the symmetry group $Sym(L_\pi)$, then $\ell$ participates in an answer set of $\pi$ iff $\ell'$ participates in an answer set of $\pi$.*

*Proof.* If $\ell$ is in the same orbit as $\ell'$ then it is symmetrical with $\ell'$ in $\pi$. Thus, there exists a symmetry $\sigma$ of $\pi$ such that $\sigma(\ell) = \ell'$. If *I* is an answer set of $\pi$ then $\sigma(I)$ is also an answer set of $\sigma(\pi) = \pi$, besides if $\ell \in I$ then $\ell' \in \sigma(I)$ which is also an answer set of $\pi$. For the converse, consider $\ell = \sigma^{-1}(\ell')$, and make a similar proof.

**Corollary 1.** *Let $\ell$ be a literal of $\pi$, if $\ell$ does not participate in any answer set of $\pi$, then each literal $\ell' \in orbit^\ell = \ell^{Sym(L_\pi)}$ does not participate in any answer set of $\pi$.*

*Proof.* The proof is a direct consequence of Theorem 2

Corollary 1 expresses an important property that we will use to break local symmetry at each node of the search tree of a SAT-based answer set procedure. That is, if a no-good is detected after assigning the value True to the current literal $\ell$, then we compute the orbit of $\ell$ and assign the value false to each literal in it, since by symmetry the value true will not lead to any answer set of the logic program.

For instance, consider the program of Example 1, and the partial interpretation $I = \{a, b, c\}$ where $c$ is the current literal under assignation. It is trivial that $I$ is not a stable model of the program. By corollary 1, we can deduce that the set $I' = \{a, b, d\}$ is not a stable model of the program too. Indeed, $I'$ is obtained by replacing the current literal $c$ in $I$ by its symmetrical literal $d$. $I$ is a no-good and by symmetry (without duplication of effort) we infer that $I'$ is a no-good.

## Symmetry detection

The most known technique to detect syntactic symmetries for CNF formulas in satisfiability is the one consisting in reducing the considered formula into a graph [28, 3, 2] whose the automorphism group is identical to the symmetry group of the original formula. We adapt the same approach here to detect the syntactic symmetries of a program $\pi$. That is, we represent the CNF formula corresponding to the completion ($Compl(\pi)$) of the logic program $\pi$ by a graph $G_\pi$ that we use to compute the symmetry group of $\pi$ by means of its automorphism group. When this graph is built, we use a graph automorphism tool like Saucy [3], Nauty [33], AUTOM [36] or the one described in [34] to compute its automorphism group which gives the symmetry group of $\pi$. Following the technique used in [28, 3, 2] to represent CNF formulas, we summarize bellow the construction of the graph which represent the logic program $\pi$. Here we focus on the case of general logic programs, but the technique could be generalized to other classes of logic programs like extended logic programs or disjunctive logic programs. Given a general logic program $\pi$ we define its associated colored oriented graph $G_\pi(V, E)$ as follows:

- Each positive literal $\ell_i$ of $Compl(\pi)$ is represented by a vertex $\ell_i \in V$ of the color 1 in $G_\pi$. The negative literal $not\ell_i$ (negation as failure) associated with $\ell_i$ is represented by a vertex $not\ell_i$ of color 2 in $G_\pi$. These two literal vertices are connected by an edge of $E$ in the graph $G_\pi$.
- Each clause $c_i$ of $Compl(\pi)$ is represented by a vertex $c_i \in V$ (a clause vertex) of color 3 in $G_\pi$. An edge connects this vertex $c_i$ to each vertex representing one of its literals.

Three colors are sufficient to make the graph for general logic programs. For the extended logic programs we should add a forth color for classical negative literals $\neg\ell_i$ and draw an edge between $\neg\ell_i$ and its associated positive literal $\ell_i$.

This is different from the recent approach which uses a body-atom graph [19]. Since our study is oriented to SAT-based ASP using the completion, we do not need to manage an oriented body-atom graph.

An important property of the graph $G_\pi$ is that it preserves the syntactic group of symmetries of $Compl(\pi)$. That is, the syntactic symmetry group of the logic program

$Compl(\pi)$ is identical to the automorphism group of its graph representation $G_\pi$, thus we could use a graph automorphism system like Saucy on $G_\pi$ to detect the syntactic symmetry group of $\pi$. The graph automorphism system returns a set of generators $Gen$ of the symmetry group from which we can deduce each symmetry of $Compl(\pi)$.

## Symmetry elimination

There are two ways to break symmetry. The first one is to deal with the global symmetry which is present in the formulation of the given problem. Global symmetry can be eliminated in a static way in a preprocessing phase of an answer set solver. A method for global symmetry elimination is introduced in [19] for the Clasp ASP system [25]. The second one is the local symmetry that could appear in the sub-problems corresponding to the different nodes of the search tree of an answer set solver. Global symmetry can be considered as the local symmetry corresponding to the root of the search tree.

Local symmetries have to be detected and eliminated dynamically at each node of the search tree. Dynamic symmetry detection in satisfiability had been studied in [5, 6] where a local syntactic symmetry search method had been given. However, this method is not complete, it detects only one symmetry $\sigma$ at each node of the search tree when failing in the assignment of the current literal $\ell$. As an alternative to this incomplete symmetry search method, a complete method which uses the tool Saucy [3] had been introduced in [12] to detect and break all the syntactic local symmetries of a constraint satisfaction problem (CSP) [35] during search. More recently local symmetry had been detected and eliminated dynamically in a SAT solver [8].

Consider the completion $Compl(\pi)$ of a logic program $\pi$, and a partial assignment $I$ of SAT-based answer set solver applied to $Compl(\pi)$. Suppose that $\ell$ is the current literal under assignment. The assignment $I$ simplifies $Compl(\pi)$ into a sub-completion $Compl(\pi)_I$ which defines a state in the search space corresponding to the current node $n_I$ of the search tree. The main idea is to maintain dynamically the graph $G_{\pi_I}$ of the sub-completion $Compl(\pi)_I$ corresponding to the current node $n_I$, then color the graph $G_{\pi_I}$ as shown in the previous section and compute its automorphism group $Aut(\pi_I)$. The sub-completion $Compl(\pi)_I$ can be viewed as the remaining sub-problem corresponding to the unsolved part. By applying an automorphism tool on this colored graph we can get the generator set $Gen$ of the symmetry sub-group existing between literals from which we can compute the orbit of the current literal $\ell$ that we will use to make the symmetry cut.

Now, we use Corollary 1 to break the local symmetry and then prune search spaces of tree search answer set methods. Indeed, if the assignment of the current literal $\ell$ defined at a given node $n_I$ of the search tree is shown to be a failure, then by symmetry, the assignment of each literal in the orbit of $\ell$ will result in a failure too. Therefore, the negated literal of each literal in the orbit of $\ell$ has to be assigned in the partial assignment $I$. Therefore we prune in the search tree, the sub-space which corresponds to the assignment of the literals of the orbit of $\ell$. That is what we call the symmetry cut.

## Local symmetry exploitation in SAT-based ASP solvers

The solver ASSAT [32] has some drawbacks: it can compute only one answer set and the formula could blow-up in space. Taking into account these disadvantages of ASSAT and the fact that each answer set of a program $\pi$ is a model of its completion

$Compl(\pi)$, Guinchiglia et al. in [27] do not use SAT solvers as black boxes, but implemented a method which is based on the DLL [18] procedure and where they include a function which checks if a generated model is an answer set or not. This method had been implemented in the Cmodels-2 system [31] and has the following advantages: it performs the search on $Compl(\pi)$ without introducing any extra variable except those used by the clause transformation of $Compl(\pi)$, deals with tight and not tight programs, and works in a polynomial space. Global symmetry breaking do not need any extra-implementation, a SAT-based answer set solver is used as a black box on the completion of the logic program and the generated symmetry breaking predicates. More recently the ASP solvers like the conflict-driven Clasp solver [25] include some materials of modern SAT solvers such as: conflict analysis via the First UIP scheme, no-good recording and deletion, backjumping, restarts, conflict-driven decision heuristics, unit propagation via watched literals, equivalence reasoning and resolution-based preprocessing [24] have shown dramatic improvements in their efficiency and compete with the best SAT solvers.

We give in the following a DLL-based answer set method in which we implement dynamic local symmetry breaking. We used as a baseline method the DLL-based answer set procedure given in [27] to show the implementation of local symmetry eliminations (local symmetry cuts).

If $I$ is an inconsistent partial interpretation in which the assignment of the value $true$ to the current literal $\ell$ is shown to be a no-good, then, all the literals in the orbit of $\ell$ computed by using the group $Sym(\pi_I)$ returned by the graph automorphism tool are symmetrical to $\ell$. Thus, we assign the value false to each literal in $\ell^{Sym(L_\pi)}$ since the value true is shown to be contradictory, and then we prune the sub-space which corresponds to the value true assignments. The other case of symmetry cut happen when the assignment $I$ is shown to be a model of $Compl(\pi)$, but is not an answer set of $\pi$. In this case, the algorithm makes a backtracking on the last choice literal $\ell$ in $I$, then according to corollary 1 assigns the value false to each literal in the orbit $\ell^{Sym(L_\pi)}$ since the value true does not lead to an answer set of $\pi$. If $\Gamma = Compl(\pi)$, then the resulting procedure called DLLAnswerSet, is given in Figure 1.

The function AutomorphismTool($\pi_I$) is a call to the automorphism tool which return the set of generators in the variable $GEN$. The function $orbit(\ell, Gen)$ is elementary, it computes the orbit (the symmetrical literals) of the literal $\ell$ from the set of generators $Gen$ returned by AutomorphismTool($\pi_I$). The set $\Gamma_\ell$ is the set of clauses obtained from $\Gamma$ by removing the clauses to which $\ell$ belongs, and by removing $\neg\ell$ from the other clauses of $\Gamma$.

The function $AnswerSetCheck(I, \pi)$ is also elementary:

- it computes the set $A = I \cap \{head(r) : r \in \pi\}$ of positive literals (atoms) in $I$ and returns $True$ if $A$ is an answer set or $\pi$, and
- return $False$, otherwise.

## Experiments

Now we shall investigate the performances of our search techniques by experimental analysis. We choose for this first implementation the graph coloring problem to show the local symmetry behavior on answer sets search vs the global symmetry. Graph coloring problem is expressed naturally as a set of rules of a general problem. For more

**Procedure** DLLAnswerSet($\Gamma, I$);
**begin**
  **if** $\Gamma = \emptyset$ **then return** $AnswerSetCheck(I, \pi)$
    **else return** False
  **else if** $\Gamma$ contains the empty clause, **then return** False
    **else**
      **if** there exists a mono-literal or a monotone literal $\ell$ **then**
        **return** DLLAnswerSet($\Gamma_\ell, I \cup \{\ell\}$)
      **begin**
        Choose an unsigned literal $\ell$ of $\Gamma$
        $Gen$=AutomorphismTool($\Gamma_I$);
        $\ell^{Sym(L_{\pi_I})}$=orbit($\ell$,$Gen$)=$\{\ell_1, \ell_2, ..., \ell_n\}$;
        **return** DLLAnswerSet($\Gamma_\ell, I \cup \{\ell\}$) **or**
        DLLAnswerSet($\Gamma_{\neg\ell \wedge \neg\ell_1 \wedge \neg\ell_2 \wedge \ldots \wedge \neg\ell_n}$,
                $I \cup \{\neg\ell, \neg\ell_1, \ldots, \neg\ell_n\}$)
      **end**
**end**

**Fig. 1.** The DLL-based answer set procedure with local symmetry elimination.

details, the reader can refer to the Lparse user's manual given on line on the Cmodels site (http://www.cs.utexas.edu/ tag/cmodels/). We expect that symmetry breaking will be more profitable in real-life applications. Here, we tested and compared two methods:

1. **Global-sym:** search with global symmetry breaking. This method uses in pre-processing phase the program SHATTER [1, 2] that detects and eliminates the global symmetries of the considered instance by adding to it symmetry breaking clauses, then apply the SAT based answer set solver defined in [27] to the resulting instance. The CPU time of *Global-sym* includes the time that SHATTER spends to compute the global symmetry.
2. **Local-sym:** search with local symmetry breaking. This method implements in the SAT based answer set solver defined in [27] the dynamic local symmetry detection and elimination strategy described in this work. The resulting method is depicted in figure 1 (the DLLAnswerSet procedure). The CPU time of *Local-sym* includes local symmetry search time.

on some random graph coloring instances. The common baseline answer set search method for both previous methods is the one given in [27]. The complexity indicators are the number of nodes of the search tree and the CPU time. Both the time needed for computing local symmetry and global symmetry are added to the total CPU time of search. The source codes are written in C and compiled on a Pentium 4, 2.8 GHZ and 1 Gb of RAM.

### The results on the graph coloring instances

Random graph coloring problems are generated with respect to the following parameters: (1) $n$ : the number of vertices, (2) $Colors$: the number of colors and (3) $d$: the

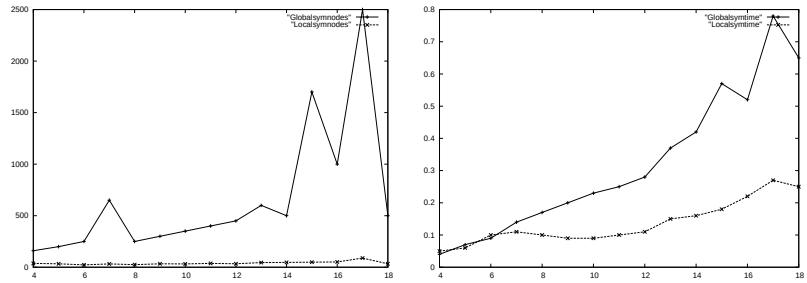



**Fig. 2.** Node and Time curves of the two symmetry methods on random graph coloring where $n = 30$ and $d = 0.5$

density which is a number between 0 and 1 expressed by the ratio : the number of constraints (the number of edges in the graph) to the number of all possible constraints (the number of possible edges in the graph). For each test corresponding to some fixed values of the parameters $n$, $Colors$ and $d$, a sample of 100 instances are randomly generated and the measures (CPU time, nodes) are taken on the average.

We reported in Figure 2 the practical results of the methods: *Global-sym*, and *Local-sym*, on the random graph coloring problem where the number of variables is $n = 30$ and where the density is ($d = 0.5$). The curves give the number of nodes respectively the cpu time with respect to the number of colors for each search method.

We can see on the node curves (the curves at the left) that *Local-sym* detects and eliminates more symmetries than the *Global-sym* method and *Global-sym* is not stable for graph coloring. From the CPU time curves (the curves at the right), we can see that *Local-sym* is in average faster than *Global-sym* even that Saucy is run at each node. Local symmetry elimination is profitable for solving random graph coloring instances and outperforms dramatically global symmetry breaking on these problems.

These are just our first results, our implementation and experiments are still in progress, we need to experiment much more and greater size instances than the ones presented here in order to confirm the advantage of symmetry breaking.

## Conclusion

We studied in this work the notion of symmetry for the answer set semantics defined for logic programs. We showed how a logic program or its completion is represented by a colored graph that can be used to compute symmetries. The syntactic symmetry group of the completion is identical to the automorphism group of the corresponding graph. Graph automorphism tools like SAUCY can be naturally used on the obtained graph to detect the syntactic symmetries. Global symmetry is eliminated statically by adding in preprocessing phase the well known lex order symmetry breaking predicates to the program completion and applying as a black box a SAT-based answer set solver on this resulting encoding. We showed how local symmetry can be detected and eliminated dynamically during search. That is, the symmetries of each sub-problem defined at a given node of the search tree and which is derived from the initial problem by considering the

partial assignment corresponding to that node. We showed that graph automorphism tools can be adapted to compute this local symmetry by maintaining dynamically the graph of the sub-program or the sub-completion defined at each node of the search tree. We proved some properties that allow us to make symmetry cuts that prune the search tree of a SAT-based answer set method. Finally, we showed how to implement these local symmetry cuts in a DLL-based answer set method.

The proposed local symmetry detection method is implemented and exploited in the tree search method $DLLAnswerSet$ to improve its efficiency. The first experimental results confirmed that local symmetry breaking is profitable for answer set solving and improves global symmetry breaking on the considered problems.

As a future work, we are looking to experiment other problems and combine both the global symmetry and local symmetry eliminations in a DLL-based answer set solver and compare the performances of the obtained methods to existing methods.

We studied the notion of symmetry for the general logic programs, but the study could naturally be generalized for extended logic programs, disjunctive logic programs or other extensions. This is another important point that we are looking to investigate in future.

## References


1. Aloul, F.A., A.Ramani, Markov, I.L., Sakallak, K.A.: Solving difficult sat instances in the presence of symmetry. In DAC pp. 1117–1137 (2003)
2. Aloul, F.A., Ramani, A., Markov, I.L., Sakallak, K.A.: Symmetry breaking for pseudo-boolean satisfiabilty. In ASPDAC'04 pp. 884–887 (2004)
3. Aloul, F.A., Ramani, A., Markov, I.L., Sakallah, K.A.: Solving difficult SAT instances in the presence of symmetry. In: Proceedings of the 39th Design Automation Conference (DAC 2002). pp. 731–736. ACM Press (2002)
4. Benhamou, B.: Study of symmetry in constraint satisfaction problems. PPCP'94
5. Benhamou, B., Sais, L.: Theoretical study of symmetries in propositional calculus and application. In CADE'11 (1992)
6. Benhamou, B., Sais, L.: Tractability through symmetries in propositional calculus. In JAR 12, 89–102 (1994)
7. Benhamou, B., Nabhani, T., Ostrowski, R., Saidi, M.R.: Dynamic symmetry breaking in the satisfiability problem. In: Proceedings of the 16th international conference on Logic for programming, artificial intelligence, and reasoning (LPAR-16) (25 april 2010), dakar, Senegal
8. Benhamou, B., Nabhani, T., Ostrowski, R., Saïdi, M.R.: Dynamic symmetry breaking in the satisfiability problem. In: Proceedings of the 16$^{th}$ international conference on Logic for Programming, Artificial intelligence, and Reasoning. LPAR-16, Dakar, Senegal (April 25 - may 1, 2010)
9. Benhamou, B., Nabhani, T., Siegel, P.: Reasoning by symmetry in non-monotonic inference. In: International Conference on Machine and Web Intelligence (ICMWI'10). pp. 264 – 269 (03 october 2010), algiers, Algeria
10. Benhamou, B., Nabhani, T., Siegel, P.: Reasoning by symmetry in non-monotonic logics. In: 13th international workshop on Non-Monotonic Reasoning, NMR'10 (14 may 2010)
11. Benhamou, B., Saidi, M.R.: Local symmetry breaking during search in csps. In: Springer (ed.) The 13th International Conference on Principles and Practice of Constraint Programming (CP 2007). LNCS, vol. 4741, pp. 195–209. Providence, USA (september 2007)
12. Benhamou, B., Saïdi, M.R.: Local symmetry breaking during search in csps. In: In CP. pp. 195–209 (2007)
13. Benhamou, B., Siegel, P.: Symmetry and non-monotonic inference. In: Symcon'08. Sydney, Australia (september 2008)

14. Besnard, P., Siegel, P.: The preferential-models approach in nonmonotonic logics - in nonstandard logic for automated reasoning. In: Academic Press. pp. 137–156. ed. P. Smets (1988)
15. Bossu, G., Siegel, P.: Nonmonotonic reasoning and databases. In: Advances in Data Base Theory. pp. 239–284 (1982)
16. Bossu, G., Siegel, P.: Saturation, nonmonotonic reasoning and the closed-world assumption. Artif. Intell. 25(1), 13–63 (1985)
17. Clark, K.: Negation as failure. In: Logic and data bases. pp. 293–322. In Herve Gallaire and Jack Minker, editors (1978)
18. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem proving. JACM, 5(7)
19. Drescher, C., Tifrea, O., Walsh, T.: Symmetry-breaking answer set solving. AI Commun. 24(2), 177–194 (2011)
20. Drescher, C., Tifrea, O., Walsh, T.: Symmetry-breaking answer set solving (2011)
21. Erdem, E., Lifschitz, V.: Tight logic programs. Thoery and Practice of Logic Programming 3, 499–518 (2003)
22. Fages, F.: Consistency ofclark's completion and existence of stable models. Thoery and Practice of Logic Programming 1, 51–60 (1994)
23. Freuder, E.: Eliminating interchangeable values in constraints satisfaction problems. AAAI-91 pp. 227–233 (1991)
24. Gebser, M., Kaufmann, B., Neumann, A., Schaub, T.: Advanced preprocessing for answer set solving. In: Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence. pp. 15–19. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008)
25. Gebser, M., Kaufmann, B., Neumann, A., Schaub, T.: clasp: A conflict-driven answer set solver. In: In LPNMR'07. pp. 260–265. Springer (2007)
26. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Logic programming: Fifth Int'l Conf. and Symp. pp. 1070–1080. In Robert Kawalski and Kenneth Bowen editors (1988)
27. Giunchiglia, E., Lierler, Y., Maratea, M.: Sat-based answer set programming. In: 19th National Conference on Artificial Intelligence,July 25-29, San Jose, California. AAAI (2004)
28. J.Crawford, Ginsberg, M.L., Luck, E., Roy, A.: Symmetry-breaking predicates for search problems. In: KR'96, pp. 148–159 (1996)
29. Kraus, S., Lehmann, D.J., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44(1-2), 167–207 (1990), citeseer.ist.psu.edu/kraus90nonmonotonic.html
30. Krishnamurty, B.: Short proofs for tricky formulas. Acta Inf. (22), 253–275 (1985)
31. Lierler, Y., Maratea, M.: Cmodels-2: Sat-based answer set solver enhanced to non-tight programs. In: Procedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR). pp. 346–350 (2004)
32. Lin, F., Y.Zhao: Assat: Computing answer sets of a logic program by sat solver. In: AAAI-02 (2002)
33. McKay, B.: Practical graph isomorphism. In: Congr. Numer. 30. pp. 45–87 (1981)
34. Mears, C., de la Banda, M.G., Wallace, M.: On implementing symmetry detection. In: SymCon'06. pp. 1–8 (2006)
35. Montanari, U.: Networks of constraints : Fundamental properties and applications to picture processing. Information Science 7 pp. 95–132 (1974)
36. Puget, J.F.: Automatic detection of variable and value symmetries. In: CP'05. pp. 474–488
37. Puget, J.F.: On the satisfiability of symmetrical constrained satisfaction problems. In: In J. Kamorowski and Z. W. Ras,editors, Proceedings of ISMIS'93, LNAI 689 (1993)
38. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
39. Siegel, P., Forget, L., Risch, V.: Preferential logics are x-logics. Journal of Logic and Computation 11(1), 71–83 (2001)

# A compression algorithm for solving efficiently Non Binary CSP using Generalized Hypertree Decomposition

Habbas Zineb[1], Amroun Kamal[2]and Singer Daniel[3]

[1] LCOMS, University of Lorraine, France,
zineb.habbas@univ-lorraine.fr
[2] University of Bejaia, Department of Computer Science, Algeria,
k_amroun25@yahoo.fr
[3] LCOMS, University of Lorraine, France,
daniel.singer@univ-lorraine.fr

Many real world problems can be formulated as Constraint Satisfaction problems (CSPs). Although, CSPs are NP-Hard in general, it has been theoretically proved that instances of non binary CSPs may be solved efficiently if they are representable as Generalized Hypertree Decomposition (GHD) of small hypertree width. Unfortunately, the main algorithm proposed for solving non binary CSPs based on GHD is not efficient in practice. This algorithm requires a large space memory to save all the intermediate results coming from the join operation. To cope with this main drawback, we use a compression algorithm based on decision trees to compress non binary CSP's with large extensional constraints. In order to exploit this compact representation, we mainly need to extend the join and semi join operations to "compressed join" and "compressed semi join" operations respectively. We prove the correctness of our approach and validate it on multiple benchmarks. Experimental results show that compressed relations and compressed operations improve significantly the practical performance of the basic algorithm for solving large non binary $CSPs$.

*Keywords* : Constraint satisfaction problems, Generalized Hypertree Decomposition, Compression

## 1 Introduction

Many important real world problems can be modelled as Constraint Satisfaction Problems (CSPs). CSPs are combinatorial in nature, so an efficient algorithm is unlikely to exist. Indeed, the usual method that guarantees to find a solution is enumerative and has an exponential time complexity in the worst case. In order to provide better theoretical complexity bounds, structural decomposition methods have received considerable interest these last decades. Numerous decomposition methods have been successfully used to characterize some tractable classes [4,11,7,1]. A structural decomposition method transforms a given instance I represented by a graph (respectively hypergraph) into a "solution equivalent" one I' having a tree-like structure (respectively hypertree-like structure). The best known complexity bounds are given by the tree-width for tree-decomposition methods and by the hypertree-width for hypertree decomposition. From theoretical viewpoint, methods based on hypertree-decomposition are better than those based on tree-decomposition [7]. In addition, theoretical time complexities for tree-decomposition as well as for hypertree-decomposition methods can really outperform classical resolution methods. However, except the recent work on BTD method [14], the practical gain of these decomposition methods has not been proved from practical viewpoint. The memory space consumption problem is the main drawback which prevents the practical efficiency of structural decomposition methods.
This work is a new attempt to exploit efficiently (generalized) hypertree decomposition for solving non binary CSPs. A basic algorithm is presented in [3] for processing generalized Hypertree decomposition with Acyclic Solving ($AS$). Its main primitive operation is the join of the relations to solve the subproblems associated with

the nodes. This operation is well known to be memory space explosive in the Data Base area. Jégou et al. [14] have introduced the method called BTD (Backtracking with Tree Decomposition) which is an enumerative search algorithm guided by an order induced by a tree decomposition. To overcome the memory explosion of the algorithm $AS$, we propose in this paper a new solving algorithm using a compressed representation of relations. For this, we use the method proposed in [16] based on Decision Trees that supposes a compact representation of relations in order to capture an exponential number of tuples in polynomial space. In consequence, the classical join and semi-join operations of the acyclic solving algorithm have to be adapted to work with compressed tuples leading to the Compressed Acyclic Solving (CAS) algorithm.
The rest of the paper is organized as follows. Section 2 is devoted to some technical background. In section 3, we present the Compressed Acyclic Solving algorithm (CAS) where we introduce the compressed join and the compressed semi-join operations. In Section 4, we compare CAS with the BTD method for some benchmarks presented in CPAI08 competition. Finally, we give our conclusions in section 5.

## 2 Background

### 2.1 Preliminaries

The notion of Constraint Satisfaction Problem (CSP) has been formally defined by U. Montanari [18]. A CSP instance is defined as a triple $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ where : $\mathcal{X} = \{X_1, ..., X_n\}$ is a finite set of n variables. $\mathcal{D} = \{D_1, ..., D_n\}$ is a set of finite domains. Each variable $X_i$ takes its value in its domain $D_i$. $\mathcal{C} = \{C_1, ..., C_m\}$ is a set of m constraints. Each constraint $C_i$ is a pair $(Scope(C_i), Rel(C_i))$ where $Scope(C_i) \subseteq \mathcal{X}$, is a subset of variables, called *the scope* of $C_i$ and $Rel(C_i) \subseteq \prod_{X_k \in Scope(C_i)} D_k$ (subset of the cartesian product ) is *the relation* of $C_i$, that specifies the legal combinations of values for the variables in $Scope(C_i)$. The size of $Scope(C_i)$ is called the arity of the constraint $C_i$. Constraints of arity 2 are called binary. A CSP where all the constraints are binary is called *binary CSP*. Constraints of arity greater than 2 are called non binary or *n-ary*. A CSP with at least one n-ary constraint is called *n-ary CSP*.
A tuple $\tau \in Rel(C_i)$ is an ordered list of values $(a_1, a_2, ..., a_k)$ where $k = |Scope(C_i)|$ such that $a_j \in D_j, j = 1, ..., k$. A *solution* to a CSP is an assignment of values to all the variables such that all the constraints are simultaneously satisfied.

In the sequel, we use indifferently the terms relation or constraint for a constraint.

**Definition 1.** ***(Hypergraph, primal graph and dual graph)*** *A hypergraph is a pair $\mathcal{H} =< V, E >$ consisting of a set of vertices $V$ and a set of hyperedges E where each hyperedge $h \in E$ is a subset of vertices of $V$. A* primal graph *of a hypergraph $\mathcal{H}$ is a graph obtained as follows : the primal graph owns the same vertices as $\mathcal{H}$ and two vertices $v_i$ and $v_j$ are connected by an edge in the primal graph if they appear in a common hyperedge of $\mathcal{H}$. A hypergraph $\mathcal{H} =< V, E >$ can be represented by a* dual graph *$\mathcal{H}^{Dual}$. The vertices of $\mathcal{H}^{Dual}$ are the hyperedges of $\mathcal{H}$. Two vertices are connected in the dual graph if their corresponding hyperedges share a vertex in $\mathcal{H}$.*

The hypergraph of a CSP instance $P =< X, D, C >$ is the hypergraph $\mathcal{H} =< V, E >$ where the set of vertices $V$ is the set of variables $X$ and the set of hyperedges $E$ corresponds to the set of constraints $C$.

**Definition 2.** ***(Hypertree)*** *Let $\mathcal{H} =< V, E >$ a hypergraph. A hypertree for $\mathcal{H}$ is a triple $< T, \chi, \lambda >$ where $T = (N, E)$ is a rooted tree, and $\chi$ and $\lambda$ are labelling functions which associate each vertex $p \in N$ with two sets $\chi(p) \subseteq vars(\mathcal{H})$ and $\lambda(p) \subseteq edges(\mathcal{H})$. If $T' = (N', E')$ is a subtree of T, we define*

$\chi(T') = \bigcup_{v \in N'} \chi(v)$. *We denote the set of vertices N of T by* vertices(T) *and the root of T by* root(T). $T_p$ *denotes the subtree of T rooted at the node p and* $Parent(p)$ *is the parent node of p.*

**Definition 3.** ***(Generalized Hypertree Decomposition)*** *A Generalized Hypertree Decomposition (GHD) [7] of a hypergraph* $\mathcal{H} =< V, E >$*, is a hypertree* $HD =< T, \chi, \lambda >$ *which satisfies the following conditions:*

1. *For each edge* $h \in E$ *, there exists* $p \in vertices(T)$ *such that* $var(h) \subseteq \chi(p)$*. We say that p covers h.*
2. *For each variable* $v \in V$*, the set* $\{p \in vertices(T) | v \in \chi(p)\}$ *induces a connected subtree of T.*
3. *For each vertex* $p \in vertices(T), \chi(p) \subseteq var(\lambda(p))$.

The *width* of a GHD $< T, \chi, \lambda >$ is $max_{p \in vertices(T)} |\lambda(p)|$. The generalized hypertree width $(ghw(\mathcal{H}))$ of hypergraph is the minimum width over all its generalized hypertree decompositions.

A hyperedge $h$ of a hypergraph $\mathcal{H} =< V, E >$ is *strongly covered* in $HD =< T, \chi, \lambda >$ if there exists $p \in vertices(T)$ such that the vertices of $h$ are contained in $\chi(p)$ and $h \in \lambda(p)$. A generalized hypertree decomposition $< T, \chi, \lambda >$ of a hypergraph $\mathcal{H}$ is *complete* if every hyperedge $h$ of $\mathcal{H}$ is strongly covered in $HD =< T, \chi, \lambda >$.

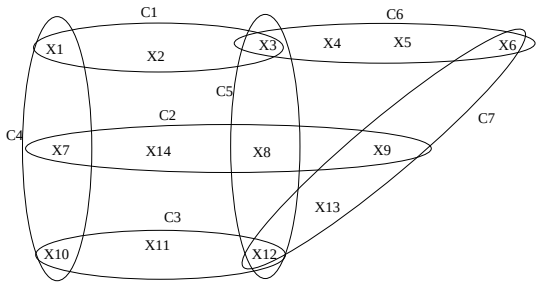
*Remark 1.* Recognizing hypergraphs with a $ghw$ at most 3 is NP complete [9]. If we have the third condition of the GHD as equality (ie. $\forall p \in vertices(T), \chi(p) = var(\lambda(p))$), we have a decomposition which has the same computational properties as a *query decomposition* [8].

*Example 1.* Let $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a CSP instance defined as follows :
$\mathcal{X} = \{X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14\}$,
$\mathcal{C} = \{C1, C2, C3, C4, C5, C6, C7\}$ where:
Scope(C1)={X1,X2,X3}, Scope(C2)={X7,X8,X14}, Scope(C3)={X10,X11,X12}, Scope(C4)={X1,X7,X10}, Scope(C5)={X3,X8,X12}, Scope(C6)={X3,X4,X5,X6} and Scope(C7)={X6,X9,X12, X13}.

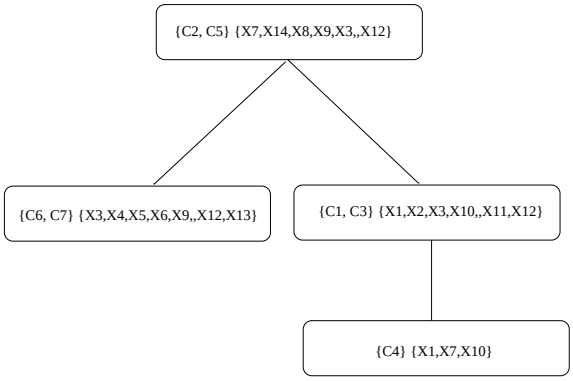
The figure 1(a) shows the constraints hypergraph of the example 1 and the figure 1(b) gives one of its generalized hypertree decompositions.

To compute a generalized hypertree decomposition of a constraints hypergraph, many heuristics are proposed in the literature: Korimort [17] proposed one heuristic based on the vertices connectivity of the given hypergraph. Samer [20] explored the use of branch decomposition for constructing hypertree decompositions. Dermaku et al. [5] proposed the following heuristics: BE (Bucket Elimination), DBE (Dual Bucket Elimination) and Hypergraph partitioning. Musliu and Schafhauser [19] explored the use of genetic algorithms for generalized hypertree decompositions, and many others heuristics and meta-heuristics.
Hereafter, we briefly recall the successful BE heuristic for generating generalized hypertree decompositions.

*The Bucket Elimination (BE) heuristic* [2] was successfully used to compute a tree decomposition of a given graph (or a primal graph of a hypergraph). BE has been extended by Dermaku et al. [5] to compute a generalized hypertree decomposition. The simple idea behind this extension derives from the fact that a generalized hypertree decomposition satisfies the properties of a tree decomposition. Consequently, for computing a generalized hypertree decomposition, BE proceeds as follows. First, it builds a tree decomposition using (basic) BE. Then, it creates the $\lambda - labels$ for each node of this tree in order to satisfy the third condition of generalized hypertree decomposition according to the *Definition* 3.




a− Constraint hypergraph of the example 1




b−Generalized Hypertree Decomposition of the example 1

**Fig. 1.** A constraints hypergraph of the example1 and its generalized hypertree decomposition. ghw= 2.

### 2.2 The Acyclic Solving Algorithm : *AS*

The Acyclic Solving (AS) algorithm introduced in [3] for processing GHD is formally described by Algorithm 1. It considers as input a complete Generalized Hypertree Decomposition associated with a given CSP instance. For solving the initial CSP represented by a complete GHD, all the problems at nodes are first solved independently by using *join* (noted $\bowtie$) and *projection* operations. Then *semi-join* operations (noted $\ltimes$) are computed in a bottom up way making the whole CSP resolution backtrack free.

## 3 Compressed Acyclic Solving algorithm: *CAS*

In this section, to cope with the main drawback of Acyclic Solving algorithm, we present its memory optimized version based on a compression strategy. Notice that similar ideas have been used in [15,6,16] in other contexts. This optimized version is referred as *CAS* for Compressed Acyclic Solving algorithm in the sequel.

**Algorithm 1** Acyclic Solving Algorithm

**Input** A Complete GHD $< T, \chi, \lambda >$ associated with a given CSP.
**Output** a solution $\mathcal{A}$ of the CSP if it is satisfiable
$\sigma = \{n_1, n_2, \ldots, n_m\}$ a node ordering with $n_1$ the root of the hypertree and each node precedes all its sons in $\sigma$
**for** each $p$ a node in $\sigma$ **do**
    $R_p = (\bowtie_{C_j \in \lambda(p)} R_j)[\chi(p)]$
**end for**
**for** i = m to 2 **do**
    Let $p_j$ the father of $p_i$ in $\sigma$
    $R_{p_j} = R_{p_j} \ltimes R_{p_i}$
**end for**
**for** i = 1 to m **do**
    Build a solution $\mathcal{A}$ by choosing a tuple $R_i$ compatible with all the previous assignments
**end for**
Return $\mathcal{A}$

### 3.1 Useful notations and definitions

Before to present formally our approach we have to introduce some notations and to define some relevant definitions.

#### 3.1.1 Notations

Let $R_1$ and $R_2$ be two relations associated to the constraints $C_1$ and $C_2$ with $S_1$ and $S_2$ their respective scopes. Let X be a subset of $S_1$.

- $\pi_X(R_1)$ denotes the projection of $R_1$ on $X$.
- $\bowtie (R_1, R_2)$ denotes the join of relations $R_1$ and $R_2$
- $\bowtie^{cr} (comp(R_1), comp(R_2))$ denotes the join of compressed relations $comp(R_1)$ and $comp(R_2)$
- $\bowtie^{ct} (ct_1, ct_2)$ denotes the join of two compressed tuples $ct_1$ and $ct_2$.
- $c\_value(x, ct)$ denotes the set of values allowed by the variable $x$ in the compressed tuple $ct$
- $tuples(ct)$ denotes the tuples accepted in the compressed tuple $ct$.
- ct[X] means the projection of the compressed tuple ct on the variables in X.
- $D_X(ct)$ denotes the cartesian product of the c_values of the variables in X in the compressed tuple $ct$.

