الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

Université Abou Bekr Belkaid
Tlemcen Algérie

جامعة أبي بكر بلقايد
تلمسان الجزائر

Neuvième Colloque sur l'Optimisation et les Systèmes d'Information

COSI'2012

12 – 15 mai 2012
Tlemcen, Algérie

Organisé par les Départements Informatique et Mathématiques
Université Abou Bekr Belkaid de Tlemcen

Actes du Neuvième Colloque sur l'Optimisation et les Systèmes d'Information- COSI'2012

12-15 Mai 2012, Tlemcen, Algérie

Université Abou Bekr Belkaïd-Tlemcen

Faculté des Sciences

Départements de Mathématiques et d'Informatique

Contents

Préface

Ces actes regroupent les articles présentés lors de la 9ème édition du Colloque sur l'Optimisation et les Systèmes d'Information (COSI 2012) qui s'est déroulé à Tlemcen, Algérie, du 12 au 15 mai 2012.

COSI est une manifestation scientifique pluridisciplinaire qui rassemble des chercheurs qui travaillent 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 2012 ont eu lieu à : Guelma (2011), Ouargla (2010), Annaba (2009), Tizi-Ouzou (2008), Oran (2007), Alger (2006), Béjaia (2005) et Tizi-Ouzou (2004).

Le comité de programme a examiné 242 articles soumis à COSI 2012. A la fin du processus d'évaluation, 37 articles longs (soit un taux d'acceptation de 15,29%) et 22 posters étaient acceptés pour publication.

Les articles contenus dans ces actes représentent de façon tout à fait homogène l'ensemble des thématiques couvertes par COSI.

Nous remercions les auteurs pour leurs excellentes contributions, les membres du comité de programme, les relecteurs externes, les membres du comité d'organisation et les sponsors.

Nous sommes particulièrement heureux que quatre chercheurs de très haut niveau aient accepté de nous présenter une conférence invitée :

- Ne votez pas : jugez ! par ***Michel Balinski*** (Ecole Polytechnique et CNRS, France).
- Fouille de données et calcul scientifique dans le cloud : vers un environnement virtuel collaboratif ubiquitaire pour la recherche par ***Karim Chine*** (Cloud Era Ltd, Cambridge, UK).
- On the power of graph searching par ***Michel Habib*** (LIAFA, Université de Paris Diderot - Paris 7, France).
- L'optimisation globale déterministe : méthodes, applications et extensions par ***Frédéric Messine*** (IRIT, ENSEEIHT, Toulouse, France).

Mohand-Saïd Hacid (Président du CP)
Méziane Aïder (Vice-Chair : Théorie des Graphes et Combinatoire)
Mourad Baiou (Vice-Chair : Recherche Opérationnelle)
Nacéra Benamrane (Vice-Chair : Traitement d'Images et Vision Artificielle)
Hélène Fargier (Vice-Chair : Intelligence Artificielle)
Hassina Seridi & Michel Schneider (Vice-Chair : Systèmes d'Information)

Organisation

Université Aboubakr Belkaïd-Tlemcen

Président d'honneur

Professeur Norreddine Ghouali
Université Aboubakr Belkaïd, Tlemcen, Algérie

Comité d'Organisation

Président

Pr. Boufeldja Tabti, Université Aboubakr Belkaïd, Tlemcen, Algérie

Vice-Président

Pr. Benmiloud Mebkhout, Université Aboubakr Belkaïd, Tlemcen, Algérie

Membres

Pr Ghalem Saïd - Vice doyen, Université Aboubakr Belkaïd, Tlemcen, Algérie
Dr Mebkhout Benmiloud -Chef de département de Mathématiques, Université Aboubakr Belkaïd, Tlemcen, Algérie
Mr Bentifour Rachid, Université Aboubakr Belkaïd, Tlemcen, Algérie
Mr Mamchaoui Mohammed, Université Aboubakr Belkaïd, Tlemcen, Algérie
Mr Menouar Mohammed El Amine, Université Aboubakr Belkaïd, Tlemcen, Algérie
Mr Benchaib Abdelatif, Université Aboubakr Belkaïd, Tlemcen, Algérie

Secrétariat

Mme Cherki Samira Ep. Bounacer
Melle Hamouni Kheira

Comité de Pilotage

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

Comité de Programme

Président

Mohand-Said Hacid, LIRIS, Université Lyon I (France)

Vice-Chairs

Méziane Aider, USTHB (Algérie)
Mourad Baiou, LIMOS (France)
Nacéra Benamrane, USTO (Algérie)
Hélène Fargier, IRIT (France)
Michel Schneider, ISIMA (France)
Hassina Seridi, Université Annaba (Algérie)

Membres

Abdelmalek Amine, Université de Saida (Algérie)
Adi Kamel, Université de Québec en Outaouais (Canada)
Ahmed-Nacer Mohaned USTHB (Algérie)
Ahmed-Ouamer Rachid Université de Tizi-Ouzou (Algérie)
Aidene Mohamed Université de Tizi-Ouzou (Algérie)
Aït Haddadene Hacène USTHB Alger (Algérie)
Aït Mohamed Otmane, université Concordia (Canada)
Alimazighi Zaia, USTHB (Algérie)
Arab Ali Cherif, Université Paris 8 (France)
Badache Nadjib CERIST (Algérie)
Barkaoui Kamel, CNAM-Paris (France)
Barki Hichem, INRIA Bordeaux (France)
Barra Vincent, UBP, Clermont-Ferrand (France)
(France)
Belbachir Hafida USTO Oran (Algérie)
Bellatreche Ladjel LISI, Université de Poitier (France)
Benamar Abdelkrim, Université Abou Bekr Belkaid Tlemcen (Algérie)
Benferhat Salem, Université d'Artois, Lens (France)
Benmammar Badr, Université Abou Bekr Belkaid Tlemcen (Algérie)
Ben Yahia Sadok Faculté des Sciences de Tunis (Tunisie)
Benatallah Boualem University of New South Wals (Australie)
Benbernou Salima Université Paris Descartes (France)
Benhamou Belaid Université d'Aix-Marseille I (France)
Berrachedi Abdelhafid USTHB Alger (Algérie)
Bessy Stéphane, Université Montpellier II (France)
Bibi Mohand Ouamer Université de Béjaia (Algérie)
Bouchekif Mohamed, Université Abou Bekr Belkaid,Tlemcen (Algérie)
Bouchemakh Isma USTHB Alger (Algérie)
Boudhar Mourad, USTHB (Algérie)
Boufaida Mahmoud Université Mentouri, Constantine (Algérie)
Boufaïda Zizette, Université Mentouri, Constantine (Algérie)
Boughanem Mohand IRIT, Toulouse (France)
Bouzeghoub Amel Telecom Sud, Paris (France)
Bouzouane Abdenour UQC (Canada)
Champion Thierry Université du Sud Toulon-Var (France)
Chaouki Babahenini Mohamed Univ. Mohamed Khider,

Biskra (Algérie)
Chellali Mustapha, Université Saad Dahlab, Blida (Algérie)
Chikh Mohammed El Amine, Univ. Abou Bekr Belkaid Tlemcen (Algérie)
Crouzeix Jean-Pierre, UBP, Clermont-Ferrand (France)
d'Orazio Laurent Université Blaise Pascal Clermont-Ferrand (France)
Didi Fedoua Université Abou Bekr Belkaid Tlemcen, (Algérie)
Djedi Nouredine Université Biskra (Algérie)
Elghazel Haytham Université Claude Bernard Lyon 1 (France)
Ellaggoune Fateh Université de Guelma (Algérie)
Engelbert Mephu, Université Blaise Pascal, Clermont-Ferrand (France)
Farah Nadir, Université Badji Mokhtar d'Annaba (Algérie)
Gourvès Laurent, Lamsade, (France)
Habib Michel, Université de Paris VII (France)
Hadjali Allel, ENSSAT, Lannion (France)
Hamadi Youssef, Microsoft Research Cambridge (Royaume Uni)
Hantouche Abderrahim CMM, Universidad de Chile (Chili)
Kanté Mamadou, Clermont-Ferrand (France)
Kazar Okba Université de Biskra (Algérie)
Kechadi Tahar UCD (Irlande)
Kedad Zoubida, UVSQ (France)
Khadir Tarek, Université Badji Mokhtar d'Annaba (Algérie)
Kheddouci Hamamache Université de Lyon 1 (France)
Lafifi Yacine Université de Guelma (Algérie)
Lacomme Philippe LIMOS, Université Blaise Pascal (France)
Laskri Mohamed Tayeb, Université Badji Mokhtar, Annaba (Algérie)
Lebbah Yahia Université D?Oran Es-Sénia (Algérie)
Leger Alain, France Télécom (France)
Lehsaini Mohamed, Université Abou Bekr Belkaid Tlemcen (Algérie)
Limouzy Vincent, Université Blaise Pascal Clermont-Ferrand (France)
Mahjoub Ridha, Université Paris Dauphine (France)
Merouani Hayett, Université Badji Mokhtar, Annaba (Algérie)
Missaoui Rokia, Universite de Quebec en Outaouais (Canada)
Nourine Rachid, Université d'Oran Es-Sénia (Algérie)
Ouanes Mohand, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Oukacha Brahim Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Ouksel Aris, UIC (USA)
Petit Jean-Marc, INSA de Lyon (France)
Rao Michael, ENS Lyon (France)
Rabhi Fethi, Université du New South Wales, Sydney (Australie)
Radjef Mohand-Said, Université Abderrahmane Mira de Bejaia (Algérie)
Raspaud André, Université Bordeaux 1 (France)
Rebaïne Djamal, Université du Québec, Chicoutimi (Canada)
Sadi Bachir, Université Mouloud Mammeri de Tizi-Ouzou (Algérie)
Sais Fatiha LRI, Université de Paris Sud (France)
Salleb-Aouissi Ansaf Columbia University (USA)
Sam Yacine, Université de Tours (Algérie)
Seridi Hamid, Université de Guelma (Algérie)
Spiteri Pierre, INP- Toulouse (France)
Taher Yehia, Tilburg University (Netherlands)
Taleb-Ahmed Abdelmalik, Université de Valenciennes (France)
Tchemisova Tatiana, University of Aveiro (Portugal)
Yebdri Mustapha, Université Abou Bekr Belkaid de Tlemcen (Algérie)
Ziou Djemel, Unversité de Sherbrooke (Canada)

Relecteurs additionnels

Mohamed Lehsaini
Katia Abbaci
Hassene Aissi
Samir Aknine
Fatiha Amirouche
Sabeur Aridhi
Mahmoud Barhamgi
Hichem Barki
Sofiane Batata
Yacine BELHOUL
Hattoibe Ben Aboubacar
Abdelkrim Benamar
Fatiha Bendali
Soumia Benkrid
Badr Benmammar
Kheir Eddine Bouazza

Omar Boucelma
Mohammed Bouchekif
Saida Boukhedouma
Abdelmadjid Boukra
Yahia Chabane
Thierry Champion
Jean-Pierre Crouzeix
Ibrahima Diarrassouba
Rahma Djiroune
Philippe Fournier-Viger
Thierry Garcia
Asma Hachemi
Assia Hachichi
Salima Hacini
Mehdi Haddad
Fayçal Hamdi
abderrahim Hantoute
Mounir Hemam
Stéphane Jean
Faiza Khan Khattak
Rania Khefifi
Selma Khouri
Ben Jabeur Lamjad
Ludovic Liétard.
Gaëlle Loosli
Philippe Marthon
Sebastien Martin
Arnaud Mary
Baraa Mohamad
Mohamed Amine Mostefai
Sarah Nait Bahloul
Cheikh Niang
Damien Nouvel
Samir Ouchani
Andre Raspaud
Lynda Said L'Hadj
Rabie Saidi
Idrissa Sarr
Vladimir Shikhman
Grégory Smits
Sahri Soror
Nouredine Tamani
Clovis Tauber
Yamina Tlili
Ronan Tournier
Tarik Zouagui

Théorie des graphes

Some properties of b-coloring vertex and edge critical graphs*

Mostafa Blidia[1], **Noureddine Ikhlef Eschouf**[2]
Frédéric Maffray[3] and **Zoham Zemir**[1]
[1]LAMDA-RO, Department of Mathematics,
University of Blida, B.P. 270, Blida, Algeria.
E-mail: zohaze@yahoo.fr, m_blidia@yahoo.fr
[2]University Yahia Farès of Medéa.
E-mail: nour_echouf@yahoo.fr
[3]C.N.R.S, Laboratoire G-SCOP, 46 Avenue
Félix Viallet, 38031. Grenoble Cedex, France

Abstract

A b-coloring is a coloring of the vertices of a graph such that each color class contains a vertex that has a neighbor in all other color classes, and the b-chromatic number $b(G)$ of a graph G is the largest integer k such that G admits a b-coloring with k colors. In this paper, we give various properties of two types of criticality with respect to b-coloring and we conclude with two open problems.

Keywords: b-coloring, vertex b^+-critical graphs, edge b^+-critical graphs

1 Introduction

A proper coloring of a simple graph G is an assignment of colors to the vertices of G such that no two adjacent vertices have the same color. The *chromatic number* of G is the minimum integer $\chi(G)$ such that G has a proper coloring with $\chi(G)$ colors.

A *b-coloring* of a graph G by k colors is a proper coloring of the vertices of G such that in each color class there exists a vertex having neighbors

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

in all the other $k-1$ colors classes. We call any such vertex a *b-vertex*. The b-chromatic number $b(G)$ of a graph G is the largest integer such that G admits a b-coloring with k colors. The concept of b-coloring has been introduced by R.W. Irving and D.F. Manlove [12, 19]. They proved that determining b(G) is NP-hard for general graph, even when it is restricted to the class of bipartite graphs [18], but it is polynomial for trees [12, 19]. The NP-completeness results have incited researchers to establish bounds on the b-chromatic number in general or to find its exact values for subclasses of graphs (see[2, 3, 5, 6, 7, 8, 10, 13, 16, 18, 17, 20, 21]).

The b-chromatic number of a graph may increase, decrease or remain unchanged when the graph G is modified by deleting a vertex or an edge. Thus, one can classify critical graphs with respect to the b-chromatic number into six classes of graphs. Let $G = (V, E)$ be a simple graph.

Class 1: vertex b^+ critical if $b(G-v) > b(G)$ for all $v \in V(G)$.

Class 2: edge b^+ critical if $b(G-e) > b(G)$ for all $e \in E(G)$.

Class 3: vertex b^- critical if $b(G-v) < b(G)$ for all $v \in V(G)$.

Class 4: edge b^- critical if $b(G-e) < b(G)$ for all $e \in E(G)$.

Class 5: vertex $b^=$ critical if $b(G-v) = b(G)$ for all $v \in V(G)$.

Class 6: edge $b^=$ critical if $b(G-e) = b(G)$ for all $e \in E(G)$.

In [4, 11], the authors characterized some graphs belonging to classes 3 and 4. In this paper we present some results concerning the first two classes.

We finish this section with some definitions and notation which are used throughout the paper. For the other necessary definitions and notation, we follow that of Berge [1]. Consider a graph $G = (V, E)$. For any $A \subset V$, let $G[A]$ denote the subgraph of G induced by A. For any vertex v of G, the *neighborhood* of v is the set $N_G(v) = \{u \in V(G) \mid (u, v) \in E\}$ (or $N(v)$ if there is no confusion), and the *closed neighborhood* of v is the set $N_G[v] = N_G(v) \cup \{v\}$. Let $\Delta(G)$ (respectively, $\delta(G)$ be the maximum (respectively, minimum) degree in G. Let $\omega(G)$ denote the size of a maximum clique of G. If G and H are two vertex-disjoint graphs, the *union* of G and H is the graph $G+H$ whose vertex-set is $V(G) \cup V(H)$ and edge-set is $E(G) \cup E(H)$. For an integer $p \geq 2$, the union of p copies of a graph G is denoted pG. The *join* of graphs G and H is the graph denoted $G \vee H$ obtained from $G+H$ by adding all edges between G and H. The *cartesian product* of two graphs G and H denoted by $G \square H$, is a simple graph with $V(G) \times V(H)$ as its vertex set and two vertices (u_1, v_1) and (u_2, v_2) are adjacent in $G \square H$ if and only if either $u_1 = u_2$ and v_1, v_2 are adjacent in H, or u_1, u_2 are adjacent in G and $v_1 = v_2$. The girth $g(G)$ of G is the length of a shortest cycle in G.