#### 3.1.2 Definitions

**Definition 4. (Compressed tuple, compressed relation)** *Let $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a CSP , consider $C_1$ a constraint in $\mathcal{C}$ with $Scope(C_1) = \{V_1, ..., V_k\}$ and $Rel(C_1) = R$ is the relation associated with it. A compressed tuple (ctuple for short) of $C_1$ is a tuple $(D'_1, ..., D'_k)$ where $D'_i \subseteq D_i \; \forall i \in \{1, \ldots k\}$. A compressed relation $R'$ associated with $R$ is a set of ctuples.*

*Example 2.* Let $C_1$ be a constraint with Scope($C_1$)=$\{X, Y, Z\}$ and let $R$ be the relation associated with $C_1$,
$R$ = $\{(0,9,1), (1,9,2), (2,9,0), (3,9,3), (4,9,2), (5,9,3), (6,9,1), (7,9,1), (8,9,1), (9,9,3), (10,9,3)\}$
The compressed relation associated with $R$ is :
$R'$ = $\{((3,5,9,10),(9),(3)), ((0,6,7,8),(9),(1)), ((1,4),(9),(2)), ((2),(9),(0))\}$

*Remark 2.* Notice that each ctuple is a compact representation of a set of tuples. The ctuple $((3,5,9,10),(9),(3))$ is equivalent to the subset of tuples $\{(3,9,3),(5,9,3),(9,9,3),(10,9,3)\}$ of the relation $R$.

**Definition 5. (Compressed CSP)** *Let $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a CSP, the compressed CSP associated with $P$ is a CSP $P' =< \mathcal{X}, \mathcal{D}, \mathcal{C}' >$ such that :$|\mathcal{C}| = |\mathcal{C}'|$ and $\forall C_i \in \mathcal{C}, \exists C'_i \in \mathcal{C}'/Scope(C_i) = Scope(C'_i)$ and the compressed relation associated with $C'_i$ is the compressed version of $Rel(C_i)$.*

**Definition 6. (Compatible compressed tuples)** *Let $ct_1$ and $ct_2$ be two compressed tuples defined respectively on two sets of variables $S_1$ and$S_2$.*

- *If there are at least two tuples $t_1 \in ct_1$ and $t_2 \in ct_2$ such that $t_1[S1 \cap S_2] = t_2[S1 \cap S_2]$, then $ct_1$ and $ct_2$ are* compatible*, else $ct_1$ and $ct_2$ are* incompatible*.*
- *$ct_1$ is called* all-compatible *with $ct_2$ if $ct_1[S_1 \cap S_2] \subseteq ct_2[S_1 \cap S_2]$.*

### 3.2 Formal presentation of *CAS*

In this subsection, we will describe formally our approach given by the algorithm 2. Details of its main components, identified in bold, are given in the sequel.

**Algorithm 2** (CAS)

1: **Input** A Complete GHD $< T, \chi, \lambda >$ associated with a given CSP.
2: **Output** a solution $\mathcal{A}$ of the CSP if it is satisfiable
3: ***Compress_Csp(CSP)***
4: $\sigma = \{n_1, n_2, \ldots, n_m\}$ a node ordering with $n_1$ the root of the hypertree and each node precedes all its sons in $\sigma$
5: **for** each n in $\sigma$ **do**
6: $R_n = ( \bowtie^{\mathbf{cr}}_{C_j \in \lambda(n)} Comp(Rel(C_j)))[\chi(n)]$
7: **end for**
8: **for** $i = m$ to 2 **do**
9: Let $n_j$ the father of $n_i$ in $\sigma$
10: $R'_{n_j} = R_{n_j} \ltimes^{cr} R_{n_i}$
11: **end for**
12: Compute the solution in a backtrack free manner
13: return $\mathcal{A}$

**3.2.1 The compression algorithm (*Compress_Csp(CSP)*)** This procedure transforms each relation $R_i$ of a constraint $C_i$ in $\mathcal{C}$ into a compressed relation $R'_i$ according to the definition 4. This procedure is implemented by using the method proposed in [16].

The compressed tuples are derived from a decision tree representing the originals tuples. A decision tree T representing tuples of a given relation is a tree where each node $v$ is labelled with a literal $l(v)$ corresponding to a possible assignment of a value to a variable. For convenience, the edge of the left child of the node $v$ is also labelled with $l(v)$ while the right child is labelled with $\neg l(v)$. For a tuple $t$, the left child of a node $v$ in the decision tree $T$ is visited if the literal is true and right one if it is false. A tuple is associated with a node $v$ if it satisfies all the literals on the path from $v$ to the root of $T$. The set of the tuples associated with $v$ is noted $U(v)$.
At a given node $v$, if $U(v) = \emptyset$, $v$ is said $empty$. If $U(v)$ describes all the possible tuples that can be associated with it, $v$ is said complete.

A literal $s$ (resp $\neg s$) is said *implied* in a node $v$ if all the tuples associated with $v$ include (resp. do not include) $s$.
However, constructing optimal decision trees is NP-Complete [12]. So, heuristic approach is used in the algorithm 3 proposed in [16]. At each node of the decision tree T, this algorithm, checks if *implied* literals

**Algorithm 3** The compression algorithm

```
1: TableToDecisionTree( Set of tuples : U, Node : v)
2: if v is empty or v is complete then
3:     return
4: end if
5: if ∃ a literal s such that s is implied then
6:     v'={Parent:v, Edgeliteral: s}
7:     TableToDecisionTree(U(v'),v')
8: else
9:     s=Choose_Literal(U(v))
10:    v1={Parent:v, Edgeliteral: s}
11:    v2={Parent:v, Edgeliteral: ¬s}
12:    TableToDecisionTree(U(v1),v1)
13:    TableToDecisionTree(U(v2),v2)
14: end if
```

exist, then it extends T with a node for each of them. On contrary, if such literals do not exist, the function *Choose_Literal*$(U(v))$ selects a literal for the node $v$ and then expands each of the two obtained nodes $v1$ and $v2$. If an empty node ($U(v) = \phi$) or a complete node is created , then it stops.

Once the decision tree T is constructed, then for each complete leaf node $v$ a compressed tuple representing $U(v)$ is created as follows. let $S(v)$ be the set of literals labelling the nodes in the path from $v$ to the root of T. For each variable $Vi$ with domain $Di$, if there is a literal $V_i = d_i \in S(v)$, then $D_i = \{d_i\}$, else $D_i = D_i - \{d_j\}$ for each literal $V_i = d_j \in S(v)$. Then the compressed tuple that describes exactly the same tuples as $v$ is $(D_1, ..., D_n)$.

For the choice of a literal by the function *CHOOSELITRAL*, a set of splitting heuristics are proposed in [16]. In this work, we have used MAXFREQ heuristic because for the benchmarks used in this paper, this heuristic returned good compressions.

**3.2.2 The compressed join operation ($\bowtie^{cr}$)** : This procedure is mainly based on the following important definitions :

**Definition 7.** ***(Join of two compressed tuples, Join of two compressed relations )***
*Let $C_1$ , $C_2$ and $C_3$ be 3 constraints with $S_1$ =Scope($C_1$), $S_2$ =Scope($C_2$), $S$=Scope($C_3$) =$S_1 \cup S_2$.*
*Let $R'_1$ and $R'_2$ be two compressed relations associated with $R_1$ of $C_1$ and $R_2$ of $C_2$. Let $ct_1$ and $ct_2$ two ctuples of $R'_1$ and $R'_2$ respectively, the join of $ct_1$ and $ct_2$ , noted $ct_1 \bowtie^{ct} ct_2$, is a ctuple $ct$ defined on S such that:*

- $\forall x \in S - S_2, c\_value(x, ct) = c\_value(x, ct_1)$.
- $\forall y \in S - S_1, c\_value(y, ct) = c\_value(y, ct_2)$.

– $\forall z \in (S_1 \cap S_2), c\_value(z, ct) = c\_value(z, ct_1) \cap c\_value(z, ct_2)$.

The Join of two compressed relations $R'_1$ and $R'_2$, denoted by $R'_1 \bowtie^{cr} R'_2$, is a Compressed relation $R'$ defined over $S$ as a union of all possible join of compressed tuples: $R'_1 \bowtie^{cr} R'_2 = \bigcup_{ct_1 \in R'_1, ct_2 \in R'_2} ct_1 \bowtie^{ct} ct_2$

*Example 3.* Let $C_1$, $C_2$ and $C_3$ be three constraints where:

– $Scope(C_1)$={x,y,u} and $R_1$={(0,9,1),(1,9,2),(2,9,0),(3,9,3),(4,9,2),(5,9,3), (6,9,1),(7,9,1),(8,9,1),(9,9,3)}.
– $Scope(C_2)$={x,y,v} and $R_2$={(0,9,3),(1,8,3),(1,9,3),(5,8,3),(6,9,3),(7,9,3)}.
– $Scope(C_3)$={x,y,u,v} and $R_3$= $R_1 \bowtie R_2$={(0,9,1,3),(6,9,1,3),(7,9,1,3),(1,9,2,3)}

The Compressed relations $R'_1$ and $R'_2$ of $R_1$ and $R_2$ are :
$R'_1$= {((3,5,9),(9),(3)),((0,6,7,8),(9),(1)),((1,4),(9),(2)),((2),(9),(0))};
$R'_2$= {((0,1,6,7),(9),(3)),((1,5),(8),(3))} and $R'_3$=$R'_1 \bowtie^{cr} R'_2$={((0,6,7),(9),(1),(3)),((1),(9),(2),(3)}.

### 3.2.3 The compressed semi-join operation ($\ltimes^{cr}$) :

Let p and q be two compressed nodes of GHD such that q is the parent of p. $R_p$ is the compressed relation obtained by the compressed join of the compressed relations associated with the constraints in $\lambda(p)$. $R_q$ is the compressed relation obtained by the compressed join of the compressed relations associated with the constraints in $\lambda(q)$. The objective of the semi-join operation (noted $R_q \ltimes^{cr} R_p$ ) is to remove from all the compressed tuples in $R_q$ all the tuples which can not be extended to a global solution for the whole CSP instance. To do this, each compressed tuple $ct$ in $R_q$ is transformed to a set of compressed tuples representing all (and only) the tuples that are accepted in $ct$ and have supports in $R_p$.

**Definition 8.** ***(A support for a tuple in a compressed tuple)***
*Let $R_q$ and $R_p$ be two compressed relations associated respectively with the nodes q and p. Let $ct_i$ be a compressed tuple in $R_p$. Let t be a tuple accepted by a compressed tuple in $R_q$. A* support *for $t$ in $ct_i$ is the set of tuples that are accepted by $ct_i$ and compatible with $t$.*

**Definition 9.** ***(Consistent compressed relations)***
*Let $R_q$ and $R_p$ be two compressed relations associated respectively with the nodes q and p. $R_q$ is* ***consistent*** *with $R_p$ if for each compressed tuple $ct_2$ in $R_q$, there is a compressed tuple $ct_1$ in $R_p$ such that $ct_2$ is* ***all-compatible*** *with $ct_1$.*

The semi-join of two compressed relations $R_q$ and $R_p$ is a compressed relations $R'_q$ ($R'_q$= $R_q \ltimes^{cr} R_p$) consistent with $R_p$ : each tuple in $R'_q$, has a support in $R_p$.
$R'_q$ is obtained as follows: each compressed tuple $ct_2 \in R_q$ is transformed as follows : with a compressed tuple $ct_1 \in R_p$, we distinguish three cases :

1. *One or more tuples accepted by $ct_2$ have supports in $ct_1$.* In this case, we partition the tuples accepted by $ct_2$ in two groups : the tuples that have supports in $ct_1$ and the others. To do this, we derive from $ct_2$ :
   (a) a compressed tuple $ct_3$ *all-compatible* with $ct_1$. $ct_3$ accepts all the tuples in $ct_2$ that have supports in $ct_1$. Then, $ct_3$ is added to $R'_q$.

(b) A set S' of compressed tuples which are *incompatible* with $ct_1$. Each tuple accepted by $ct_2$ which has no support in $ct_1$ is accepted by a compressed tuple in S'. S' is derived from $ct_2$ as follows : for each variable y in $(\chi(q) \cap \chi(p))$, a new compressed tuple $ct_j$ is derived then created (from $ct_2$).
However, the tuples accepted by the compressed tuples in S' can have supports in the other compressed tuples in $R_p$. So, in order to search supports for them after $ct1$ in $R_p$, the elements of S' are inserted at the end of $R_q$. Then, $ct_2$ is removed from $R_q$ and the next compressed tuple in $R_q$ is treated.

2. *There is no support in $ct_1$ for any tuple in $ct_2$*:
In this case, we explore the next compressed tuple in $R_p$ which has not all the c_values of the variables in $(\chi(q) \cap \chi(p))$ included in their corresponding ones in $ct_1$. Because, if a compressed tuple $ct$ in $R_p$ has all the c_values of the variables in S included in those of $ct_1$, then $ct$ behaves like $ct_1$ with $ct_2$.

3. *There is no support in $R_p$ for any tuple in $ct_2$* :
In this case, we remove from $R_q$ each compressed tuple $ct_j$ s.t $\forall v \in (\chi(q) \cap \chi(p)), ct_j[v] \subseteq ct_2[v]$, because all the tuples accepted by such compressed tuples have no support in $R_p$. The process is repeated until $R_q$ becomes empty.

### 3.3 Theoretical properties of *CAS*

In order to prove the correctness of CAS, we have to establish some necessary lemmas.

**Lemma 1.** *Under the same hypotheses of definition 7, Let $ct_1$ and $ct_2$ be two compressed tuples. If $T_1$=tuples($ct_1$) and $T_2$=tuples($ct_2$), then $T_1 \bowtie T_2$ = tuples($ct_1 \bowtie^{ct} ct_2$)).*

**Lemma 2.** *If $ct_1$ and $ct_2$ are compatible then :*

1. *$ct_3$ is all-compatible with $ct_1$.*
2. *All the compressed tuples in S' are incompatible with $ct_1$.*
3. *The set of the tuples accepted by $ct_2$ is the union of the tuples accepted by the compressed tuple $ct_3$ and those accepted by the compressed tuples in S'.*

**Lemma 3.** *Let M be the set of all the tuples accepted in $R_q$ and not in $R'_q$. Each partial solution which includes a tuple in M for the variables in $\chi(q)$ can not be extended to a global solution of the considered CSP instance.*

**Proposition 1.** *The CAS algorithm is correct and complete w.r.t the AS algorithm*

*Proof.* AS, we have to prove the correctness of all its components. The procedure Compress_Csp has already been proved to be correct and complete in [16]. The second component of CAS which consists in compressed join operation is correct and complete thanks to the lemma 1. The third component of CAS which consists in compressed semi-semi operation is correct and complete thank to lemma 3

## 4 Experiments

We implemented the Compressed Acyclic Solving algorithm (CAS) using C++ language. The experiments are run on a Linux based HP Compaq 6720s 1,86 Ghz and 2 GO of RAM. The tests have been executed on benchmarks downloaded from the site[4]. For each instance the time out (TO) is fixed to 1800 seconds.

[4] http://www.cril.univ-artois.fr/CPAI08/

We used the Bucket Elimination (BE) heuristic [5] to compute a GHD of any CSP instance. BE is the best heuristic giving a nearly optimal generalized hypertree decomposition within a reasonable CPU time for the families of benchmarks used in this paper. In all the tables, $|V|$ is the number of the variables, $|E|$ is the number of constraints, w is the ghw of the decomposition returned by BE for the considered constraints hypergraph, r is the maximum cardinality of the constraint relations, $\rho$ is the compression ratio and G is the gain of compression. Notice that our time results include the time of de decomposition using BE, the time of completion and the time of CAS.

### 4.1 Memory gain

Before reporting the results of the experiments, we first present two notions to measure the efficiency of the compression algorithm.

**Definition 10.** *(**The compression gain** ) : Let $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a CSP instance and let $P' =< \mathcal{X}, \mathcal{D}, \mathcal{C}' >$ be its compressed representation. Let $C_i$ be a constraint, $R_i$ its relation in $P$, $R'_i$ its compressed relation in $P'$ and let $ct_j$ be a compressed tuple in $R'_i$. Let 's consider $G_{ct_j} = \sum_{k=1}^{|Scope(C_i)|} |c\_value(x_k, ct_j)|$. So, we have :*

- *The compression gain with respect $R'_i$: $G_{C_i} = \frac{|scope(C_i)|\times|R_i| - \sum_{i=1}^{|R'_i|} G_{ct_i}}{|scope(C_i)|\times|R_i|}$*
- **The compression gain***: $G = \frac{\sum_{i=1}^{|\mathcal{C}|} G_{C_i}}{|\mathcal{C}|}$.*

**Remarque 1** *$G$ is called the memory gain. It allows us to measure the memory savings that have been achieved when the constraints relations are compressed. Of course, more the gain is close to 0, more the compression is bad. The optimal gain is reached when the gain $G = 1$. Notice that we have supposed that each relation $R_i$ is associated with only one constraint.*

**Definition 11.** *(**Compression ratio**) : Let $P =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a CSP instance and let $P' =< \mathcal{X}, \mathcal{D}, \mathcal{C}' >$ its compressed representation. Let $R_i$ be the relation of $C_i$ and $R'_i$ its compressed version. The compression ratio $\rho$ measures the performance of the compression algorithm w.r.t the total number of tuples covered by the instance.*
*More formally $\rho$ is defined as follows : $\rho = \frac{\sum_{i=1}^{m} |R'_i|}{\sum_{i=1}^{m} |R_i|}$. m is the number of the relations of the CSP instances.*

**Remarque 2** *$\rho$ is a measure indicating the degree of compression w.r.t to constraints relations. More this number is close to zero and greater is the compression ratio. The optimal ratio is reached when $\rho = \frac{m}{\sum_{i=1}^{m} |R_i|}$ : this means that each compressed relation is represented by a unique compressed tuple.*

### 4.2 Results analysis

We consider for our experiments the benchmarks of Modified Renault used in [13]. These benchmarks correspond to very structured non binary CSPs. The class *Modified Renault* (abbreviation *ren* in the table 1) contains different instances involving domains containing up to 42 possible values. The greatest constraint relation contains 48721 tuples. The *original Renault* problem is obtained from a Renault Megane configuration problem.

In the sequel, we report the memory gain of compression algorithm and we compare the results we obtained by using CAS with the best results obtained by different variants of BTD. Concerning the BTD variants, the CPU times for this family are those given in [13]. The authors of this last paper used a PC Pentium IV, 3,2

GHZ with 1 MO of RAM and running under LINUX).

Our first results are reported in the table 1. The memory compression is about 50%, which is appreciable. Moreover CPU times for both CAS algorithm and BTD are often comparable. We believe that this result is indeed promising because, as far as we know BTD method is the best known approach to date for solving non binary CSPs using tree decomposition.

| Problems | Size | | | | Memory gain | | Time(s) | |
|---|---|---|---|---|---|---|---|---|
| | $\|V\|$ | $\|E\|$ | $r$ | w | $\rho$ | G | CAS | Best_BTD |
| $Ren3$ | 111 | 147 | 48721 | 3 | 0,03 | 0,55 | 6,14 | 10,67 |
| $Ren6$ | 111 | 147 | 48721 | 3 | 0,03 | 0,55 | 4,98 | 2,70 |
| $Ren12$ | 111 | 149 | 48721 | 3 | 0,02 | 0,55 | 5,09 | 10,49 |
| $Ren16$ | 111 | 149 | 48721 | 3 | 0,03 | 0,56 | 5,10 | 3,65 |
| $Ren17$ | 111 | 149 | 48721 | 4 | 0,02 | 0,55 | 7,48 | 3,41 |
| $Ren18$ | 111 | 149 | 48721 | 3 | 0,02 | 0,56 | 4,69 | 10,46 |
| $Ren19$ | 111 | 149 | 48721 | 3 | 0,03 | 0,55 | 4,67 | 7,74 |
| $Ren23$ | 111 | 159 | 48721 | 3 | 0,04 | 0,54 | 8,26 | 2,81 |
| $Ren24$ | 111 | 159 | 48721 | 4 | 0,04 | 0,54 | 7,89 | 7,63 |
| $Ren30$ | 111 | 154 | 48721 | 3 | 0,03 | 0,55 | 91,71 | 3,80 |
| $Ren35$ | 111 | 154 | 48721 | 4 | 0,03 | 0,56 | 44,60 | 7,32 |
| $Ren36$ | 111 | 154 | 48721 | 4 | 0,03 | 0,55 | 20,40 | 1,78 |
| $Ren37$ | 111 | 154 | 48721 | 4 | 0,02 | 0,56 | 4,47 | 13,68 |
| $Ren39$ | 111 | 154 | 48721 | 5 | 0,02 | 0,56 | 4,70 | 1,79 |
| $Ren40$ | 108 | 149 | 48721 | 4 | 0,02 | 0,56 | 4,55 | 5,86 |
| $Ren42$ | 108 | 149 | 48721 | 3 | 0,02 | 0,56 | 5,33 | 2,48 |
| $Ren47$ | 108 | 149 | 48721 | 4 | 0,02 | 0,56 | 5,17 | 53,71 |

**Table 1.** Comparison between BTD and $CAS$ : Modified Renault benchmarks

For the benchmarks used in this paper, CAS performs better than Gottlob method. The last method suffers to solve all these benchmarks. Obviously, CAS behaves better when the compression ratio $\rho$ is important (approaching $\frac{|C|}{\sum_{i=1}^{|C|}|R_i|}$). Also, CAS depends in the quality of the decomposition. We have tested the decomposition returned by det k decomp [10] for the benchmark Renault modified 30 and we have obtained a time resolution of 14 seconds instead of 91,71 required by the decomposition returned by BE. CAS as for Gottlob method behaves good when the hypertree width or the number of classical tuples is small in all the relations.

## 5 Conclusion

In this paper we have exploited the algorithm proposed in [16] for compressing large relational constraints. As a result, we have experimentally shown that compression algorithm delivers an appreciable memory compression gain which is about 50% for the used benchmarks. Moreover, we demonstrated that this compression algorithm is useful in practice, by using it to improve the performance of existing Acyclic Solving algorithm for solving non binary CSPs. Indeed CAS algorithm gave quite encouraging results which are generally comparable to

the results obtained when the best algorithm known to date is used. Future works will include a larger study of compression algorithm based on different heuristics used by the decision tree approach to derive the compressed relations.

## References


1. DAVID COHEN, PETER JEAVONS, M. G. A unified theory of structural tractability for constraint satisfaction and spread cut decomposition. In *Proceedings of IJCAI'05* (2005).
2. DECHTER, R. unifying framework for reasoning. *Artificial Intelligence 113* (1999).
3. DECHTER, R. *Constraint Processing*. Morgan Kaufmann, 2003.
4. DECHTER, R., AND PEARL, J. Tree-clustering schemes for constraint-processing. In *Proceedings of the sixth National Conference on Artificial Intelligence (AAAI-88)* (Saint Paul, MN, 1988), pp. 150–154.
5. DERMAKU, A., GANZOW, T., GOTTLOB, G., MCMAHAN, B., MUSLIU, N., AND SAMER, M. Heuristic methods for hypertree decompositions. Tech. rep., DBAI-R, 2005.
6. FOCCACI, F., AND MILANO, M. Global cut framework for removing symmetries. In *Proceedings of seventh international conference On priciples and Practice of Constraint programming* (2005).
7. GOTTLOB, G., LEONE, N., AND SCARCELLO, F. A comparison of structural csp decomposition methods. *Artificial Intelligence 124* (2000), 243–282.
8. GOTTLOB, G., LEONE, N., AND SCARCELLO, F. Robbers, marshals and guards : Theoretic and logical characterizations of hypertree width. *Journal of the ACM* (2002).
9. GOTTLOB, G., MIKLOS, Z., AND SCHWENTICK, T. Generalized hypertree decomposition : Np - hardness and tractable variants. In *Proceedings of the 26 th ACM SIGMOD SICACT SIGART Symposium on principles of databases systems* (2007).
10. GOTTLOB, G., AND SAMER, M. A backtraching based algorithm for computing hypertree decompositions. *arXivics.DS 0701083vl* (2007).
11. GYSSENS, M., JEAVONS, P. G., AND COHEN, D. A. Decomposing constraint satisfaction problems using database techniques. *Artificial Intelligence 66* (1994), 57–89.
12. HYAFI, L., AND RIVEST, R. Constructing optimal decision binary decision trees is np complete. *Information processing letters 5* (1976), 15–17.
13. JÉGOU, P., NDIAYE, S. N., AND TERRIOUX, C. Combined strategies for decomposition-based methods for solving csps. In *Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2009)* (2009), pp. 184–192.
14. JÉGOU, P., AND TERRIOUX, C. Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence, 146* (2003), 43–75.
15. KATSIRELOS, G., AND BACCHUS, F. Generalized nogoods in csps. In *Proceedings of the twentieth national conference On Artificial Intelligence* (2001).
16. KATSIRELOS, G., AND WALSH, T. A compression algorithm for large arity extentional constraints. In *Proceedings of CP'07* (2007), pp. 379–393.
17. KORIMORT, T. Heuristic hypertree decomposition. *AURORA TR 2003-18* (2003).
18. MONTANARI, U. Networks of constraints: Fundamental properties and applications to pictures processing. *Information Sciences 7* (1974), 95–132.
19. MUSLIU, N., AND SCHAFHAUSER, W. Genetic algorithms for generalized hypertree decompositions. *European Journal of Industrial Engineering 1*, 3 (2005), 317–340.
20. SAMER, M. Hypertree-decomposition via branch-decomposition. In *Proceedings of the 19th international joint conference on Artificial intelligence* (Edinburgh, Scotland, 2005), pp. 1535–1536.

# Endogenous Formation of Coalitions in Research and Development: Modeling and Study of Stability Conditions

Razika Sait[1], A.Hakim Hammoudi[2], Mohammed Said Radjef[1]
Soraya Ait Aissa[1], and Mira Birem[1]

[1]Laboratory of Modeling and Optimization of Systems (LAMOS)
Bejaia University, Algeria
[2]National Institute of Agronomic Research-Food and Social Sciences
(INRA-ALISS) France

{zika_univ_bejaia@yahoo.fr,hammoudi@ivry.inra.fr,radjefms@yahoo.fr}

**Abstract.** The approach of endogenous formation of coalition considers that coalitions result of an individual choice of each player. This paper study the impact of Research and Development (R&D) externalities on the stability of coalitions that arise endogenously between $N$ firms in order to coordinate their activities in R&D. Sequential game with three stages has been constructed where at the first stage, the $N$ firms decide simultaneously whether or not to conduct R&D jointly in a $n$-coalition. At the second stage, the $(N - n + 1)$ firms engage in a non-cooperative game in which they decide simultaneously their levels of R&D investment, but at the third stage, the $N$ firms remain non-cooperative rivals in the product market.

**Keywords:** Nash equilibrium, Endogenous formation of coalitions, Research and Development, Coalition, Stable coalition.

## 1 Introduction

In recent years, the application of the game theory to the topic of Research and Development (R&D) cooperation has become a very active field of research in applied microeconomics [3]. In this context, there is a large theoretical literature on R&D cooperation and competition following the pioneering papers studied by D'Aspremont and Jacquemin [4]. Most of this papers ([8], [5], [1], [6], [2] ...) are interested only on the study of the collective incentives of two firms to cooperate in R&D by considering a two-stage game. But little attention has been devoted to oligopolistic market in which there are several firms [10]. So, the originality of our contribution to the field of endogenous formation of coalition approach with externalities is to generalize the model of D'Aspremont [4]. The problem seems more complex with several firms, since one should wonder with which one wants to enter into partnership and what non-partners are going to do. A model based on three-stage game has developed where at the first, the $N$ firms decide

simultaneously whether or not to conduct R&D jointly in a $n$-coalition. At the second one, the $(N-n+1)$ firms engage in a non-cooperative game in which they decide simultaneously their levels of R&D investment but at the third stage, the $N$ firms decide simultaneously their levels of production.

## 2 Basic hypothesis of the model

Consider an industry where $N$ symmetric firms are in competition in a homogeneous market. These firms are firstly engaged in costly R&D investment in order to reduce their unit production cost, denoted $c \geq 0$, and then compete as rivals in the product market assuming that:

- The market inverse demand function is linear, $P(Q) = a - Q$, where $Q$ is the industry supply.
- $\beta \in [0, 1]$ represents the proportion of knowledge diffusion, known in the economic literature by R&D spillover.
- $f_i$ is the R&D cost function of a firm $i$. Supposed to be quadratic on its R&D investment, denoted $r_i \geq 0$. It is defined by:

$$f_i(r_i) = \frac{r_i^2}{2}, \quad i = 1, ..., N. \tag{1}$$

In the following section, we study the case where there is no cooperation between firms. In section four, we study the case where at the first stage a cooperation in R&D can be formed endogenously. Numerical results representing the impact of R&D spillovers on the profitability and the stability of the R&D coalition are provided and discussed in section 5 and 6.

## 3 The firms do not cooperate in R&D

### 3.1 Formulation of the game

Let $I = \{1, 2, \ldots, N\}$ be the set of firms representing the players of the associated game. The model is a two stage non-cooperative game:

1. At the first stage, each firm $i$ chooses its own level R&D effort, $r_i \geq 0, \ \ i \in I$, which contributes in reducing its own marginal cost of production and assumed to affect over to the others firms reducing the rival's constant marginal cost by $\beta r_i$.
2. At the second stage the $N$ firms choose simultaneously their output levels $q_i \geq 0, \ \ i \in I$.

Thus, the maximization problem of firm $i$ $(i \in I)$ is given by:

$$\underset{q_i, r_i \geq 0}{max} \ \ q_i(P(Q) - c - r_i - \beta \sum_{j=1, j \neq i}^{N} r_j) - \frac{r_i^2}{2}, \quad i \in I \tag{2}$$

### 3.2 Equilibrium

We use backward induction in order to compute the subgame perfect Nash equilibria in pure strategies [9].

#### 3.2.1 Nash equilibrium of the second stage

Applying the first-order and second-order necessary and sufficient optimality conditions to the problem (2), the level output equilibrium, denoted $q_i^*$, is given by:

$$q_i^*(r_i, r_{-i}) = \frac{a - c + (1-\beta)r_i + \sum_{j=1}^{N} r_j}{N+1}, \quad i \in I \tag{3}$$

Then, the firm $i$'s profit function in the second stage, denoted $\pi_i^2$ is given by:

$$\pi_i^2(r_i, r_{-i}) = \frac{[a - c + (2\beta - 1)\sum_{j=1}^{N} r_j + (N+1)(1-\beta)r_i]^2}{(N+1)^2} - \frac{r_i^2}{2}, \quad i \in I \tag{4}$$

#### 3.2.2 Nash equilibrium of the first stage

At the first stage of the game, the firms choose simultaneously their R&D investment levels $r_i \geq 0, i \in I$. Given the symmetry of $N$ firms, solving the maximization problem of firm $i$ ($i \in I$) at this stage gives the same R&D investment's level equilibrium, denoted $r_i^{nc} \geq 0$, for each firm by:

$$r_i^{nc}(\beta) = \frac{2(N(1-\beta)+\beta)(a-c)}{D + (N-1)E}, \quad i \in I \tag{5}$$

where $D = (N+1)^2 - 2(N(1-\beta)+\beta)^2$ and $E = -2(N(1-\beta)+\beta)(2\beta - 1)$
Therefore, the non-cooperative profit of firm $i$ in the first stage is defined by:

$$\pi_{nc}(1) = \frac{(a-c)^2[(N+1)^2 - 2\sigma^2]}{[(N+1)^2 - 2\sigma^2 - 2\sigma(N-1)(2\beta-1)]^2}, \quad i \in I \tag{6}$$

where $\sigma = [N(1-\beta)+\beta]$

## 4 The firms form a R&D coalition

### 4.1 Formulation of the game

In this case, a three stage non-cooperative game is considered:

1. At the first stage, the $N$ firms announce simultaneously whether or not to conduct R&D in a $n$-coalition which considered as one firm. Let $C_n = \{1, 2, \ldots, n\}$ be the subset of $n$ first firms form a R&D coalition and allow full sharing of information. And noted by $F_g = \{n+1, n+2, \ldots, N\}$ the subset of firms that choose their investment levels independently, called the fringe.

2. At the second stage, the $n$ firms obtain an identical reduction of the production costs by a joint effort of research $r_c \geq 0$. At this stage, the $(N-n+1)$ firms engage in a non-cooperative game in which they decide simultaneously their R&D level. We suppose that there is R&D spillovers, between R&D coalition and the firms of the fringe.
3. At the third stage, the $N$ firms compete; they decide simultaneously their levels of production in order to maximize their individual profit which is defined for:

Case $\mathbf{1} : \mathbf{i} \in \mathbf{C_n}$ :

$$\pi_i(n, r_c, r_{-c}, q_i, q_{-i}) = q_i(P(Q) - \beta \sum_{j=n+1}^{N} r_j) - \frac{1}{n}(\frac{r_c^2}{2}), \quad i = \overline{1,n} \tag{7}$$

where $r_{-c} = (r_{n+1}, r_{n+2}, \ldots, r_N)$

Case $\mathbf{2} : \mathbf{i} \in \mathbf{F_g}$ :

$$\pi_i(n, r_i, r_{-i}, q_i, q_{-i}) = q_i(P(Q) - \beta(r_c + \sum_{k=n+1 \ k\neq i}^{N} r_k)) - \frac{r_i^2}{2}, \quad i = \overline{n+1, N} \tag{8}$$

where $r_{-i} = (r_c, r_{n+1}, r_{n+2}, \ldots, r_{i-1}, r_{i+1}, \ldots, r_N)$

### 4.2 Equilibrium

The Nash equilibrium of this market is given by computing the subgame perfect Nash equilibria [9] in pure strategies which are obtained by backward induction.

#### 4.2.1 Output level equilibrium

The Nash equilibrium of this stage, denoted $q^* = (q_1^*, \ldots, q_N^*)$, is given by:

$$q_i^*(n, r_c, r_{-c}) = \frac{a - c + \theta(n) r_c + (2\beta - 1) \sum_{j=n+1}^{N} r_j}{N+1}, \quad i = \overline{1,n} \tag{9}$$

$$q_i^*(n, r_i, r_{-i}) = \frac{a - c - \lambda(n) r_c + (2\beta - 1) \sum_{j=n+1}^{N} r_j + (N+1)(1-\beta) r_i}{N+1}, \quad i = \overline{n+1, N} \tag{10}$$

where $\theta(n) = (N-n)(1-\beta) + 1; \quad \lambda(n) = n(1-\beta) - \beta$.
So, the firm $i$'s profit function of this stage, denoted $\pi_i^3$, is written as follows:

$$\pi_i^3(n, r_c, r_{-c}) = \frac{[a - c + \theta(n) r_c + (2\beta - 1) \sum_{j=n+1}^{N} r_j]^2}{(N+1)^2} - \frac{r_c^2}{2n}, \quad i = \overline{1,n} \tag{11}$$

$$\pi_i^3(n, r_i, r_{-i}) = \frac{[a-c-\lambda(n)r_c+(2\beta-1)\sum_{j=n+1}^{N} r_j+(N+1)(1-\beta)r_i]^2}{(N+1)^2} - \frac{r_i^2}{2}, \quad i = \overline{n+1, N} \tag{12}$$

#### 4.2.2 Nash equilibrium of the second stage

The market competition in the second stage consist in a confrontation of the $(N - n + 1)$ firms:

- Coalition firms $C_n$ determine their collaborative effort $r_c \geq 0$.
- The $(N-n)$ fringe firms, which choose their investment levels simultaneously.

Thus, the Nash equilibrium levels of R&D investment for each firm coalition is given by:

$$r_c^*(n) = \frac{b_c(D-E)}{A(D-E)+(N-n)(AE-BC)} \tag{13}$$

and the R&D investment's level for each firm of the fringe is given by:

$$r_j^*(n) = \frac{Ab_{nc} - Cb_c}{A(D-E)+(N-n)(AE-BC)}, \quad j = \overline{n+1, N} \tag{14}$$

where:

$$\begin{aligned} &b_c = 2n\theta(n)(a-c),\\ &b_{nc} = b_j = 2\ \sigma(a-c),\\ &A = (N+1)^2 - 2n(\theta(n))^2,\\ &B = -2n\theta(n)(2\beta-1),\\ &C = 2\sigma\lambda(n). \end{aligned}$$

Then, the function profit of each firm coalition of the second stage, denoted $\pi_c^i$, is given by:

$$\pi_c^i(n) = \frac{(a-c)^2[\alpha_1 - 2n^3\alpha_2 + 4n^2\alpha_3 - 2n\alpha_4]}{[Y_1 - 4n^5Y_2 + 4n^4Y_3 + 2n^3Y_4 + 4n^2Y_5 - 2nY_6]^2}, \quad i = \overline{1, n} \tag{15}$$

where

- $\alpha_1 = [(N+1)^3 - 2\sigma(N+1)^2(1-\beta)]^2$,
- $\alpha_2 = [(N+1)^2(1-\beta) - 2\sigma(N+1)^2(1-\beta)^2]^2$,
- $\alpha_3 = (N(1-\beta)+1)(1-\beta)[(N+1)^2 - 2\sigma(N+1)(1-\beta)]^2$,
- $\alpha_4 = [(N(1-\beta)+1)((N+1)^2 - 2\sigma(N+1)(1-\beta))]^2$,
- $Y_1 = (N+1)^4 - 2\sigma(N+1)^3(1-3\beta)$,
- $Y_2 = N\sigma(1-\beta)^3(2\beta-1)$,
- $Y_3 = \sigma(1-\beta)^2(2\beta-1)(2+N(2-\beta))$,
- $Y_4 = 2\sigma(N(1-\beta)+1)(1-\beta)(2\beta-1)[(2-6\beta) - N(1-\beta)] + 2\sigma(1-\beta)^2[N(1-2\beta-2\beta^2) + (1-3\beta)] - (N+1)^2(1-\beta)^2$,
- $Y_5 = (N(1-\beta)+1)(1-\beta)[(N+1)^2 - 2\sigma(N(1-\beta-4\beta^2) - (1-3\beta))] + \sigma(N(1-\beta)+1)^2 (2\beta-1)(N-1)(1-\beta)$,
- $Y_6 = (N(1-\beta)+1)^2[(N+1)^2 - 2\sigma(N+1)(1-3\beta) + 2N\beta(2\beta-1)] - \sigma(N+1)^2(2\beta-1)$.