2 Known results

Let G be a graph with decreasing degree sequence $d(v_1) \geq d(v_2) \geq ... \geq d(v_n)$ and let $m(G) = max\{i : d(v_i) \geq i - 1\}$. Irving and Manlove [12, 19] proved that for any graph G, $b(G) \leq m(G)$ and they show that for a tree T the inequality $m(T) - 1 \leq b(G) \leq m(T)$ is satisfied. It is obvious that for each graph G with maximum degree $\Delta(G)$, $\chi(G) \leq b(G) \leq \Delta(G) + 1$. The first investigations on the graphs for which $b(G) = \Delta(G) + 1$ are due to A. El Sahili and M. Kouider [8].

Theorem 1 *[8] Let G be a d-regular graph with girth at least 5 and containing no cycles of order 6. Then the b-chromatic number of G is $d + 1$.*

They also conjectured that every d-regular graph G with girth at least 5 satisfies $b(G) = d + 1$. However, Blidia et al. disproved this conjecture in [5] by showing that the Petersen graph, which is 3-regular of girth 5, has b-chromatic number 3. Also, they have reformulated this conjecture as follows.

Conjecture 2 *[5] Let G be a d-regular graph with girth at least 5, different from the Petersen graph. Then the b-chromatic number of G is $d + 1$.*

They also showed that the conjecture 2 is true for small values of d ($d \leq 6$).

Theorem 3 *Let G be a d-regular graph with girth $g(G) \geq 5$, different from the Petersen graph, and with $d \leq 6$. Then $b(G) = d + 1$.*

In [13], Jakovac and Klavzar [13] showed that, except for the graphs in Figure 1, b-chromatic number of connected cubic graph is 4.

Theorem 4 *[13] Let G be a connected cubic graph. Then $b(G) = 4$ unless G is P, $K_3\Box K_2$, $K_{3,3}$ or G_1 (see Figure 1). In these cases, $b(P) = b(K_3\Box K_2) = b(G_1) = 3$ and $b(K_{3,3}) = 2$.*

In [10], Hoàng and Kouider showed the following result.

Lemma 5 *[10] Let G_1, G_2 be two vertex-disjoint graphs. Then the join $G_1 \vee G_2$ has $b(G_1 \vee G_2) = b(G_1) + b(G_2)$.*

The following Theorem on graphs of girth greater than 5 was proved in [15].

Theorem 6 *[15] If G is a graph with girth $g(G) \geq 6$, then $b(G) \geq \delta(G)$.*

3 Vertex b^+-critical graphs

As usual, we say that a vertex is simplicial if its neighborhood induces a clique.

Proposition 7 *Let G be a vertex b^+-critical graph. Then*
i) $b(G) \leq \delta(G) - 1$.
ii) $b(G) \leq m(G) - 1$.
iii) Vertex b^+-critical graphs do not contain simplicial vertices.
iv) $g(G) \leq 5$.

Proof. *i)* Let $v \in V(G)$ be a vertex of minimum degree $\delta(G)$. Let $b(G - v) = k$ and consider a b-coloring c of $G - v$ with k colors. Suppose that $b(G) \geq \delta(G)$. Then $k > \delta(G) = d_G(v)$. Let π be a coloring of G with k colors obtained from c as follows: $\pi(u) = c(u)$ for every vertex u of $G - v$. As $d_G(v) \leq k - 1$, one can color v with a color that does not appear in its neighborhood. Then π is a proper coloring of G with k colors. Moreover, each b-vertex of c is a b-vertex of π. Therefore, π is a b-coloring of G with k colors. So, $b(G) \geq k$, a contradiction.

ii) Suppose that $b(G) = m(G)$. As $m(G - v) \leq m(G)$ for every vertex v of G, it follows that $b(G - v) \leq m(G) = b(G)$, a contradiction.

iii) Suppose that G contains a simplicial vertex v. Then $b(G - v) = k > b(G) \geq \omega(G) > d_G(v)$. With an argument similar to that used in (i), one can show that G admits a b-coloring with k colors. So $b(G) \geq k$, a contradiction.

iv) Is a direct consequence of Theorem 6 and item (i). ■

The following corollary is immediate.

Corollary 8 *If G is a Δ-regular vertex b^+-critical graph, then $b(G) \leq \Delta(G) - 1$.*

Petersen graph P, $K_3 \square K_2$ and graph G_1 of the Figure 1 are not vertex b^+-critical since $b(P) = b(K_3 \square K_2) = b(G_1) = 3$ and $\Delta(P) = \Delta(K_3 \square K_2) = \Delta(G_1) = 3$. Also, it is easy to verify that $K_{3,3}$ is not vertex b^+-critical. So Theorem 4 and Proposition 7 (i) imply the following result.

Corollary 9 *Connected cubic graphs are not vertex b^+-critical .*

Recall that a graph G is chordal [9, 22] if every cycle of length at least four in G has a chord (an edge between non-consecutive vertices of the cycle). It is well known that any chordal graph contains at least one simplicial vertex. Hence, the following result is immediate.

Corollary 10 *Chordal graphs are not vertex b^+-critical.*

Proposition 11 *Let G_1, G_2 be two vertex-disjoint and vertex b^+-critical graphs. Then the join $G_1 \vee G_2$ is a vertex b^+-critical graph.*

Proof. Let $G = G_1 \vee G_2$ and v be the removed vertex from G. Without loss of generality, we may suppose that $v \in V(G_1)$. By Lemma 5, $b(G-v) = b((G_1 - v) \vee G_2) = b(G_1 - v) + b(G_2) > b(G_1) + b(G_2) = b(G)$. ■

The following result is due to R. Javadi and B. Omoomi [14].

Proposition 12 *[14] $b(C_3 \square C_3) = 3$.*

Proposition 13 *$C_3 \square C_3$ is a vertex b^+-critical graph.*

Proof. Let $G = C_3 \square C_3$. By Proposition 12, $b(G) = 3$. Figure 2, item (b) shows a b-coloring of $G - v$ with 4 colors where v is a vertex of G. Thus up to symmetry, $b(G - u) \geq 4 > b(G) = 3$, for every vertex u of G. So, G is vertex b^+-critical graph. ■

The following result is a direct consequence of Propositions 11 and 13.

Corollary 14 *The join of $l > 1$ copies of $C_3 \square C_3$ is vertex b^+-critical garph.*

4 Edge b^+-critical graphs

Proposition 15 *Let G be an edge b^+-critical graph. Then*
i) $b(G) \leq \Delta(G)$.
ii) $b(G) \leq m(G) - 1$.
iii) Edge b^+-critical graphs do not contain simplicial vertices.

Proof. *i)* Suppose that $b(G) = \Delta(G) + 1$. As $\Delta(G - e) \leq \Delta(G)$, it follows that $b(G - e) \leq \Delta(G) + 1 = b(G)$, a contradiction.

ii) We use a similar proof of Proposition 7 (ii) by replacing vertex v by edge e.

iii) Assume that G contains a simplicial vertex x. Let $e = xy$ be the removed edge from G such that $y \in N(x)$. Let $k = b(G-e)$ and consider a b-coloring c of $G-e$ with k colors. The vertices x and y have the same color, otherwise, by adding the edge e, c remains a b-coloring of G with k colors and thus $b(G) \geq k$, a contradiction. Consequently, any b-vertex of c is adjacent to a vertex (different from x) of color $c(x)$. If $\omega(G) \geq k$, then $b(G) \geq \omega(G) \geq k$,

a contradiction. If $\omega(G) < k$, then $d_G(x) = |N_G(x)| \leq \omega(G) - 1 < k - 1$. Therefore, one can recolor x with a missing color in its neighborhood. But in this case, by adding the edge e, c remains a b-coloring of G with k colors, a contradiction. ■

The following corollary is immediate.

Corollary 16 *Chordal graphs are not edge b^+-critical.*

Proposition 17 *The Petersen graph and $C_3\Box C_3$ are edge b^+-critical graphs.*

Proof. Let P be the Petersen graph and let $G \in \{P,\ C_3\Box C_3\}$. By Theorem 4 and Proposition 12, $b(G) = 3$. Figure 2, items (c) and (e) shows a b-coloring of $G - e$ with 4 colors where e is an edge of G. Since all edges of G play the same role, $b(G - e') \geq 4 > b(G) = 3$, for every edge e' of G. So, P and $C_3\Box C_3$ are edge b^+-critical. ■

Proposition 18 *Let P be the Petersen graph. Then $P \vee P$ and $P \vee (C_3\Box C_3)$ are edge b^+-critical.*

Proof. Let $H \in \{P,\ C_3\Box C_3\}$ and $G = P \vee H$. Set $E = E_1 \cup E_2 \cup E_3$ such that $E_1 = \{uv \in E(G) : u, v \in V(P)\}$, $E_2 = \{uv \in E(G) : u, v \in V(H)\}$ and $E_3 = \{uv \in E(G) : u \in V(P),\ v \in V(H)\}$. Let e be the removed edge from G. Since all edges of $E_1 \cup E_2$ (respectively, E_3) play the same role, there are two types of edges to consider.