## 5 Study of the collective rationality

We said that there is a collective rationality to cooperate, or the coalition $C_n$ is profitable, if all members get a better profit in the coalition $C_n$ rather than if there is no cooperation. Formally:

$$\pi_c^i(n) \geq \pi_{nc}(1), \quad \forall\, i \in C_n. \tag{16}$$

The numerical results applied for $a = 100$, $c = 50$ and $N = 5, 10, 15$ are represented in Table 1.

**Table 1.** Spillover impact on the profitability of the coalition

| n | $N = 5$ | $N = 10$ | N=15 |
|---|---|---|---|
| 2 | $\beta \in [0.4, 0.7]$ | $\beta \in [0.5, 0.8]$ | $\beta \in [0.5, 0.9]$ |
| 3 | $\beta \in [0.4, 0.6]$ | $\beta \in [0.6, 0.8]$ | $\beta \in [0.6, 0.9]$ |
| 4 | $\beta \in [0.2, 0.5]$ | $\beta \in [0.6, 0.7]$ | $\beta \in [0.6, 0.8]$ |
| 5 | $\beta \in [0.2, 0.4]$ | $[0.6, 0.7]$ | $\beta \in [0.6, 0.8]$ |
| 6 | - | $\beta \in [0.5, 0.6]$ | $\beta \in [0.7, 0.8]$ |
| 7 | - | $\beta \in [0.4, 0.5]$ | $\beta \in [0.6, 0.7]$ |
| 8 | - | $\beta \in [0.4, 0.5]$ | $\beta \in [0.6, 0.7]$ |
| 9 | - | $\beta \in [0.3, 0.4]$ | $\beta = 0.6$ |
| 10 | - | $\beta = 0.3$ | $\beta \in [0.5, 0.6]$ |
| 11 | - | - | $\beta = 0.5$ |
| 12 | - | - | $\beta \in [0.4, 0.5]$ |
| 13 | - | - | $\beta = 0.4$ |
| 14 | - | - | $\beta \in [0.3, 0.4]$ |
| 14 | - | - | $\beta = 0.3$ |

### 5.1 Results interpretation

According to the results in Table 1, we note that:

- The profitability of the coalition $C_n,\ n = \overline{2, N}$, depends on the number of firms in the market, $N$, and the number $n$ of firms in cooperation. It depends also with the level of spillovers $\beta$.
- No coalition $C_n,\ n = \overline{2, N}$ is profitable if the spillover rate is lower than a some critical value $\beta_n^{min}$.
- No coalition $C_n,\ n = \overline{2, N}$ is profitable if the spillover rate is greater than a some critical value $\beta_n^{max}$.
- When the number of firms engaging, $n$, increases, the coalition $C_n$ becomes profitable if the level of spillovers is low. In other words, the more the effects of spillovers are important, the firms have little incentive to cooperate in R& D.
- For $N \geq 6$, the grand coalition is profitable only if $\beta = 0.3$.

## 6 Study of the stability

Due to the free-rider behavior, the study of the profitability is not enough to describe a sustainable R&D coalition. This is why we study the stability in this section which is considered as a major concept in coalition formation.

**Definition 1.** *A coalition $C_n$ is said to be stable if no coalition member has any interest in leaving (internal stability) and no fringe firm has an incentive to join (external stability) ([4], [12]).*

The numerical results applied to $a = 100$, $c = 50$ and $N = 5, 10, 15$ are summarized in Table 2.

**Table 2.** Spillovers impact on the stability of the coalition

| n | N=5 | N=10 | N=15 |
|---|---|---|---|
| 2 | not stable | not stable | not stable |
| 3 | $\beta = 0.4$ | not stable | not stable |
| 4 | $\beta = 0.3$ | not stable | not stable |
| 5 | $\beta \in [0.5, 0.8]$ | not stable | $\beta = 0.6$ |
| 6 | - | not stable | not stable |
| 7 | - | not stable | not stable |
| 8 | - | not stable | not stable |
| 9 | - | $\beta = 0.3$ | not stable |
| 10 | - | $[0.5, 0.9]$ | not stable |
| 11 | - | - | not stable |
| 12 | - | - | $\beta = 0.4$ |
| 13 | - | - | not stable |
| 14 | - | - | $\beta = 0.3$ |
| 15 | - | - | $\beta \in [0.5, 0.9]$ |

### 6.1 Results interpretation

The numerical results show that:

- The stability of the coalition depends on the number of cooperating firm, $n$, the number of the firms on the market, $N$, and the degree of R&D spillovers, $\beta$.
- Excepting $N = 5$, the grand coalition, $C_n$, is stable only if $\beta \in [0.5, 0.9]$. In other words, the grand coalition is stable only if the levels of R&D spillovers rate are large enough to internalize the R&D spillovers.

## 7 Conclusion

This paper develops a general version of the [4] model which still allows for the calculation of specific equilibria and therefore enables a comparison between cooperative and non-cooperative R&D. Analysis of this generalization shows that the conceptual point of view is very complex because the profitability alone or the stability alone is not enough to describe a sustainable R&D coalition. Our main results show that:

- Internalizing spillovers through R&D coalition is beneficial because firms would otherwise spend less on R&D coalition due to free-rider behavior.
- We can have profitability but not internal stability. In this case, there is a collective rationality to cooperate, but if there are no contractual commitments the coalition can not be formed.

## References


1. Aloysius, J.A.: Cooperative and Noncooperative R&D in Duopoly with Spillovers. The American Economic Review. 136, 591-602 (1990)
2. Aloysius, J.A.: Research joint ventures: Acooperative game for competitors. European Journal of Operational Research. 136, 591-602 (1992)
3. Amir, A.: Modelling Imperfectly Appropriate R&D via spillovers. International Journal of Industrial Organization. 18, 1013–1032 (2000)
4. D'Aspremont, C., Jacquemin, A.: Cooperative and noncooperative R&D in duopoly with spillovers. The American Economic Review. 78, 1133–1137 (1988)
5. De Bondt, R., Veugelers, R.: Strategic investment with spillovers. European Journal of Political Economy. 7, 345–366 (1991)
6. Hinloopen, J.: Subsidizing cooperative and noncooperative R&D in duopoly with spillovers. Journal of Economics. 66, 151–175 (1997)
7. Kamien, M., Muller, E., Zang, I.: Research Joint Ventures and R&D Cartels. American Economic Review. 82, 1293–1306 (1992)
8. Katz, M.: An anlysis of cooperative research and development. Rand journal of Economics. 17, 527-543 (1986)
9. Selten, R.: Reexamination of the perfectness concept for equilibrium points in extensive games. International Journal of Game Theory. 4, 25–55 (1975)
10. Suzumura, K.: Cooperative and Noncooperative R&D in an Oligopoly with Spillovers. The American Economic Review. 82, 1307–1320 (1992)
11. Tesoriere, A.: Cooperative and noncooperative R&D in duopoly with spillovers. The American Economic Review. 78, 1133–1137 (1997)
12. Von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (USA) (1944).

# Objets 3D et optimisation

# 3D objects representation and recognition by using topological invariants

Salah Derdar[1], Djemel Ziou[2], Nadir Farah[1], Med Tarek Khadir[1]
derdar@labged.net, djemel.ziou@usherbrooke.ca, nadir.farah@labged.net, and khadir@labged.net

[1] Département d'Informatique
Laboratoire LabGED
Université d'Annaba
Po-Box 12, 23000, Annaba, Algérie
[2] Département d'informatique
Université de Sherbrooke
Sherbrooke J1K 2R1, Qc, Canada

**Abstract.** Holes, tunnels and cavities of 3D objects are topological features which can be used for object representation and recognition. In this work, an algebraic topology approach is proposed for counting their numbers and for localizing them. A summarized complex based representation of a 3D object is build and translated into algebraic language. Then, the homology group is constructed, where its ranks are the number of holes, tunnels, and cavities. The localization of these invariants is based on the reconstruction of cubical generators of the homology group and allows to find contours of the invariants. It follows that the resulting algorithm handles easily an object composed by several connected components, holes, tunnels, and cavities. The proposed algorithm is validated by using various 2D and 3D images.

## 1 Introduction

Object representation is an important issue in computer vision and computer graphics. Among used features, the invariants under a continuous deformations are relevant for the classification of objects and spaces. Most existing algorithms for their estimation are based on discrete geometry [27] [22]. More recently, concepts of algebraic topology such as homology groups are also used for the counting the topological invariants [24]. The discrete geometry and algebraic topology are two different areas of knowledge. In algebraic topology, objects are expressed in algebraic language while in discrete geometry, all reasoning is done at the combinatoric level. With the evolution of imaging technology on 3D objects, and increasing the huge amount of data processed, homology groups seems to be suitable for object representation, allowing the computation of some non common features such as the holes and cavities [31]. Many computer application areas involve algebraic topology including image processing, image analysis, computer graphics, molecular modelling [24]. Also, homology groups are built from low-level primitives (pixels, edges, surfaces, ...) output of some image processing algorithms such as the acquisition [40], binarization [39], edge detection [38]. [20] [24] [8]. These concepts were used for feature extraction and image representation [2] [4] [5] [1] [3] [31]. A model for the images where roots are from physics and algebraic topology is proposed by Ziou and Allili [32]. This model includes the image support, quantities, and the processing operations. The image support is seen as cubical and simplicial complexes, or any other geometrical primitive. The image quantities are described by co-chains and the operations by co-boundaries and hodge operator. Furthermore, in this model, the homology theory is used for computing the Euler number as well as the number of connected components and holes [1] [3] [31]. The matching of topological invariants was also studied by using the same formalism [8]. In this paper, we are interested in the localization of the cycles in 2- and 3-dimensional images that are connected components, holes, tunnels, and cavities. In all these works, the image support is considered as a cubical complex. As consequence, the homology groups estimation is time and memory consuming. To overcome this drawback, we propose the use of an image support which does not fulfill one of the standard cubical complex requirements at the geometrical level and we will show that it is also a complex at the algebraic level. It follows that the

concepts of homology groups can be reused. Compared to the state of art [2] [8] [1] [3] [31], this extension of the cubical complex concepts has several benefits which are: 1) computing homology and localizing cycles in 2D and 3D images which is less time and memory consuming, 2) giving an efficient representation in the case of the uniform topological space (i.e. there are many connected components in a same space), 3) reducing the number of cubes instead of computing the homology groups and localizing their homology generators in the base where the number of cubes is large, 4) representing 3D objects and high-dimensional data. So, we propose an extension of the cubical complex concept called the summarized complex and use it for the characterization of 2D and 3D objects by estimating the number and the position of connected components, holes, tunnels and cavities. Indeed, the new approach is composed of two successive tasks called the labelling and localization tasks. The localization task consists firstly of computing the homology groups and secondly reconstructing the cycle chains that represent the corresponding cubical generators. By definition, a cubical generator of a calculated homology group is algebraically a cycle chain of generators and geometrically a set of cubes that are visually plotted in the original complex. The useful cycles that can be localized are holes in 2-dimensional images and tunnels and cavities in 3-dimensional images.

The paper is organized in five sections. In section 2, we present the new structure of the summarized complex with high cubical representation. We discuss the homology computation with the reduction algorithm in section 3. The approach for localizing the cycles in 2D and 3D images is presented in section 4. Section 5 gives the computational results.

## 2 Problem statement

Cubical grid is often used by many algorithms in graphics and computer vision. Pixels and voxels in an image can be seen as cubes [32]. More generally, an image was defined as a support formed by cubes and quantities assigned to these cubes [31][3]. For an image of dimension $n$, the support is a $n$-cubical complex. For the sake of simplicity and without loss of generality, let us consider a binary $n$-image (i.e., image of dimension $n$). The pixels belonging to the object are set to one and those of the background are zero. The support can be seen as the cubes formed by pixels having the value one. In the case of $n$-dimensional images, the image support is a set of unit $n$-cubes which denoted $n$-pixels. When $n$=0, the image is a set of vertices; when $n$=1, a set of edges; when $n$=2, a set of squares; when $n$=3, a set of cubes, and so on [3]. Any two $n$-pixels are either disjoint or intersect in a common $p$-pixel where $p < n$. This subdivision of the image support which is achieved via a cubic tessellation is enough to apply concepts of the cubical homology theory including the homology [4] [5] [11].

### 2.1 High cubical representation

In this paper, a new subdivision of image support is given based on a new geometric entity called the high cube and denoted $n$-hcube. The high cube is a set of adjacent cubes that are regrouped in the rectangular form according to their positional coordinates. More precisely, a high cube is conceived to reduce the size of the complex and is characterized by two arrays. The first one gives the positional coordinates, then, the second array presents the high cube scale according to canonical axes. Indeed, the smallest high cube is a simple cube with scale equal to one. This new subdivision is conceived in the goal to extract a minimum number of high cubes from the image support. Recall that, construct a complex from a small number of high cubes gives more advantages and benefits in the localization task. At the geometrical level, the image support we propose here does not fulfill the requirement of a common face between two cubes of the standard cubical complex. In this subdivision, any two high cubes are either disjoint or partially intersect in a common face which is not considered as a high cube in this complex. Figure 1(i) describes such support where two 2-hcubes $A$ and $B$ are intersected in a partial face $b_3 \cap a_4$ where $a_4$ and $b_3$ are respectively faces of 2-hcubes $A$ and $B$. In image processing, a such support can be the output of trees based segmentation algorithms [35] [37] [36] [34]. As the last requirement needed in a standard cubical complex is not fulfilled, this new subdivision gives a non cubical complex called the summarized complex. In other words, a non cubical complex at combinatorial level can be a cubical complex at algebraic level after some specific manipulations. However, for generating easily

the corresponding summarized chain complex, we will perform the labelling task which is composed of three successive steps. We will firstly construct the summarized complex, secondly, standardize the boundaries of the constructed complex by using a new method called the splitting process, and the last step consists of generating the summarized chain complex. The generated chain complex can be used firstly in computing homology groups and secondly in localizing these groups by determining simply its cycle chain. The two last steps describe together the localization task.

### 2.2 Splitting process

This process is applied on invalid summarized complex in the goal to standardize the boundaries of their high cubes and to give a valid one. Recall that, the boundary standardization conducts to modify the set of all high cubes, by modifying the boundary of the old high cubes and adding new high cubes of small dimension. For example, if we take a high cube of dimension 2, its boundary is constituted of four 1-faces. After applying the splitting process, this high cube can have a new boundary constituted of more than four 1-faces. Thus, the modified summarized complex can be formed by a new set of high cubes. Even it is not a cubical complex in combinatorial level, but it is considered a correct cubical complex in algebraic level, i.e. it fulfills all requirements that needed in a standard cubical complex. Formally, we can say that the splitting process is of level $q$, when, it is performed on all high cubes of dimension $q$ for $q$=$n$-1 to 1 where $n$ is the dimension of the given complex. For example, in figure 1(i) the two 2-hcubes $A$ and $C$ that have respectively the boundaries given by 1-chains $+a_1$-$a_2$-$a_3$+$a_4$ and $+c_1$-$c_2$-$c_3$+$c_4$ are two adjacent high cubes. Algebraically, between these two 2-hcubes, there is not a common face in their boundaries. Geometrically, these two high cubes are partially intersected or have a partial face (see figure 1(ii)). This partial face is the intersection between two 1-hcubes $a_4$ and $c_3$. To remedy the existence of the partial face, the splitting process is performed between 1-hcubes $a_4$ and $c_3$ and gives a 1-hcube $c_3$ as a common face between the two 2-hcubes $A$ and $C$. For more details, $a_4$ is split into $+a_5$+$c_3$ and $c_3$ is maintained not split. Consequently, the new boundary chain of 2-hcubes $A$ is $\partial_2 A$=$+a_1$-$a_2$-$a_3$+$a_5$+$c_3$. Also, the two 2-hcubes $A$ and $B$ have also two partially intersected faces $a_5$ and $b_3$ (see figure 1(iii)). After two successive steps of splitting process, $a_5$ is split into $+b_3$+$a_6$ and $b_3$ is maintained not split. Then, the new boundary chain of 2-hcubes $A$ is $\partial_2 A$=$+a_1$-$a_2$-$a_3$+$b_3$+$a_6$+$c_3$. However, we conclude that only the set of all 1-hcubes is modified. Automatically, the boundary set of all 2-hcubes are standardized and also modified. The set of all 0-hcubes stays always unchanged. The splitting process

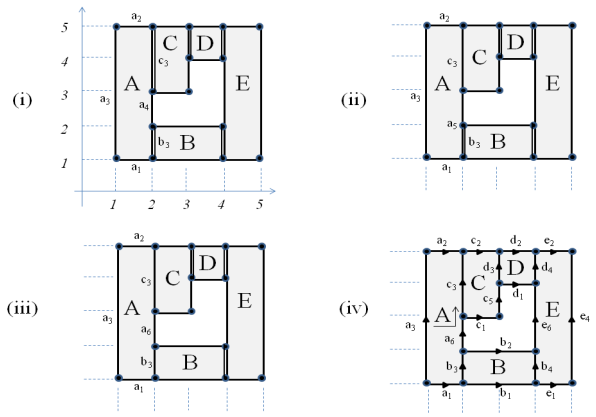



**Fig. 1.** The splitting process performed on invalid summarized complex: (i) an invalid summarized complex, (ii) the splitting process between 1-hcubes $a_4$ and $c_3$ and which gives $c_3$ as a common face, (iii) the splitting process between 1-hcubes $a_5$ and $b_3$ and which gives $b_3$ as a common face, (iv) a valid oriented summarized complex after applying the splitting process.

algorithm accepts an invalid summarized complex as input data and gives a valid summarized complex in the output. The computational complexity of the splitting process is $O(M^2)$ where $M$ is the maximum size of the set of all high cubes of level $q$ for $q$=$n$-1 to 1.

After standardizing the boundaries of the summarized complex by the splitting process in geometrical level, we will now generate the summarized chain complex in algebraically level. Then, define formally a valid summarized complex $\mathcal{K}^s$ constructed after applying the splitting process. We denote $\mathcal{K}^s_q$ is a set of all high cubes of dimension $q$ with an explicit orientation for all $0 \leq q \leq n$. The example in figure 1(iv) gives an oriented summarized 2-complex with a uniform choice of orientations along its boundary edges. By this, we mean that we orient the edges so that the 'to' summit of each edge is the 'from' summit of the next edge. The orientation of a high cube depends directly of the orientation of its faces. In figure 1(iv), we see a 2-hcube $A$ is oriented in anti-clockwise direction and has a same orientation as its 1-faces which are the oriented edges $a_1$, $a_2$, $a_3$, $a_6$, $b_3$ and $c_3$. These oriented edges or 1-faces have been linked by six summit which are not illustrated in the figure for the sake of clarity. Furthermore, the boundary of the $q$-hcube $\delta$ allows us to write the relationship between a $q$-hcube and its $(q-1)$-faces in algebraic form. It is by definition the alternating sum of its oriented $(q-1)$-faces. For example, the boundary of $A$ in figure 1(iv) is $\partial_2(Q)$=$a_1$+$b_3$+$a_6$+$c_3$-$a_2$-$a_3$ where these 1-faces are oriented according to anti-clockwise direction.

To achieve the third step of the labelling task, a chain complex has to be generated from the summarized complex. It is composed of two types of elements groups and homomorphisms between them. Chain groups are naturally generated as linear combinations of high cubes with coefficients in a given abelian group (e.g. $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$). The summarized complex can be written in algebraic form by getting generators from all high cubes of $\mathcal{K}^s_q$ and putting them in the canonical basis $E^s_q$. These generators of $E^s_q$ are given by affecting an unique integer number to each high cube in $\mathcal{K}^s_q$. For example, the complex in figure 1(iv) can generate a summarized chain complex with three canonical bases $E^s_0$, $E^s_1$ and $E^s_2$ where their elements are labels of all high cubes constructed after applying the splitting process. Then, the two 2-hcubes $A$ and $B$ are labelled respectively to $\hat{A}$ and $\hat{B}$ in $E^s_2$, and son on.

Indeed, given a subdivided space $X^s$ in terms of a summarized complex, a $q$-chain $c$ in $X^s$ is a formal sum of integer multiples of elements of $E^s_q$. It is a linear combination of the form $\sum_{i=0}^{N_q} \alpha_i \hat{\delta}_i$ where $\hat{\delta}_i \in E^s_q$ and $\alpha_i$ are integers, and $N_q$ is the number of elements in $E^s_q$. The set of $q$-chains denoted by $C_q(X^s)$ defines a free abelian group with common basis $E^s_q$, such that $C_q(X^s)$=0 if $q > n$ or $q < 0$.

## 3 Homology computation

The homology groups is a primordial topological invariants given by performing the homology computation. Thus, these groups are computed by using the algorithms based on the collapsing process and that are cited in[1] [31] [3] [19]. The homology theory concepts have been illustrated in more detail in these papers, when, the rank of non trivial homology groups $H_0$ and $H_1$ gives respectively the number of connected components and holes in the case of 2D image. Thus, in 3D image, the rank of non trivial homology groups $H_0$, $H_1$ and $H_2$ gives respectively the number of connected components, tunnels and cavities. Here, we focus only on results from applying the homology computation on the summarized complex. We start this section by explaining firstly the reduction or collapsing process on a standard cubical complex. After we illustrate how to use this process on the summarized complex.

### 3.1 Reduction algorithm

The homology groups can be obtained by a sequence of collapsing operations by eliminating successively the cubes of the chain complex such that the homology is preserved at each step. Then instead of computing the homology of the initial chain complex, we compute the homology of a reduced complex because they are equal; i.e. $H(\mathcal{C})\cong\mathcal{C}^f$ where the superscript $f$ indicates that the chain complex is final after a sequence of reduction operations. In the literature, there exists several algorithms for homology computation. Among these, we note that are based on Smith normal form [10][25] and on collapsing process [19] [31]. It should be noted that algorithms which are based on collapsing process are more efficient [31]. Recall that, the collapsing

concept, is originated from the graph theory and the algebraic topology [14] [19]. A detailed description of an example of the collapsing process is found in [3][31]. The collpsing is performed in some iterations in such a way that the homology of the chain complex is always maintained. Formally, let us define $(C, \partial)$ as

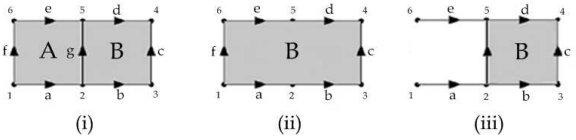



**Fig. 2.** Collapsing process: (i) initial cubical complex of dimension 2, (ii) reduced cubical complex after applying interior face collapsing of $A$ by $g$, (iii) reduced cubical complex after applying exterior face collapsing of $A$ by $f$

a finite generated chain complex of dimension $n$, $E_q$ is a canonical basis of $C_q$ for each $q$, and $\langle \cdot, \cdot \rangle$ is the scalar product associated to that canonical basis. Given $A \in E_q$, $a \in E_{q-1}$ such that $\langle \partial A, a \rangle \neq 0$ , $a$ is a face of $A$ such that $\partial A = \lambda a + r$, and $\lambda = \langle \partial A, a \rangle$ (i.e. $a$ is incident to $A$) where $\lambda \in \{-1, 1\}$ and $r$ is the remaining of the boundary of $A$ with the exception of $a$. When $\lambda$ =1, signify that $a$ is a face of $A$ with the same orientation and when $\lambda$ =-1, signify that $a$ is a face with the opposite orientation.
Now, let us define the reduction in formally manner like in [3][31][1], we define the new boundary of the cube $B$ denoted $\overline{\partial}_q B$ with new bases $\overline{E}_q$ and $\overline{E}_{q-1}$ that are calculated by removing respectively $A$ from $E_q$ and $a$ from $E_{q-1}$.
This means that the boundary map is only updated for cubes of dimension $q$ which have $a$ in their boundaries and for cubes of dimension $(q+1)$ which have $A$ in their boundaries. The boundary map stays unchanged for cubes of dimension $i \notin \{q-1, q\}$. If the second cube $B$ intersects with $A$ and both shares a face $a$, then the collapsing of $A$ by $a$ is an interior face collapsing otherwise it is an exterior face collapsing.

### 3.2 Homology of the summarized complex

Before applying the homology on the summarized complex, a chain complex will be generated by the labelling task. Formally, given a topological space $X^s$ that represents the new subdivision, we define $C(X^s)$ the generated summarized chain complex. By linearity, the boundary operator $\partial_q$ can be extended to $q$-chains. The boundary operator connects two chain groups $C_q(X^s)$ and $C_{q-1}(X^s)$, then, we can obtain $\partial_q : C_q(X^s) \longrightarrow C_{q-1}(X^s)$, such that, the property $\partial_q \circ \partial_{q+1} = 0$ for all $q$ (i.e. the boundary of the boundary of a $q$-chain is null) is satisfied and $\partial_0 = 0$ since $C_{-1}(X^s) = 0$. The $q$-chain groups can be put into a sequence, related by boundary operator $\partial$. This sequence is called a generated free summarized chain complex and denoted by $(C^s, \partial)$. A chain $c \in C_q^s$ is called a $q$-cycle if $\partial_q(c)$=0. If $c = \partial_{q+1}(d)$ for some $d \in C_{q+1}^s$ then $c$ is called a $q$-boundary. Define the $q$-th homology group to be the quotient group of $q$-cycles and $q$-boundaries, denoted by $H_q(X^s)$.

An example of computing the homology of the summarized 2-complex is given in figure 3. Then, figure 3(i) gives the initial summarized complex. In this context, the homology is computed by using the previous reduction algorithm, After applying this algorithm, two homology classes in 19 iterations of face collapsing as shown in figure 4(vii). Theses iterations are repartitioned in five iterations of exterior collapsing of level 2 (see figure 3(ii)), eight iterations of exterior collapsing of level 1 (see figure 3(iii)) and six iterations of interior collapsing of level 1 (see figures from 4(i) to 4(vii)). In the same time of computing the homology, a new structure is used as a stack to store some generators collapsed or have a modified boundary.

Now, we can explain the role and the structure of the stack used during the collapsing process. These generators are stored in the stack in the form of a record structure denoted element. The element structure is composed of: iteration number $\#it$ and a list of subelements. The subelement structure is also composed of two fields: *key* that describes the generator that is collapsed or has a new boundary during an interior face

collapsing and a chain called *elementary cycle chain* that is composed of an alternative sum of generators that gives really the new boundary chain of the generator referenced by the field *key*. Thus, each element of the stack is created at each iteration of interior collapsing. Therefore, because we are interested only on localizing 1-cycles, then we consider uniquely six iterations of interior face collapsing of level 1 and then we get a stack busy with six elements. These elements are illustrated in table 1.

| $\#it$ | subelements |
|---|---|
| 19 | $d_2 \rightarrow d_2$+$c_2$ |
| 18 | $c_2 \rightarrow c_2$-$d_4$ |
| | $d_2 \rightarrow d_2$ |
| 17 | $d_4 \rightarrow d_4$-$c_3$ |
| | $c_2 \rightarrow c_2$ |
| | $d_2 \rightarrow d_2$ |

– – –

| $\#it$ | subelements |
|---|---|
| 16 | $c_3 \rightarrow c_3$-$e_6$ |
| | $d_4 \rightarrow d_4$ |
| | $c_2 \rightarrow c_2$ |
| | $d_2 \rightarrow d_2$ |
| 15 | $e_6 \rightarrow e_6$-$a_6$ |
| | $c_3 \rightarrow c_3$ |
| | $d_4 \rightarrow d_4$ |
| | $c_2 \rightarrow c_2$ |
| | $d_2 \rightarrow d_2$ |

| $\#it$ | subelements |
|---|---|
| 14 | $a_6 \rightarrow a_6$-$b_2$ |
| | $e_6 \rightarrow e_6$ |
| | $c_3 \rightarrow c_3$ |
| | $d_4 \rightarrow d_4$ |
| | $c_2 \rightarrow c_2$ |
| | $d_2 \rightarrow d_2$ |

**Table 1.** Stack with six elements which are constituted of subelements, each subelement is referenced by a key is collapsed or has a modified boundary chain during a six iterations of interior face collapsing from $\#it$=19 to $\#it$=14

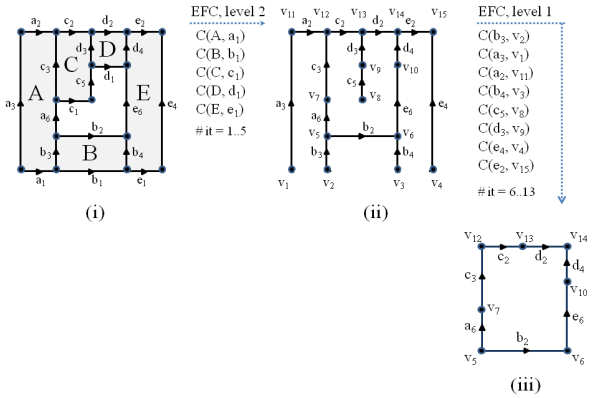



**Fig. 3.** The collapsing process from iteration $\#it$=1 to iteration $\#it$=13: (i) the initial summarized cubical 2-complex with one connected component and one hole, (ii) the reduced complex after a sequence of exterior face collapsing of level two, (iii) the reduced complex after a sequence of exterior face collapsing of level one.

## 4 LOCALIZATION OF CYCLES

The algebraic topology based representation we propose allows to estimate the number of homology groups and to localize them. This localization has become a tool to validate the robustness and consistency of some algorithms such as image matching algorithm applied between two images [8]. It could be also used for object recognition, detection, tracking of moved objects in a sequence of images, ... etc. Recall that the localization has been tackled in the case of 2D images with one object. Among the invariants of interest on which we

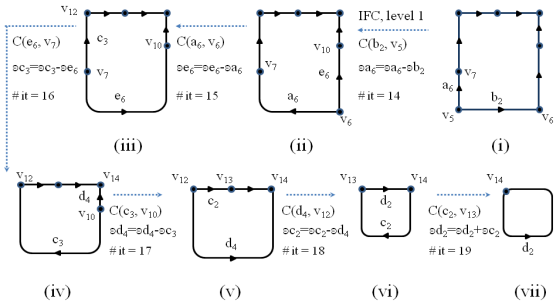



**Fig. 4.** The collapsing process from iteration $\#it$=14 to iteration $\#it$=19: (i) the reduced summarized complex before starting the interior face collapsing of level 1, (ii) interior collapsing of $b_2$ by $v_5$, (iii) interior collapsing of $a_6$ by $v_6$, (iv) interior collapsing of $e_6$ by $v_7$, (v) interior collapsing of $c_3$ by $v_{10}$, (vi) interior collapsing of $d_4$ by $v_{12}$, (vii) interior collapsing of $c_2$ by $v_{13}$ which gives two homology classes: $v_{14}$ for $H_0$ and $d_2$ for $H_1$.

want to apply the localization task, there are the cycles. The localization of these invariants has become an interest topic in computer graphics and image analysis. In this section, the second step of the localization task and an algorithm reserved to localize these invariants is given. This algorithm consists of spotting the cubical generators from a set of generators stored in the stack and which have been collected during the first step which is the homology computation. A subset of these generators are used to form a cycle chain that represents one cubical generator of existing holes, tunnels or cavities. This subset is determined by a method called the reconstruction process. Thus, two factors influence greatly specifically on the running time of this reconstruction process: 1) the cubical generator may have a long cycle chain. 2) The stack that used in this process can be more long. To continue successfully the localization task, the reconstruction process is applied on stack of generators.

**Reconstruction process** The localization task is achieved by reconstructing the cycle chains of their homology groups. In the last subsection, we demonstrate the first step of the localization task which is the homology computation algorithm . Unfortunately these algebraic invariants computed during this step alone are not sufficient to localize the homology groups. In order to achieve this localization very easily, we must get the cycle chains by reading the chains saved in the field *elementary cycle chain* of the stack's structure. Then, the reading is achieved from the summit to the bottom, i.e. return from the final complex to the reduced complex before applying the first interior face collapsing. Figures 4 shows an example of the reconstruction with a stack that contains six elements (i.e. there are six iterations of interior face collapsing). Thus, figures 4(i) to 4(vii) give the reduced complex during these six iterations. Figure 4(vii) illustrates the last iteration of collapsing where two homology classes are computed: $v_{14}$ for the group $H_0$ and $d_2$ for the group $H_1$. Then, in this example, we interest only of the computation of the cycle chain that represents the homology class $d_2$. Firstly, we initialize an empty chain $chz$ by a unique generator $d_2$ and after we read all elements in the stack from the summit to the bottom and replace only the generators of the chain $chz$ by the chain *elementary cycle chain*, such that there exist at least a matching between the key generator of the stack that represents the chain *elementary cycle chain* and a generator in the chain $chz$.

Now, we describe in detail the reconstruction process and show how to read the value *elementary cycle chain* referenced by the value *key* from iteration $\#it$=19 to iteration $\#it$=14.
- at $\#it$=19, the stack's summit contains only one key $d_2$, then, we take the value *elementary cycle chain* of this key which matches with the unique generator that includes in $chz$. Because $chz$=$+d_2$, and at this iteration, $\partial_1^{19} d_2$=$\partial_1^{18}(d_2 + c_2)$. Thus, the new value of cycle chain $chz$ is $+d_2 + c_2$.
- at $\#it$=18, the stack's summit contains two keys $d_2$ and $c_2$, then, we take the value *elementary cycle chain* of these keys which match with the generators that includes in $chz$. Because $chz$=$+d_2 + c_2$, and at this

iteration, $\partial_1^{18} d_2 = \partial_1^{17}(d_2 + c_2)$ and $\partial_1^{18} c_2 = \partial_1^{17}(c_2 - d_4)$. Thus, the new value of cycle chain $chz$ is $+d_2 + c_2 - d_4$. We repeat the process until we reach the element of the stack referenced by $\#it$=14. In this case, the summit of the stack contains six subelements that associated to six keys $a_6$, $e_6$, $c_3$, $d_4$, $c_2$ and $d_2$. Then, we take only the chain *elementary cycle chain* of these keys which match correctly to generators that include in the chain $chz$ which is evaluated in the last iteration. Thus, at this iteration, $\partial_1^{14} a_6 = \partial_1^{13}(a_6 - b_2)$, $\partial_1^{14} e_6 = \partial_1^{13} e_6$, $\partial_1^{14} c_3 = \partial_1^{13} c_3$, $\partial_1^{14} d_4 = \partial_1^{13} d_4$, $\partial_1^{14} c_2 = \partial_1^{13} c_2$ and $\partial_1^{14} d_2 = \partial_1^{13} d_2$. However, the new value of cycle chain $chz$ is $+d_2 + c_2 - d_4 + c_3 - e_6 + a_6 - b_2$. Indeed, the superscript in boundary operator denotes the iteration number $\#it$ whereas the subscript is the group of the chain complex on which the operator is applied. Note that $\partial_1^0$ indicates the initial state of the boundary map before starting the collapsing process. Note that the computational complexity of the reconstruction process is $O(npm)$, where $n$ is size of the stack, $p$ is the element's size in the stack and $m$ is the maximum length of the result cycle chain. For comparison, in the case of the standard cubical complex $n$, $p$ and $m$ are greater than the same values when applying the summarized complex.

## 5 EXPERIMENTAL RESULTS

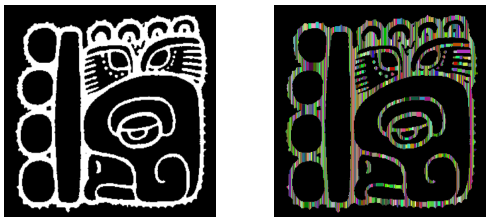
The proposed algorithm has been validated on various synthetic images on Intel Core i3-2310M 2.10 Ghz with 6 Go of RAM and Eclipse IDE with Java J2SE 1.5. We will present firstly the topological invariant localization in the case of 2D images. Figure 5(a) shows a 512x512 gray scale image. Figure 5(b) shows the visualization in color image of the summarized complex which gives 2136 high cubes. The result of homology computation is shown in the table 2. Figures 6(a) and 6(b) describe respectively the cubical generator of the first hole of the image given in figure 5(a) and a small window where the localization of this hole is more visualized. Then, figures 7(a) and 7(b) present respectively the cubical generator of a second hole of figure 5(a) and a small window where the localization of this hole is more visualized. The last table confirms

| dimension | # hcubes | homology |
|---|---|---|
| dim 0 | 7187 *0-hcubes* | $\mathcal{H}_0 = Z^{12}$ |
| dim 1 | 9341 *1-hcubes* | $\mathcal{H}_1 = Z^{30}$ |
| dim 2 | 2136 *2-hcubes* | $\mathcal{H}_2 = 0$ |

**Table 2.** The homology computation by dimension of the summarized complex of 2D image

that the using of the summarized complex with 2136 2-hcubes gives a profit in time consuming and memory resources than using the standard cubical complex with 64248 2-cubes, effectively when we perform the homology computation and the localization task. As result, the running of these processes on 30 holes requires one hour, 44 minutes and 42 seconds in the case of the standard cubical complex. Then, the same processes require only nine minutes and 51 seconds when applying the summarized complex. Now, we present the topological invariant localization in the case of 3D images. We start firstly by localizing the cavities in these images. Figures 8(a), 8(b) and 8(c) describe respectively the initial 3D image, a vertical and a horizontal sections or cuts of this image. The last two figures prove the existence a cavity in the initial image. The image in 8(a) conducts to generate an standard cubical complex with 22920 cubes and a summarized complex with 336 high cubes after applying the compacting and splitting processes. The repartition of high cubes and the homology computation by dimension are given respectively in table 3 for the summarized complexes. Figures 9(a) and 9(b) describe respectively the 3D cubical generator of the unique cavity of the 3D image that is given in figure 8(a) in the standard and the summarized complexes. The homology computation and the localization task require 13 minutes and 48 seconds when applying the standard cubical complex. Then, the same processes require only two seconds when applying the summarized complex.

Secondly, we present also the localization of tunnels in 3D images. Figure 10(a) describes the initial 3D image to treat by the above algorithm. The image in 10(a) conducts to generate an standard cubical complex



(a) (b)

**Fig. 5.** (a) Initial binary image after thresholding process which gives 64248 pixels or cubes, (b) the visualization in color image of the summarized complex which gives 2136 high cubes.

(a) (b)

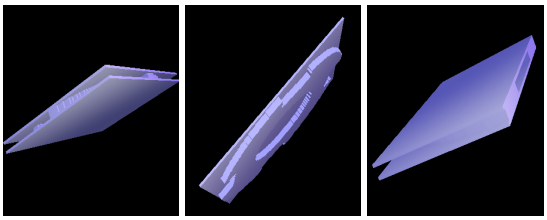
**Fig. 6.** (a) Localization of the first hole by representing its cubical generator en red color, (b) Small window of the image where the localization of this hole is more visualized.

(a) (b)

**Fig. 7.** (a) Localization of the second hole by representing its cubical generator en red color, (b) Small window of the image where the localization of this hole is more visualized.

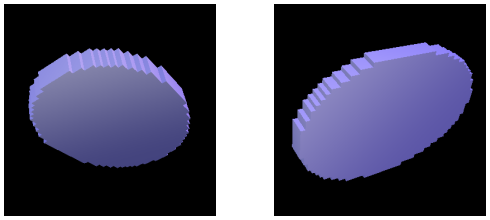
| dimension | # hcubes | homology |
|---|---|---|
| dim 0 | 1691 *0-hcubes* | $\mathcal{H}_0 = Z^1$ |
| dim 1 | 3550 *1-hcubes* | $\mathcal{H}_1 = 0$ |
| dim 2 | 2197 *2-hcubes* | $\mathcal{H}_2 = Z^1$ |
| dim 3 | 336 *2-hcubes* | $\mathcal{H}_3 = 0$ |

**Table 3.** The homology computation by dimension of the summarized complex of 3D image



(a) (b) (c)

**Fig. 8.** (a) Initial 3D image which composed of 22920 cubes, (b) Horizontal section or cut of the initial 3D image, (c) Vertical section or cut of the initial 3D image.



(a) (b)

**Fig. 9.** Localization of the unique cavity (a) by representing its 3D cubical generator in the standard cubical complex, (b) by representing its 3D cubical generator in the summarized complex.

with 3086 cubes and a summarized complex with 93 high cubes after applying the compacting and splitting processes. The repartition of cubes and the homology computation by dimension are given in table 4 for the summarized complex.

Figures 10(b) and 10(c) describe respectively the 3D cubical generator of the unique tunnel of the 3D image

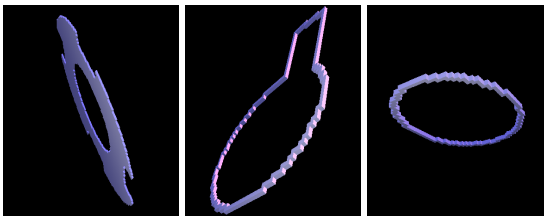
| dimension | # hcubes | homology |
|---|---|---|
| dim 0 | 662 *0-hcubes* | $\mathcal{H}_0 = Z^1$ |
| dim 1 | 1179 *1-hcubes* | $\mathcal{H}_1 = 0$ |
| dim 2 | 610 *2-hcubes* | $\mathcal{H}_2 = Z^1$ |
| dim 3 | 93 *2-hcubes* | $\mathcal{H}_3 = 0$ |

**Table 4.** The homology computation by dimension of the summarized complex of 3D image

that is given in figure 10(a) in the standard and summarized complexes. The homology computation and the localization task require 15 seconds when applying the standard cubical complex. Then, the same processes require only one second when applying the summarized complex.

## 6 Conclusion

In this paper a new approach for localizing the topological invariants on using the cubical homology theory is proposed. This localization adopts an algorithm of two successive stages, the first one computes the homology classes, then the second stage is devoted to localize holes, tunnels and cavities by identifying their cubical generators. In order to reduce the computational complexity of this algorithm, the homology computation



(a) (b) (c)

**Fig. 10.** (a) Initial 3D image which composed of 3086 cubes, (b) Localization of the unique tunnel by representing its 3D cubical generator in the standard cubical complex, (c) Localization of the unique tunnel in the summarized complex.

is applied on summarized complex followed by the reconstruction process that treats really the localization task. This algorithm consists of localizing three types of invariants: holes, tunnels and cavities using the new structure of the summarized complex before computing the cycle chain of the corresponding cubical generator. The localization of homology cycles concerns a cubical generator for holes when using 2-dimensional images and for tunnels and cavities when using 3-dimensional images. This localization makes possible the images to be visualized through significant information. The experimental results were given on both 2D and 3D images.

## References


1. M. Allili and D. Ziou. Computational Homology Approach for Topology Descriptors and Shape Representation. In Proceedings of International Conference on Image and Signal Processing (ICISP2003), pages 508-516, Agadir, Morocco, June 2003.
2. M. Allili, K. Mischaikow, and A. Tannenbaum. Cubical Homology and the Topological Classification of 2d and 3d Imagery. In IEEE International Conference on Image Processing, pages 173-176, 2001.
3. M. Allili and D. Ziou. Topological Feature Extraction in Binary Images. In Proceedings of the 6th IEEE International Symposium on Signal Processing and its Applications, pages 651-654, Malaysia, 2001.
4. M.-F. Auclair-Fortier, P. Poulin, D. Ziou, and M. Allili. A Computational Algebraic Topology Model for the Deformation of Curves. In Proceedings of 2nd International Workwhop on Articulated Motion and Deformable Objects (AMDO 2002), pages 56-67, 2002.
5. M.-F. Auclair-Fortier, D. Ziou, and M. Allili. A Global Cat Approach for Graylevel Diffusion. In 7th IEEE International Symposium on Signal Processing and its Applications (ISSPA 2003), pages 453-456, 2003.
6. R.J. Campbell and P.J. Flynn. A Survey of Free-Form Object Representation and Recognition Techniques. Computer Vision and Image Understanding, 81:166-210, 2001.
7. Hui Chen and Bir Bhanu. 3d Free Form Object Recognition In Range Images Using Local Surface Patches. Pattern Recognition Letters, 28(10):1252-1262, 2007.
8. S. Derdar and M. Allili and and D. Ziou. Image Matching Using Algebraic Topology. In Proceedings of the IS&T / SPIE 18th Annual Symposium on Electronic Imaging, volume 6066, San Jose, California, USA, January 2006.
9. S. Derdar and M. Allili and D. Ziou. Topological Feature Extraction Using Algebraic Topology. In Proc. SPIE–IS&T Electronic Imaging , Vision Geometry XV, volume 6499, San Jose, California, USA, January 2007.
10. J.G. Dumas and F. Heckenbach and B. D. Saunders and V. Welker. Computing Simplicial Homology Based on Efficient Smith Normal Form Algorithms.
11. R. Egli and N.F. Stewart. A Framework for System Specification Using Chains on Cell Complexes. Computer-Aided Design, 31(11):669-681, 1999.
12. T. Fabry and D. Smeets and and D. Vandermeulen. Surface Representations for 3D Face Recognition. In Milos Oravec, editor, Face Recognition, chapter 15. ISBN: 978–953–307–060–5, InTech, 2010.
13. R. Ghrist. Barcodes: The Persistent Topology of TData. Bulletin of the American Mathematical Society, 45:61-75, 2008.
14. P. J. Giblin. Graphs, Surfaces and Homology. Chapman and Hall, 1977.

15. S. Haker and G. Sapiro and A. Tannenbaum. Knowledge-Based Segmentation of Sas Data With Learned Priors. IEEE Transactions on Image Processing, 9:298-302, 2000.
16. S. Haker and G. Sapiro and A. Tannenbaum and D. Wasburn. Missile Tracking Using Knowledge-Based Adaptive Thresholding. In Proceedings of the International Conference on Image Processing, pages 786-789. IEEE, 2001.
17. H. Hoppe and T. DeRose and T. Duchamp and J. McDonald and W. Stuetzle. Surface Reconstruction From Unorganized Points. In Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pages 71-78, New York, NY, USA, 1992. ACM, SIGGRAPH 92.
18. A. Johnson and M. Hebert. Using Spin Images For Efficient Object Recognition In Cluttered 3D Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5):433-449, 1999.
19. T. Kaczynski and M. Mrozek and M. Slusarek. Homology Computation by Reduction of Chain Complexes. Computers and Math. Appl., 34(4):59-70, 1998.
20. I. Karaka and O. Ege. Some Results On Simplicial Homology Groups of 2D Digital Images. International Journal of Information and Computer Science, 1(8):198-203, 2012.
21. J. J. Koenderink and A. J. Van Doorn. Internal Representation of Solid Shape With Respect to Vision. Biological Cybernetics, 32(4):211-216, 1979.
22. T.Y. Kong and A. Rosenfeld. Digital Topology: Introduction and Survey. CVGIP, 48:357-393, 1989.
23. D. Lowe. Fitting Parameterized 3D Models to Images. IEEE Transactions On Pattern Analysis and Machine Intelligence, 13:441-450, 1991.
24. M. Niethammer and A.N. Stein and W.D. Kalies and P. Pilarczyk and K. Mischaikow and A. Tannenbaum. Analysis of Blood Vessel Topology by Cubical Homology. In Proceedings of the International Conference on Image Processing (2002), pages 969-972. IEEE, 2002.
25. S. Peltier and A. Ion and Y. Haxhimusa and W. Kropatsch. Computing Homology Group Generators of Images Using Irregular Graph Pyramids. Technical report priptr-111, Vienna University of Technology, Faculty of Informatics, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, 2007.
26. S. Peltier and A. Ion and W. G. Kropatsch and G. Damiand and Y. Haxhimusa. Directly Computing the Generators of Image Homology Using Graph Pyramids. Image and Vision Computing, 27(7):846-853, 2009.
27. A. Rosenfeld and A.C. Kak. Digital, Pictures and Processing. Academic Press, 1982.
28. Chin–seng Chua and Ray Jarvis. Point Signatures: A New Representation For 3D Object Recognition. International Journal of Computer Vision - IJCV, 25(1):63-85, 1997.
29. F. Stein and G. Medioni. Structural Indexing: Efficient 3–D Object Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):125-145, 1992.
30. S. Ulman and R. Basri. Recognition by Linear Combinations of Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(10):992-1006, 1991.
31. D. Ziou and M. Allili. Generating Cubical Complexes from Image Data and Computation of the Euler Number. Pattern Recognition, 35(12):2833-2839, 2002.
32. D. Ziou and M. Allili. Image Model: A New Perspective for Processing Images. In Proceedings of the IS&T / SPIE 16th International Symposium on Electronic Imaging, pages 123-133, San Jose, California, USA, January 2004. Science and Technology.
33. A. Zomorodian and G. Carlsson. Localized Homology. Computational Geometry: Theory and Applications archive, 41(3):126-148, November 2008.
34. R. Gonzlez-Daz and B. Medrano and P. Real and J. Snchez-Pelez. Algebraic Topological Analysis of Time-Sequence of Digital Images. Lecture Notes in Computer Science, 45:208219, 2005. 34
35. S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. JACM, vol. 23, pp. 368–388, April, 1976.
36. Y. Deng and B.S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. IEEE Trans. Pattern Anal. Machine Intell., vol. 23, no. 8, pp. 800–810, Aug. 2001.
37. G. M. Hunter and K. Steiglitz. Operations on images using quad trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2, April 1979: 145–154.
38. J. Canny. A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, Nov. 1986.
39. R. Gonzalez and R. Woods. Digital Image Processing, Addison-Wesley Publishing Company, 1992.
40. R. M. Haralick and L. G. Shapiro. Computer and Robot Vision, Addison-Wesley. Co, New York, ISBN: 0201569434, 1993.

# Une interface 3D pour OLAP en réalité virtuelle

Sébastien Lafon[1], Fatma Bouali[2 1], Christiane Guinot[1], Gilles Venturini[1]

[1] Université François-Rabelais de Tours, Laboratoire d'Informatique
[2] Université de Lille2, IUT, Dpt STID

**Résumé** Nous présentons dans cet article une nouvelle interface visuelle et interactive pour explorer des cubes OLAP en réalité virtuelle. En premier lieu nous introduirons un état de l'art des visualisations en 3D de cubes OLAP où nous détaillons leurs avantages et leurs points faibles. Ensuite, nous détaillons notre approche, avec en particulier la représentation de plusieurs mesures et l'utilisation d'opérateurs OLAP directement dans la représentation 3D. Enfin nous exposons les résultats d'une évaluation utilisateur sur un ensemble de tâches ainsi que les conclusions qui en ont été tirées.

## 1 Introduction

OLAP (Online Analytical Processing), décrit notamment dans [1] [2], est un ensemble d'outils permettant de réaliser une analyse multidimensionnelle de données volumineuses [3]. Pour cela les données sont représentées selon plusieurs dimensions, chacune d'entres elles étant divisée en membres. Ces données sont généralement symbolisées par un cube ou hypercube OLAP subdivisé en cellules représentant une mesure au croisement de chaque dimension. OLAP met à disposition plusieurs opérateurs permettant d'interagir avec l'hypercube pour pouvoir ainsi préciser l'analyse. Ces opérateurs permettent de modifier l'apparence de l'hypercube pour l'adapter à ce que l'on souhaite visualiser, de naviguer à travers les hiérarchies des dimensions pour afficher plus ou moins de détails, et d'extraire des données utiles [4].

La majorité des représentations OLAP se font à l'aide de tableaux croisés dynamiques. En effet, comme précisé dans [1], avant OLAP la majorité des outils de visualisation de données utilisaient des tableaux, les utilisateurs sont donc habitués à ce type de représentation. C'est pourquoi beaucoup d'outils OLAP utilisent des tableaux croisés auxquels sont rajoutées des fonctionnalités spécifiques à OLAP (opérateurs, ...). Cependant cette représentation sous forme de tableaux n'est pas adaptée pour les données multidimensionnelles. Comme décrit dans [5], le problème de représentation de ce type de données vient justement de leur caractère multidimensionnel. En effet, OLAP permet d'effectuer une analyse en observant une ou plusieurs mesures représentées selon une ou plusieurs dimensions. La visualisation à l'aide de tableaux est adaptée lorsque les mesures sont définies par une ou deux dimensions. Dès que ce nombre de dimensions est supérieur, il est nécessaire de regrouper des dimensions sur les lignes ou les colonnes du tableau, ce qui rend l'analyse beaucoup plus complexe lorsque les données sont volumineuses.

D'autres interfaces ont donc été développées pour OLAP, et notre objectif est de contribuer à ce domaine en nous concentrant plus spécialement sur des visualisations 3D utilisant la réalité virtuelle (écran stéréoscopique, matériel d'interaction) à l'instar des travaux de [6]. Nous avons cherché à savoir notamment si les développements matériels récents dans le domaine de la 3D stéréoscopique peuvent contribuer à ce type d'interfaces. La suite de cet article est organisée comme suit : la section 2 introduit un état de l'art des visualisations en 3D de cubes OLAP. La section 3 présente la visualisation OLAP que nous proposons, VR4OLAP. La section 4 aborde l'évaluation utilisateur réalisée pour tester notre application. Enfin la section 5 conclut sur les perspectives de ce travail.

## 2 Etat de l'art

Il existe plusieurs visualisations de cubes OLAP en dehors des tableaux croisés (voir par exemple [7]), et nous nous intéressons ici à celles mettant en oeuvre des visualisations.

Dans [8] est décrit le système DBMiner qui propose un large panel de fonctions de fouille de données, dont OLAP. Les données OLAP dans DBMiner sont affichées en 3D dans une représentation à 3 axes avec des cubes de tailles différentes espacés les uns des autres afin de permettre une meilleure lisibilité des données se trouvant au centre du cube. Les membres de chaque dimension sont également affichés dans la visualisation. On peut choisir les dimensions sur chaque axe et jusqu'à deux mesures via une interface extérieure à la visualisation. L'une des deux mesures sera représentée par la couleur et l'autre par la taille du cube affiché. Lorsque la souris reste assez longtemps sur un cube, les membres associés changent de couleur pour permettre à l'utilisateur de les situer, et une légende avec les valeurs du cube apparaît en haut de la visualisation. Si un clic est effectué sur un cube, un panneau apparaît détaillant les valeurs du cube. Les opérateurs OLAP disponibles sont le roll-up, le drill-down, le slice, et le dice, exécutables via une fenêtre externe à la visualisation.

Dans [6], un logiciel de visualisation de données est introduit et celui-ci a la particularité de présenter les données dans un environnement en réalité virtuelle. Ce système se nomme DIVE-ON (Datamining in an Immersed Virtual Environment Over a Network) et permet d'immerger l'utilisateur dans la visualisation grâce à la projection de celle-ci tout autour de l'utilisateur en stéréoscopie. Cet environnement de réalité virtuelle utilisé s'appelle CAVE (Cave Automatic Virtual Environment), il place ainsi l'utilisateur entre 3 écrans (d'environs trois mètres sur trois) disposé en face de lui, à sa gauche et à sa droite. La réalité virtuelle permet à l'utilisateur d'avoir une activité sensori-motrice et cognitive dans un monde artificiel, ici c'est le déplacement dans le cube 3D et la possibilité d'exécuter des opérations pour modifier le cube. Ces actions se font à l'aide d'un gant permettant de suivre les mouvements de l'utilisateur, et d'un casque qui permet au logiciel de savoir où l'utilisateur regarde. Les données sont représentées sous forme de cubes ou de sphères espacés, de tailles et de couleurs différentes suivant les valeurs ou la nature des données. Une grille représentant

la structure externe du cube est représentée afin de ne pas perdre l'orientation des données. Pour utiliser les opérateurs OLAP, l'utilisateur dispose d'un menu latéral à partir duquel il peut effectuer les opérateurs drill-down, roll-up, slice et dice. Ce menu latéral apparaît lorsque l'utilisateur appuie sur l'un des boutons du gant. De plus, grâce au casque que ce dernier porte, le menu est toujours affiché face à lui. L'utilisateur peut également pointer un objet et afficher grâce à un second bouton sur le gant une boîte de dialogue affichant des informations sur l'objet pointé. Un troisième bouton sur le gant permet de se déplacer dans la visualisation selon deux modes de déplacement : un mode fournissant une carte montrant à l'utilisateur où il se trouve et lui permettant de cibler sa destination, et un mode permettant à l'utilisateur de pointer directement dans la visualisation l'endroit qu'il veut atteindre.

Dans [9], l'outils DIVA (Data warehouse Interface for Visual Analysis) est présenté. Cet outil est dédié à la visualisation et l'analyse OLAP. DIVA est intégré à une interface Web, son but étant de fournir à l'utilisateur une interface légère et simple pour exécuter les requêtes et opérateurs OLAP de manière transparente. L'avantage de cette visualisation en 3D est que l'on peut se déplacer librement dans la scène pour analyser les données sous n'importe quel angle de vue. Les valeurs sont inscrites sur les cubes et l'effet de transparence sur les cubes permet une meilleure analyse. Cependant, l'intérieur du cube est invisible et la couleur des cubes est insignifiante. On peut noter l'effet de plan qui permet de rappeler l'orientation du cube. Les opérateurs OLAP disponibles sont le drill-down, le roll-up, le slice et le dice. Alors que les opérateurs slice et dice sont intégrés à la visualisation, le drill-down et le roll-up sont exécutés via un panneau en dehors de la visualisation. Ainsi par exemple, pour effectuer un roll-up, il faut choisir la dimension sur laquelle on veut agréger les données. Une fois la dimension choisie, dans la liste "FROM" du panneau on sélectionne de quelle hiérarchie on part, et on indique dans la liste "TO" le niveau de la hiérarchie que l'on souhaite atteindre. Il reste ensuite à cliquer sur le bouton "Roll Up" pour valider. L'exécution des opérateurs roll-up et drill-down n'est donc pas des plus intuitives.

Une autre solution intéressante est Miner3D [10], ce logiciel propriétaire permet de visualiser en 3D les données d'un dataWarehouse. Le cube peut être construit selon différentes visualisations : une visualisation bars chart, qui représente les données sous forme de rectangles dont la taille varie en fonction de la valeur de la mesure, une visualisation représentant les données sous forme de cubes ou sous forme de sphères dont la taille et la couleur varient en fonction de la valeur,... Il est possible d'afficher jusqu'à 5 dimensions et de visualiser dynamiquement les mesures au cours d'une période. Le logiciel va ainsi afficher à intervalle régulier le cube dont les valeurs des mesures correspondent à l'intervalle actuel. Lorsque l'on passe la souris sur une donnée, la valeur de la mesure correspondante s'affiche. Il est possible d'exécuter les opérateurs drill-down, roll-up, slice et dice.

Pour synthétiser, dans les visualisations présentées ici, il n'y en a aucune qui implémente beaucoup d'opérateurs OLAP, on trouve le plus souvent le slice, le

dice et le drill-down. La mise en place de plus d'opérateurs permettrait à l'utilisateur de mieux naviguer dans le cube. De plus les opérateurs gagneraient en intuitivité s'ils étaient directement incorporés dans la visualisation, comme par exemple l'opérateur slice dans l'outil DIVA. Concernant les membres, toutes les visualisations présentées les affichent ce qui permet à l'utilisateur de savoir facilement à quoi correspondent les données qu'il analyse. Il est également intéressant d'afficher les valeurs des mesures lorsque l'on clique sur une donnée comme dans la plupart des visualisations, pour avoir ainsi la valeur précise des données qu'on observe. Dans les outils DBMiner et DIVE-ON, les données affichées sont espacées pour éviter que celles au premier plan n'occluent pas celle aux plans suivants. Enfin, l'affichage en stéréoscopie utilisé dans DIVE-ON est intéressant car il permet à l'utilisateur de mieux percevoir la profondeur. Enfin, à notre connaissance, aucune évaluation utilisateur n'a été réalisée sur ces visualisations pour valider l'efficacité de celles-ci.

Toutes ces remarques nous ont aidées à concevoir une nouvelle visualisation, VR4OLAP (Virtual Reality for OLAP), afin de proposer à l'utilisateur un environnement 3D et interactif le plus complet possible pour explorer des données OLAP mais aussi les présenter à d'autres personnes.

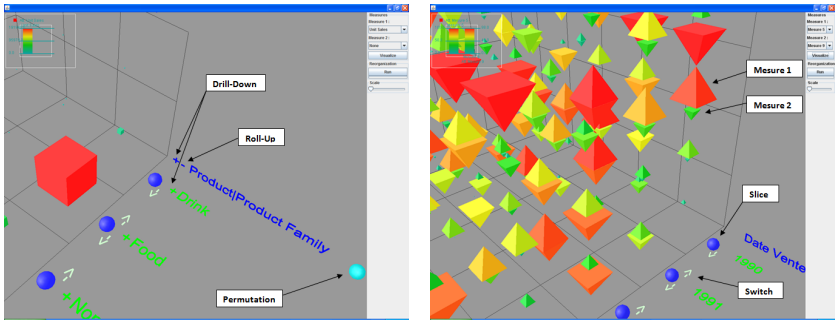
## 3 Visualisation OLAP proposée : VR4OLAP

### 3.1 Choix des données à visualiser

Les données à visualiser dans notre outil sont gérées à l'aide du serveur OLAP Mondrian, un serveur Open Source se présentant sous la forme d'une librairie Java. Mondrian fait partie de la catégorie des serveurs R-OLAP, c'est-à-dire qu'il permet d'accéder à des données contenues dans une base de données relationnelle classique qui est structurée pour réagir comme une base OLAP. Il exécute des requêtes écrites avec le langage MDX pour récupérer les données. Pour fonctionner, il faut fournir au serveur OLAP Mondrian un fichier XML décrivant le schéma multidimensionnel de la base de données sur laquelle le serveur se connecte. C'est dans ce fichier que sont définis les différents cubes disponibles, les dimensions associées ainsi que leur hiérarchie, et les mesures pouvant être visualisées. Mondrian permet de se connecter à de nombreuses bases de données différentes. Actuellement notre application permet de se connecter à des bases Access et MySQL. Une fois la connexion à la base réussie, la fenêtre de choix du cube apparaît sous la forme d'une interface 2D. Cette fenêtre permet à l'utilisateur de choisir les données qui seront affichées en 3D (cube choisi, dimensions pour ce cube et une ou deux mesures).

### 3.2 Définition de la visualisation 3D

La visualisation 3D est définie de la manière suivante (voir figure 1). Trois axes sont utilisés pour représenter les trois dimensions choisies. Chacun de ces axes porte le nom de la dimension qui lui correspond, ainsi que les noms des




(a) (b)

**Figure 1.** Visualisation de une (a) ou deux (b) mesures

membres de cette dimension. Les membres affichés pour une dimension dépendent des opérations de développement réalisées sur cette dimension (voir la section suivante). Ensuite, si une seule mesure a été sélectionnée, les valeurs de cette mesure sont représentées par un cube dont la taille et la couleur dépendent de la mesure. Si deux mesures ont été choisies, alors chaque cellule du cube est représentée par un pyramidion (voir figure 1(b)). Cet objet 3D est composé de deux pyramides dont la hauteur et la couleur dépendent de chacune des valeurs des deux mesures choisies. En plus de ces éléments, une grille est affichée pour aider l'utilisateur à mieux situer les données par rapport aux membres des dimensions. Mentionnons également qu'il est possible, via une fenêtre 2D externe à la visualisation, de changer toutes les couleurs de la scène 3D, ce qui permet à l'utilisateur de personnaliser la visualisation de ses données. L'utilisateur peut mettre en route ou non l'affichage stéréoscopique, et d'un point de vue matériel, l'environnement de visualisation stéréoscopique peut être un grand écran 3D immersif ou bien un écran LCD 3D.

### 3.3 Interactions

Nous avons ensuite représenté les opérateurs OLAP et autres interactions dans cette visualisation à l'aide de signes visuels supplémentaires sur lesquels l'utilisateur peut directement interagir (voir figure 1). Suivant l'interaction sélectionnée, le système génère la requête MDX correspondante et regénère dynamiquement un nouveau cube.

Une des interactions les plus simples consiste, lorsque l'on clique sur une donnée avec la souris, à faire apparaître une fenêtre "pop-up" qui affiche la valeur de la mesure et les membres correspondants à la donnée cliquée. Ensuite viennent les opérateurs OLAP. L'opérateur drill-down est représenté dans la visualisation par le symbole "+". Ce symbole se trouve sous les noms des membres et sous le nom des dimensions. Lorsque l'on clique sur un "+" se trouvant sous un membre, un drill-down est effectué sur ce membre et lorsque l'on clique sur un

"+" sous le nom d'une dimension, alors un drill-down est effectué sur chaque membre de cette dimension. De plus, lorsqu'un membre ou une dimension ne permet pas de faire un drill-down alors le symbole est masqué. De manière similaire, l'opérateur roll-up est représenté par un symbole "-" sous les noms des dimensions. Lorsque l'on clique dessus, on remonte d'un cran dans la hiérarchie de la dimension correspondante. De même, lorsqu'une dimension ne permet pas de faire un roll-up alors le symbole est masqué. Les opérateurs de sélection Slice et Dice sont représentés par des sphères se trouvant à côté des noms des membres des axes. Lorsque l'on clique dessus, seules sont affichées les données liées au membre correspondant à la sphère cliquée. Les autres données sont masquées, cependant les noms des autres membres du même axe et les sphères correspondantes sont toujours affichés pour que l'utilisateur puisse également les sélectionner s'il le souhaite. Pour désélectionner un membre, il suffit de recliquer sur la sphère correspondante. L'opérateur de permutation switch est représenté dans la visualisation par les symboles "-¿" et "¡-" à côté des noms des membres. Lorsque l'on clique dessus, le membre correspondant est permutée avec sa voisine de gauche ou de droite (selon le sens de la flèche). Il est également possible de permuter deux dimensions en cliquant sur la sphère se trouvant à côté du nom de la troisième dimension.

D'autres interactions sont proposées à l'utilisateur. A droite de la visualisation se trouve un bandeau à partir duquel l'utilisateur peut effectuer différentes opérations. Il peut tout d'abord changer les mesures affichées, et le système répond dynamiquement en modifiant la visualisation. L'utilisateur dispose d'un curseur pour faire varier dynamiquement la taille des cubes ou pyramidions afin de mieux les voir s'ils sont trop petits. Ensuite, l'utilisateur peut lancer un algorithme de réorganisation. En effet, dans nos travaux précédents [11] nous nous sommes intéressés à la réorganisation d'une dimension afin de placer côte à côte des membres ayant des valeurs de mesures similaires. Cet algorithme était cependant limité à une réorganisation linéaire des dimensions et ne pouvait tenir compte d'éventuelles hiérarchies. Le nouvel algorithme utilisé dans VR4OLAP (qui par manque de place ne sera pas décrit ici, voir [12]) se sert d'un algorithme génétique pour réorganiser les membres de chaque dimension tout en respectant leurs hiérarchies. Il réalise donc des permutations d'arbres et de sous arbres afin de maximiser une mesure de lisibilité du cube. Plusieurs versions de cet algorithme sont utilisables (réorganiser la hiérarchie telle qu'elle est affichée et développée, réorganiser toute la hiérarchie, réorganiser niveau par niveau en agrégeant les mesures). Son exécution est incrémentale, par pas durant moins de 1 minute. L'utilisateur peut donc cliquer une première fois pour obtenir un résultat et donc un nouveau cube, et s'il souhaite continuer et améliorer ce résultat, il peut cliquer à nouveau. Pendant l'exécution de l'algorithme, il peut continuer à utiliser la visualisation. Ainsi, l'utilisation de l'algorithme génétique ne vient pas rallonger outre mesure l'utilisateur dans son exploration du cube. Les résultats sont visuellement très intéressants (voir l'évaluation utilisateur). Enfin, les déplacements dans la visualisation se font soit à l'aide du clavier et de la souris, soit à l'aide d'un SpacePilot (souris à 6 degrés de liberté).

## 4 Evaluation utilisateur

### 4.1 Présentation

Afin de cerner l'efficacité et les voies d'amélioration de notre visualisation, et aussi de la comparer à une approche concurrente, nous avons réalisé une évaluation utilisateur selon le protocole suivant. Tout d'abord nous avons recruté des utilisateurs/testeurs ayant déjà des connaissances en OLAP. 7 utilisateurs ont ainsi été choisis au département STID (STatistique et Informatique Décisionnelle) de l'IUT de l'Université de Lille 2. Nous avons sélectionné alors différentes interfaces à tester : VR4OLAP en 3D-monoscopique, VR4OLAP en 3D-stéréoscopique, VR4OLAP en 3D-stéréoscopique avec réorganisation lorsque cette dernière a un sens, et une méthode classique appelée JPivot. Cet outil 2D est un logiciel open-source disposant d'une interface web et représentant les cubes OLAP à l'aide de tableaux croisés dynamiques. JPivot permet d'effectuer de façon interactive plusieurs opérateurs OLAP (drill-down, roll-up, slice, dice, ...). Il affiche la valeur des mesures directement sous la forme d'un nombre.

Nous avons défini un ensemble de questions/tâches représentatives afin de cerner les points forts et points faibles de ces visualisations en termes de fouille de données. Voici les questions qui ont été posées :

- Q1 : trouver une valeur du cube pour des attributs donnés. Pour cela, l'utilisateur partira d'un cube non développé et devra pour aller chercher une valeur cible en utilisant les opérateurs drill-down et roll-up.
- Q2 : afficher un certain cube à partir d'un autre le contenant (utilisation des opérateurs de sélection slice et dice).
- Q3 : trouver, dans un cube contenant deux mesures, la cellule dans laquelle ces deux mesures sont égales.
- Q4 : trouver dans une dimension du cube deux membres âyant le même comportement vis à vis de la mesure (2 "tranches" du cube identiques).
- Q5 : même question que la précédente (deux membres égaux) mais en s'aidant avec la réorganisation (VR4OLAP stéréo uniquement).
- Q6 : trouver dans un cube un membre d'une dimension plus atypique que les autres (les membres appartiennent à des classes, les membres d'une même classe ont les mêmes valeurs, et le membre atypique est celui n'appartenant à aucune classe).
- Q7 : même question que la précédente (trouver le membre atypique) mais en s'aidant de l'algorithme de réorganisation (VR4OLAP stéréo uniquement).
- Q8 : trouver le nombre de classes des membres d'une dimension.
- Q9 : même question que la précédente (trouver le nombre de classe de membres d'une dimension) mais en s'aidant de la réorganisation (VR4OLAP stéréo uniquement).

Nous avons utilisé plusieurs cubes de données : pour les deux premières questions, la base utilisée est celle fournie avec Mondrian (base Access MondrianFoodMart), et pour les autres questions nous avons utilisé une base Access remplie par nous même spécialement pour l'évaluation. Une randomisation a lieu afin d'éviter les

effets d'apprentissage (plusieurs fois le même cube, ou le même ordre de test 2D-3D, ou les questions dans le même ordre, etc).

A chaque question et pour chaque méthode testée, la réponse donnée par l'utilisateur est notée pour pouvoir plus tard la comparer avec la valeur attendue et ainsi mesurer la qualité de la réponse sous la forme d'une mesure de similarité avec la bonne réponse ($\in [0, 1]$). De plus, pour chacune des questions, le temps de réponse a été mesuré. Un questionnaire préalable permet de connaître le niveau de la personne en Informatique, en 3D et en OLAP. Un questionnaire final permet de connaître les impressions "à chaud" de l'utilisateur. Avant de répondre aux questions, nous laissons l'utilisateur interagir avec les visualisations dans le but qu'il se familiarise avec elles et obtienne les compétences nécessaires pour répondre aux questions posées par la suite.

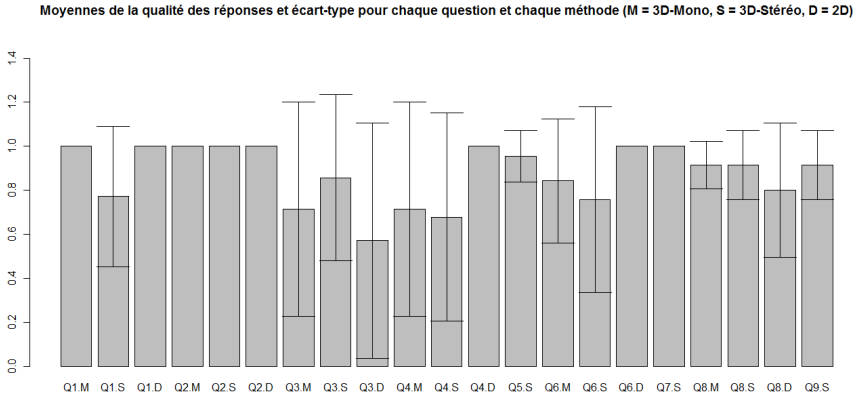
### 4.2 Résultats et discussion

Les résultats des questions Q1 à Q9 sont présentés figure 2. Nous pouvons en retirer les analyses et conclusions suivantes. En ce qui concerne les opérateurs OLAP (questions Q1 et Q2), les utilisateurs répondent avec une qualité comparable pour la 2D et la 3D-mono, mais la qualité baisse un peu en 3D-stereo. Les temps sont un peu plus courts pour la 3D. L'implémentation des opérateurs OLAP dans la représentation 3D semble globalement aussi efficace que celle de la 2D, ce qui est encourageant pour notre approche si l'on considère que les utilisateurs sont souvent plus habitués à la 2D qu'à la 3D.

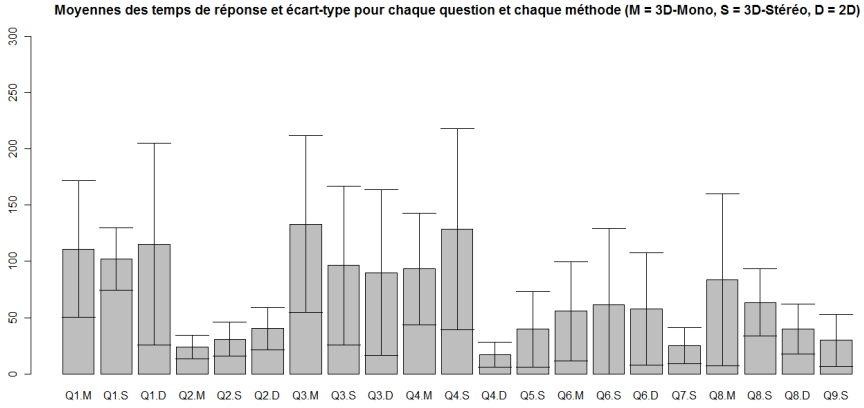
Pour la représentation de deux mesures (question Q3), on note que les meilleurs temps de réponses sont obtenus en 2D, mais par contre la qualité de la réponse est meilleure pour la 3D que la 2D. La représentation de deux mesures sous forme de pyramidion possède donc un avantage par rapport à la représentation 2D sous forme de deux nombres. Un attribut visuel "taille" permet plus facilement la comparaison entre deux valeurs [13].

Pour la question Q4, les résultats sont nettement en faveur de la 2D, aussi bien en qualité que pour le temps de réponse. En effet, nous avons pu observer que le fait de représenter explicitement la valeur sous la forme d'un nombre simplifie cette tâche par rapport à la 3D. Dans la représentation 3D, la comparaison entre valeurs éloignées spatialement les unes des autres est difficile sur la base de la couleur et de la taille des cubes, par rapport à des chiffres qui doivent se mémoriser plus facilement. Egalement, il peut y avoir des occlusions en 3D (mais en principe, nous avons remarqué que celles-ci sont limitées car les côtés des cubes aident beaucoup à résoudre cette tâche).