Case 1: $e \in E_1 \cup E_2$.

Up to symmetry, we may suppose that $e \in E_1$. By Lemma 4, $b(G - e) = b((P - e) \vee H) = b(P - e) + b(H)$. Proposition 17 implies that $b(G - e) > b(P) + b(H) = b(P \vee H) = b(G)$. So, G is edge b^+-critical

Case 2: $e \in E_3$.

Let c_1 be a particular proper coloring of P with colors $1, 2, 3, 4$, as shown in Figure 3, item (a), and let c_2 be a particular proper coloring of H with colors $4, 5, 6, 7$, as shown in Figure 3, item (b) or item (c). Let $G - e$ be the join of P and H minus the edge $e = uv$ where $u \in V(P)$ and $v \in V(H)$, (see [Figure 3, items (a) and (b)] or [Figure 3, items (a) and (c)]). Let c be a coloring of $G - e$ defined as follows: for every vertex u of $G - e$, set $c(u) = c_1(u)$ if $u \notin H$, otherwise set $c(u) = c_2(u)$. It is a simple exercice to verify that c is a b-coloring of $G - e$ with 7 colors. Thus $b(G - e) \geq 7$. Theorem 4, Lemma 5 and Proposition 12 imply that $b(G) = b(P \vee H) = b(P) + b(H) = 6 < b(G - e)$. So, G is edge b^+-critical. ■

Using Propositions 17 and 18, we obtain the following result.

Corollary 19 *Let H be the join of $l > 1$ copies of the Petersen graph. Then H and $(C_3 \square C_3) \vee H$ are edge b^+-critical.*

Proof. Let $G = G_1 \vee G_2 \vee G_3 ... \vee G_l$ $(l \geq 2)$ with $G_1 \in \{P, C_3 \square C_3\}$ and $G_i = P$ $(2 \leq i \leq l)$. Theorem 4 and Proposition imply that $b(G_i) = 3$ for each $i \in \{1, ..., l\}$. Also, by Lemma 5, we have

$$b(G) = b(G_1 \vee G_2 \vee G_3 ... \vee G_l) = \sum_{i=1}^{l} b(G_i) = 3l$$

Let $e = uv$ be the removed edge from G. So, there are three cases to consider.

i) $u, v \in V(G_1)$. Lemma 5 implies that

$$\begin{aligned} b(G-e) &= b((G_1 - e) \vee G_2 \vee ... \vee G_l) \\ &= b(G_1 - e) + \sum_{i=2}^{l} b(G_i) \\ &= b(G_1 - e) + 3(l-1). \end{aligned}$$

By Proposition 17, $b(G_1 - e) > b(G_1) = 3$. Thus $b(G - e) > 3l = b(G)$.

ii) $u \in V(G_1)$ and $v \notin V(G_1)$. Without loss of generality, we may suppose that $v \in V(G_2)$. Then

$$\begin{aligned} b(G-e) &= b((G_1 \vee G_2 - e)) \vee G_3 \vee ... \vee G_l) \\ &= b((G_1 \vee G_2) - e) + \sum_{i=3}^{l} b(G_i) \\ &= b((G_1 \vee G_2) - e) + 3(l-2) \end{aligned}$$

Proposition 18 and Lemma 5 imply that $b((G_1 \vee G_2) - e) > b(G_1 \vee G_2) = b(G_1) + b(G_2) = 6$. So $b(G - e) > 3l = b(G)$.

iii) $u, v \notin V(G_1)$. Without loss of generality, we may suppose that $u \in V(G_2)$. If $v \in V(G_2)$, then

$$\begin{aligned} b(G-e) &= b(G_1 \vee (G_2 - e) \vee G_3 \vee ... \vee G_l) \\ &= b(G_1) + b(G_2 - e) + \sum_{i=3}^{l} b(G_i) \\ &= 3 + b(G_2 - e) + 3(l-2) \end{aligned}$$

By Proposition 17, $b(G_2 - e) > b(G_2) = 3$. Thus $b(G - e) > 3l = b(G)$. So $b(G - e) > 3l = b(G)$.
Suppose now that $v \notin V(G_2)$. Without loss of generality, we may suppose that $v \in V(G_3)$. Then

$$\begin{aligned} b(G-e) &= b(G_1 \vee (G_2 \vee G_3 - e) \vee G_4 \vee ... \vee G_l) \\ &= b(G_1) + b(G_2 \vee G_3 - e) + \sum_{i=4}^{l} b(G_i) \\ &= 3 + b(G_2 \vee G_3 - e) + 3(l-3) \end{aligned}$$

Proposition 18 and Lemma 5 imply that

$$b((G_2 \vee G_3) - e) > b(G_2 \vee G_3) = b(G_2) + b(G_3) = 6.$$

So,$b(G - e) > 3l = b(G)$. Hence, G is edge b^+-critical ■

The following two propositions are a direct consequence of Theorem 4, Propositions 15 (i),17 and Theorem 3.

Proposition 20 *The Petersen graph is the only connected cubic edge b^+-critical graph.*

Proposition 21 *The Petersen graph is the only d-regular ($d \leq 6$) edge b^+-critical graph with girth at least* 5.

Based on the results of M. Blidia et al. [5] on d-regular graphs with girth at least 5, we state the following conjecture.

Conjecture 22 *Any d-regular graph with girth at least* 5, *different from Petersen graph , is not edge b^+-critical.*

5 Open questions

Problem 23 *Is it true that $C_3 \square C_3$ and the join of $l > 1$ copies of $C_3 \square C_3$ are the only vertex b^+-critical graphs?.*

Problem 24 *Let H be the join of $l > 1$ copies of the Petersen graph. Is it true that the only graphs edge b^+-critical are P, H and $(C_3 \square C_3) \vee H$?.*

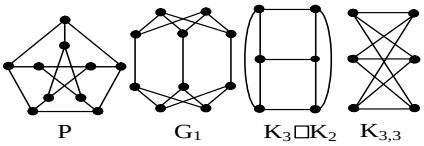


Figure 1: Cubic graphs whose b-chromatic number is less than 4.

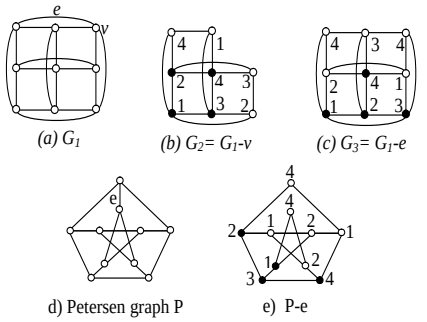


Figure 2: b-coloring of $C_3 \square C_3 - v$, $C_3 \square C_3 - e$ and $P - e$ with 4 colors.

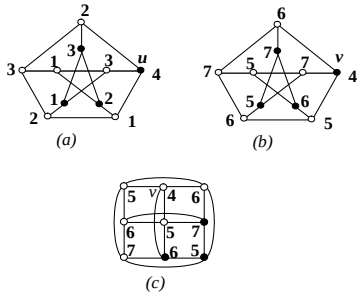


Figure 3: Coloring of P and $C_3 \square C_3$ with 4 colors.

References

[1] C. Berge. *Graphs*. North Holland, 1985.

[2] F. Bonomo, G. Durán, F. Maffray, J. Marenco, M. Valencia-Pabon. On the b-coloring of cographs and P_4-sparse graphs. Graphs and Combinatorics 25(2009)153 − 167.

[3] M. Blidia, N. Ikhlef Eschouf, F. Maffray. b-coloring of some bipartie graphs. To apear in Australasian Journal of Combinatorics.

[4] M. Blidia, N. Ikhlef Eschouf, F. Maffray. On vertex b-critical trees. To apear in Opuscula Mathematica.