Pour la détection de membre atypique (Q6), les temps de réponse sont assez comparables pour cette tâche, avec un avantage net en qualité pour la 2D grâce à la lecture directe des valeurs sous forme de nombre. Pour la détection du nombre de classes (Q8), les réponses sont meilleures en 3D qu'en 2D, mais les temps sont plus courts en 2D. Nous avons observé le même phénomène que pour la question Q4 : en 2D, les utilisateurs se contentent d'observer la tête de colonne pour déterminer le nombre de classes. Nous aurions du complexifier cette tâche en créant des classes pas totalement homogènes.




(a)




(b)

**Figure 2.** Représentation des réponses des utilisateurs (qualité en haut, temps en bas).

En ce qui concerne la réorganisation (questions Q5, Q7 et Q9), on observe que celle-ci égalise ou améliore les performances de la 3D-stéréo sans réorganisation. La réorganisation augmente la qualité des réponses et diminue le temps nécessaire pour répondre (souvent divisé par 2), même par rapport à la 2D (questions Q7 et Q9). Le fait de faire apparaître des régularités dans la visualisation augmente de beaucoup la faculté d'analyse (un sujet largement débattu en réorganisation linéaire de matrices par exemple [14]). Les individus atypiques apparaissent nettement mieux, de même que les classes existantes dans les valeurs visualisées.

| Question | 2D-mono | 3D-mono | 3D-stéréo |
|---|---|---|---|
| Facile à utiliser | 3.5 (0.9) | 3.0 (0.8) | 2.8 (0.6) |
| Se repérer dans les données | 3.2 (0.4) | 3.8 (0.8) | 3.4 (1.1) |
| Attractif | 1.7 (0.4) | 3.7 (0.4) | 4.0 (0.8) |
| Adapté pour présentation | 2.8 (0.8) | 3.8 (1.0) | 2.8 (1.0) |

**Table 1.** Réponses des utilisateurs à différentes questions (scores de 1 à 5, moyenne sur 7 utilisateurs, écarts-types entre parenthèses).

Enfin, nous rapportons les informations obtenues via le questionnaire informel final (voir table 1). On constate que la 3D et la 2D obtiennent des notes comparables en ce qui concerne le repérage de l'utilisateur au sein des données et la facilité d'utilisation. Compte tenu de la nouveauté de la visualisation 3D, on aurait pu s'attendre à une notation défavorable pour notre approche mais ce n'est pas le cas. En particulier, l'apprentissage de l'utilisation du SpacePilot est difficile au début, même si nous savons par expérience que ce périphérique est très efficace une fois maîtrisé. Plus précisément, dans la phase d'apprentissage "libre", les utilisateurs ont passé trois fois plus de temps à tester le SpacePilot que le clavier (358 secondes en moyenne contre 156 secondes). Nous avons noté aussi que les déplacements clavier-souris pouvaient être améliorés avec un zoom plus rapide en 3D. Malgré cela, la 2D et la 3D sont notées sur ces points de manière équivalente. On note aussi que les utilisateurs trouvent VR4OLAP beaucoup plus attractif que l'approche 2D, et qu'ils l'utiliseraient plus volontiers que la 2D pour présenter des résultats. Ce point peut être important si l'on considère qu'OLAP est aussi utilisé pour présenter les conclusions d'une analyse à des décideurs. On constate aussi que les utilisateurs ont souvent moins bien noté la 3D-stéréo que la 3D-mono, ce qui traduit les difficultés qu'ils ont pu rencontré (adaptation, fatigue, etc).

En conclusion de cette étude, il faut retenir les points suivants : des performances équivalentes ont été observées entre 2D et 3D, notamment dans la navigation dans le cube avec les opérateurs OLAP, ou encore la visualisation de deux mesures. Ce point ne peut donc prouver la supérioté de la 3D-mono sur la 2D, ou encore la 3D-stereo sur la 2D, ou l'inverse. Néanmoins, il faut rappeler que les utilisateurs sont peu familiers avec la 3D. Avec un apprentissage plus

long (plusieurs séances), il serait peut être plus facile de montrer l'apport de la 3D en général (ou inversement), mais cela compliquerait beaucoup le protocole qui ne serait peut être plus acceptable pour des utilisateurs bénévoles. Ensuite, des performances meilleures en 2D sont observées notamment en détection de similarité. Pourtant, la 3D-mono est souvent considérée comme plus attractive que la 2D. Egalement, la réorganisation apporte un gain net dans la résolution des tâches liées à la similarité. Enfin, la 3D-stereo ne s'est pas montré meilleure en performance que la 3D-mono. Nous pensions qu'avec du matériel récent et plus accessible que le CAVE utilisé dans [6] (et qui n'avait pas été évalué par des utilisateurs) une différence aurait pu apparaître, mais cela n'a pas été le cas.

## 5 Conclusion

Nous avons présenté dans cet article VR4OLAP, une nouvelle interface pour OLAP. Les principaux éléments qui la caractérisent sont une représentation 3D, l'inclusion d'un grand nombre d'opérateurs OLAP sous la forme d'objets clicquables, l'utilisation d'un algorithme de réorganisation et enfin la visualisation sur un écran stéréoscopique avec du matériel d'interaction. Nous avons aussi présenté les résultats d'une évaluation utilisateur, ce qui est nouveau à notre connaissance pour ce type d'interface. Ces résultats n'ont pas permis de montrer un avantage de la 3D-stéréo sur la 3D-mono. Ils laissent plus d'ouverture sur la comparaison 2D-3D si l'on considère le manque d'expérience des utilisateurs en 3D par rapport à la 2D. Cependant, l'engouement des utilisateurs pour la 3D est motivant et souligne le défi représenté par les interfaces 3D : faire correspondre à cet engouement une facilité d'utilisation et une efficacité dans la résolution des tâches.

Outre les perspectives suggérées par nos testeurs, nous souhaitons développer à la fois le côté visualisation et le côté interaction de VR4OLAP. Dans le premier cas, nous allons ajouter des représentations plus complexes, avec par exemple des images (placées sur les cubes) pour représenter des informations supplémentaires sur les données (par exemple, photos de produits), et nous allons étudier également comment représenter plus de deux mesures. Pour les interactions, nous sommes en train de mettre en place des opérations de sélection de données, afin de proposer ensuite de nouveaux résultats à l'utilisateur (par exemple ne garder que le cube qui correspond aux données sélectionnées, ou encore préciser des informations sur ces données, etc).

## Références


1. Codd, E., Codd, S., Salley, C. : Providing olap to user-analysts : An it mandate. Technical report, E.F. Codd and Associates (1993)
2. Chaudhuri, Q., Dayal, U. : An overview of data warehousing and olap technology. ACM SIGMOD Record **26** (1997) 65–74
3. Chaudhuri, S., Dayal, U., Narasayya, V. : An overview of business intelligence technology. Commun. ACM **54**(8) (2011) 88–98

4. Han, J., Kamber, M., Pei, J. : Data Mining : Concepts and Techniques. 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2011)
5. Cuzzocrea, A., Mansmann, S. : Olap visualization : models, issues, and techniques. Encyclopedia of Data Warehousing and Mining, 2nd ed. (2009) 1439–1446
6. Ammoura, A., Zaïane, O., Goebel, R. : Towards a novel olap interface for distributed data warehouses. Data Warehousing and Knowledge Discovery, Third International Conference, DaWaK 2001 **2114** (2001) 174–185
7. Mansmann, S., Scholl, M.H. : Visual olap : a new paradigm for exploring multidimensional aggregates. In : IADIS International Conference Computer Graphics and Visualization 2008. (2008) 59–66
8. Han, J., Chiang, J., Chee, S., Chen, J., Chen, Q., Cheng, S., Gong, W., Kamber, M., Koperski, K., Liu, G., Lu, Y., Stefanovic, N., Winstone, L., Xia, B., Zaiane, O.R., Zhang, S., Zhu, H. : Dbminer : A system for data mining in relational databases and data warehouses. CASCON'97 : Meeting of Minds (1997) 249–260
9. Bulusu, P. : Diva-data warehouse interface for visual analysis. master thesis. university of florida (2003)
10. Miner3D : http ://www.miner3d.com/products/visual_olap.html. (2010)
11. Sureau, F., Bouali, F., Venturini, G. : Optimisation heuristique et génétique de visualisations 2d et 3d dans olap : premiers résultats. RNTI, 5ème journées francophones sur les entrepôts de données et l'analyse en ligne (EDA'09) (2009) 62–75
12. Lafon, S., Bouali, F., Guinot, C., Venturini, G. : Réorganisation hiérarchique de visualisations dans olap. Revue des Nouvelles Technologies de l'Information **Extraction et Gestion des Connaissances, RNTI-E-23** (2012) 287–298
13. Mackinlay, J. : Automating the design of graphical presentations of relational information. ACM Trans. Graph. **5**(2) (April 1986) 110–141
14. Bertin, J. : La graphique et le traitement graphique de l'information. Nouvelle Bibliothèque Scientifique. (1977)

# The adaptive method with hybrid direction for solving linear programs

Mohand Bentobache and Mohand Ouamer Bibi

L.A.M.O.S., Laboratory of Modelling and Optimization of Systems,
University of Bejaia, 06000, Algeria
mbentobache@yahoo.com, mobibi.dz@gmail.com

**Abstract.** In [2,3,5,6], a new search direction for the adaptive method, called hybrid direction, was suggested. For testing optimality, the optimality estimate was defined and used. However, a suboptimality criterion was not given and the updating formula, when we change the support, is not derived. In this paper, we overcome all the difficulties encountered in previous works. Indeed, by using the suboptimality estimate of the current solution, we derive a more general updating formula for the suboptimality estimate when we change the feasible solution and when we change the support too. Hence, the updating formula given in [9] is a special case of our formula. Finally, a numerical example is given for illustration purpose.
**Keywords:** Linear programming, Adaptive method, Hybrid direction, Suboptimality estimate.

## 1 Introduction

In [9], the authors developed the support method which is a generalization of the simplex method [8] for solving Linear Programming (LP) problems. The principle of this method is to start by a support feasible solution comprising a basis and a feasible solution and to go through interior or extreme points to achieve an optimal one. Later, they have developed the adaptive method to solve, particularly, linear optimal control problems [10]. This method is generalized to solve general linear and convex quadratic problems [4,7,11,12].

In [2,3,5,6], we suggested a new search direction for the adaptive method, this direction is called a hybrid direction because it takes for some solution components extreme values in order to bring them to their bounds and it takes for others the reduced gradient values. For testing optimality, the optimality estimate is defined and used. However, a suboptimality criterion was not given and the updating formula, when we change the support, was not derived.

In this work, we overcome all the difficulties encountered in previous works. Indeed, by using the suboptimality estimate of the current solution, we derive a more general updating formula for the suboptimality estimate when we change the feasible solution and when we change the support too. Hence, the updating formula given in [9] is a special case of our formula.

The paper is organized as follows: in Section 2, we give some definitions. In Section 3, we present the main theory of the suggested algorithm and a numerical example is given for illustration purpose. Finally, Section 4 concludes the paper and provides some perspectives.

## 2 Problem statement and definitions

Consider the linear programming problem with bounded variables presented in the following standard form:

$$\begin{aligned} &\max z = c^T x, \\ &\text{subject to } Ax = b,\ l \leq x \leq u, \end{aligned} \tag{1}$$

where $c$ and $x$ are $n$-vectors; $b$ an $m$-vector; $A$ an $(m \times n)$-matrix with $rank A = m < n$; $l$ and $u$ are finite-valued $n$-vectors. We define the following sets of indices:

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

• If $v$ is an arbitrary $n$-vector and $M$ an arbitrary $(m \times n)$-matrix, then we can write

$$v = v(J) = (v_j, j \in J) \text{ and } M = M(I, J) = (m_{ij}, i \in I,\ j \in J).$$

Moreover, we can partition $v$ and $M$ as follows:

$$v^T = (v_B^T, v_N^T),\ \text{where } v_B = v(J_B) = (v_j, j \in J_B),\ v_N = v(J_N) = (v_j, j \in J_N);$$

$$M = (M_B, M_N),\ \text{where } M_B = M(I, J_B),\ M_N = M(I, J_N).$$

• A vector $x$ verifying the constraints of problem (1) is called a *feasible solution*.
• A feasible solution $x^0$ is called *optimal* if

$$z(x^0) = c^T x^0 = \max\ c^T x,$$

where $x$ is taken from the set of all feasible solutions of the problem (1).
• A feasible solution $x^\epsilon$ is said to be $\epsilon$-*optimal* or *suboptimal* if

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

where $x^0$ is an optimal solution for the problem (1) and $\epsilon$ is a positive number chosen beforehand.
• We consider the index subset $J_B \subset J$ such that $|J_B| = |I| = m$. Then the set $J_B$ is called a *support* if $\det(A_B) \neq 0$.
• The pair $\{x, J_B\}$ comprising a feasible solution $x$ and a support $J_B$ will be called a *support feasible solution* (SFS).
• An SFS is called *nondegenerate* if $l_j < x_j < u_j,\ j \in J_B$.
• We define the $m$-vector of multipliers $\pi$ and the $n$-vector of reduced costs $\Delta$ as follows:

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

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

**Theorem 1** *(The optimality criterion [9]). Let $\{x, J_B\}$ be an SFS for the problem (1). Then the relations:*

$$\begin{cases} \Delta_j \geq 0, \ \textit{for } x_j = l_j, \\ \Delta_j \leq 0, \ \textit{for } x_j = u_j, \\ \Delta_j = 0, \ \textit{for } l_j < x_j < u_j, \ j \in J_N, \end{cases} \tag{2}$$

*are sufficient and, in the case of nondegeneracy of the SFS $\{x, J_B\}$, also necessary for the optimality of the feasible solution $x$.*

For an SFS $\{x, J_B\}$, we make the following partition: $J_N = J_N^{++} \cup J_N^{--} \cup J_N^0$, where

$J_N^{++} = \{j \in J_N : \Delta_j > 0\}, J_N^{--} = \{j \in J_N : \Delta_j < 0\}$ and $J_N^0 = \{j \in J_N : \Delta_j = 0\}$.

• The quantity $\beta(x, J_B)$ defined by:

$$\beta = \beta(x, J_B) = \sum_{j \in J_N^{++}} \Delta_j (x_j - l_j) + \sum_{J_N^{--}} \Delta_j (x_j - u_j), \tag{3}$$

is called the *suboptimality estimate* [9]. Thus, we have the following results [9]:

**Theorem 2** *(Sufficient condition for suboptimality). Let $\{x, J_B\}$ be an SFS for the problem (1) and $\epsilon$ an arbitrary positive number. If $\beta(x, J_B) \leq \epsilon$, then the feasible solution $x$ is $\epsilon$-optimal.*

**Corollary 1** *Let $\{x, J_B\}$ be an SFS for the problem (1). The condition $\beta(x, J_B) = 0$ is sufficient for the optimality of the feasible solution $x$.*

## 3 Adaptive method with hybrid direction (AMHD)

Let $\{x, J_B\}$ be an SFS for the problem (1) and $\eta \in [0, 1]$. Let $x^+ \in \mathbb{R}^n_+$ and $x^- \in \mathbb{R}^n_-$ be two vectors defined as follows:

$$x^+ = \eta(x - l) \text{ and } x^- = \eta(x - u).$$

We introduce the following set of indices:

$$\begin{aligned} &J_N^+ = \{j \in J_N : \Delta_j > x_j^+\}, \ J_N^- = \{j \in J_N : \Delta_j < x_j^-\}, \\ &J_N^{P^+} = \{j \in J_N : 0 < \Delta_j \leq x_j^+\}, \ J_N^{P^-} = \{j \in J_N : x_j^- \leq \Delta_j < 0\}, \\ &J_N^P = \{j \in J_N : x_j^- \leq \Delta_j \leq x_j^+\} = J_N^{P^+} \cup J_N^{P^-} \cup J_N^0. \end{aligned} \tag{4}$$

Thus,

$$J_N^{++} = J_N^+ \cup J_N^{P^+}, \ J_N^{--} = J_N^- \cup J_N^{P^-}, \ J_N = J_N^+ \cup J_N^- \cup J_N^P.$$

Let us define the non-negative quantities $\gamma = \gamma(\eta, x, J_B)$ and $\mu$ as follows:

$$\gamma = \begin{cases} \frac{1}{\eta}[\sum_{j\in J_N^+} \Delta_j x_j^+ + \sum_{j\in J_N^-} \Delta_j x_j^- + \sum_{j\in J_N^{P^+}\cup J_N^{P^-}} \Delta_j^2], \text{ if } \eta > 0; \\ \beta(x, J_B), \qquad\qquad\qquad\qquad\qquad\qquad \text{if } \eta = 0, \end{cases} \tag{5}$$

$$\mu = \begin{cases} \frac{1}{\eta}[\sum_{j\in J_N^{P^+}} \Delta_j(x_j^+ - \Delta_j) + \sum_{j\in J_N^{P^-}} \Delta_j(x_j^- - \Delta_j)], \text{ if } \eta > 0; \\ 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } \eta = 0. \end{cases} \tag{6}$$

The quantity $\gamma(\eta, x, J_B)$ is called the optimality estimate [2,3,5,6].

**Remark 1** *When $\eta \longrightarrow 0$, we get $J_N^{P^+} = J_N^{P^-} = \emptyset$. Then $\lim_{\eta\longrightarrow 0} \mu = 0$.*

**Lemma 1** *For all $\eta \geq 0$, the optimality estimate can be written as follows:*

$$\gamma = \beta - \mu \leq \beta. \tag{7}$$

*Proof.* For $\eta = 0$, we have $\mu = 0$ and $\gamma = \beta$, so $\gamma = \beta - \mu$. For $\eta > 0$,

$$\begin{aligned} \beta = \beta(x, J_B) &= \sum_{j\in J_N^{+^+}} \Delta_j(x_j - l_j) + \sum_{j\in J_N^{-^-}} \Delta_j(x_j - u_j) \\ &= \frac{1}{\eta} \sum_{j\in J_N^+\cup J_N^{P^+}} \Delta_j x_j^+ + \frac{1}{\eta} \sum_{j\in J_N^-\cup J_N^{P^-}} \Delta_j x_j^- \\ &= \frac{1}{\eta} \sum_{j\in J_N^+} \Delta_j x_j^+ + \frac{1}{\eta} \sum_{j\in J_N^-} \Delta_j x_j^- + \frac{1}{\eta} \sum_{j\in J_N^{P^+}} \Delta_j x_j^+ + \frac{1}{\eta} \sum_{j\in J_N^{P^-}} \Delta_j x_j^- \\ &= \gamma - \frac{1}{\eta} \sum_{j\in J_N^{P^+}\cup J_N^{P^-}} \Delta_j^2 + \frac{1}{\eta} \sum_{j\in J_N^{P^+}} \Delta_j x_j^+ + \frac{1}{\eta} \sum_{j\in J_N^{P^-}} \Delta_j x_j^- \\ &= \gamma + \frac{1}{\eta} \sum_{j\in J_N^{P^+}} \Delta_j(x_j^+ - \Delta_j) + \frac{1}{\eta} \sum_{j\in J_N^{P^-}} \Delta_j(x_j^- - \Delta_j) \\ &= \gamma + \mu. \end{aligned}$$

Since $\mu \geq 0$, we get $\gamma = \beta - \mu \leq \beta$. □

### 3.1 Changing the feasible solution

Let $\{x, J_B\}$ be an SFS for the problem (1) and $\eta \in [0, 1]$. We define the feasible direction $d$ as follows:

$$\begin{aligned} &d_j = l_j - x_j, \text{ if } j \in J_N^+; \\ &d_j = u_j - x_j, \text{ if } j \in J_N^-; \\ &d_j = \tfrac{-\Delta_j}{\eta}, \text{ if } j \in J_N^P, \eta \neq 0; \\ &d_j = 0, \text{ if } j \in J_N^P, \ \eta = 0; \\ &d_B = -A_B^{-1} A_N d_N. \end{aligned} \tag{8}$$

This direction, with respect to the standard direction of the adaptive method is called a hybrid direction. Contrarily to the direction used in the adaptive method, which takes only extreme or zero values, the hybrid direction takes extreme values for components with relatively big values of the reduced cost vector and it takes for others the reduced gradient values.

In order to improve the objective function while remaining in the feasible region, we compute the step length $\theta^0$ along the direction $d$ as follows:
$\theta^0 = \min\{\theta_{j_1}, 1\}$, where

$$\theta_{j_1} = \min\{\theta_j, j \in J_B\} \text{ and } \theta_j = \begin{cases} (u_j - x_j)/d_j, \text{ if } d_j > 0; \\ (l_j - x_j)/d_j, \text{ if } d_j < 0; \\ \infty, \qquad\qquad \text{if } d_j = 0. \end{cases} \tag{9}$$

Then the new feasible solution is $\bar{x} = x + \theta^0 d$. Since $Ad = 0$, then the vector $d$ is a feasible direction. Furthermore, in [5], we proved that $d$ is an ascent direction: the objective function increment is given by

$$\bar{z} - z = \theta^0 \gamma(\eta, x, J_B) \geq 0. \tag{10}$$

By replacing the expression of $\gamma(\eta, x, J_B)$ in (10), we get

$$\bar{z} - z = \theta^0(\beta - \mu). \tag{11}$$

**Lemma 2** *For all $\eta \geq 0$, the quantity $\bar{\beta} = \beta(\bar{x}, J_B)$ can be computed as follows:*

$$\bar{\beta} = (1 - \theta^0)\beta + \theta^0\mu \leq \beta. \tag{12}$$

*Proof.* For $\eta > 0$, we have

$$\begin{aligned}
\bar{\beta} &= \sum_{j \in J_N^{++}} \Delta_j(\bar{x}_j - l_j) + \sum_{j \in J_N^{--}} \Delta_j(\bar{x}_j - u_j) \\
&= \sum_{j \in J_N^{++}} \Delta_j(x_j - l_j) + \sum_{j \in J_N^{--}} \Delta_j(x_j - u_j) + \theta^0 \sum_{j \in J_N^{++}} \Delta_j d_j + \theta^0 \sum_{j \in J_N^{--}} \Delta_j d_j \\
&= \beta + \theta^0 \sum_{j \in J_N^{+} \cup J_N^{P+}} \Delta_j d_j + \theta^0 \sum_{j \in J_N^{-} \cup J_N^{P-}} \Delta_j d_j \\
&= \beta + \theta^0 \sum_{j \in J_N^{+}} \Delta_j d_j + \theta^0 \sum_{j \in J_N^{-}} \Delta_j d_j + \theta^0 \sum_{j \in J_N^{P+}} \Delta_j d_j + \theta^0 \sum_{j \in J_N^{P-}} \Delta_j d_j \\
&= \beta - \theta^0 \sum_{j \in J_N^{+}} \Delta_j(x_j - l_j) - \theta^0 \sum_{j \in J_N^{-}} \Delta_j(x_j - u_j) - \frac{\theta^0}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 \\
&= \beta - \frac{\theta^0}{\eta} \sum_{j \in J_N^{+}} \Delta_j x_j^+ - \frac{\theta^0}{\eta} \sum_{j \in J_N^{-}} \Delta_j x_j^- - \frac{\theta^0}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 \\
&= \beta - \theta^0 \gamma \leq \beta.
\end{aligned}$$

Since for $\eta > 0$, $\gamma = \beta - \mu$, then we find

$$\bar{\beta} = (1 - \theta^0)\beta + \theta^0\mu.$$

For $\eta = 0$, we have $\mu = 0$. Hence, we find the classical updating formula of $\beta(x, J_B)$ [9]:

$$\bar{\beta} = \beta(\bar{x}, J_B) = (1 - \theta^0)\beta(x, J_B). \square$$

**Theorem 3** *(Sufficient conditions for optimality and suboptimality of $\bar{x}$)*
*If $\theta^0 = 1$ and $\mu = 0$, then the feasible solution $\bar{x}$ is optimal.*
*If $\theta^0 = 1$ and $\mu \leq \epsilon$, then the feasible solution $\bar{x}$ is $\epsilon$-optimal.*

*Proof.* We assume that $\theta^0 = 1$ and $\mu = 0$. Following Lemma 2, $\bar{\beta} = 0$. By using Corollary 1, we deduce the optimality of $\bar{x}$. If $\theta^0 = 1$ and $\mu \leq \epsilon$, then following Lemma 2, $\bar{\beta} = \mu \leq \epsilon$. By using Theorem 2, we deduce the suboptimality of $\bar{x}$. $\square$

**Remark 2** *The condition $\mu = 0$ holds in the following cases:*
***(a)*** *when $\eta = 0$;*
***(b)*** *when $\eta > 0$ and $J_N^{P^+} \cup J_N^{P^-} = \emptyset$;*
***(c)*** *when $\eta > 0$, $J_N^{P^+} \cup J_N^{P^-} \neq \emptyset$, and* $\begin{cases} \Delta_j = x_j^+, \text{ for all } j \in J_N^{P^+}; \\ \Delta_j = x_j^-, \text{ for all } j \in J_N^{P^-}. \end{cases}$

If $\theta^0 = 1$ and $\mu > \epsilon$, then we switch to the iterations of the adaptive method by setting $\eta = 0$. If $\theta^0 = \theta_{j_1} < 1$ and $\bar{\beta} > \epsilon$, then we change the support $J_B$.

### 3.2 Changing the support

We define the $n$-vector $\kappa$ and the real number $\alpha_0$ as follows:

$$\kappa = x + d \text{ and } \alpha_0 = \kappa_{j_1} - \bar{x}_{j_1},$$

where $j_1$ is the leaving index computed in (9). So the dual direction is

$$t_{j_1} = -sign(\alpha_0);\ t_j = 0,\ j \neq j_1,\ j \in J_B;\ t_N^T = t_B^T A_B^{-1} A_N. \tag{13}$$

**Remark 3** *We have $\alpha_0 = \kappa_{j_1} - \bar{x}_{j_1} = x_{j_1} + d_{j_1} - x_{j_1} - \theta^0 d_{j_1} = (1 - \theta^0)d_{j_1}$. Since $0 \leq \theta^0 < 1$, then $t_{j_1} = -sign(\alpha_0) = -sign(d_{j_1})$.*

**Remark 4** *The dual direction $t$ and the primal direction $d$ are orthogonal. Indeed,*

$$t^T d = t_N^T d_N + t_B^T d_B = (t_B^T A_B^{-1} A_N) d_N + t_B^T (-A_B^{-1} A_N d_N) = 0.$$

Let us define the following sets:

$$J_N^{0^+} = \{j \in J_N^0 : t_j > 0\} \text{ and } J_N^{0^-} = \{j \in J_N^0 : t_j < 0\}, \tag{14}$$

and the quantity:

$$\alpha = -|\alpha_0| + \sum_{j \in J_N^{0+} \cup J_N^{P+}} t_j(\kappa_j - l_j) + \sum_{j \in J_N^{0-} \cup J_N^{P-}} t_j(\kappa_j - u_j). \tag{15}$$

The new reduced cost vector and the new support are computed as follows:

$$\bar{\Delta} = \Delta + \sigma^0 t \text{ and } \bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\},$$

where

$$\sigma^0 = \sigma_{j_0} = \min_{j \in J_N} \{\sigma_j\}, \text{ with } \sigma_j = \begin{cases} \frac{-\Delta_j}{t_j}, \text{ if } \Delta_j t_j < 0; \\ 0, \quad \text{if } j \in J_N^{0-} \text{ and } \kappa_j \neq u_j; \\ 0, \quad \text{if } j \in J_N^{0+} \text{ and } \kappa_j \neq l_j; \\ \infty, \quad \text{otherwise.} \end{cases} \tag{16}$$

**Remark 5** *If $\sigma^0 = \infty$, then problem (1) is infeasible.*

We assume that $\sigma^0 < \infty$. The suboptimality estimate corresponding to the new feasible solution and the new support is given by

$$\bar{\bar{\beta}} = \beta(\bar{x}, \bar{J}_B) = \sum_{j \in \bar{J}_N, \bar{\Delta}_j > 0} \bar{\Delta}_j(\bar{x}_j - l_j) + \sum_{j \in \bar{J}_N, \bar{\Delta}_j < 0} \bar{\Delta}_j(\bar{x}_j - u_j).$$

**Lemma 3** *The suboptimality estimate $\beta(\bar{x}, \bar{J}_B)$ can be written as follows:*

$$\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0 \alpha. \tag{17}$$

*Proof.* We have $\bar{J}_N = (J_N \setminus \{j_0\}) \cup \{j_1\}$. If $\sigma^0 = 0$, then $j_0 \in J_N^{0+}$ or $j_0 \in J_N^{0-} \Rightarrow \Delta_{j_0} = 0 \Rightarrow \bar{\Delta}_{j_0} = 0$. If $\sigma^0 > 0$, then $\bar{\Delta}_{j_0} = \Delta_{j_0} + \sigma^0 t_{j_0} = \Delta_{j_0} - \frac{\Delta_{j_0}}{t_{j_0}} t_{j_0} = 0$. Therefore,

$$\bar{\bar{\beta}} = \begin{cases} \sum_{j \in J_N, \bar{\Delta}_j > 0} \bar{\Delta}_j(\bar{x}_j - l_j) + \sum_{j \in J_N, \bar{\Delta}_j < 0} \bar{\Delta}_j(\bar{x}_j - u_j) + \bar{\Delta}_{j_1}(\bar{x}_{j_1} - l_{j_1}), \text{ if } \bar{\Delta}_{j_1} > 0; \\ \sum_{j \in J_N, \bar{\Delta}_j > 0} \bar{\Delta}_j(\bar{x}_j - l_j) + \sum_{j \in J_N, \bar{\Delta}_j < 0} \bar{\Delta}_j(\bar{x}_j - u_j) + \bar{\Delta}_{j_1}(\bar{x}_{j_1} - u_{j_1}), \text{ if } \bar{\Delta}_{j_1} < 0. \end{cases}$$

Since $\Delta_{j_1} = 0$ ($j_1 \in J_B$), two cases can occur:
• If $\bar{\Delta}_{j_1} > 0$, then $t_{j_1} > 0 \Rightarrow -\text{sign}(d_{j_1}) > 0 \Rightarrow d_{j_1} < 0$. Hence,

$$\bar{x}_{j_1} = x_{j_1} + \theta^0 d_{j_1} = x_{j_1} + \frac{(l_{j_1} - x_{j_1})}{d_{j_1}} d_{j_1} = l_{j_1} \Rightarrow \bar{\Delta}_{j_1}(\bar{x}_{j_1} - l_{j_1}) = 0.$$

• If $\bar{\Delta}_{j_1} < 0$, then $t_{j_1} < 0 \Rightarrow d_{j_1} > 0$. Hence,

$$\bar{x}_{j_1} = x_{j_1} + \theta^0 d_{j_1} = x_{j_1} + \frac{(u_{j_1} - x_{j_1})}{d_{j_1}} d_{j_1} = u_{j_1} \Rightarrow \bar{\Delta}_{j_1}(\bar{x}_{j_1} - u_{j_1}) = 0.$$

Since $\sigma^0$ is chosen in such a way that $\Delta_j$ and $\bar{\Delta}_j$ keep the same sign, we deduce that

$$\bar{\Delta}_j > 0 \Leftrightarrow [(\Delta_j > 0) \text{ or } (\Delta_j = 0 \text{ and } t_j > 0)],$$

and

$$\bar{\Delta}_j < 0 \Leftrightarrow [(\Delta_j < 0) \text{ or } (\Delta_j = 0 \text{ and } t_j < 0)].$$

Hence,

$$\begin{aligned}
\bar{\bar{\beta}} &= \sum_{j\in J_N, \bar{\Delta}_j>0} \bar{\Delta}_j(\bar{x}_j - l_j) + \sum_{j\in J_N, \bar{\Delta}_j<0} \bar{\Delta}_j(\bar{x}_j - u_j) \\
&= \sum_{j\in J_N, \Delta_j>0} (\Delta_j + \sigma^0 t_j)(\bar{x}_j - l_j) + \sum_{j\in J_N, \Delta_j<0} (\Delta_j + \sigma^0 t_j)(\bar{x}_j - u_j) \\
&+ \sum_{j\in J_N, \Delta_j=0, t_j>0} (\Delta_j + \sigma^0 t_j)(\bar{x}_j - l_j) + \sum_{j\in J_N, \Delta_j=0, t_j<0} (\Delta_j + \sigma^0 t_j)(\bar{x}_j - u_j) \\
&= \sum_{j\in J_N, \Delta_j>0} \Delta_j(\bar{x}_j - l_j) + \sum_{j\in J_N, \Delta_j<0} \Delta_j(\bar{x}_j - u_j) \\
&+ \sigma^0 \sum_{j\in J_N, \Delta_j=0, t_j>0} t_j(\bar{x}_j - l_j) + \sigma^0 \sum_{j\in J_N, \Delta_j=0, t_j<0} t_j(\bar{x}_j - u_j) \\
&+ \sigma^0 \sum_{j\in J_N, \Delta_j>0} t_j(\bar{x}_j - l_j) + \sigma^0 \sum_{j\in J_N, \Delta_j<0} t_j(\bar{x}_j - u_j) \\
&= \bar{\beta} + \sigma^0 [\sum_{j\in J_N^{0+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{0-}} t_j(\bar{x}_j - u_j)] \\
&\quad + \sigma^0 [\sum_{j\in J_N^{+}{}^{+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{-}{}^{-}} t_j(\bar{x}_j - u_j)] \\
&= \bar{\beta} + \sigma^0(\alpha_1 + \alpha_2),
\end{aligned}$$

where

$$\alpha_1 = \sum_{j\in J_N^{0+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{0-}} t_j(\bar{x}_j - u_j)$$

and

$$\alpha_2 = \sum_{j\in J_N^{+}{}^{+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{-}{}^{-}} t_j(\bar{x}_j - u_j).$$

For $j \in J_N^{0+} \cup J_N^{0-}$, we have $d_j = \frac{-\Delta_j}{\eta} = 0 \Rightarrow \bar{x}_j = x_j = \kappa_j$. Then

$$\alpha_1 = \sum_{j\in J_N^{0+}} t_j(\kappa_j - l_j) + \sum_{j\in J_N^{0-}} t_j(\kappa_j - u_j),$$

and

$$\begin{aligned}
\alpha_2 &= \sum_{j\in J_N^+} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^-} t_j(\bar{x}_j - u_j) + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= \sum_{j\in J_N^+} t_j(x_j - l_j) + \sum_{j\in J_N^-} t_j(x_j - u_j) + \theta^0 \sum_{j\in J_N^+} t_j d_j + \theta^0 \sum_{j\in J_N^-} t_j d_j \\
&\quad + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= \sum_{j\in J_N^+} t_j(-d_j) + \sum_{j\in J_N^-} t_j(-d_j) + \theta^0 \sum_{j\in J_N^+} t_j d_j + \theta^0 \sum_{j\in J_N^-} t_j d_j \\
&\quad + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= -(1-\theta^0)[\sum_{j\in J_N^+} t_j d_j + \sum_{j\in J_N^-} t_j d_j] + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= -(1-\theta^0)[\sum_{j\in J} t_j d_j - \sum_{j\in J_N^0} t_j d_j - \sum_{j\in J_N^{P+}\cup J_N^{P-}} t_j d_j - \sum_{j\in J_B} t_j d_j] \\
&\quad + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= -(1-\theta^0)[0 - 0 - \sum_{j\in J_N^{P+}\cup J_N^{P-}} t_j d_j - t_{j_1} d_{j_1}] + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= -(1-\theta^0)|d_{j_1}| + (1-\theta^0) \sum_{j\in J_N^{P+}\cup J_N^{P-}} t_j d_j + \sum_{j\in J_N^{P+}} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\bar{x}_j - u_j) \\
&= -(1-\theta^0)|d_{j_1}| + \sum_{j\in J_N^{P+}} t_j[(\bar{x}_j - l_j) + (1-\theta^0)d_j] + \sum_{j\in J_N^{P-}} t_j[(\bar{x}_j - u_j) + (1-\theta^0)d_j] \\
&= -(1-\theta^0)|d_{j_1}| + \sum_{j\in J_N^{P+}} t_j[x_j + \theta^0 d_j - l_j + d_j - \theta^0 d_j] \\
&\quad + \sum_{j\in J_N^{P-}} t_j[x_j + \theta^0 d_j - u_j + d_j - \theta^0 d_j] \\
&= -(1-\theta^0)|d_{j_1}| + \sum_{j\in J_N^{P+}} t_j(\kappa_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\kappa_j - u_j).
\end{aligned}$$

Since $(1-\theta^0)|d_{j_1}| = |\alpha_0|$, then we obtain

$$\alpha_2 = -|\alpha_0| + \sum_{j\in J_N^{P+}} t_j(\kappa_j - l_j) + \sum_{j\in J_N^{P-}} t_j(\kappa_j - u_j).$$

Therefore,

$$\alpha_1 + \alpha_2 = -|\alpha_0| + \sum_{j \in J_N^{0^+} \cup J_N^{P^+}} t_j(\kappa_j - l_j) + \sum_{j \in J_N^{0^-} \cup J_N^{P^-}} t_j(\kappa_j - u_j) = \alpha. \ \square$$

**Remark 6** *If $J_N^P = \emptyset$, then $\alpha = -|\alpha_0|$. So we find the classical updating formula of $\beta(\bar{x}, J_B)$ [9]: $\bar{\beta}(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) - \sigma^0|\alpha_0|$.*

### 3.3 Scheme of the adaptive method with hybrid direction and short step rule

Let $\{x, J_B\}$ be an initial SFS for the problem (1), $\epsilon$ be a non-negative number and $\eta \in [0, 1]$. The scheme of the adaptive method with hybrid direction and short step rule is described in the following steps:

**Algorithm 1**

**(1)** *compute* $\pi^T = c_B^T A_B^{-1}$, $\Delta_N^T = \pi^T A_N - c_N^T$;
**(2)** *compute the suboptimality estimate* $\beta$ *with (3);*
**(3)** *if* $\beta = 0$, *then the algorithm stops with the optimal SFS* $\{x, J_B\}$;
**(4)** *if* $\beta \leq \epsilon$, *then the algorithm stops with the* $\epsilon$*-optimal SFS* $\{x, J_B\}$;
**(5)** *compute the vectors* $x^+ = \eta(x - l)$ *and* $x^- = \eta(x - u)$;
**(6)** *compute the sets* $J_N^+$, $J_N^-$, $J_N^{P^+}$ *and* $J_N^{P^-}$ *with (4);*
**(7)** *compute* $\mu$ *with (6);*
**(8)** *compute the primal search direction* $d$ *with (8);*
**(9)** *compute the primal step length* $\theta^0$ *with (9);*
**(10)** *compute* $\bar{x} = x + \theta^0 d$ *and* $\bar{z} = z + \theta^0(\beta - \mu)$;
**(11)** *if* $\theta^0 = 1$, *then*
  **(11.1)** *if* $\mu = 0$, *then* $\bar{x}$ *is optimal. Stop;*
  **(11.2)** *if* $\mu \leq \epsilon$, *then* $\bar{x}$ *is* $\epsilon$*-optimal. Stop;*
  **(11.3)** *else, set* $\eta = 0$, $x = \bar{x}$, $z = \bar{z}$ *and go to step (5);*
**(12)** *compute the suboptimality estimate* $\bar{\beta} = \beta(\bar{x}, J_B) = (1 - \theta^0)\beta + \theta^0\mu$;
**(13)** *if* $\bar{\beta} \leq \epsilon$, *then the algorithm stops with the* $\epsilon$*-optimal SFS* $\{\bar{x}, J_B\}$;
**(14)** *compute* $\kappa = x + d$ *and* $\alpha_0 = \kappa_{j_1} - \bar{x}_{j_1}$;
**(15)** *compute the dual direction* $t$ *with (13);*
**(16)** *compute the sets* $J_N^{0^+}$, $J_N^{0^-}$, *and* $\alpha$ *with relations (14) and (15);*
**(17)** *compute the dual step length* $\sigma^0$ *and the entering index* $j_0$ *with (16);*
**(18)** *if* $\sigma^0 = \infty$, *then problem (1) is infeasible; else, compute the new suboptimality estimate* $\bar{\bar{\beta}} = \beta(\bar{x}, \bar{J}_B) = \bar{\beta} + \sigma^0\alpha$;
**(19)** *compute the new reduced costs vector and the new support:* $\bar{\Delta} = \Delta + \sigma^0 t$, $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$ *and* $\bar{J}_N = (J_N \setminus \{j_0\}) \cup \{j_1\}$;
**(20)** *set* $x = \bar{x}$, $J_B = \bar{J}_B$, $J_N = \bar{J}_N$, $z = \bar{z}$, $\Delta = \bar{\Delta}$, $\beta = \bar{\bar{\beta}}$ *and go to step (3).*

**Numerical example**

Let us solve the following LP problem with AMHD using short step rule:

$$\max \ c^T x, \ \text{s.t.} \ Ax = b, \ l \leq x \leq u,$$

where

$$A = \begin{pmatrix} 3 & -1 & 1 & 1 & 0 \\ -1 & -4 & 1 & 0 & 1 \end{pmatrix},\ b = (1,2)^T,\ c = (1,-3,1,0,0)^T,\ l = 0_{\mathbb{R}^5},\ u = (1,4,5,5,19)^T.$$

We set $\eta = 1/2$ and we start by the feasible solution

$$x = (0,0,0,1,2)^T,\ \text{with } z = 0.$$

**Iteration 1:** the results of this iteration are as follows:

$J_B = \{4,5\},\ J_N = \{1,2,3\},\ \pi^T = c_B^T A_B^{-1} = (0,0),\ \Delta_N^T = \pi^T A_N - c_N^T = (-1,3,-1);$
$\Delta = (-1,3,-1,0,0)^T,\ J_N^{++} = \{2\},\ J_N^{--} = \{1,3\},\ \beta = \beta(x,J_B) = 6 > 0;$
$x^+ = \eta(x-l) = (0,0,0,1/2,1)^T,\ x^- = \eta(x-u) = (-1/2,-2,-5/2,-2,-17/2)^T;$
$J_N^+ = \{2\},\ J_N^- = \{1\},\ J_N^{P^+} = \emptyset,\ J_N^{P^-} = \{3\},\ \mu = 3;$
$d_N = (1,0,2)^T,\ d_B = (-5,-1)^T,\ d = (1,0,2,-5,-1)^T;$
$\theta^0 = \min\{\theta_4,\theta_5,1\} = \min\{1/5,2,1\} = 1/5 \Rightarrow j_1 = 4;$
$\bar{x} = x + \theta^0 d = (1/5,0,2/5,0,9/5)^T,\ \bar{z} = c^T\bar{x} = z + \theta^0(\beta - \mu) = 3/5;$
$\bar{\beta} = (1-\theta^0)\beta + \theta^0\mu = 27/5 > 0,\ \kappa = x + d = (1,0,2,-4,1)^T,\ \alpha_0 = \kappa_{j_1} - \bar{x}_{j_1} = -4;$
$t_B^T = (1,0),\ t_N^T = (3,-1,1),\ t = (3,-1,1,1,0)^T,\ J_N^{0^+} = J_N^{0^-} = \emptyset,\ \alpha = -7;$
$\sigma^0 = \min\{\sigma_1,\sigma_2,\sigma_3\} = \min\{1/3,3,1\} = 1/3 \Rightarrow j_0 = 1;$
$\bar{\bar{\beta}} = \bar{\beta} + \sigma^0\alpha = 46/15 > 0,\ \bar{\Delta} = (0,8/3,-2/3,1/3,0)^T,\ \bar{J}_B = \{1,5\},\ \bar{J}_N = \{4,2,3\}.$

**Iteration 2:** the results of this iteration are as follows:

$J_B = \{1,5\},\ J_N = \{4,2,3\},\ x = (1/5,0,2/5,0,9/5)^T,\ z = 3/5;$
$\beta = \beta(x,J_B) = 46/15,\ \Delta = (0,8/3,-2/3,1/3,0)^T;$
$x^+ = (1/10,0,1/5,0,9/10)^T,\ x^- = (-2/5,-2,-23/10,-5/2,-43/5)^T;$
$J_N^+ = \{4,2\},\ J_N^- = \emptyset,\ J_N^{P^+} = \emptyset,\ J_N^{P^-} = \{3\},\ \mu = 98/45;$
$d_N = (0,0,4/3)^T,\ d_B = (-4/9,-16/9)^T,\ d = (-4/9,0,4/3,0,-16/9)^T;$
$\theta^0 = \min\{\theta_1,\theta_5,1\} = \min\{9/20,81/80,1\} = 9/20 \Rightarrow j_1 = 1;$
$\bar{x} = (0,0,1,0,1)^T,\ \bar{z} = 1,\ \bar{\beta} = \beta(\bar{x},J_B) = 8/3 > 0;$
$\kappa = (-11/45,0,26/15,0,1/45)^T,\ \alpha_0 = -11/45;$
$t_B^T = (1,0),\ t_N^T = (1/3,-1/3,1/3),\ t = (1,-1/3,1/3,1/3,0)^T,\ J_N^{0^+} = J_N^{0^-} = \emptyset;$
$\alpha = -4/3,\ \sigma^0 = \min\{\sigma_4,\sigma_2,\sigma_3\} = \min\{\infty,8,2\} = 2 \Rightarrow j_0 = 3,\ \bar{\bar{\beta}} = \bar{\beta} + \sigma^0\alpha = 0.$

Therefore, the optimal solution and the optimal value are:

$$x^0 = (0,0,1,0,1)^T \text{ and } z^0 = 1.$$

The different iterations of the adaptive method (AMHD with $\eta = 0$) for solving the previous example are as follows:

$x^{(1)} = (0,0,0,1,2)^T,\ z^{(1)} = 0,\ \beta^{(1)} = 6;$
$x^{(2)} = (1/8,0,5/8,0,3/2)^T,\ z^{(2)} = 3/4,\ \beta^{(2)} = 35/12;$
$x^{(3)} = (0,0,1,0,1)^T,\ z^{(3)} = 1,\ \beta^{(3)} = 0.$

## 4 Conclusion

Contrarily to [2,3,5,6] where we use the optimality estimate to test the optimality of the current feasible solution, in this work the conditions used to characterize the optimality of the current solution are based on the suboptimality estimate. A general formula for updating the suboptimality estimate is derived and an algorithm called the adaptive method with hybrid direction is described. In future works, we will modify the long step rule described in [9] to use it in our method. Furthermore, we will apply some crash procedure like that presented in [1] in order to initialize AMHD with a good initial SFS, then we will implement our method and will test its performance on practical LP test problems [13].

## References


1. M. Bentobache and M. O. Bibi, *A Two-phase Support Method for Solving Linear Programs: Numerical Experiments*, Mathematical Problems in Engineering, vol. 2012, Article ID 482193, 28 pages doi:10.1155/2012/482193.
2. M. Bentobache, On mathematical methods of linear and quadratic programming, PhD thesis, University of Bejaia, 2013.
3. M. Bentobache and M.O. Bibi, *Adaptive method with hybrid direction: theory and numerical experiments*, Proceedings of Optimization 2011, Universidade Nova de Lisboa, Portugal, 24-27 July 2011, pp. 112.
4. M. O. Bibi, Support method for solving a linear-quadratic problem with polyhedral constraints on control, *Optimization*, vol. 37, no. 2, pp. 139–147, 1996.
5. M. O. Bibi and M. Bentobache, The adaptive method with hybrid direction for solving linear programming problems with bounded variables, In: Proceedings of COSI'2011, University of Guelma, Algeria, pp. 80-91, 24-27 April 2011.
6. M. O. Bibi and M. Bentobache, *A hybrid direction algorithm for solving linear programs*, Proceedings of DIMACOS'11, University of Mohammedia, Morocco, 5-8 May 2011, pp. 28–30.
7. B. Brahmi and M. O. Bibi, Dual support method for solving convex quadratic programs, Optimization, Vol. 59, No. 6, 2010, pp. 851–872.
8. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.
9. R. Gabasov and F. M. Kirillova, *Methods of linear programming*, Vol. 1, 2 and 3, Edition of the Minsk University, 1977, 1978 and 1980 (in Russian).
10. R. Gabasov, F.M. Kirillova and S.V. Prischepova, *Optimal Feedback Control*, Springer-Verlag, London, 1995.
11. E. A. Kostina and O. I. Kostyukova, An algorithm for solving quadratic programming problems with linear equality and inequality constraints, *Computational Mathematics and Mathematical Physics*, vol. 41, no. 7, pp. 960––973, 2001.
12. E. A. Kostina, The long step rule in the bounded-variable dual simplex method: Numerical experiments, *Mathematical Methods of Operations Research*, vol. 55, pp. 413–429, 2002.
13. *Netlib test problems*; available at http://www.netlib.org/lp/data.

# Ordonnancement, gestion de la production

# An Integer Chance Constrained Model for Production Planning

Fatima BELLAHCENE

Operational Research and Mathematics Decision Aid Laboratory (LAROMAD), Faculty of Sciences, Mouloud Mammeri University, Tizi-Ouzou 15000, Algeria.
f_bellahcene@yahoo.fr

**Abstract.** The focus in this paper is on a special integer chance constrained stochastic program that arises from an application to production planning and in which the objective function is the expectation of a linear combination of the random variables. Under the assumption that the random variables are independent and normally distributed, the usual route used in stochastic programming is followed by transforming the stochastic model into an equivalent deterministic convex integer nonlinear optimization program. This program is solved using the discrete polyblock approximation algorithm which exploits its special structure. A numerical example is included for illustration.

**keywords:** Monotone optimization; Stochastic programming; Domain cut; Polyblock approximation.

**AMS subject classification**: 90-XX, 90B50

## 1 Introduction

In practice, every economic activity involves discrete decision variables and a piece of uncertainty. When the problem parameters are treated as random variables an integer stochastic programming problem must be solved. This last is usually transformed into an integer equivalent deterministic nonlinear problem. Different methods to carry out this transformation are presented in the works of Kall and Wallace [9], Prekopa [14] or Liu and Iwamura [12]. One of the most applied in problems with continuous random variables is the Chance Constrained Programming CCP method in which it is required that the constraints hold with, at least, a given probability. Its application lead us to obtain a nonconvex feasible set, which makes the solution of the resulting problem difficult.

Different algorithms have been proposed for solving (mixed) integer nonlinear programming (MINLP) problems. In the early 70's, Geoffrion [8] introduced the Generalized Benders Decomposition (GBD) method. The GBD method uses duality theory to derive one single constraint that combines the linearizations derived from all the original problem constraints and solves a mixed integer linear programming (MILP) master problem. In the 80's, Duran and Grossmann [6] introduced the Outer Approximation decomposition algorithm. The OA method is very similar to the GBD method, differing only in the definition of the (MILP) master problem. Specifically, instead of combining

the linearizations derived from all the original problem constraints, it uses linearizations for each nonlinear constraint. The fundamental insight behind the algorithm is that (MINLP) is equivalent to a mixed integer linear program (MILP) of finite size. This latter algorithm was subsequently improved in the 90's by Fletcher and Leyffer [7]. The NLP-Based Branch-and-Bound algorithm was first proposed by Quesada and Grossmann [15]. The method is an extension of the OA method but instead of solving a sequence of master problems, the master problem is dynamically updated in a single branch-and-bound tree that closely resembles the branch-and-cut method for MILP. In the same period, a related method called the Extended Cutting Plane (ECP) method which is an extension of Kelley's cutting plane method [10] for solving convex NLPs was proposed by Westerlund and Pettersson [20]. The main feature of the ECP method is that it does not require the use of an NLP solver. The algorithm is based on the iterative solution of a reduced master problem (RMP). Linearizations of the most violated constraint at the optimal solution of (RMP) are added at every iteration. There are also many modern software packages implementing the cited algorithms (For more details, see for example Abhishek and al. [1] and Bonami et al. [2]). We believe that our review covers all significant methods, although we may have missed some of the pertinent literature.

The focus in this paper is on a special integer stochastic program with a chance constraint in which, with a given probability, a linear combination of random variables is bounded above. The objective is to maximize the expectation of a linear function of the random variables. The stochastic program is first reduced to an equivalent deterministic convex integer nonlinear program with monotonic objective and constraints functions. This program can be easily solved by using one of the cited methods but its special structure encouraged us to design the discrete Polyblock method [17] to solve it rather simply and accurately. The Polyblock algorithm does not require linearizations or duality or any other properties except monotonicity of the functions. Its efficiency has been demonstrated in various applications such as multiplicative programming, linear and polynomial fractional programming (see for example [13]). At each stage of the technique, the feasible domain is divided in two subdomains and each of them is analyzed in order to discard the one not containing a point with the objective value better than the incumbent solution.

First, in section 2, we state the considered stochastic problem and reformulate it as an integer monotonic optimization problem. In section 3, we review some basic concepts and results of monotonic optimization. The discrete version of the polyblock approximation method is outlined in section 4 followed by a numerical example which shows how the method works in practice.

## 2 Problem statement

Consider a chance constrained program in which, with a given degree of probability $\alpha$ ($\alpha$ close to 1), a linear combination of random variables $Y_i$, $i = 1, \ldots, n$ is bounded above by $b_u$. The decision variables $x_i$, $i = 1, \ldots, n$ are assumed to be integers and bounded. Each random variable $Y_i$ is normally distributed and linear in $x_i$. The objective

is to maximize the expectation of a linear function of the $Y_i$. This problem is formulated as:

$$\begin{cases} \text{maximize } E\left(\sum_{i=1}^{n}\beta_i Y_i\right) \\ \text{subject to } P\left(\sum_{i=1}^{n} Y_i \le b_u\right) \ge \alpha \\ x \in X = \{x \in \mathbb{Z}_+^n \mid l_i \le x_i \le d_i,\ i = 1, ..., n\} \end{cases} \tag{P1}$$

where $l_i$ and $d_i$ are integer numbers with $l_i < d_i$ for $i = 1, \ldots, n$ and $\beta_i \ge 0$ for $i = 1, \ldots, n$.

Such a decision problem may arise in manufacturing, where $x_i$ is the number of items produced and $Y_i$ is the number of nondefective items. Similar problems may arise in other applications such as: higher education when an institution wishes to maximize the desirability of a class of students while satisfying a constraint on the number of students who enroll [3,5,11]. In this setting, the decision variables $x_i$ represent the number of students of a given type who are admitted, while the random variable $Y_i$ represents the number of such students who enroll. In airline or hotel overbooking problems, $x_i$ represent the number of reservations accepted and $Y_i$ is the number of passengers or guests and marketing, where $x_i$ is the number of solicitations issued and $Y_i$ is the number of responses.

In each of these applications, the random variables $Y_i$ can be characterized as binomial random variables with parameters $p_i$ and $x_i$, and since the number of items $x_i$ is typically large, the binomial random variables are easily approximated by normal random variables with mean $\mu_i = p_i x_i$ and variance $\sigma_i^2 = p_i(1 - p_i)x_i = v_i x_i$. Under the assumption that the item types act independently, $\sum_{i=1}^{n} Y_i$ is approximately normal with mean and variance

$$\mu = \sum_{i=1}^{n}\mu_i = \sum_{i=1}^{n} p_i x_i \quad \text{and} \quad \sigma^2 = \sum_{i=1}^{n}\sigma_i^2 = \sum_{i=1}^{n} v_i x_i \tag{1}$$

The objective function can be written as :

$$E\left(\sum_{i=1}^{n}\beta_i Y_i\right) = \sum_{i=1}^{n}\beta_i p_i x_i = \sum_{i=1}^{n} c_i x_i \tag{2}$$

Furthermore, as introduced by Charnes and cooper [4], the upper-bound chance constraint in (P1) may be written as

$$P\left(\sum_{i=1}^{n} Y_i \le b_u\right) \ge \alpha \Longleftrightarrow \Phi\left(\frac{b_u - \mu}{\sigma}\right) \ge \alpha \Longleftrightarrow \mu - b_u + \sigma\Phi^{-1}(\alpha) \le 0$$

where $\Phi$ is the cumulative normal distribution function.

Using this approximation, the stochastic program (P1) is reduced to the following integer nonlinear monotonic program:

$$\begin{cases} \text{maximize } f(x) = \sum_{i=1}^{n} c_i x_i \\ \text{subject to } g(x) = -b_u + \mu + \sigma \Phi^{-1}(\alpha) \leq 0 \\ x \in X = \{x \in \mathbb{Z}_+^n \mid l_i \leq x_i \leq d_i,\ i = 1, ..., n\} \end{cases} \tag{P2}$$

Taking into account the expressions (2), the constraint in (P2) can be written as:

$$g(x) = -b_u + \sum_{i=1}^{n} p_i x_i + \Phi^{-1}(\alpha) \sqrt{\sum_{i=1}^{n} v_i x_i} \leq 0$$

$g$ is a convex and increasing function of $x_i$ on $[l_i, d_i]$ (see, for example, Shing and Nagasawa [16]).

Note that additional linear or nonlinear increasing constraints can be added to the model (P1) and reported in (P2) without influencing the resolution process.

## 3 Characteristics of monotonic programs

In the following, we review some properties of monotonic functions from the general results in [18, 19]. We also include some proofs, although they can be found in these references.

For any two vectors $x, y \in \mathbb{R}^n$ we write $x \leq y$ to mean $x_i \leq y_i$ for every $i = 1, \ldots, n$. If $l \leq d$ then the box $[l, d]$ is the set of all $x \in \mathbb{R}^n$ satisfying $l \leq x \leq d$. When $x \leq y$ we also say that $y$ dominates $x$.A function $f : \mathbb{R}_+^n \longmapsto \mathbb{R}$ is said to be increasing if $f(y) \geq f(x)$ whenever $y \geq x \geq 0$. A set $S \subset [l, d]$ is said to be normal if

$$l \leq x \leq y\ ,\ \ y \in S \implies x \in S \tag{3}$$

The normal hull of $S$ is the smallest normal set containing $S$.

**Proposition 1.** *The normal hull of $S$ is the set $S^{\urcorner} = \bigcup_{z \in S} [l, z]$. If $S$ is compact so is $S^{\urcorner}$.*

*Proof.* Let $P = \bigcup_{z \in S} [l, z]$. $P$ is normal and $P \supset S$, hence $P \supset S^{\urcorner}$. Conversely, if $x \in P$ then $x \in [l, z]$ for some $z \in S \subset S^{\urcorner}$, hence $x \in S^{\urcorner}$ by normality of $S^{\urcorner}$, so that $P \subset S^{\urcorner}$ and, therefore $P = S^{\urcorner}$. If $S$ is compact then $S$ is contained in a ball B centered at 0, and if $x^k \in S^{\urcorner}$, $k = 1, \ldots$ then since $x^k \in \left[l, z^k\right] \subset B$, there exists a subsequence $\{k_\nu\} \subset \{1, 2, \ldots\}$ such that $z^{k_\nu} \longrightarrow z^0 \in S,\ x^{k_\nu} \longrightarrow x^0 \in \left[l, z^0\right]$, hence $x^0 \in S^{\urcorner}$, proving the compactness of $S^{\urcorner}$.

**Definition 1.** *A polyblock $P$ is the normal hull of a finite set $T \subset [l, d]$ called its vertex set.*

By proposition 1, $P = \bigcup_{z \in T} [l, z]$. The intersection of finitely many polyblocks is a polyblock. A vertex $z$ of a polyblock is proper if there is no vertex $z' \neq z$ "dominating" $z$ i.e. such that $z' \geq z$ and improper otherwise. Improper vertices can be deleted without changing the polyblock, so a polyblock is fully determined by its proper vertices.

**Proposition 2.** *The maximum of an increasing function f over a polyblock is achieved at a proper vertex of this polyblock.*

*Proof.* Let $x^*$ be a maximizer of $f(x)$ over a polyblock $P$. Since a polyblock is the normal hull of its proper vertices, there exists a proper vertex $z$ of $P$ such that $x^* \in [l, z]$. Then $f(z) \geq f(x^*)$ because $z \geq x^*$, so $z$ must be also an optimal solution.

## 4 The discrete polyblock method

Consider the following optimization program:

$$\begin{cases} max\ f(x) \\ subject\ to\ \ g_i(x) \leq 0, \ \ i = 1, \ldots, m \\ x \in X = \{x \in Z_+^n \mid l_j \leq x_j \leq d_j\ ;\ j = 1, ..., n\} \end{cases}$$

From property (3) and the monotonocity of the $g_i's$, the set $S = \{x \in X \mid g_i(x) \leq 0\ , i = 1, ..., m\}$ defined above is normal. Let

$$G(x) = \max_{i=1,\ldots,m} \{g_i(x)\} \tag{4}$$

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 $\beta \notin S$. Since $G(\alpha) < 0$ and $G(\beta) > 0$, there must exist a point $x_b$ in $X$ that satisfies $G(x_b) = 0$ ( i.e., $g_i(x_b) \leq 0$, $i = 1, ..., m$ and there exists at least one $i$ such that $g_i(x_b) = 0$ ). $x_b$ is the intersection point of the line $x = \lambda^* \alpha + (1 - \lambda^*)\beta,\ 0 \leq \lambda^* \leq 1$ and the boundary $\Gamma$, where

$$\lambda^* = \sup\{\lambda \in [0, 1] \mid \lambda\alpha + (1 - \lambda)\beta \in S\} \tag{5}$$

Bisection method or Newton's method can be used to find the boundary point $x_b$.

Suppose that $x_b$ is not integral. Denote by $\lfloor x_b \rfloor$ the integer vector with its $j$-th component being the maximum integer less than or equal to $x_{b,j}$, $j = 1, ..., n$ and denote by $\lceil x_b \rceil$ the integer vector with its $j$-th component being the minimum integer greater than or equal to $x_{b,j}$, $j = 1, ..., n$.

Let $x^F = \lfloor x_b \rfloor$ and $x^I = \lceil x_b \rceil$. It is easy to see that $x^F$ is a feasible point ($x^F \in S$) and $x^I$ is infeasible ($x^I \notin S$). The monotonicity of $f$ and $g_i$ implies that there are no feasible points better than $x^F$ in $\langle \alpha, x^F \rangle$ and there are no feasible points in $\langle x^I, \beta \rangle$. Therefore, 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.

The following theorem which can be found in Xun and al. [17, p. 172] shows how to cut a revised domain into sub-boxes.

**Theorem 1.** *: 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.*

$$A \setminus B = \bigcup_{i=1}^{n} \left( \prod_{k=1}^{i-1} \langle \alpha_k, \gamma_k \rangle \ \langle \gamma_i + 1, \beta_i \rangle \times \prod_{k=i+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{6}$$

$$A \setminus C = \bigcup_{i=1}^{n} \left( \prod_{k=1}^{i-1} \langle \gamma_k, \beta_k \rangle \ \langle \alpha_i, \gamma_i - 1 \rangle \times \prod_{k=i+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{7}$$

The polyblock method consists of finding a feasible point $x^F$ and an infeasible point $x^I$ and generating integer boxes using the formulas (6) and (7). The best feasible solution obtained during the generation of integer boxes is kept as an incumbent solution. Moreover, by the monotonicity of the problem, an integer box with the function value of its upper bound point less than to the function value of the incumbent $x^F$ can be discarded.

### 4.1 Algorithm

Step 0 : (Initialization).

Let $l = (l_1, ..., l_n)$, $d = (d_1, ..., d_n)$.

If $l$ is infeasible, then problem (P2) has no feasible solution;

If $d$ is feasible, then $d$ is the optimal solution to (P2), 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 a box $\langle \alpha, \beta \rangle \in X^k$ with the maximum objective function value of the upper bound point. Set $X^k = X^k \setminus \langle \alpha, \beta \rangle$.

Find the root $\lambda^*$ of the following equation:

$$\lambda^* = \sup\{\lambda \in [0, 1] \mid \lambda\alpha + (1 - \lambda)\beta \in S\}$$

Set $x_b = \lambda^*\alpha + (1 - \lambda^*)\beta$ and $x^F = \lfloor x_b \rfloor$.

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 (7) 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 (6) to partition set $\Omega_2 = \langle \widehat{\alpha}, \widehat{\beta} \rangle \setminus \langle \widehat{\alpha}, x^F \rangle$.

(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 $\beta$ is feasible, remove $\langle \alpha, \beta \rangle$ from $Y^k$. Furthermore if $f(\beta) > f(x_{opt})$ set $x_{opt} = \beta$.

(b) If $\alpha$ 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 $\alpha$ is feasible, $\beta$ 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).

Remove 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.

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 $X^k$. The algorithm proceeds successively by refining the partition and removing integer boxes that do not contain promising solutions and finally terminates in a finite number of iterations.

*Remark 1.* As said in [17], two box-selection strategies can be used in Step 1. The first strategy is to select the integer box in $X^k$ with the maximum objective function value of the upper bound point. The second strategy is to select the last integer box included in $X^k$.

### 4.2 Illustrative example

Let us consider the following example with a similar structure to that of problem (P1), with : $\alpha = 0,998$ ; $b_u = 18$ ; $p_1 = 0.80$ ; $p_2 = 0.90$ ; $\beta_1 = 400$ ; $\beta_2 = 200$ ; $l_1 = 1$ ; $d_1 = 5$ ; $l_2 = 9$ ; $d_2 = 15$.

The termination step in the bisection method is $\varepsilon = 0,002$.

Problem (P2) is formulated as :

$max\ f(x) = 320x_1 + 180x_2$

subject to

$0.80x_1 + 0.90x_2 + 2,9\sqrt{0.16x_1 + 0.09x_2} - 18 \leq 0$

$x_1x_2 - 3x_1 - x_2 - 20 \leq 0$

$6x_1 + x_2 - 35 \leq 0$

$x \in X = \left\{ x \in Z^2 \mid 1 \leq x_1 \leq 5\ ,\ 9 \leq x_2 \leq 15 \right\}$

The iterations of the algorithm are described as follows:

**Iteration 1**:

let $l = (1, 9)$ ; $d = (5, 15)$ ; $x_{opt} = (1, 9)$ ; $f_{opt} = 1940$

$X^1 = \langle l, d \rangle = \langle (1, 9), (5, 15) \rangle$ ; $k = 1$.

Select $\langle \alpha, \beta \rangle = \langle l, d \rangle = \langle (1, 9), (5, 15) \rangle$ and set $X^1 = X^1 \setminus \langle \alpha, \beta \rangle = \varnothing$.

$g_1\left[\lambda(1, 9) + (1 - \lambda)(5, 15)\right] = 0 \Longrightarrow -0.5 - 8.6\lambda + 2.9\sqrt{2.15 - 1.18\lambda} = 0$

$g_2\left[\lambda(1, 9) + (1 - \lambda)(5, 15)\right] = 0 \Longrightarrow 24\lambda^2 - 72\lambda + 25 = 0$

$g_3\left[\lambda(1, 9) + (1 - \lambda)(5, 15)\right] = 0 \Longrightarrow 10 - 30\lambda = 0$

The bisection procedure finds out $\lambda_1 = 0.3789$ ; $\lambda_2 = 0.4023$ ; $\lambda_3 = 0.3320$

$\lambda^* = 0.4023$ and $x_b = (3.3908\ , 12.586)$ ;

$x^F = \lfloor x_b \rfloor = (3, 12)$ and $x^I = \lceil x_b \rceil = (4, 13)$.
Since $f(x^F) = 3120 > f_{opt} = 1940$ set $x_{opt} = (3, 12)$ and $f_{opt} = 3120$.

Partition the set $\Omega_1 = \langle \alpha, \beta \rangle \setminus \langle x^I, \beta \rangle$ into two integer boxes.
$\Omega_1 = \langle (1, 9), (5, 15) \rangle \setminus \langle (4, 13), (5, 15) \rangle = \{ \langle (1, 9), (3, 15) \rangle ; \langle (4, 9), (5, 12) \rangle \}$
Since $(3, 12) \in \langle (1, 9), (3, 15) \rangle$ set $\Omega_1 = \langle (4, 9), (5, 12) \rangle$ .
$\Omega_2 = \langle (1, 9), (3, 15) \rangle \setminus \langle (1, 9), x^F \rangle = \langle (1, 9), (3, 15) \rangle \setminus \langle (1, 9), (3, 12) \rangle = \langle (1, 13), (3, 15) \rangle$
Set $Y^1 = \Omega_1 \cup \Omega_2 = \{ \langle (4, 9), (5, 12) \rangle ; \langle (1, 13), (3, 15) \rangle \}$
We get $Z^1 = Y^1$.

Set $X^2 = X^1 \cup Z^1 = \{ \langle (4, 9), (5, 12) \rangle ; \langle (1, 13), (3, 15) \rangle \}$.

**Iteration 2**:
Since $f(5, 12) > f((3, 15)$, select $\langle \alpha, \beta \rangle = \langle (4, 9), (5, 12) \rangle$ and set $X^2 = \langle (1, 13), (3, 15) \rangle$.
$g_1 [\lambda(4, 9) + (1 - \lambda)(5, 12)] = 0 \Longrightarrow -3.2 - 3.5\lambda + 2.9\sqrt{1.88 - 0.43\lambda} = 0$
$g_2 [\lambda(4, 9) + (1 - \lambda)(5, 12)] = 0 \Longrightarrow 3x^2 - 23x + 13 = 0$
$g_3 [\lambda(4, 9) + (1 - \lambda)(5, 12)] = 0 \Longrightarrow 7 - 9x = 0$
The bisection procedure finds out $\lambda_1 = 0.1992$ ; $\lambda_2 = 0.6133$ ; $\lambda_3 = 0.7773$
$\lambda^* = 0.7773$ and $x_b = (4.2227, 9.6681)$ ;
$x^F = \lfloor x_b \rfloor = (4, 9)$ and $x^I = \lceil x_b \rceil = (5, 10)$.
$f(x^F) = 2900 < f_{opt}$ , set $x_{opt} = (3, 12)$ and $f_{opt} = 3120$.

Partition the set $\Omega_1 = \langle \alpha, \beta \rangle \setminus \langle x^I, \beta \rangle$ into two integer boxes.
$\Omega_1 = \langle (4, 9), (5, 12) \rangle \setminus \langle (5, 10), (5, 12) \rangle = \{ \langle (4, 9), (4, 12) \rangle ; \langle (5, 9), (5, 11) \rangle \}$
Since $x^F = (4, 9) \in \langle (4, 9), (4, 12) \rangle$, set $\Omega_1 = \langle (5, 9), (5, 11) \rangle$
$\Omega_2 = \langle (4, 9), (4, 12) \rangle \setminus \langle (4, 9), x^F \rangle = \langle (4, 9), (4, 12) \rangle \setminus \langle (4, 9), (4, 9) \rangle = \langle (4, 10), (4, 12) \rangle$
Set $Y^2 = \Omega_1 \cup \Omega_2 = \{ \langle (5, 9), (5, 11) \rangle ; \langle (4, 10), (4, 12) \rangle \}$
$(5, 9)$ is infeasible remove $\langle (5, 9), (5, 11) \rangle$ from $Y^2$ we get $Z^2 = \langle (4, 10), (4, 12) \rangle$
For $\langle (1, 13), (3, 15) \rangle \in X^2$ we have $f(3, 15) = 3660 > f_{opt}$
Then $X^3 = X^2 \cup Z^2 = \{ \langle (1, 13), (3, 15) \rangle ; \langle (4, 10), (4, 12) \rangle \}$.

**Iteration 3**:
Since $f(3, 15) > f((4, 12)$, select $\langle \alpha, \beta \rangle = \langle (1, 13), (3, 15) \rangle$ and set $X^3 = \langle (4, 10), (4, 12) \rangle$.
$g_1 [\lambda(1, 13) + (1 - \lambda)(3, 15)] = 0 \Longrightarrow -2.1 - 3.4\lambda + 2.9\sqrt{1.83 - 0.5\lambda} = 0$
$g_2 [\lambda(1, 13) + (1 - \lambda)(3, 15)] = 0 \Longrightarrow 4x^2 - 28x + 1 = 0$
$g_3 [\lambda(1, 13) + (1 - \lambda)(3, 15)] = 0 \Longrightarrow -2 - 14x = 0$
The bisection procedure finds out $\lambda_1 = 0.4648$ ; $\lambda_2 = 0.0352$ ; $\lambda_3 = 0.9961$
$\lambda^* = 0.9961$ and $x_b = (1.0078, 13.008)$;
$x^F = \lfloor x_b \rfloor = (1, 13)$ and $x^I = \lceil x_b \rceil = (2, 14)$.
$f(x^F) = 2660 < f_{opt}$ , set $x_{opt} = (3, 12)$ and $f_{opt} = 3120$.

Partition the set $\Omega_1 = \langle \alpha, \beta \rangle \setminus \langle x^I, \beta \rangle$ into two integer boxes.
$\Omega_1 = \langle (1, 13), (3, 15) \rangle \setminus \langle (2, 14), (3, 15) \rangle = \{ \langle (1, 13), (1, 15) \rangle ; \langle (2, 13), (3, 13) \rangle \}$
Since $x^F = (1, 13) \in \langle (1, 13), (1, 15) \rangle$, set $\Omega_1 = \langle (2, 13), (3, 13) \rangle$
$\Omega_2 = \langle (1, 13), (1, 15) \rangle \setminus \langle (1, 13), x^F \rangle = \langle (1, 14), (1, 15) \rangle$
Set $Y^3 = \Omega_1 \cup \Omega_2 = \{ \langle (2, 13), (3, 13) \rangle ; \langle (1, 14), (1, 15) \rangle \}$
$(3, 13)$ is feasible and $f(3, 13) = 3300 > f_{opt} = 3120$
Set $x_{opt} = (3, 13)$ and $f_{opt} = 3300$. Remove $\langle (2, 13), (3, 13) \rangle$ from $Y^3$.

$f(1,15) = 3020 < f_{opt} = 3300$, remove $\langle (1,14),(1,15)\rangle$ from $Y^3$.

We obtain $Z^3 = \emptyset$.
For $\langle (4,10),(4,12)\rangle \in X^3$, we have $f(4,12) = 3440 > f_{opt} = 3300$
Set $X^4 = X^3 \cup Z^3 = \langle (4,10),(4,12)\rangle$.

**Iteration 4**:
select $\langle \alpha, \beta\rangle = \langle (4,10),(4,12)\rangle$ and set $X^4 = \emptyset$.
$g_1\left[\lambda(4,10) + (1-\lambda)(4,12)\right] = 0 \Longrightarrow -4 - 1.8\lambda + 2.9\sqrt{1.72 - 0.18\lambda} = 0$
$g_2\left[\lambda(4,10) + (1-\lambda)(4,12)\right] = 0 \Longrightarrow 4 - 6x = 0$
$g_3\left[\lambda(4,10) + (1-\lambda)(4,12)\right] = 0 \Longrightarrow 1 - 2x = 0$
The bisection procedure finds out $\lambda_1 = 0.9961$ ; $\lambda_2 = 0.6680$ ; $\lambda_3 = 0.9961$,
$\lambda^* = 0.9961$ and $x_b = (4, 10.008)$;
$x^F = \lfloor x_b \rfloor = (4,10)$ and $x^I = \lceil x_b \rceil = (5,11)$.
$f(x^F) = 3080 < f_{opt}$
$\Omega_1 = \langle \alpha, \beta\rangle \setminus \langle x^I, \beta\rangle = \langle (4,10),(4,12)\rangle \setminus \langle (5,11),(4,12)\rangle = \langle (4,10),(4,12)\rangle$
Since $x^F = (4,10) \in \langle (4,10),(4,12)\rangle$ set $\Omega_1 = \emptyset$
$\Omega_2 = \langle (4,10),(4,12)\rangle \setminus \langle (4,10),x^F\rangle = \langle (4,10),(4,12)\rangle \setminus \langle (4,10),(4,12)\rangle = \emptyset$
Finally, we get $Z^4 = Y^4 = \varnothing$ and $X^5 = \varnothing$.

Stop, the incumbent $x_{opt} = (3,13)$ is the optimal solution to the problem with $f_{opt} = 3300$.