[5] M. Blidia, F. Maffray, Z. Zemir. On b-colorings in regular graphs. Discrete Applied Mathematics 157 (2009) 1787 − 1793.

[6] V. Compos, C. Linhares, F. Maffray, A. Sliva. b-chromatic number of Cacti. Electronic Notes In Discrete Mathematics, 35 : 281 − 286, 2009.

[7] S. Cabello, M. Jakovac. On the b-chromatic number of regular graphs. Discrete applied mathematics, 2011, vol. 159, no 13, pp. 1303 − 1310.

[8] A. El Sahili, M. Kouider. About b-colorings of regular graphs, Res. Rep. 1432, LRI, Univ. Orsay, France, 2006.

[9] M.C. Golumbic. Algorithmic Graph Theory and Perfect Graphs, Annals of Discrete Mathematics 57, 2nd Edition, North Holland, 2004.

[10] C.T. Hoàng, M. Kouider. On the b-dominating coloring of graphs. Discrete Appl. Math. 152 (2005) 176 − 186.

[11] N. Ikhlef Eschouf. Characterization of some b-chromatic edge critical graphs. Australasian Journal of Combinatorics 47 (2010), Pages 21−35.

[12] R.W. Irving, D.F. Manlove. The b-chromatic number of graphs. Discrete Appl. Math. 91(1999) 127 − 141.

[13] M. Jakovac, S. Klavžar. The b-Chromatic Number of Cubic Graphs. Graphs and Combinatorics (2010) 26: 107 − 118.

[14] R. Javadi, B. Omoomi. On b-coloring of cartesian product of graphs. To apear in Ars Combinatoria.

[15] M.Kouider, b-chromatic number of a graph, subgraphs and degrees Rapport interne LRI 1392, Univ. Orsay, France, 2004.

[16] M. Kouider, M. Mahéo. Some bounds for the b-chromatic number of a graph. Discrete Math. 256 (2002) $267 - 277$.

[17] M. Kouider, M. Zaker. Bounds for the b-chromatic number of some families of graphs. Discrete mathematics, vol. 306, no 7, pp. $617 - 623$, 2006.

[18] J. Kratochvíl, Z. Tuza, M. Voigt. On the b-chromatic number of graphs. Lecture Notes in Computer Science 2573 (2002), $310 - 320$.

[19] D.F. Manlove. Minimaximal and maximinimal optimization problems: a partial order-based approach. PhD thesis, technical report tr-$1998-27$ of the Computing Science Department of Glasgow University, 1998.

[20] S. Shaebani. On The b-chromatic number of regular graphs without 4-Cycle. arXiv : 1103.152 v1 [math.CO] 08 Mars 2011.

[21] S. Shaebani. The b-Chromatic Number of Regular Graphs via The Edge Connectivity. arXiv:1105.2909 v1 [math.CO] 14 May 2011.

[22] J. Ramirez-Alfonsin, B. Reed. Perfect Graphs. Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, 2001.

Sur le Nombre de Bondage Localisateur

Widad DALI[1] and Mostafa BLIDIA[2]

[1] Faculté des Sciences Commerciales, Économiques et de Gestions
Université d'Alger 3, Alger, Algérie
widdal@yahoo.fr
[2] Faculté des Sciences, Université U.S.D.B., B.P. 270 Blida, Algérie
m_blidia@yahoo.fr

Résumé Dans ce travail nous nous intéressons à un paramètre de domination, le nombre de domination localisatrice γ_L. Nous définissons par rapport à ce paramètre un autre paramètre de domination le nombre de bondage localisateur $b_L(G)$. Nous donnons des valeurs exactes et des bornes supérieures du $b_L(G)$ pour certaines classes de graphes.

Mots clés : Domination localisatrice et nombre de bondage.

1 Introduction

En 1987 Slater introduit les ensembles dominants localisateurs (voir [2], [1]). La principale motivation de l'introduction de ce concept est l'étude de la protection contre les incendies.

Soit $G = (V, E)$ un graphe simple. Le voisinage ouvert d'un sommet $u \in V$ est $N(u) = \{v \in V : uv \in E\}$, $|N(u)| = deg(u)$. Un ensemble $D \subseteq V$ est un dominant localisateur (E.D.L.) dans un graphe simple G si pour tout sommet $u \in V - D, N(u) \cap D \neq \emptyset$ et pour toute paire de sommets u et v dans $V - D$ les ensembles $N(u) \cap D$ et $N(v) \cap D$ sont différents.
La cardinalité d'un ensemble dominant localisateur ayant la plus petite taille dans G est appelée nombre de domination localisatrice, notée $\gamma_L(G)$.

Dans le but de l'étude de la vulnérabilité des réseaux de communication, Fink, Jacobson, Kinch et Roberts [3] introduisent un nouveau paramètre de domination le nombre de bondage d'un graphe simple non vide.

Le nombre de bondage $b(G)$ d'un graphe $G = (V, E)$ est la cardinalité du plus petit ensemble d'arêtes $E' \subseteq E$ pour lequel le nombre de domination du graphe partiel $G - E'$ augmente (c.à.d., $b(G) = min\{|E'|, E' \subseteq E : \gamma(G - E') > \gamma(G)\}$). L'introduction du nombre de bondage a motivé l'étude des nombres de bondage par rapport à d'autres paramètres comme le nombre de bondage pair et le nombre de bondage restreint. Pour plus de detail voir [4], [5].

Dans ce travail nous définissons le nombre de bondage par rapport au nombre de domination localisatrice pour les graphes simples et non vides. Le nombre de bondage localisateur $b_L(G)$ d'un graphe $G = (V, E)$ est la cardinalité du plus

petit ensemble d'arêtes $E^{'} \subseteq E$ pour lequel le nombre de domination localisatrice du graphe partiel $G - E^{'}$ augmente. $b_L(G) = min\{|E^{'}|, E^{'} \subseteq E : \gamma_L(G - E^{'}) > \gamma_L(G)\}$.

Comme tout graphe $G = (V, E)$ non vide admet au moins un graphe partiel $H = (V, \emptyset)$ tel que $\gamma_L(H) > \gamma_L(G)$. Alors, le nombre de bondage localisateur $b_L(G)$ d'un graphe G est bien défini.

2 Valeurs Exactes du Nombre de Bondage Localisateur

Nous calculons dans cette partie des valeurs exactes du nombre de bondage localisateur pour certaines classes de graphes de structures simples.

2.1 Le Nombre de Bondage Localisateur dans les Chaînes et les Cycles

Proposition 1. *Le nombre de bondage localisateur d'une chaîne P_n, $n \geq 2$ est*

$$b_L(P_n) = \begin{cases} 2 \text{ } si \, n \equiv 3[5], \\ 1 \quad sinon. \end{cases}$$

Preuve. Nous rappelons que le nombre de domination localisatrice d'une chaîne P_n, $n \geq 2$ est (voir [2])

$$\gamma_L(P_n) = \begin{cases} 2\lceil n/5 \rceil - 1 \text{ si } n \equiv 1[5] \text{ } ou \text{ } n \equiv 2[5], \\ 2\lceil n/5 \rceil \qquad\qquad\quad \text{sinon.} \end{cases}$$

La propriété est facilement vérifiée pour $n \leq 5$.
Soit maintenant $n \geq 6$. En enlevant une arête d'une chaîne P_n, considérons le cas où le graphe partiel H obtenu est composé de deux chaînes notées respectivement P_{n_1} et P_{n_2} avec $n_1 + n_2 = n$ et $n_1 \geq 2$, $n_2 \geq 2$. Nous discutons les cas suivants,

Si $n \equiv 0[5]$, alors $\gamma_L(P_n) = 2\lceil n/5 \rceil$ et on choisit P_{n_1} et P_{n_2} avec $n_1 \equiv 2[5]$ et $n_2 \equiv 3[5]$, calculons $\gamma_L(H)$.
$\gamma_L(H) = \gamma_L(P_{n_1}) + \gamma_L(P_{n_2}) = 2\lceil n_1/5 \rceil - 1 + 2\lceil n_2/5 \rceil = 2(n_1 + 3)/5 - 1 + 2(n_2 + 2)/5 = 2(n/5) + 1 = 2\lceil n/5 \rceil + 1 > \gamma_L(P_n)$.
D'où $b_L(P_n) = 1$, pour $n \equiv 0[5]$.