## 5 Conclusion

The discrete polyblock method for the solution of integer stochastic programming problems of expectation type is considered in this paper. The disadvantage of this procedure relates to the fact that the algorithm apply only for monotone functions. There are several advantages of the technique however. The information to be supplied by the decision maker is at its least. The method is simple to use since it exploit only the monotonic properties of the objective and constraints functions and no linearization or duality property is needed which distinguishes it from other known methods. Future research will be devoted to the testing of the approach, both from the algorithmic and the DSS design point of view.

## References


1. Abhishek K., Leyffer S. and Linderoth J. FilMINT: an outer approximation-based solver for convex mixed-integer nonlinear programs, INFORMS Journal of Computing, 22, (2010), 555-567.
2. Bonami P., Kilinc M., et Linderoth J. T., Algorithms and Software for Solving Convex Mixed Integer Nonlinear Programs. Dans Mixed Integer Nonlinear Programming, IMA Volumes in Mathematics and its Applications, Volume 154, éditeurs Jon Lee et Sven Leyffer, Springer New York, (2012), 1-39.
3. Bruggink T.H., and Gambhir V., Statistical models for college admission and enrollment: A case study for a selective liberal arts college, Research in Higher Education 37, (1996), 221–240.

4. Charnes A., Cooper W.W., Deterministic equivalents for optimizing and satisfying under chance constraints, Operations Research 11, (1963), 18–39.
5. DePaolo C.A., A stochastic programming model for optimal college enrollments, Doctoral Dissertation, Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ, (2001).
6. Duran M. A. and Grossmann I., An outer-approximation algorithm for a class of mixed-integer nonlinear programs, Mathematical Programming 36, (1986), 307–339.
7. Fletcher R. and Leyffer S., Solving mixed integer nonlinear programs by outer approximation, Mathematical Programming 66, (1994),327–349.
8. Geoffrion A., Generalized Benders decomposition, Journal of Optimization Theory and Applications 10, (1972) 237–260.
9. Kall P., Wallace S.W., Stochastic Programming, John Wiley and Sons, Chichester, (1994).
10. Kelley J. E., The cutting plane method for solving convex programs, Journal of SIAM 8, (1960) ,703–712.
11. Lee S.M., Moore L.J., Optimizing university admissions planning, Decision Sciences 5, (1974), 405–414.
12. Liu B., Iwamura K., Modelling stochastic decision systems using dependent-chance programming, European Journal of Operational Research 101, (1997), 193–203.
13. Phuong N. T. H. and Tuy, H., A unified monotonic approach to generalized linear fractional programming, Journal of Global Optimization, vol. 26, (2004), 229–259.
14. Prekopa A., Stochastic Programming, Kluwer Academic Publishers, Dordrecht, (1995).
15. Quesada I. and Grossmann I. E., An LP/NLP based branch-and-bound algorithm for convex MINLP optimization problems, Computers and Chemical Engineering 16, (1992), 937–947.
16. Shing C., Nagasawa H., Interactive decision system in stochastic multi-objective portfolio selection, International Journal of Production Economics 60-61, (1999), 187–193.
17. Sun X. L. and Li J. L., Nonlinear integer programming, Springer, (2006).
18. Tawarmalani M., and Sahinidis, N. V., Global optimization of mixed integer nonlinear programs: A theoretical and computational study, Mathematical Programming 99, (2004), 563–591.
19. Tuy H., Monotonic optimization: Problems and solution approaches, SIAM Journal of Optimization 11 (2), (2000), 464–494.
20. Westerlund T. and Pettersson F., A cutting plane method for solving convex MINLP problems, Computers and Chemical Engineering, 19, (1995), 131–136.

# PSO for the two machines flow shop with coupled-tasks

Nadjat Meziani[1], Ammar Oulamara [2], and Mourad Boudhar[3]

[1]University of Abderrahmane Mira Bejaia, Algeria
[2]University of Lorraine, Metz, France
[3]USTHB University Algiers, Algeria
ro_nadjet07@yahoo.fr, Ammar.Oulamara@loria.fr and mboudhar@yahoo.fr

**Abstract.** In this work, we are interested to solve the flow shop problem on two machines with coupled-tasks such as each task is composed of two operations on the first machine and one operation on the second machine in order to minimize the makespan. The two operations on the first machine are separated by an exact time lag. The proposed method consists in the application of the particle swarm optimization (PSO) in which we have introduced the local search method to improve its performance. Numerical experiments are conducted and their results are compared with those of the other heuristics used to solve the same problem.

**Key words:** particle swarm optimization, scheduling, makespan, coupled-tasks, heuristics.

## 1 Introduction

The flow shop scheduling problem which consists to deal $n$ tasks on $m$ machines in the same order is one of the frequently problem encountered an industrial processes workshop, manufacturing systems and assembly workshops. In the literature, the authors studied several types of the flow shop problems with different variants and proposed exact methods, heuristics and metaheuristics for their resolution. In this article, we present a metaheuristic, particle swarm optimization (PSO) to solve the flow shop scheduling problem with coupled-tasks. The problem consists to deal $n$ tasks on two machines in order to minimizing the makespan such as each task is composed on two operations on the first machine separated by an exact time lag and only one operation on the second machine.

Many researches have reported on the application of this method in several areas and especially on combinatorial optimization problems such as scheduling problems in which we find few works in the literature. Tasgetiren et al. in [12] have proposed a PSO for both problems of permutation flow shop to minimize the makespan and maximum lateness respectively. Also, to solve the permutation flow shop to minimize the makespan, Zhingang Lia et al. [14] presented the PSO algorithm combined with the crossover operator and in [16] have proposed a new

PSO combined with mutation and crossover operators. In [13], Xianpeng et al. presented an algorithm for PSO to solve the same problem with the blocking constraint to minimize the makespan. For the no-wait flow shop problem, authors in [9] introduced a discrete PSO in order to minimize the makespan and the total flow.

## 2 Description of the problem

The coupled-tasks scheduling problem was introduced for the first time by Shapiro in [10]. Each coupled-tasks consists in two different operations which are carried out on one machine in the order, separated by an time interval known as a time lag. A coupled-tasks is noted by the triplet $(a_i, L_i, bi)$, represent the processing time of the first operation $(a_i)$, the time lag which runs out between the completion time of the first operation and the starting processing of the second operation $(L_i)$ and the processing time of the second operation $(b_i)$. During the time lag, the machine is inactive and another task can be treated.

The motivation of the coupled-tasks problem stems from a scheduling problem of radar tasks which consists in the emission of the pulses and the reception of answers after the time interval. This problem appears also in workshops chemical productions where one machine must carry out several operations of the same task and an exact delay is imposed between the execution of each two consecutive operations due to the chemical reactions.

Orman et al. studied in [8] the coupled-tasks problem with one machine in order to minimize the $C_{max}$. As these problems are difficult, the problem $1/Coup - Task, a_i = a, L_i = l, b_i = b/C_{max}$ was left open by these authors and for which others were interested. In [1], Ahr et al. proposed an exact algorithm using the dynamic programming which allows to resolve the problem for small instances where $L$ is fixed. This algorithm was adapted by Brauner et al. in [3] to resolve a coupled-tasks problem motivated by the time management problems of cyclic production with robots. Few works were realized by adding constraints to the coupled-tasks problem. Blazewicz et al [2] proved that the polynomial problem $1/Coup - task, a_i = b_i = 1, L_i = l/C_{max}$ is NP-hard by adding a precedence constraint between the coupled-tasks. In [15], Yu et al. proved that the problem on two machines $F2/Coup - Task, a_i = b_i = 1, L_i/C_{max}$ is NP-hard. Simonin et al. studied in [11] the coupled tasks problem in the presence of the treatment tasks. In [7], Meziani et al. studied the flow shop problem on two machines with the coupled tasks on the first machine. They proved that the problem $F2/Coup - Task(1), a_i, b_i = L_i = p, c_i/C_{max}$ is NP-hard and they developed heuristics based in the interleaving of tasks to solve the problem with numerical experiments. They also presented some polynomial subproblems. In the following of the article, we use the discrete PSO method to resolve the problem $F2/Coup - Task(1), a_i, b_i = L_i = p, c_i/C_{max}$ studied in [7]. Then, the problem consists to process $n$ tasks on two machines where each task is composed

of two operations of $a_i$ and $b_i$ processing times separated by an exact time lag $L_i$ on the first machine and one operation of $c_i$ processing time on the second machine. Our objective is to minimize the makespan $C_{max}$.

## 3 An introduction to PSO

The particle swarm optimization (PSO) is an optimization method recently introduced by Eberhat and Kennedy [5]. This method is inspired by the collective behavior of insects colonies such as fish schools and birds flocks. Indeed, it is surprising how these animals move in one direction, sometimes splitting into two groups to avoid a predator and reforming the original group. Each individual uses local information to which it can access on the displacement of its nearest neighbors to decide on its own movement. Simple rules such as "stay relatively close to other individuals", "move in the same way", "go at the same speed", are sufficient to maintain group cohesion and allow complex collective behaviors and adapted.
The PSO is designed for optimizing continuous nonlinear functions. It based on a population called swarm, consisting of a set of individuals randomly arranged initially. Each individual is a particle representing a potential solution to the problem optimization and assuming that the fitness function is called $Z$ which to be minimized. Let $N_p$ the number of particles or the size of the swarm. Each particle $k$ can be represented by the following characteristics:

1. Each particle $k$ is represented in $n$-dimension search space by its position vector $X_k^t = (X_{k1}^t, X_{k2}^t, \ldots, X_{kn}^t)$ and its velocity vector $V_k^t = (V_{k1}^t, V_{k2}^t, \ldots, V_{kn}^t)$ at iteration $t$, where $X_{kd}^t$: the position of the particle $k$ at the dimension $d = \overline{1,n}$ and $V_{kd}^t$: the velocity of the particle $k$ at the dimension $d = \overline{1,n}$.
2. Each particle $k$ remember its best position visited until iteration $t$ denoted by $P_k^t = (P_{k1}^t, P_{k2}^t, \ldots, P_{kn}^t)$.
3. Each particle $k$ remember the best position of the best particle in the swarm denoted by $P_g^t = (P_{g1}^t, P_{g2}^t, \ldots, P_{gn}^t)$.
4. Each particle update its velocity vector value at iteration $t$ using the following formula: $V_{kd}^t = wV_{kd}^{t-1} + c_1 r_1 (P_{kd}^{t-1} - X_{kd}^{t-1}) + c_2 r_2 (P_{gd}^{t-1} - X_{kd}^{t-1})$;
   $w$: inertia weight that defines the capacity of each particle to improve the convergence of the method.
   $c_1, c_2$: two positives constants used to decide whether the particles prefer moving toward to $P_k^t$ or $P_g^t$ position.
   $r_1, r_2$: random real numbers uniformly distributed in $[0, 1]$.
5. The position vector of the particle $k$ at iteration $t$ is given by: $X_{kd}^t = X_{kd}^{t-1} + V_{kd}^t$;

6. The best position of each particle is updating using:

$$P_k^t = \begin{cases} P_k^{t-1} \text{ if } Z(X_k^t) \geq Z(P_k^{t-1}) \\ X_k^t \quad \text{if } Z(X_k^t) < Z(P_k^{t-1}) \end{cases}$$

7. And the global best position found so far in the swarm population is obtained as:

$$P_g^t = \begin{cases} argmin_{p_k^t} \quad Z(P_k^t) \text{ if } \; Z(P_k^t) < Z(P_g^{t-1}) \\ P_g^{t-1} \qquad\qquad\qquad \text{otherwise} \end{cases}$$

The application of this method is not limited on continuous space search since many optimization problems are defined on discrete domains. However, Kennedy and Eberhat have developed [6] a discrete version of the PSO which differs from the first in two characteristics. The first characteristic, the particle is encoded into binary variables $\{0, 1\}$. The second characteristic, the velocity must be transformed into a probability reflecting the chance that the binary variable takes the value "1". Subsequently, especially in workshops scheduling, other authors have proposed different versions of the discrete PSO for some problems cited in [9][13].

## 4 The proposed method

### 4.1 Solution representation

One of the key issues in designing the PSO algorithm lies in its solution representation where particles bear the necessary information related to the problem domain or hand. In our proposed PSO algorithm, each particle is represented by an array whose length is equal to the number of tasks, in which each element indicates the position of the task on the machine.

| task | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ |
|---|---|---|---|---|---|---|
| position | 4 | 1 | 6 | 5 | 3 | 2 |

**Fig. 1.** The solution representation

### 4.2 Update of particle

The behavior of a particle is a compromise between three choices: find its own position ($X_k^t$), to go towards its personnel best position ($P_k^t$) and to go towards the best position of the particle in the whole population ($P_g^t$). The position of the particle at iteration $t$ can be updated as follows:
$X_i^t = c_2 \otimes F_3(c_1 \otimes F_2(w \otimes F_1(X_k^{t-1}), P_k^{t-1}), P_g^{t-1})$

The update equation consists of three components. The first component is $\lambda_k^t = w \otimes F_1(X_k^{t-1})$ which represents the velocity of the particle and $F_1$ is the mutation operator with the probability of $w$ such as:

$$\lambda_k^t = \begin{cases} F_1(X_k^{t-1}) \text{ if } \; r \leq w \\ X_k^{t-1} \qquad\quad \text{otherwise} \end{cases}$$

with $r$ a random real number uniformly distributed in $[0, 1]$.

The second component is $\delta_k^t = c_1 \otimes F_2(\lambda_k^t, P_k^{t-1})$, which is the cognition part of the particle. In this component, $F_2$ represents the crossover operator with the probability of $c_1$. Note that $\lambda_k^t$ and $P_k^{t-1}$ will be the first and second parent for the crossover operator, respectively. It results either in $\delta_k^t = F_2(\lambda_k^t, P_k^{t-1})$ or $\delta_k^t = \lambda_k^t$ depending on the choice of a uniform random number.

The third component is $X_k^t = c_2 \otimes F_3(\delta_k^t, P_g^{t-1})$ which is the social part of the particle. In this component, $F_3$ represents the crossover operator with the probability of $c_2$. Note that $\delta_k^t$ and $P_g^{t-1}$ will be the first and second parents for the crossover operator, respectively. Then $X_k^t = F_3(\delta_k^t, P_g^{t-1})$ or $X_k^t = \delta_k^t$ depending on the choice of a uniform random number.

In this article, we apply the gbest model of Kennedy and Eberhart for the discrete PSO. The pseudocode of the DPSO algorithm is given in algorithm1.

```
Begin
   Initialize parameters
   Initialize population
   Evaluate
   Repeat
     Find the personal best (P_i^t)
     Find the global best (P_g^t)
     For each particle
       Apply velocity component
       Apply cognition component
       Apply social component
       Calculate the fitness value
     Apply the local search to the global best
   Until( Maximum iteration is reached)
 End.
```

Algorithm1: DPSO algorithm with local search

### 4.3 Mutation

There are several formulas for mutation operators applied an the genetic algorithms, among them we choose the mutation in the reverse order that we applied to all particles. In sequence, we randomly generate the segment length in which we must reverse the order of the genes to find the new sequence. The following figure illustrates how we proceed with this operator.

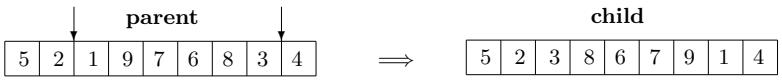



**Fig. 2.** Inversion Mutation

### 4.4 Crossover

In the equation calculation of the particle position, we propose to use mutation and crossover operators used in the genetic algorithms. The crossover operator generates new sequence by combining two other sequences or parents. An illustration of the crossover operator operation used is given in the following figure.

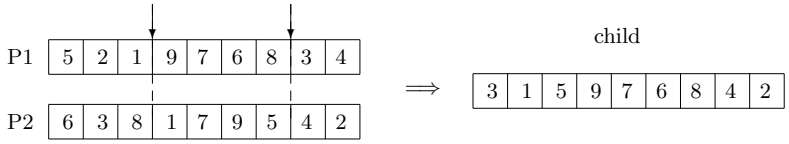



**Fig. 3.** One segment crossover

Two points crossover are randomly generated on the first parent ($P_1$). In the new sequence called child, we keep this block of ($P_1$) and the remaining elements are complemented by those of the second parent ($P_2$) in their order.

## 5 Local search

In this section, we introduce the local search method in the proposed discrete PSO to improve its performance. The local search algorithm used is the following:

**Begin**
$s_0 \longleftarrow X_g$;
$s \longleftarrow$ Swap$(i, j, s_0)$; iter:=0;
**While** $iter < n$ **do**
**begin**
$s_1 \longleftarrow$ Insert$(k, l, s)$;
**If** $Z(s_1) \leq Z(s_0)$ **then**
**begin**
$s \longleftarrow s_1$;
**end;**
iter:=iter+1;
**end;**
**If** $Z(s) \leq Z(s_0)$ **then**
**begin**
$X_g \longleftarrow s$;
**End;**
**End.**

The best solution found by the DPSO algorithm is used as an initial solution in the local search algorithm to diversify the solution. The performance of this method depends on the choice of the neighborhood structure for which we apply the two following procedures:

- Swap the position of two tasks $i$ and $j$ randomly chosen in the best solution vector.
- Insert a task of position $i$ in a new position $j$ in the best solution found vector.

The initial solution is replaced by the final solution $s$ found if the value of its fitness is better than the initial solution.

## 6 Heuristics

In this section, we propose heuristics based on the interleaving of tasks for solving the problem $F2/Coup-Task(1), a_i, L_i = b_i = p, c_i/C_{max}$ studied by Meziani et al. in [7].

**Heuristic** $H_1$

1. Construct the sets $K = \{J_i/a_i \leq p\}$ and $S = \{J_i/a_i > p\}$.
2. Apply Johnson rule[4] on the tasks of $S$.
3. Order the tasks of $K$ according to the $LPT$ rule of $a_i$.
4. Interleave a task of $K$ with one of $S$.
5. Interleave the remainder tasks of $K$ or order the remainder tasks of $S$ at the end of the schedule.

**Heuristic** $H_2$

1. Construct the sets $K = \{J_i/a_i \leq p\}$ and $S = \{J_i/a_i > p\}$.

2. Apply the rule of Johnson on the tasks of $S$.
3. Order the tasks of $K$ according to the $LPT$ rule of $a_i$.
4. Interleave tasks of the set $K$ between them; if there is one task of $K$ not interleaved, interleave it with one task of $S$.
5. Process tasks of $S$ according to Johnson rule[4].

**Heuristic** $H_3$

1. Construct the sets $K = \{J_i/a_i \leq p\}$ and $S = \{J_i/a_i > p\}$.
2. Order the tasks of $K$ and $S$ according the $LPT$ rule of $c_i$.
3. Interleave a task of $K$ with one of $S$.
4. If it remains tasks of $K$, interleave between them and if it remains tasks of $S$, process them according to their order at the end of the scheduling.

## 7 Numerical experimentations

To evaluate the performance of the proposed metaheuristic and heuristics, we generated 100 instances uniformly distributed. For each instance, the number of tasks and processing times take their values in $\{10, 20, 50, 100, 250\}$ and $[1.50], [1.100], [50.\ 100]$ respectively, then we varies the value of the time lag ($L_i = p$) by assigning it the following values: $10, 30, 50, 70$ and $100$. For the PSO, we fix the number of iterations to 50 as a stopping criterion, the number of particles $N_p = 50$ and probabilities of mutation $w = 0.2$ and crossover $c_1 = c_2 = 0.8$. For each tasks $n$ value, we calculate the number of times that the total completion time $C_{max}$ is better, the lower bound coincides with the $C_{max}$, the average execution time ($\overline{time}$) of each heuristic (in milliseconds), the average deviation and the maximum deviation performance. This latter is given by $\overline{dev} = \frac{C_{max}(H)-LB}{LB}$ such that $LB$ is a lower bound given by:

$$LB = \max\left\{ LB_1 = \sum_{i=1}^{\frac{n_1}{2}}(a_i + 3p) + \sum_{i=1}^{n_2}(a_i + 2p) + \min_{1\leq i\leq n}\{c_i\}, LB_2 = \min_{1\leq i\leq n}\{a_i + 2p\} + \sum_{i=1}^{n} c_i \right\}$$

Where $n_1$ and $n_2$ are the number of the interleaved and non-interleaved tasks respectively.

According to the results reported in the above tables, for the processing time $a_i, c_i \in [1, 50], p = 10$ (table1) and $a_i, c_i \in [1, 100], p = 30$ (table2), the PSO provides better results for $C_{max}$ when $n = 10, 20$, and by increasing the number of tasks, the heuristic $H_3$ leads better values. The optimum is gotten by the PSO and the heuristic $H_3$ in the majority of cases. For $a_i, c_i \in [50, 100], p = 50$ (table3), the heuristic $H_3$ provides better values of $C_{max}$ compared to the PSO and the optimum coincides with the lower bound in many cases for both methods. By increasing the number of tasks, the number of times where $C_{max}$ is better increases by using the PSO for $a_i, c_i \in [1, 50], p = 30$ (table1) and $a_i, c_i \in [1, 100], p = 70$ (table2). For a small number of tasks $n = 10, 20$ and for $a_i, c_i \in [1, 50], p = 50$ (table1) and $a_i, c_i \in [50, 100], p = 100$ (table3), the best values of $C_{max}$ are provided by the PSO and with the increase of the number of tasks, the heuristics $H_1$ gives best results and the optimum is attained in some

**Table 1.** Results of the tests

| | | $a_i, c_i \in [1,50], p=10$ | | | | $a_i, c_i \in [1,50], p=30$ | | | | $a_i, c_i \in [1,50], p=50$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO |
| n=10 | $C_{max}$ | 0 | 0 | 96 | 100 | 0 | 0 | 48 | 100 | 23 | 0 | 0 | 100 |
| | opt | 0 | 0 | 70 | 70 | 0 | 0 | 30 | 51 | 13 | 0 | 0 | 50 |
| | $\overline{time}$ | 0.03 | 0.04 | 0.01 | 134.11 | 0.03 | 0.02 | 0.01 | 135.5 | 0.01 | 0.02 | 0.01 | 133.24 |
| | $\overline{dev}$ | 0.3801 | 0.4108 | 0.0076 | 0.0054 | 0.3922 | 0.4694 | 0.0226 | 0.0044 | 0.0221 | 0.1102 | 0.0664 | 0.0039 |
| | mdev | 0.4891 | 0.5448 | 0.1018 | 0.0557 | 0.5475 | 0.6367 | 0.1077 | 0.0388 | 0.05157 | 0.1473 | 0.1548 | 0.02867 |
| n=20 | $C_{max}$ | 0 | 0 | 96 | 99 | 0 | 0 | 19 | 99 | 18 | 0 | 0 | 93 |
| | opt | 0 | 0 | 85 | 83 | 0 | 0 | 12 | 38 | 4 | 0 | 0 | 20 |
| | $\overline{time}$ | 0.02 | 0.05 | 0.01 | 196.79 | 0.02 | 0.02 | 0.01 | 196.33 | 0.05 | 0.04 | 0.02 | 193.18 |
| | $\overline{dev}$ | 0.3806 | 0.4086 | 0.0008 | 0.0010 | 0.3839 | 0.4791 | 0.0243 | 0.0027 | 0.0137 | 0.0583 | 0.0756 | 0.0038 |
| | mdev | 0.4275 | 0.4525 | 0.0145 | 0.0201 | 0.5182 | 0.6329 | 0.0801 | 0.0183 | 0.0297 | 0.0741 | 0.1395 | 0.0178 |
| n=50 | $C_{max}$ | 0 | 0 | 100 | 89 | 0 | 0 | 7 | 99 | 76 | 0 | 0 | 28 |
| | opt | 0 | 0 | 87 | 77 | 0 | 0 | 3 | 8 | 2 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.09 | 0.04 | 0.04 | 443.39 | 0.11 | 0.01 | 0.06 | 473.84 | 0.13 | 0.02 | 0.02 | 450.95 |
| | $\overline{dev}$ | 0.3789 | 0.4096 | 0.0001 | 0.0005 | 0.3838 | 0.4964 | 0.0234 | 0.0038 | 0.0058 | 0.0239 | 0.0731 | 0.0088 |
| | mdev | 0.4134 | 0.4525 | 0.0015 | 0.0126 | 0.5029 | 0.6160 | 0.0567 | 0.01385 | 0.01191 | 0.0306 | 0.111 | 0.01741 |
| n=100 | $C_{max}$ | 0 | 0 | 100 | 76 | 0 | 0 | 1 | 100 | 100 | 0 | 0 | 0 |
| | opt | 0 | 0 | 94 | 72 | 0 | 0 | 1 | 3 | 1 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.09 | 0.08 | 0.09 | 1196.5 | 0.13 | 0.07 | 0.09 | 1263.1 | 0.13 | 0.12 | 0.11 | 1248.1 |
| | $\overline{dev}$ | 0.3793 | 0.4106 | 0.0001 | 0.0007 | 0.3871 | 0.5110 | 0.0209 | 0.0053 | 0.0027 | 0.0123 | 0.0751 | 0.0153 |
| | mdev | 0.4045 | 0.4385 | 0.0003 | 0.0104 | 0.04657 | 0.6211 | 0.0417 | 0.0166 | 0.00598 | 0.0149 | 0.1004 | 0.0266 |
| n=250 | $C_{max}$ | 0 | 0 | 100 | 74 | 0 | 0 | 0 | 100 | 100 | 0 | 0 | 0 |
| | opt | 0 | 0 | 95 | 70 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.34 | 0.3 | 0.36 | 7032.8 | 0.39 | 0.24 | 0.32 | 7061 | 0.52 | 0.41 | 0.48 | 7456.7 |
| | $\overline{dev}$ | 0.3811 | 0.4127 | 0.0001 | 0.0003 | 0.3836 | 0.5104 | 0.0227 | 0.0091 | 0.0011 | 0.0047 | 0.0758 | 0.0286 |
| | mdev | 0.3963 | 0.4297 | 0.0002 | 0.0041 | 0.4425 | 0.5844 | 0.03976 | 0.0178 | 0.00237 | 0.0061 | 0.0917 | 0.03728 |

**Table 2.** Results of the tests

| | | $a_i, c_i \in [1,100], p=30$ | | | | $a_i, c_i \in [1,100], p=50$ | | | | $a_i, c_i \in [1,100], p=70$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO |
| n=10 | $C_{max}$ | 0 | 0 | 97 | 100 | 0 | 0 | 63 | 100 | 1 | 0 | 26 | 100 |
| | opt | 0 | 0 | 70 | 71 | 0 | 0 | 36 | 43 | 1 | 0 | 12 | 48 |
| | $\overline{time}$ | 0.03 | 0.02 | 0.01 | 134.26 | 0.02 | 0.01 | 0.02 | 133.59 | 0.03 | 0.03 | 0.02 | 130.08 |
| | $\overline{dev}$ | 0.4351 | 0.4825 | 0.0058 | 0.0047 | 0.4437 | 0.5138 | 0.0178 | 0.0075 | 0.3337 | 0.3833 | 0.03572 | 0.0048 |
| | mdev | 0.5428 | 0.5849 | 0.0815 | 0.0368 | 0.5420 | 0.6168 | 0.08 | 0.0497 | 0.5538 | 0.4051 | 0.1084 | 0.0356 |
| n=20 | $C_{max}$ | 0 | 0 | 95 | 97 | 0 | 0 | 67 | 98 | 0 | 0 | 10 | 100 |
| | opt | 0 | 0 | 74 | 72 | 0 | 0 | 38 | 47 | 0 | 0 | 2 | 26 |
| | $\overline{time}$ | 0.02 | 0.03 | 0.02 | 197.2 | 0.04 | 0.03 | 0.02 | 202.2 | 0.04 | 0.01 | 0.01 | 204 |
| | $\overline{dev}$ | 0.4326 | 0.4877 | 0.0008 | 0.0009 | 0.4416 | 0.5394 | 0.0082 | 0.0023 | 0.3134 | 0.3931 | 0.0362 | 0.0031 |
| | mdev | 0.4937 | 0.5472 | 0.0123 | 0.0114 | 0.5181 | 0.6212 | 0.0502 | 0.0229 | 0.5030 | 0.6373 | 0.1149 | 0.0186 |
| n=50 | $C_{max}$ | 0 | 0 | 100 | 92 | 0 | 0 | 56 | 95 | 0 | 0 | 1 | 100 |
| | opt | 0 | 0 | 81 | 75 | 0 | 0 | 34 | 38 | 0 | 0 | 1 | 2 |
| | $\overline{time}$ | 0.05 | 0.08 | 0.05 | 465.56 | 0.05 | 0.06 | 0.03 | 470.85 | 0.05 | 0.03 | 0.01 | 455.18 |
| | $\overline{dev}$ | 0.4319 | 0.4935 | 0.0009 | 0.0004 | 0.4402 | 0.5585 | 0.0055 | 0.0012 | 0.3153 | 0.4112 | 0.0359 | 0.0061 |
| | mdev | 0.4698 | 0.5305 | 0.0017 | 0.0144 | 0.4987 | 0.5948 | 0.0401 | 0.0136 | 0.4743 | 0.6071 | 0.0623 | 0.0159 |
| n=100 | $C_{max}$ | 0 | 0 | 100 | 88 | 0 | 0 | 54 | 88 | 0 | 0 | 0 | 100 |
| | opt | 0 | 0 | 79 | 70 | 0 | 0 | 37 | 35 | 0 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.15 | 0.09 | 0.01 | 1256.27 | 0.11 | 0.14 | 0.08 | 1242.3 | 0.08 | 0.09 | 0.12 | 1235.19 |
| | $\overline{dev}$ | 0.4337 | 0.4951 | 0.0003 | 0.0001 | 0.4453 | 0.5704 | 0.0033 | 0.0009 | 0.3140 | 0.4163 | 0.0358 | 0.0102 |
| | mdev | 0.4568 | 0.5143 | 0.0004 | 0.0051 | 0.4913 | 0.5997 | 0.0224 | 0.0087 | 0.4139 | 0.5455 | 0.0549 | 0.0195 |
| n=250 | $C_{max}$ | 0 | 0 | 100 | 73 | 0 | 0 | 51 | 90 | 0 | 0 | 0 | 100 |
| | opt | 0 | 0 | 84 | 59 | 0 | 0 | 40 | 36 | 0 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.45 | 0.23 | 0.27 | 7521.4 | 0.35 | 0.32 | 0.35 | 7519.22 | 0.38 | 0.32 | 0.35 | 7261.35 |
| | $\overline{dev}$ | 0.4335 | 0.4946 | 0.0001 | 0.0003 | 0.4414 | 0.5731 | 0.0029 | 0.0011 | 0.3114 | 0.4174 | 0.0386 | 0.0169 |
| | mdev | 0.4491 | 0.5127 | 0.0004 | 0.0029 | 0.4794 | 0.5929 | 0.0177 | 0.0075 | 0.03634 | 0.4921 | 0.0563 | 0.0261 |

**Table 3.** Results of the tests

| | | $a_i, c_i \in [50,100], p = 50$ | | | | $a_i, c_i \in [50,100], p = 70$ | | | | $a_i, c_i \in [50,100], p = 100$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO | $H_1$ | $H_2$ | $H_3$ | PSO |
| n=10 | $C_{max}$ | 0 | 0 | 100 | 98 | 0 | 0 | 84 | 100 | 21 | 0 | 0 | 99 |
| | opt | 0 | 0 | 87 | 95 | 0 | 0 | 60 | 60 | 13 | 0 | 0 | 51 |
| | $\overline{time}$ | 0.01 | 0.01 | 0.03 | 133.32 | 0.02 | 0.03 | 0.01 | 136.2 | 0.06 | 0.02 | 0.01 | 132.3 |
| | $\overline{dev}$ | 0.5499 | 0.5262 | 0.0008 | 0.0001 | 0.4804 | 0.5908 | 0.0029 | 0.0018 | 0.0109 | 0.1033 | 0.0320 | 0.0017 |
| | mdev | 0.5824 | 0.5562 | 0.0042 | 0.6140 | 0.6141 | 0.6641 | 0.0254 | 0.0176 | 0.0264 | 0.1201 | 0.0684 | 0.0159 |
| n=20 | $C_{max}$ | 0 | 0 | 100 | 94 | 0 | 0 | 88 | 100 | 19 | 0 | 0 | 88 |
| | opt | 0 | 0 | 90 | 89 | 0 | 0 | 56 | 60 | 5 | 0 | 0 | 23 |
| | $\overline{time}$ | 0.02 | 0.03 | 0.02 | 195.98 | 0.05 | 0.04 | 0.03 | 199.96 | 0.04 | 0 | 0.03 | 202.67 |
| | $\overline{dev}$ | 0.5606 | 0.5529 | 0.0001 | 0.0004 | 0.4968 | 0.6383 | 0.0011 | 0.0005 | 0.0061 | 0.0529 | 0.0327 | 0.0017 |
| | mdev | 0.5884 | 0.5881 | 0.0012 | 0.0014 | 0.5814 | 0.6895 | 0.0096 | 0.0039 | 0.0135 | 0.0612 | 0.0595 | 0.0086 |
| n=50 | $C_{max}$ | 0 | 0 | 100 | 83 | 0 | 0 | 88 | 95 | 76 | 0 | 0 | 26 |
| | opt | 0 | 0 | 92 | 75 | 0 | 0 | 63 | 58 | 4 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.08 | 0.04 | 0.03 | 461.7 | 0.03 | 0.06 | 0.03 | 466.52 | 0.09 | 0.09 | 0.01 | 475.5 |
| | $\overline{dev}$ | 0.5651 | 0.5667 | 0.0001 | 0.0001 | 0.5006 | 0.6591 | 0.0003 | 0.0001 | 0.00264 | 0.0216 | 0.0318 | 0.0042 |
| | mdev | 0.5853 | 0.5872 | 0.0008 | 0.0041 | 0.5571 | 0.6858 | 0.0057 | 0.0013 | 0.0053 | 0.0246 | 0.0503 | 0.0106 |
| n=100 | $C_{max}$ | 0 | 0 | 100 | 71 | 0 | 0 | 98 | 85 | 100 | 0 | 0 | 0 |
| | opt | 0 | 0 | 90 | 63 | 0 | 0 | 72 | 61 | 4 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.14 | 0.09 | 0.16 | 1192.5 | 0.05 | 0.1 | 0.07 | 1235 | 0.08 | 0.12 | 0.11 | 1240.8 |
| | $\overline{dev}$ | 0.5662 | 0.5711 | 0.0001 | 0.0002 | 0.5021 | 0.6682 | 0.0006 | 0.0002 | 0.0012 | 0.0109 | 0.0346 | 0.0072 |
| | mdev | 0.5793 | 0.5834 | 0.0003 | 0.0021 | 0.5389 | 0.6869 | 0.0034 | 0.0025 | 0.0026 | 0.0124 | 0.0463 | 0.0148 |
| n=250 | $C_{max}$ | 0 | 0 | 100 | 59 | 0 | 0 | 100 | 85 | 100 | 0 | 0 | 0 |
| | opt | 0 | 0 | 92 | 50 | 0 | 0 | 98 | 81 | 4 | 0 | 0 | 0 |
| | $\overline{time}$ | 0.57 | 0.44 | 0.41 | 7412.79 | 0.43 | 0.25 | 0.28 | 7090.3 | 0.47 | 0.42 | 0.39 | 7342 |
| | $\overline{dev}$ | 0.5674 | 0.5727 | 0.0001 | 0.0002 | 0.5077 | 0.6701 | 0.0001 | 0.0002 | 0.0005 | 0.0054 | 0.0348 | 0.0132 |
| | mdev | 0.5761 | 0.5820 | 0.0002 | 0.0008 | 0.5346 | 0.6835 | 0.0003 | 0.0003 | 0.0011 | 0.00493 | 0.0452 | 0.0168 |

cases by both methods. We note for $a_i, c_i \in [1,100], p = 50$ (table2) that PSO leads better values of $C_{max}$ compared to the other heuristics. For the number of tasks $n = 10, 20, 50$ and $a_i, c_i \in [50,100], p = 70$ (table3), we note that the PSO provides better results for $C_{max}$ which decreases with the increasing of tasks number that allow for the heuristic $H_3$ becomes the best. For this latest two cases, the optimum coincides with the lower bound by using PSO and the heuristic $H_3$ in the most cases. Also, we note that a metaheuristic (PSO) requires a higher execution time compared to other heuristics which is due to the number of the particles and iterations used in the method.

## 8 Conclusion

In this paper, we proposed a metaheuristic discrete particle swarm optimization (DPSO) to solve the problem of the coupled-tasks on two machines. The problem consists in a flow shop with two machines with coupled tasks on the first machine such as each task consists in two operations on the first machine separated by an exact time lag and one operation on the second machine in order to minimize the makespan. In the metaheuristic we used mutation and crossover operators used in the genetic algorithms to diversify the population and introduced local search method to improve its performance. Also, we realized several numerical tests and compared the results with those obtained by other heuristics already used to solve the same problem. Our perspective is to use another metaheuristic

and Branch and Bound method to resolve the problem and compare their results with those found by the PSO and heuristics.