Si $n \equiv 1[5]$, alors $\gamma_L(P_n) = 2\lceil n/5 \rceil - 1$ et on choisit P_{n_1} et P_{n_2} avec $n_1 \equiv 3[5]$ et $n_2 \equiv 3[5]$, calculons $\gamma_L(H)$.
$\gamma_L(H) = \gamma_L(P_{n_1}) + \gamma_L(P_{n_2}) = 2\lceil n_1/5 \rceil + 2\lceil n_2/5 \rceil = 2(n_1 + 2)/5 + 2(n_2 + 2)/5 = 2(n + 4)/5 = 2\lceil n/5 \rceil > \gamma_L(P_n)$.
D'où $b_L(P_n) = 1$, pour $n \equiv 1[5]$.

Si $n \equiv 2[5]$, alors $\gamma_L(P_n) = 2\lceil n/5 \rceil - 1$ et on choisit P_{n_1} et P_{n_2} avec $n_1 \equiv 3[5]$ et $n_2 \equiv 4[5]$, calculons $\gamma_L(H)$.

$\gamma_L(H) = \gamma_L(P_{n_1}) + \gamma_L(P_{n_2}) = 2\lceil n_1/5\rceil + 2\lceil n_2/5\rceil = 2(n_1+2)/5 + 2(n_2+1)/5 = 2(n+3)/5 = 2\lceil n/5\rceil > \gamma_L(P_n)$.
D'où $b_L(P_n) = 1$, pour $n \equiv 2[5]$.

Si $n \equiv 3[5]$, pour toute partition de P_n en deux composantes connexes P_{n_1} et P_{n_2} telle que $n_1 + n_2 = n$ et $\begin{cases} n_1 \equiv 0[5] \quad \text{et } n_2 \equiv 3[5], \\ \text{ou, } n_1 \equiv 1[5] \text{ et } n_2 \equiv 2[5], \\ \text{ou, } n_1 \equiv 4[5] \text{ et } n_2 \equiv 4[5]. \end{cases}$
On a $\gamma_L(P_{n_1}) + \gamma_L(P_{n_2}) = \gamma_L(P_n)$, donc $b_L(P_n) \geq 2$.
En enlevant deux arêtes de P_n, considérons le graphe partiel H composé de deux sommets isolés et de la chaîne P_{n_3}, $n_3 \geq 2, n_3 \equiv 1[5]$ avec $1 + 1 + n_3 = n$, calculons $\gamma_L(H)$.
$\gamma_L(H) = 2 + \gamma_L(P_{n_3}) = 2 + 2\lceil n_3/5\rceil - 1 = 2 + 2(n_3+4)/5 - 1 = 2(n+2)/5 + 1 = 2\lceil n/5\rceil + 1 > \gamma_L(P_n)$, ainsi $b_L(P_n) \leq 2$.
D'où $b_L(P_n) = 2$, pour $n \equiv 3[5]$.

Si $n \equiv 4[5]$, alors $\gamma_L(P_n) = 2\lceil n/5\rceil$ et on choisit P_{n_1} et P_{n_2} avec $n_1 \equiv 1[5]$ et $n_2 \equiv 3[5]$, calculons $\gamma_L(H)$.
$\gamma_L(H) = \gamma_L(P_{n_1}) + \gamma_L(P_{n_2}) = 2\lceil n_1/5\rceil - 1 + 2\lceil n_2/5\rceil - 2 = 2(n_1+4)/5 - 1 + 2(n_2+2)/5 - 2 = 2(n+3)/5 + 1 = 2\lceil n/5\rceil + 1 > \gamma_L(P_n)$.
D'où $b_L(P_n) = 1$, pour $n \equiv 4[5]$. ■

Comme conséquence de la Proposition 1, on a le corollaire suivant.

Corollaire 1. *Le nombre de bondage localisateur d'un cycle C_n, $n \geq 3$ est*

$$b_L(C_n) = \begin{cases} 3 \textit{ si } n \equiv 3[5], \\ 2 \quad \textit{sinon.} \end{cases}$$

Preuve. Nous rappelons que le nombre de domination localisatrice d'un cycle C_n, $n \geq 3$ est (voir [2])

$$\gamma_L(C_n) = \begin{cases} 2\lceil n/5\rceil - 1 \text{ si } n \equiv 1[5] \textit{ ou } n \equiv 2[5], \\ 2\lceil n/5\rceil \qquad\qquad \text{sinon.} \end{cases}$$

La propriété est facilement vérifiée pour $n \leq 5$.
Soit maintenant $n \geq 6$. Pour toute arête e de C_n, $\gamma_L(C_n - e) = \gamma_L(P_n) = \gamma_L(C_n)$, donc $b_L(C_n) \geq 2$.
Si $n \equiv 3[5]$. De la proposition 1, $b_L(P_n) = 2$. Donc pour tout couple d'arêtes $\{e, e'\}$ de C_n, on a $\gamma_L(C_n - \{e, e'\}) = \gamma_L(P_n - e') = \gamma_L(P_n) = \gamma_L(C_n)$, ainsi $b_L(C_n) \geq 3$.
D'autre part, considérons trois arêtes e, e' et e'' de C_n telles que
$\gamma_L(C_n - \{e, e', e''\}) = \gamma_L(P_n - \{e', e''\}) > \gamma_L(P_n) = \gamma_L(C_n)$, ainsi $b_L(C_n) \leq 3$.
D'où $b_L(C_n) = 3$, pour $n \equiv 3[5]$.
Si $n \not\equiv 3[5]$. De la proposition 1, $b_L(P_n) = 1$. Considérons deux arêtes e et e' de C_n telles que $\gamma_L(C_n - \{e, e'\}) = \gamma_L(P_n - e') > \gamma_L(P_n) = \gamma_L(C_n)$, ainsi $b_L(C_n) \leq 2$.
D'où $b_L(C_n) = 2$, pour $n \not\equiv 3[5]$. ■

2.2 Le Nombre de Bondage Localisateur dans les Graphes Complets et les Graphes Multipartis Complets

Proposition 2. *Le nombre de bondage localisateur d'un graphe complet K_n, $n \geq 2$ est $b_L(K_n) = n(n-1)/2$.*

Preuve. Comme le nombre de domination localisatrice d'un graphe complet K_n, $n \geq 2$ est $\gamma_L(K_n) = n-1$ (voir [2]). Le seul cas où le nombre de domination localisatrice peut être augmenté est d'avoir un graphe partiel H tel que $\gamma_L(H) = n$, ce qui correspond au nombre de domination localisatrice du graphe partiel vide de K_n. Donc pour augmenter $\gamma_L(K_n)$ il faut enlever toutes les arêtes du graphe complet, ainsi $b_L(K_n) = n(n-1)/2$. ■

Théorème 1. *Le nombre de bondage localisateur d'un graphe t-partie complet $K_{n_1,n_2,...,n_t}$, $t \geq 2$ tel que $n_1 \leq n_2 \leq ... \leq n_t$ est*

$$b_L(K_{n_1,n_2,...,n_t}) = \begin{cases} n_2(n_1-1) & si\ t=2\ et\ n_1 \geq 2, \\ n_1 \sum_{2}^{t} n_i & si\ t \geq 3\ et\ n_1 \geq 2, \\ n_2 & si\ t=2\ et\ n_1 = 1, \\ min\{m \sum_{i=m+1}^{t} n_i + m(m-1)/2, & \\ \quad (n_{m+1}-1) \sum_{i=1, i\neq m+1}^{t} n_i\} & si\ t \geq 3,\ n_m = 1,\ n_{m+1} \geq 2\ et\ 1 \leq m < t. \end{cases}$$

Preuve. Soit $K_{n_1,n_2,...,n_t} = (V, E)$, $t \geq 2$ un graphe t-partie complet tel que $n_1 \leq n_2 \leq ... \leq n_t$. On note par N_i la partie de $K_{n_1,n_2,...,n_t}$ de cardinalité n_i, $i = \overline{1,t}$. Nous rappelons que le nombre de domination localisatrice d'un graphe t-partie complet est ([2])

$$\gamma_L(K_{n_1,n_2,...,n_t}) = \begin{cases} \sum_{i=1}^{t}(n_i - 1) & \text{si } n_1 \geq 2, \\ n_2 & \text{si } t=2\ et\ n_1 = 1. \end{cases}$$

et on peut conclure que,

$$\gamma_L(K_{n_1,n_2,...,n_t}) = \sum_{i=m+1}^{t} (n_i - 1) + (m-1) \text{ si } t \geq 3,\ n_m = 1,\ n_{m+1} \geq 2\ et\ 1 \leq m < t.$$

$1^{er} cas.$ $n_1 \geq 2$.

Si $t = 2$. Observons d'abord que le graphe partiel H obtenu de K_{n_1,n_2} en enlevant les $n_2(n_1-1)$ arêtes incidentes à $n_1 - 1$ sommets de N_1 est isomorphe au graphe $\underbrace{K_1 \cup ... \cup K_1}_{n_1-1 fois} \cup K_{1,n_2}$, ce qui fait que $\gamma_L(H) = n_1 + n_2 - 1 > \gamma_L(K_{n_1,n_2})$, donc $b_L(K_{n_1,n_2}) \leq n_2(n_1-1)$. D'autre part, dans

tout graphe partiel H obtenu en enlevant au plus $n_2(n_1-1)-1$ arêtes de K_{n_1,n_2}, il existe au moins deux sommets non isolés x et y et au moins un sommet de degré supérieur ou égale à deux dans N_1, sans perte de généralité soit $deg(x) \geq 2$ et dans ce cas le sous ensemble $(N_1\backslash\{x\}) \cup (N_2\backslash\{y^{'}\})$ où $y^{'} \in N_{K_{n_1,n_2}}(y)$, est un E.D.L. du graphe H, donc $\gamma_L(H) \leq \gamma_L(K_{n_1,n_2})$, alors $b_L(K_{n_1,n_2}) \geq n_2(n_1-1)$, ainsi $b_L(K_{n_1,n_2}) = n_2(n_1-1)$.
Si $t > 2$. Comme tout $\gamma_L(K_{n_1,n_2,\ldots,n_t})$-ensemble contient exactement $n_i - 1$ sommets de chaque partie N_i, pour augmenter le nombre de domination localisatrice du graphe t-partie complet, il faut enlever le minimum d'arêtes pour obtenir un graphe partiel H tel que tout $\gamma_L(H)$-ensemble contient au moins tous les sommets d'une partie N_i et $n_j - 1$ sommets de chaque partie N_j, $j \neq i$, $i,j = \overline{1,t}$. Pour cela, il faut supprimer le minimum d'arêtes de telle manière à obtenir un graphe partiel qui contient n_i sommets isolés d'une partie N_i, et comme

$$n_1 \sum_{2}^{t} n_i \leq n_k \sum_{1,i\neq k}^{t} n_i, \forall k \neq 1,$$

on a,

$$min\{n_j \sum_{i\neq j} n_i, i,j = 1,t\} = n_1 \sum_{2}^{t} n_i.$$

Donc le minimum d'arêtes à enlever est atteint pour la partie N_1, en effet, si nous isolons les sommets de la partie N_1, le graphe partiel H obtenu est isomorphe au graphe $\underbrace{K_1 \cup \ldots \cup K_1}_{n_1 fois} \cup K_{n_2,\ldots,n_t}$, et il est de nombre de domination localisatrice strictement supérieur à celui de $K_{n_1,n_2,\ldots,n_t}$, ainsi $b_L(K_{n_1,n_2,\ldots,n_t}) = n_1 \sum_{2}^{t} n_i$.

$2^{ème} cas.$ $n_1 = 1$.
Si $t = 2$. On a d'une part $\gamma_L(K_{1,n_2} - E) = n_2 + 1 > \gamma_L(K_{1,n_2})$ et d'autre part, $\forall E^{'} \subset E$ on a $\gamma_L(K_{1,n_2} - E^{'}) = \gamma_L(K_{1,n_2})$, il s'ensuit que $b_L(K_{1,n_2}) = |E| = n_2$.
Si $t > 2$. Posons $M = \bigcup_{i=1}^{m} N_i$, $|M| = m$, comme tout $\gamma_L(K_{n_1,n_2,\ldots,n_t})$-ensemble contient respectivement $n_i - 1$ et $m - 1$ sommets des parties $N_i, i \geq m+1$ et M, afin d'augmenter $\gamma_L(K_{n_1,n_2,\ldots,n_t})$, il faut enlever le minimum d'arêtes pour obtenir un graphe partiel H tel que tout $\gamma_L(H)$-ensemble doit comprendre au moins tous les sommets de la partie M ou d'une partie $N_i, i \geq m+1$. Pour cela, dans le premier cas il faut isoler tout les sommets de la partie M, ceci revient donc à enlever les $m \sum_{i=m+1}^{t} n_i + m(m-1)/2$ arêtes, d'où on a

$\gamma_L(H) = \gamma_L(\underbrace{K_1 \cup ... \cup K_1}_{m fois} \cup K_{n_{m+1},...,n_t}) = m + \gamma_L(K_{n_{m+1},...,n_t}) > \gamma_L(K_{n_1,n_2,...,n_t})$.

Ou bien, isoler tout les sommets sauf un d'une partie $N_i, i \geq m+1$, et comme

$$-\sum_{i=1,i\neq m+1}^{t} n_i \leq -\sum_{i=1,i\neq k}^{t} n_i, \forall k \geq m+2,$$

et

$$n_{m+1}\sum_{i=1,i\neq m+1}^{t} n_i \leq n_k \sum_{i=1,i\neq k}^{t} n_i, \forall k \geq m+2,$$

on a,

$$min\{(n_j - 1)\sum_{i\neq j} n_i, \ \begin{matrix} i = \overline{1,t} \\ j = \overline{m+1,t.} \end{matrix} \} = (n_{m+1} - 1) \sum_{i\neq m+1} n_i.$$

En effet, le graphe partiel $H = \underbrace{K_1 \cup ... \cup K_1}_{n_{m+1}-1 fois} \cup K_{n_1,...,n_m,1,n_{m+2},...,n_t}$ obtenu de $K_{n_1,n_2,...,n_t}$ en supprimant les $(n_{m+1} - 1) \sum_{i\neq m+1} n_i$ arêtes incidentes aux sommets de la partie N_{m+1} est tel que $\gamma_L(H) = \sum_{i=m+1}^{t} (n_i - 1) + m > \gamma_L(K_{n_1,n_2,...,n_t})$.

Ainsi, il reste à choisir le nombre minimum entre $m\sum_{i=m+1}^{t} n_i + m(m-1)/2$ et $(n_{m+1} - 1) \sum_{i\neq m+1} n_i$, pour obtenir la valeur du nombre de bondage localisateur. Par conséquent,

$b_L(K_{n_1,n_2,...,n_t}) = min\{m \sum_{i=1,i\neq m+1}^{t} n_i + m(m-1)/2, (n_{m+1}-1) \sum_{i=1,i\neq m+1}^{t} n_i\}$.

■

3 Bornes Supérieures du Nombre de Bondage Localisateur

Nous commençons cette partie en établissant des bornes des bornes supérieures du nombre de bondage localisateur dans la classe des arbres.

3.1 Le Nombre de Bondage Localisateur dans les Arbres

Nous rappelons qu'un sommet $u \in V$ est dit pendant si et seulement si $deg(u) = 1$, le voisin d'un sommet pendant est dit sommet support et un arbre non trivial est un arbre d'ordre $n \geq 2$.

Théorème 2. *Si T est un arbre non trivial, alors $b_L(T) \leq \Delta(T)$.*

Preuve. Soient, $T = (V, E)$ un arbre non trivial et v un sommet support de T. Nous notons par $L_v = \{w_1, w_2, ..., w_k\}$, $k \geq 1$ l'ensemble des sommets pendants voisins à v et par E_v l'ensemble des arêtes incidentes à v. Considérons le graphe partiel $H = (T - E_v)$. Tout dominant localisateur minimum D_L de H contient le sous ensemble de sommets $\{v\} \cup L_v$. L'ensemble $D_L \backslash \{w_1\}$ est un dominant localisateur de l'arbre T de cardinalité strictement inférieure au nombre de domination localisatrice du graphe H, d'où $b_L(T) \leq |E_v| = d_T(v) \leq \Delta(T)$. ■

Comme conséquence immédiate du Théorème 2 on a le corollaire suivant.