## References


1. Ahr, D., Bksi, J., Galambos, G., Oswald, M., Reinelt, G.: An exact algorithm for scheduling identical coupled tasks. Math. Met. Oper. Res. 59, 193-203 (2004)
2. Blazewicz, J., Ecker, K., Kis, T., Potts, C.N., Tanas, M., Whitehead, J.: Scheduling of coupled tasks with unit processing times. J. Sched. 13, 453-461 (2010)
3. Brauner, N., Finke, G., Lehoux-Lebacque, V., Potts, C., Whitehead, J.: Scheduling of coupled tasks and one-machine no-wait robotic cells. Comp. Oper. Res. 36(2), 301-307 (2009)
4. Johnson, S.M.: Optimal two and three stage production schedules with setup time included. Nav. Res. Logi. Quar. 1, 61-67 (1954)
5. Kennedy J., Eberhart R.C.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp. 1942-1948, Piscataway (1995)
6. Kennedy J., Eberhart R.C.: A discrete binary version of the particle swarm algorithm. In: Proceedings of the world multiconference on systemics, cybernetics and informatics, pp. 4104-4109, Piscatawary (1997)
7. Meziani N., Oulamara A., Boudhar M.: Minimizing the makespan on two-machines fowshop scheduling problem with coupled-tasks. Neuvime Colloque sur l'optimisation et les systmes d'Information (COSI'2012),Tlemcen (Algrie), 192-203 (2012)
8. Orman, A.J., Potts, C.N.: On the complexity of coupled-Task Scheduling. Disc. App. Math. 72, 141-154 (1997)
9. Quan-Ke P., Tasgetiren M.F., Liang Y.C.: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Comp. and Oper. Res. 35, 2807–2839 (2008)
10. Shapiro, R.D.: Scheduling Coupled Tasks. Nav. Res. Logi. quar. 20, 489-498 (1980)
11. Simonin, G., Giroudeau, R., Knig, J.C.: Polynomial-time algorithms for scheduling problem for coupled-tasks in presence of treatment tasks. Elect. Not. Disc. Math. 36, 647-654 (2010)
12. Tasgetiren M.F., Sevkli M., Liang Y.C., Gencyilmaz G.: Particle swarm optimization algorithm for makespan and maximum lateness minimization in permutation flowshop sequencing problem. In: 4th international symposium on intelligent manufacturing systems, pp. 431-441, Turkey (2004)
13. Xianpeng W., Lixin T.: A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking,App. Soft Comp. 12, 652–662 (2012)
14. Zhigang L., Xingsheng G., Bin J.: A novel particle swarm optimization algorithm for permutation flow shop scheduling to minimize makespan. Chaos, Solitons and Fractals, 35, 851-861, 2008.
15. Yu, W., Hoogeveen, H., Lenstra, J.K.: Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard. J. Sched. 7, 333-348 (2004)
16. Zhigang L., Xingsheng G., Bin J.: A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. Appl. Mathe. and Comput. 175, 773–785 (2006)

# Résolution d'un problème de contrôle optimal avec une contrainte sur l'état final et sur l'état par la méthode de relaxation

Titouche Saliha [1], Spiteri Pierre [2], Messine Frédéric [2], Aidene Mohamed[1]

[1] L2CSP Laboratoire de Conception et de conduite de Systèmes de Production, Tizi-Ouzou. Algérie.
[2] ENSEEIHT-IRIT, Université de Toulouse, France.
titouchesaliha@yahoo.fr, Pierre.Spiteri@enseeiht.fr,
frederic.messine@enseeiht.fr, aidene@mail.ummto.dz

**Résumé** Dans ce travail, nous mettons en œuvre une méthode numérique pour déterminer la solution d'un problème de contrôle optimal quadratique avec contraintes sur l'état et l'état final. La convergence de la méthode itérative étudiée est analysée. Nous comparons ensuite, sur un exemple, la solution analytique à la solution numérique calculée en utilisant la méthode de relaxation couplée à la méthode de tir.

**Mots clés:** méthode de relaxation, méthode de tir, contrôle optimal, sous différentiel.

## 1 Introduction

Dans cette étude, nous présentons une méthode numérique pour résoudre un problème de contrôle optimal avec un temps terminal fixé et une contrainte sur l'état ainsi que sur l'état final. Nous considérons, sous le même formalisme, deux cas distincts de problèmes de contrôle optimal: le cas sans et avec contrainte sur l'état. Dans les deux cas, en vue d'une résolution numérique et en utilisant la notion de sous différentiel [1] et [3] pour prendre en compte si nécessaire, la projection sur le convexe des contraintes, nous reformulerons les équations d'optimalité issues du principe de minimum de Pontryagin ; ces dernières forment un système algébro-différentiel où l'équation d'état est munie d'une condition initiale et d'une condition finale. Par contre, l'équation d'état adjoint n'est munie d'aucune condition initiale ou terminale utilisable de manière algorithmique. Pour déterminer la condition initiale sur l'état adjoint, nous utiliserons dans cette étude, la méthode de tir [7], couplée à la méthode de relaxation (voir [2],[4] et [5]). Sous des hypothèses convenables, nous analysons la convergence de la méthode itérative considérée et nous terminons en exposant des résultats d'expérimentations numériques.

## 2 Position du problème

### 2.1 Cas sans contrainte sur l'etat

Soit le système dynamique suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ x(0) = x_0,\ x(T) = x_f,\ t \in [0, T], \\ x \in X_{ad}, u \in U, \end{cases} \tag{1}$$

où $x(t)$ est un n-vecteur représentant l'état du système à l'instant t, $X_{ad}$ est l'ensemble des trajectoires admissibles, $X_{ad}$ étant un ensemble convexe fermé prenant en compte les contraintes sur l'état, $x(0) = x_0$ est la condition initiale, $x(T) = x_f$ est l'état final. $u(t)$ est un r-vecteur représentant la commande agissant sur le système à l'instant $t \in [0, T]$; $U$ est l'ensemble des commandes admissibles qui ici est un ensemble ouvert. $A,\ B$ sont des $n \times n$ et $n \times r$ matrices données.
On cherche une commande admissible $\hat{u}$ qui transfère le système d'un état initial $x_0$ vers un état final $x_f$ fixé et minimisant la fonction coût $J$ définie par :

$$J(u) = \tfrac{1}{2} \int_0^T [(x - x_d)^t Q(x - x_d) + u^t N u] dt,$$

où $x_d$ représente un état désiré et les matrices $Q$ et $N$ sont symétriques, $Q$ est définie non-négative et $N$ est définie positive. L'Hamiltonien du système est donné par :

$$H(x, p, u, t) = \frac{1}{2}[(x - x_d)^t Q(x - x_d) + u^t N u] + p^t[Ax + Bu],$$

où $p$ est le vecteur d'état adjoint. Cherchons la commande minimisant l'Hamiltonien, cela revient à chercher $\hat{u}$, tel que :

$$H(\hat{x}, \hat{p}, \hat{u}) \leq H(x, p, u);\ \forall u \in U,\ \forall t \in [0, T].$$

Les équation d'optimalité s'écrivent donc :

$$\begin{cases} \frac{dx}{dt} = \frac{\partial H}{\partial p} = Ax + Bu;\ x(0) = x_0,\ x(T) = x_f,\ \forall t \in [0, T], \\ -\frac{dp}{dt} = \frac{\partial H}{\partial x} = A^t p + Q(x - x_d),\ p(0) \text{ à déterminer}, \\ \frac{\partial H}{\partial u} = 0 = Nu + B^t p. \end{cases} \tag{2}$$

Ces équations sont connues sous le nom d'équations d'Hamilton-Pontryagin.

### 2.2 Condition de transversalité sur p

De manière générale, lorsque l'on prend en compte un coût terminal, le critère à minimiser s'écrit :

$$J = g(T, x(T)) + \int_0^T f^0(x(t), u(t), t) dt,$$

où $g$ est le coût terminal, l'état final étant fixé. Conformément à [7], soient $M_0$ et $M_1$ deux sous ensembles de $\mathbb{R}^n$; on cherche à déterminer une trajectoire reliant

$M_0$ à $M_1$ tout en minimisant le coût. Si de plus $M_0$ et $M_1$ sont des variétés de $\mathbb{R}^n$ ayant des espaces tangents $T_{x(0)}M_0$ et $T_{x(T)}M_1$ respectivement en $x(0) \in M_0$ et en $x(T) \in M_1$, alors le vecteur $p(t)$ peut être construit de manière à vérifier les conditions de transversalité :

$$p(0) \perp T_{x(0)}M_0, \tag{3}$$

$$p(T) - p^0 \nabla_x g(T, x(T)) \perp T_{x(T)}M_1, \tag{4}$$

où $p^0$ est un réel tel que $p^0 < 0$ conduit au principe du maximum de Pontryagin et $p^0 > 0$ conduit au principe de minimum de Pontryagin [7]. Si $M_0 = \{x_0\}$, la condition (3) devient vide et si la variété $M_1$ s'écrit sous la forme :

$$M_1 = \{x \in \mathbb{R}^n / F_1(x) = ... = F_q(x) = 0\},$$

où les $F_i$ sont des fonctions de classe $C^1$ sur $\mathbb{R}^n$, alors l'espace tangent à $M_1$ en un point $x \in M_1$ est :

$$T_x M_1 = \{k \in \mathbb{R}^n / \nabla F_i(x)k = 0,\ i = 1, ..., q\},$$

et la condition (4) s'écrit :

$$\exists k_1, ..., k_q \in \mathbb{R} / p(T) = \sum_{i=1}^{q} k_i \nabla_x F_i(x(T)) + p^0 \nabla_x g(T, x(T)),$$

où $k_i$ sont les multiplicateurs de *Lagrange*. Dans notre problème, $g(T, x(T)) = 0$; donc la condition de transversalité sur le vecteur adjoint s'écrit :

$$p(T) = \sum_{i=1}^{q} k_i \nabla_x F_i(x(T)), k_i \in \mathbb{R}.$$

**Remarque 1** *On aboutit à la résolution d'un système algébro-différentiel; l'équation d'état décrivant le système physique est munie d'une condition initiale $x(0) = x_0$ et d'une condition finale $x(T) = x_f$. Par contre, la seconde équation correspondant à l'équation d'état adjoint, n'est munie d'aucune condition initiale ni d'aucune condition terminale utilisable pratiquement. On va donc utiliser la méthode de tir pour calculer la valeur de $p(0)$.*

### 2.3 La méthode de tir simple

La méthode de tir permet d'obtenir la valeur de $p(0)$ nécessaire à la résolution du problème à résoudre qui est caractérisé par l'application du principe du maximum ou minimum de Pontryagin. Si on est capable, à partir de la condition de minimisation de l'Hamiltonien d'exprimer le contrôle extrémal en fonction de $(x(t), p(t))$ alors le système extrémal est un système différentiel de la forme $\dot{z}(t) = F(t, z(t))$, où $z(t) = (x(t), p(t))$. Avec un intégrateur numérique à partir de $z_0$ on obtient : $\tilde{z}_i^{z_0} \sim z(t_i)$, où les $t_i$ sont les temps discrétisés par l'intégrateur. Or dans $z_0 = (x_0, p_0)$ les $x_0$ sont donnés (condition initiales du problème). Donc en faisant varier $p_0$ on obtiendra des $\tilde{z}_i^{z_0}$ différents. Ce qui nous intéresse sont les $\tilde{z}_N^{z_0} \sim z(T)$ (au temps final) or $\tilde{z}_N^{z_0} = (\tilde{x}_N^{z_0}, \tilde{p}_N^{z_0})$ et seuls les $\tilde{x}_N^{z_0}$ sont importants. Comme ils dépendent que de $p_0$ on les notera $\tilde{x}_N^{p_0}$. Définissons $G$ la fonction implicite qui en donnant $p_0$ par calcul numérique via un intégrateur retourne

$\tilde{x}_N^{p_0} - x_f$ :

$$G : \mathbb{R}^n \to \mathbb{R}^n \text{ et } G(p_0) = \tilde{x}_N^{p_0} - x_f.$$

Avec $G$, on définit un système non linéaire implicite de $n$ équations à $n$ inconnues :

$$G(p_0) = 0.$$

Pour le résoudre, on utilisera la méthode de Newton. Le principe de la méthode est le suivant : à une étape k donné, soit $p_0^k$ une approximation d'un zéro $p_0$ de $G$; donc $p_0$ s'écrit $p_0 = p_0^k + \Delta p_0^k$, et on a alors :

$$0 = G(p_0) = G(p_0^k + \Delta p_0^k) = G(p_0^k) + \frac{\partial G}{\partial p_0}(p_0^k).(p_0 - p_0^k) + o(p_0 - p_0^k),$$

ce qui entraîne la résolution de

$$\frac{\partial G}{\partial p_0}(p_0^k).(p_0 - p_0^k) = -G(p_0^k),$$

où $\frac{\partial G}{\partial p_0}(p_0^k)$ est la matrice Jacobienne de l'application $p_0 \to G(p_0)$ calculée quand $p_0 = p_0^k$; or on ne connait la fonction $p_0 \to G(p_0)$ que numériquement. On va donc utiliser un procédé de dérivation numérique basé sur la méthode des différences finies. Pour éviter le calcul de $\frac{\partial G}{\partial p_0}(p_0^k)$, il suffit de trouver une approximation de $\frac{\partial G}{\partial p_0}(p_0^k)$; conformément à [**?**], on utilise deux approximations par différences finies.

$$\frac{\partial G_i}{\partial p_{0_j}}(p_0^k) \approx \frac{1}{h_{ij}}[G_i(p_0 + \sum_{k=1}^{j} h_{ik}e^k) - G_i(p_0 + \sum_{k=1}^{j-1} h_{ik}e^k)],$$

ou bien

$$\frac{\partial G_i}{\partial p_{0_j}}(p_o^k) \approx \frac{1}{h_{ij}}[G_i(p_0 + h_{ij}e^j) - G_i(p_0)],$$

où les $h_{ij}$ sont des paramètres de discrétisation correspondant à la $i^{eme}$ équation et à la $j^{eme}$ variable, et $e^k$ est le $k^{eme}$ vecteur de la base canonique; notons que, classiquement, on peut toujours choisir les valeurs de $h_{ij}$ égales entre elles à une valeur $h$. Soit $\Delta_{ij}(p_0, h)$ une approximation par différences finies consistante; alors, on a :

$$\lim_{h \to 0} \Delta_{ij}(p_0, h) = \frac{\partial G_i}{\partial p_{0j}}(p_0), i, j = 1, ..., n.$$

On pose,

$$J(p_0, h) = (\Delta_{ij}(p_0, h)),$$

qui est une approximation de la matrice Jacobienne. De manière générale, on a à considerer à chaque itération :

$$p_0^{k+1} = p_0^k - J(p_0^k, h^k)^{-1}.G(p_0^k).$$

Le problème de la convergence de ce processus itératif est résolu grace à un résultat du livre d'Ortega et Rheinboldt [6]; en effet, si les pas de discrétisation $h_{ij}$ sont petits et tendent vers zéro, on est assuré de la convergence de ce processus.

### 2.4 Cas avec contraintes sur l'état

Dans ce cas, l'ensemble des trajectoires admissibles $X_{ad}$ est un ensemble convexe. On va reformuler les conditions nécessaires d'optimalité. Pour cela on

va donc utiliser la notion de sous-différentiabilité pour obtenir des conditions d'optimalité. Auparavant, on va faire quelques rappels mathématiques.

**Définition 1** *Soit $\chi$ une fonction convexe dans $E$ et $\mu$ un point de $E$, on note par $\partial\chi(\mu)$ l'ensemble des $\mu' \in E'$ tel que*

$$\chi(\upsilon) \geq \chi(\mu) + < \upsilon - \mu, \mu' >, \text{ pour tout } \upsilon \in E, \tag{5}$$

*où $<,>$ est le produit de dualité de $E$ dans $E'$ et $E'$ est l'espace topologique dual de $E$; un tel élément $\mu'$ est appelé sous-gradiant de $\chi$ en $\mu$, et $\partial\chi(\mu)$ est appelé le sous différenciel de $\chi$ en $\mu$.*

**Remarque 2** *Le produit de dualité de $E$ et $E'$ est une application bilinéaire de $E \times E'$ dans $\mathbb{R}$ ou $\mathbb{C}$. Si $E$ est un espace de Hilbert, alors $<,>$ est le produit scalaire de $E$.*

**Remarque 3** *Soit $\chi$ une application convexe différentiable (Fréchêt différentiable) en $\mu$; alors $\partial\chi(\mu)$ est un opérateur univoque qui coïncide avec la différentielle (au sens de Fréchet) de $\chi$ en $\mu$. On montre que $\partial\chi(\mu)$ est un ensemble convexe fermé (éventuellement vide voir* [1]*).*

**Lemme 1 ([?])** *$\mu \in E$ est tel que $\chi(\mu) = \min_{\upsilon \in E}(\chi(\upsilon))$ si et seulement si $0 \in \partial\chi(\mu)$. De plus le sous-différentiel $\partial\chi(\mu)$ est un opérateur monotone (en général multivoque) de $E$ dans $E'$.*

**Définition 2** *Soit $K$ un sous ensemble convexe fermé de $E$ modelisant les contraintes. On appelle fonction indicatrice de $K$, la fonctionnelle $\psi_K$ définie par :*

$$\Psi_K(\mu) = \begin{cases} 0, & si\ \mu \in K, \\ +\infty, & sinon. \end{cases}$$

*On montre que $\Psi_K(\mu)$ est convexe [?].*

**Consequence 1** *Il résulte de Lemme 1 que chercher le minimum de $\chi$ sur $K \subset E$ revient à résoudre une équation multivoque $0 \in A(\upsilon)$, où $A = \partial(\chi + \Psi_K)$, $\Psi_K$ fonction indicatrice du convexe $K$. En utilisant la définition du sous différentiel, on a (voir [1]),*

$$\partial\Psi_K(\upsilon) = \{\upsilon' \in E' / < \upsilon - w, \upsilon' > \geq 0, \text{ pour tout } w \in K\}.$$

*Ce qui montre que $D(\partial\Psi_K) = D(\Psi_K) = K$ et $\partial\Psi_K(\upsilon) = \{0\}$ pour tout $\upsilon \in int(K)$. Par ailleurs, si $\upsilon$ se trouve sur la frontière de $K$, alors $\partial\Psi_K(\upsilon)$ est confondu avec le cône normal à $K$ au point $\upsilon$.*

### 2.5 Application à notre problème

Considérons le domaine fermé $X_{ad} = \{x \in \mathbb{R} / x_{min} \leq x \leq x_{max}\}$ et notons $\Psi_{X_{ad}}$ la fonction indicatrice de $X_{ad}$ qui vérifie :

$$\Psi_{X_{ad}} = \begin{cases} 0, & si\ x \in X_{ad}, \\ +\infty, & sinon. \end{cases}$$

Le sous-différentiel est alors donné par :

$$\Psi_{X_{ad}} = \begin{cases} ]-\infty, 0], & si\ x = x_{min}, \\ 0, & si\ x_{min} < x < x_{max}, \\ [0, +\infty[, & si\ x = x_{max}, \\ \emptyset, & sinon, \end{cases}$$

et admet le graphe représenté par la Figure 1. On remarque que le sous différentiel

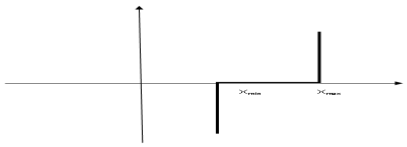


**Fig. 1.** Sous-différentiel de la fonction $\Psi_{X_{ad}}$

$\Psi_{X_{ad}}$ est bien monotone.
Appliquons le Lemme 1; on cherche $\hat{u}$ qui minimise l'Hamiltonien $H$; ceci peut s'écrire sous la forme :

$$0 \in \partial H(\hat{u}).$$

Comme H est un opérateur continu ([3]), on a la nouvelle formulation des conditions nécessaires d'optimalité :

$$\begin{cases} 0 \in \frac{dx}{dt} + \partial\Psi_{X_{ad}} - Ax - B\hat{u};\ x(0) = x_0,\ x(T) = x_f,\ \forall t \in [0, T], \\ -\frac{dp}{dt} = A^t p + Q(x - x_d),\ p(0) \text{ à déterminer}, \\ N\hat{u} + B^t p = 0. \end{cases}$$

## 3 Méthode de résolution numérique

Pour la résolution du problème, avec et sans contraintes, nous effectuons le couplage de la méthode de relaxation (voir [2],[4] et [5]) avec la méthode de tir [7]. Cette dernière étant destinée à calculer $p(0)$ nécessaire à la résolution du système algébro-différentiel obtenu par application du principe de minimum de Pontryagin.

### 3.1 Cas avec contrainte

Les étapes de la méthode de résolution numérique sont résumées ci-dessous :

1. Approximation de la commande initiale $u^0$, et de l'état adjoint initial $p^0(0)$, pour $t \in [0, T]$,
2. $r \leftarrow 0$,
3. **Tant que** $\mid u^{(r+1)} - u^{(r)} \mid > \epsilon$ (où $\epsilon$ définit le seuil de convergence) **faire** :

- Détermination de l'état $x^{(r)}$ et de l'état adjoint $p^{(r)}$ pour $t \in [0,T]$ composante par composante par intégration numérique des équations d'état avec projection sur le convexe $X_{ad}$ :

$$\begin{cases} \frac{d\bar{x}}{dt} = A\bar{x} + Bu^{(r)}, 0 < t \leq T, \\ \bar{x}(0) = x_0, \end{cases} \quad et \quad x^{(r)} = Proj(\bar{x}), \tag{6}$$

où $Proj(.)$ est l'opérateur de projection sur le convexe fermé $X_{ad}$.
Puis on calcule le vecteur d'état adjoint :

$$\begin{cases} -\frac{dp^{(r)}}{dt} = A^t p^{(r)} + Q(x^{(r)} - x_d), \\ p^{(r)}(0), \end{cases} \tag{7}$$

où $p^{(r)}(0)$ est calculé par la méthode de tir,

- Détermination de la commande $u^{(r+1)}$ :

$$u^{(r+1)} \leftarrow -N^{-1}B^t p^{(r)}, \tag{8}$$

- $Convergence \leftarrow \mid u^{(r+1)} - u^{(r)} \mid$,
- Détermination de la fonction de tir :

$$G(p) = x^{(r)}(T) - x_f,$$

- Solution de l'équation de tir par la méthode de Newton et détermination de la nouvelle valeur de $p(0)$ :

$$p^{(r+1)}(0) \leftarrow p^{(r)}(0) + correction,$$

- $r \leftarrow r + 1$.

**Fin de tant que**

**Remarque 4** *Les étapes* (8), (9) *et* (10) *de la boucle correspondent à la méthode de relaxation alors que les étapes suivantes correspondent à la mise en œuvre de la méthode de tir.*

### 3.2 Cas sans contrainte

La démarche est analogue sauf que l'étape (8) est remplacée par

$$\begin{cases} \frac{dx^{(r)}}{dt} = Ax^{(r)} + Bu^{(r)}, 0 < t \leq T, \\ x(0) = x_0. \end{cases}$$

## 4 Convergence de la méthode

Ecrivons les équation d'optimalité sous forme matricielle :

$$\begin{pmatrix} \frac{dx}{dt} + \partial\Psi_{X_{ad}} \\ -\frac{dp}{dt} \\ 0 \end{pmatrix} + \begin{pmatrix} \bar{A} & 0 & -B \\ -Q & \bar{A}^t & 0 \\ 0 & B^t & N \end{pmatrix} \begin{pmatrix} x \\ p \\ u \end{pmatrix} \ni \begin{pmatrix} 0 \\ -Qx_d \\ 0 \end{pmatrix},$$

à laquelle il faut adjoindre des conditions initiales pour $x$ et $p$ et où $\bar{A} = -A,\ N = kI,\ I$ étant la matrice identité.

**Proposition 1** *Si les conditions suivantes sont vérifiées :*
- $\bar{A}$ *est une M-matrice* ($\bar{a}_{ij} \leq 0$ *si* $i \neq j$ *et* $\bar{A}^{-1} \geq 0$)
- $k \geq k_0 > 0$
- $p^2(0) - p^2(T) > 0$,

*alors l'algorithme permettant de calculer numériquement la loi de commande optimale, par la méthode itérative considérée, converge quelque soit la donnée initiale* $u^0$.

**Preuve 1** *Le résultat découle de la propriété des M-matrices et de la monotonie du sous gradient et des opérateurs de dérivation, compte tenu des hypothèses considérés.*

**Remarque 5** *Les M-matrices ont de nombreuses propriétés importantes; notamment le rayon spectral de la matrice de Jacobi associée à* $J = I - \bar{D}^{-1}.\bar{A}$ *est inférieur à* 1*; propriété que nous utiliserons dans la suite. Ici* $\bar{D}$ *est la diagonale de* $\bar{A}$.

**Remarque 6** *La preuve de convergence est valable dans le cas avec et sans contrainte sur l'etat. En effet, dans ce dernier cas le sous-différentiel de la fonction indicatrice est nul et le raisonement est encore valable.*

## 5 Exemple numérique

### 5.1 Cas sans contrainte

On considère le système en anneau suivant :

$$\begin{cases} \textit{Déterminer} \quad \hat{u} \in U, \quad \textit{tel que}, \\ J(\hat{u}) \leq J(u), \forall u \in U. \end{cases} \tag{9}$$

où

$$J(u) = \tfrac{1}{2} \int_0^T [\| x - x_d \|^2 + k \| u \|^2] dt,$$

sous les contraintes suivantes :

$$\begin{cases} \dot{x}_i = -wx_i + ax_{i+1} + bu_i, \ \ i \in \{1, 2, ..., n-1\} \textit{ et } n \geq 2 \\ \dot{x}_n = ax_1 - wx_n + bu_n, \\ x(0) = x_0 \ \ x(T) = x_1, \end{cases} \tag{10}$$

où $a, b$ et $w$ sont des constantes réelles positives. L'Hamiltonien relatif à ce problème est donné par:

$$H(x,p,u,t) = \frac{1}{2}(\| x - x_d \|^2 + k \| u \|^2) + \sum_{i=1}^{n-1} p_i(-wx_i + ax_{i+1} + bu_i) + p_n(ax_1 - wx_n + bu_n).$$

Les équation d'optimalité s'écrivent :

$$\begin{cases} \dot{x}_i = -wx_i + ax_{i+1} + bu_i, \ i \in \{1,2,...,n-1\} \\ \dot{x}_n = ax_1 - wx_n + bu_n, \\ \dot{p}_1 = -x_1 + wp_1 - ap_n + x_{1d}, \\ \dot{p}_i = -x_i - ap_{i-1} + wp_i + x_{id}, \ i \in \{2,...,n\}, \\ ku_i + bp_i = 0, \ i \in \{1,...,n\}. \end{cases}$$

**5.1.1. Solution numérique**

La solution numérique est calculée pour $n = 5$. En posant $z(t) = (x(t), p(t))$ notre système devient :

$$\begin{cases} \dot{z}_i = -wz_i + az_{i+1} + bu_i, \ i \in \{1,...,4\}, \\ \dot{z}_5 = az_1 - wz_5 + bu_5, \\ \dot{z}_6 = -z_1 + wz_6 - az_{10} + x_{1d}, \\ \dot{z}_i = -z_{(i-5)} - az_{(i-1)} + wz_i + x_{(i-5)\,d}, \ i \in \{7,...,10\} \\ z_i(0) = 0.5, \ i \in \{1,...,5\}, \\ z_i(0) \in \mathbb{R}, \ \ i \in \{6,...,10\}. \end{cases}$$

Soit $z(t)$ une solution du système au temps $t$ avec les conditions initiales $z(0) = (z_1(0), \ldots, z_i(0), \ldots, z_{10}(0))$. Pour $T = 4$, on doit avoir :

$$z_i(t = 4, z(0)) = \begin{cases} 0.5, & \text{pour } i = \overline{1,5}; \\ z_i(0), & \text{pour } i = \overline{6,10}. \end{cases}$$

où $z_i(0)$ pour $i = \overline{6,10}$ sont à déterminer. On construit une fonction de tir qui est une équation algébrique non linéaire de la variable $p$ à l'instant $t = 4$; cette fonction de tir est calculée par une procédure d'intégration numérique d'équations différentielles ordinnaires (Euler, Runge-Kutta,etc); la fonction de tir s'écrit :

$$G(z) = \bar{z} - I \times 0.5, \ \text{ où } \ \bar{z} = (\bar{z}_i = z_i \ \text{ pour } \ 1 \leq i \leq 5).$$

**Le problème à résoudre s'écrit alors :** Déterminer $p(0)$ tel que $G(z(0))$ donne le $x(T)$ désiré. L'algorithme de résolution numérique de ce problème sera alors complètement défini, si l'on se donne :

1. l'algorithme d'intégration d'un système différentiel à valeur initiale (par exemple une procédure d'Euler ou de Runge-Kutta), pour calculer la fonction de tir $G$ (ici 'ode45' de Matlab qui est une méthode de Runge-Kutta 4/5 à pas variable).
2. l'algorithme de résolution de $G(z) = 0$ qui dans notre cas utilise la méthode de quasi-Newton ('fsolve' de Matlab).

**5.1.2. Solution exacte**

Dans le cas où $n = 2$, pour calculer de manière analytique la commande optimale $u(t)$, et sa trajectoire correspondante $x(t)$ du problème $(13) - (14)$, nous avons utilisé les équations d'optimalité ainsi que la condition de transversalité sur $p(t)$; puisque il n'y a pas de coût terminal, la condition de transversalité sur $p(t)$ s'écrit :

$$\exists k_1, k_2 \in \mathbb{R}/p(T) = \sum_{i=1}^{2} k_i \nabla F_i(x(t)).$$

On pose $F_1(x) = x_1(T) - 0.5,\ F_2(x) = x_2(T) - 0.5,\ p(T) = (k_1, k_2)$ où $k_1, k_2$ sont les multiplicateurs de Lagrange. La solution des équations d'état $x_1(t),\ x_2(t)$ est donnée par :

$$\begin{aligned} x_1(t) = {} & \lambda\left(\frac{w^2-a^2+c_1^2-\frac{b^2}{k}}{2w} - c_1\right)e^{c_1 t} + \beta\left(\frac{w^2-a^2+c_1^2-\frac{b^2}{k}}{2w} + c_1\right)e^{-c_1 t} \\ & + \gamma\left(\frac{w^2-a^2+c_2^2-\frac{b^2}{k}}{2w} - c_2\right)e^{c_2 t} + \alpha\left(\frac{w^2-a^2+c_2^2-\frac{b^2}{k}}{2w} + c_2\right)e^{-c_2 t} + \nu', \end{aligned}$$

et

$$\begin{aligned} x_2(t) = {} & \lambda\left(\frac{w^2-a^2-c_1^2+\frac{b^2}{k}}{2w} - c_1\frac{w^2+a^2-c_1^2+\frac{b^2}{k}}{2aw}\right)e^{c_1 t} + \beta\left(\frac{w^2-a^2-c_1^2+\frac{b^2}{k}}{2w}\right. \\ & \left. + c_1\frac{w^2+a^2-c_1^2+\frac{b^2}{k}}{2aw}\right)e^{-c_1 t} + \gamma\left(\frac{w^2-a^2-c_2^2+\frac{b^2}{k}}{2w} - c_2\frac{w^2+a^2-c_2^2+\frac{b^2}{k}}{2aw}\right)e^{c_2 t} \\ & + \alpha\left(\frac{w^2-a^2-c_2^2+\frac{b^2}{k}}{2w} + c_2\frac{w^2+a^2-c_2^2+\frac{b^2}{k}}{2aw}\right)e^{-c_2 t} + \nu''. \end{aligned}$$

L'expression de $u_1(t)$ *et* $u_2(t)$ sont données par:

$$u_1(t) = -\frac{b^2}{k}[\lambda e^{c_1 t} + \beta e^{-c_1 t} + \gamma e^{c_2 t} + \alpha e^{-c_2 t} + \nu].$$

$$\begin{aligned} u_2(t) = {} & -\frac{b^2}{k}\left[\left(\frac{w^2+a^2-c_1^2+\frac{b^2}{k}}{2aw}\right)e^{c_1 t}\lambda + \left(\frac{w^2+a^2-c_1^2+\frac{b^2}{k}}{2aw}\right)e^{-c_1 t}\beta\right. \\ & \left. + \left(\frac{w^2+a^2-c_2^2+\frac{b^2}{k}}{2aw}\right)e^{c_2 t}\gamma + \left(\frac{w^2+a^2-c_2^2+\frac{b^2}{k}}{2aw}\right)e^{-c_2 t}\alpha + \nu'''\right]. \end{aligned}$$

Où

$$\begin{aligned} & c_1 = \pm\sqrt{(a-w)^2+\tfrac{b^2}{k}}, && c_2 = \pm\sqrt{(a+w)^2+\tfrac{b^2}{k}} \\ & \nu = \frac{w(a^2-w^2-\frac{b^2}{k})x_{1d} + a(a^2-w^2+\frac{b^2}{k})x_{2d}}{(a^2+w^2+\frac{b^2}{k})^2 - 4a^2w^2}, && \nu' = \left(\frac{w^2-a^2-\frac{b^2}{k}}{2w}\right)\nu + \frac{1}{2}x_{1d} + \frac{a}{2w}x_{2d} \\ & \nu'' = \left(\frac{w^2-a^2+\frac{b^2}{k}}{2a}\right)\nu + \frac{1}{2}x_{2d} + \frac{w}{2a}x_{1d}, && \nu''' = \left(\frac{w^2+a^2+\frac{b^2}{k}}{2aw}\right)\nu + \frac{1}{2a}x_{1d} - \frac{1}{2w}x_{2d}. \end{aligned}$$

Les constantes étant déterminées par les conditions aux limites suivantes :
$x_1(0) = 0.5, x_2(0) = 0.5, x_1(T) = 0.5, x_2(T) = 0.5, p_1(T) = k_1, p_2(T) = k_2$.
La solution exacte est représentée sur la Figure 2.

**5.1.3 Comparaison des deux approches.**

Dans le cas $n = 2$, les expérience numériques ont été réalisées pour $w = 2; a = 0.5; b = 1; T = 2; x_{1d} = x_{2d} = 0.2$. On déduit que la solution exacte et la solution numérique sont concordantes (voir Figure 2 et Figure 3). Les performances de la

procédure numérique sont résumées dans le tableau 1 ci dessous, pour différentes valeurs de $k$. Notons que la convergence est rapide et que de plus, le temps de calcul est très faible.

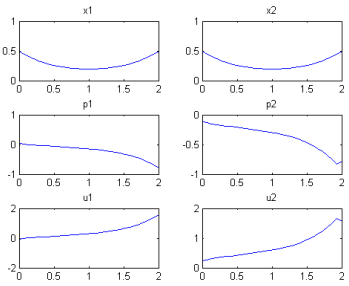



**Fig. 2.** Solution exacte sans contraintes

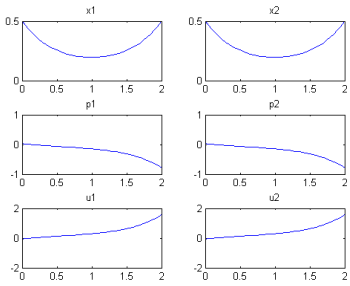



**Fig. 3.** Solution numerique sans contraintes

### 5.2 Cas avec contrainte

En présence de contrainte sur l'état pour $n > 2$, le calcul de la solution exacte par une méthode analytique est difficile à effectuer. Nous nous limiterons ici à la recherche d'une solution numérique. Les données sont les mêmes que celles considérées dans le cas sans contrainte sauf que ici $n = 5$ et $T = 4$. L'algorithme est analogue à celui proposé précédement en rajoutant les contraintes sur l'état $x(t)$ définies par:

$$si\ x > x_{max},\ alors\ x = x_{max}\ sinon\ si\ x < x_{min}\ alors\ x = x_{min}.$$

Pour $x_{min_i} = 0.35$ *et* $x_{max_i} = 0.5$ pour $i \in \{1, ..., 5\}$, les résultats numériques sont présentés dans le tableau 2 et les solutions sont dessinées sur les Figures 4 et 5. Comme précédement la convergence, exprimée en nombre d'itérations est rapide et les temps de calculs très faibles. Notons que dans les résultats obtenus, les contraintes sont saturées, compte-tenu des paramètres utilisées.

| k | temps | nombre d'itération |
|---|---|---|
| 0.5 | 0.1248 | 2 |
| 1 | 0.1716 | 2 |
| 1.5 | 0.0468 | 1 |
| 2 | 0.1092 | 1 |
| 2.5 | 0.1092 | 1 |

Tableau 1

| k | temps | nombre d'itération |
|---|---|---|
| 0.5 | 0.1872 | 2 |
| 1 | 0.1716 | 2 |
| 1.5 | 0.1872 | 2 |
| 2 | 0.1716 | 2 |
| 2.5 | 0.2340 | 2 |

Tableau 2

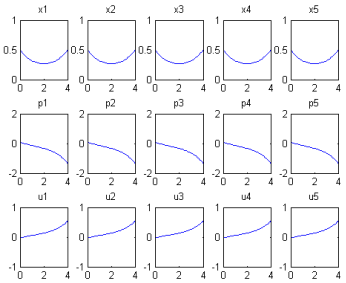



**Fig. 4.** Solution numérique sans contrainte

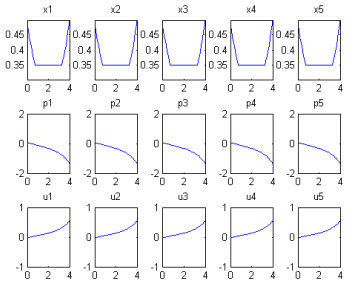



**Fig. 5.** Solution numérique avec contrainte

## 6 Conclusion

Dans ce travail, nous avons proposé, sous certaines hypothèses, l'utilisation d'un algorithme de relaxation pour la résolution d'un problème de contrôle optimal non linéaire dans le cas où il y a une contrainte sur l'état et l'état final. Nous avons testé cette méthode sur un problème modè et il s'avère que la convergence est rapide et que les temps de calculs sont petits.

## References

1. V. Barbu, Nonlinear semigroups and differential equations in Banach spaces, Noordhoff International Publishing,1976.
2. D.Gien, B.Lang, J.C.Miellou, L.Raffort, P.Spiteri, Commande optimale de systèmes complexes, RAIRO Automatique, Systems Analysis and Control vol. 18, 1984, pp.209-224.
3. P.J. Laurent, Approximation et optimisation, Collection Enseignement des sciences, 1972.
4. B.Lang, J.C.Miellou, P.Spiteri, Asynchronous relaxation algorithms for optimal control problems. Mathematical and Computers in Simulations 28(1986) 227-242.
5. J.C.Miellou, P.Spiteri, A parallel asynchronous relaxation algorithm for optimal control problem. Proceeding of the International Conference on Mathematical Analysis and its Appllications, Kowait 1985.
6. J.M.Ortega and W.C.Rheinboldt, Iterative solution of nonlinear equations in several variables. Academic Press, New York, 1970.
7. E.Trelat, Contrôle optimal: théorie et applications, Vuibert, collection Mathématiques Concrètes, 2005.

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