Corollaire 2. *Si T est un arbre non trivial avec $l(T)$ sommets pendants, alors $b_L(T) \leq l(T)$.*

Corollaire 3. *Si T est un arbre non trivial qui admet un sommet support de degré deux, alors $b_L(T) \leq 2$.*

Preuve. Soit $S(T)$ l'ensemble des sommets supports de l'arbre T. Pour tout sommet $v \in S(T)$, $\gamma_L(T - E_v) > \gamma_L(T)$, où E_v est l'ensemble des arêtes incidentes à v. Donc $b_L(T) \leq min\{d_T(v), v \in S(T)\}$, et comme T admet un sommet support v de degré deux, $min\{d_T(v), v \in S(T)\} = 2$, ainsi $b_L(T) \leq 2$. ■

Nous présentons les résultats obtenus suite à notre étude des arbres dans cette inéquation $b_L(T) \leq min\{d_T(v), v \in S(T)\} \leq \Delta(T) \leq l(T)$. En considérant que $b_L(T) \leq l(T)$, cette borne n'est pas atteinte pour tous les arbres. La question qu'on se pose est, pour quels arbres cette borne est atteinte ?

Proposition 3. *$b_L(T) = l(T)$ si et seulement si T est une chaîne P_n, $n \equiv 3[5]$ ou T est une étoile $K_{1,n}$, $n \geq 2$.*

Preuve. La borne du nombre de bondage localisateur est atteinte si et seulement si $b_L(T) = min\{d_T(v), v \in S(T)\} = \Delta(T) = l(T)$.
La condition suffisante.
Soit $T = P_n$, $n \equiv 3[5]$. De la proposition 1, $b_L(P_n) = 2$. D'autre part, $min\{d_{P_n}(v), v \in S(P_n)\} = \Delta(P_n) = l(P_n) = 2$. D'où l'égalité.
Soit $T = K_{1,n}$, $n \geq 2$. Du Théorème 1, $b_L(K_{1,n}) = n$. D'autre part, $min\{d_{K_{1,n}}(v), v \in S(K_{1,n})\} = \Delta(K_{1,n}) = l(K_{1,n}) = n$. D'où l'égalité.
La condition nécessaire.
Pour $l(T) = 2$, T est une chaîne. De la proposition 1, $b_L(P_n) = 2 \Leftrightarrow n \equiv 3[5]$.
Pour $l(T) \geq 3$. Soit T un arbre tel que $b_L(T) = min\{d_T(v), v \in S(T)\} = \Delta(T) = l(T)$ et supposons que $T \neq K_{1,n}$, $n \geq 2$. Donc T admet au moins deux supports v et $v^{'}$ et forcément $d_T(v) = d_T(v^{'}) = \Delta(T)$, et comme v et $v^{'}$ sont reliés par une unique chaîne, T a au moins $2(\Delta(T)-1)$ sommets pendants, $l(T) \geq 2(\Delta(T)-1)$ mais $\Delta(T) = l(T)$. Il résulte que $l(T) \leq 2$, contradiction.
Ainsi la borne du nombre de bondage localisateur n'est atteinte que si T est une chaîne P_n, $n \equiv 3[5]$ ou T est une étoile $K_{1,n}$, $n \geq 2$. ■

Nous terminons cette section en donnant des bornes supérieures du nombre de bondage localisateur pour des graphes de structure plus générale.

3.2 Le Nombre de Bondage Localisateur dans les Graphes

Observation 1. *Si k arêtes peuvent être enlevées d'un graphe G, avec tout graphe partiel H de G vérifie $\gamma_L(H) \geq \gamma_L(G)$, pour obtenir un graphe partiel H avec $b_L(H) = 1$, alors $b_L(G) \leq k + 1$.*

Théorème 3. *Si G est un graphe tel que tout graphe partiel H vérifie $\gamma_L(H) \geq \gamma_L(G)$, alors*
$b_L(G) \leq min\{deg(u) + deg(v) - 1$, u et v sont des sommets voisins dans $G\}$.

Preuve. Soient u et v deux sommets voisins d'un graphe $G = (V, E)$. Nous notons par E_x l'ensemble des arêtes incidentes à un sommet $x \in V$. Considérons le graphe partiel $H = G - E^{'}$ avec $E^{'} = (E_u \cup E_v) - \{uv\}$. Il est clair que $b_L(H) = 1$ et comme $|(E_u \cup E_v) - \{uv\}| = deg(u) + deg(v) - 2$, il s'ensuit de l'observation 1 que $b_L(G) \leq deg(u) + deg(v) - 1$. Ainsi
$b_L(G) \leq min\{deg(u) + deg(v) - 1$, u et v sont des sommets voisins dans $G\}$. ■

Corollaire 4. *Si G est un graphe sans sommets isolés tel que tout graphe partiel H vérifie $\gamma_L(H) \geq \gamma_L(G)$, alors $b_L(G) \leq \delta(G) + \Delta(G) - 1$.*

Preuve. Soient u et v deux sommets adjacents d'un graphe $G = (V, E)$ tels que $deg(u) = \delta(G)$. Nous déduisons du théorème 1 que $b_L(G) \leq \delta(G) + deg(v) - 1$, ceci implique que $b_L(G) \leq \delta(G) + \Delta(G) - 1$. ■

Proposition 4. *Si H est un graphe partiel d'un arbre T non trivial, alors $\gamma_L(H) \geq \gamma_L(T)$.*

Preuve. . Soit $T = (V, E)$ un arbre non trivial. Supposons qu'il existe au moins un graphe partiel $H = T - E^{'}, E^{'} \subseteq E$ tel que $\gamma_L(H) < \gamma_L(T)$. Considérons un $\gamma_L(H)$-ensemble D_L. En rajoutant l'ensemble des arêtes $E^{'}$ à H, il est clair que pour toute paire de sommets u et v de $V \backslash D_L$ les ensembles $N_G(u) \cap D_L$ et $N_G(v) \cap D_L$ sont différents et non vides. Alors D_L est un ensemble dominant localisateur de l'arbre T. D'où $\gamma_L(T) \leq |D_L| = \gamma_L(H) < \gamma_L(T)$, contradiction. ■

Du Corollaire 4 et de la Proposition 4, nous pouvons déduire que $b_L(T) \leq \Delta(T)$ et $b_L(P_n) \leq 2$.

Proposition 5. *Si G est un graphe qui admet un sommet v tel que $\gamma_L(G - v) \geq \gamma_L(G)$, alors $b_L(G) \leq \Delta(G)$.*

Preuve. Soit $G = (V, E)$ un graphe et soit v un sommet de G tel que $\gamma_L(G - v) \geq \gamma_L(G)$. On a $\gamma_L(G - v) = \gamma_L(G - E_v) - 1$, avec E_v est l'ensemble des arêtes incidentes à v. Il résulte que $\gamma_L(G - E_v) \geq \gamma_L(G) + 1 > \gamma_L(G)$.
Ainsi $b_L(G) \leq |E_v| = deg(v) \leq \Delta(G)$. ■

Comme conséquence immédiate de la Proposition 4, on a le corollaire suivant,

Corollaire 5. *Si $b_L(G) > \Delta(G)$, alors $\gamma_L(G-v) < \gamma_L(G)$, pour tout sommet v de G.*

Remarquons que la Proposition 5 s'applique au cas des cycles C_n lorsque $n \not\equiv 3[5]$, par contre le Corollaire 5 s'applique au cas des cycles C_n lorsque $n \equiv 3[5]$ et au cas des graphes complets d'ordre strictement supérieur à deux.

4 Conclusion

Dans ce travail nous nous sommes intéressés à la domination localisatrice. Nous avons défini le nombres de bondage localisateur $b_L(G)$. Nous avons donné des valeurs exactes du $b_L(G)$ pour les chaînes, les cycles, les graphes complets et les graphes multipartis complets. Nous avons également établi des bornes supérieures du $b_L(G)$ pour les arbres et pour les graphes dans quelques cas particuliers. En conclusion, il serait intéressant de déterminer des valeurs exactes et des bornes inférieures ou supérieures du $b_L(G)$ pour d'autres classes de graphes.

Références

1. P.J. Slater, Domination and acyclic in graphs, *Networks*, 17 (1987) 55-64.
2. P.J. Slater, Dominating and reference sets in graphs, *J.Mathematical and Physical Sciences*, 22 (1988) 445-455.
3. J.F. Fink, M.S. Jacobson, L.F. Kinch and J. Roberts, The bondage number of graph, *Discrete Mathematics*, 86 (1990) 47-57.
4. J. Raczek, Paired bondage number, *Discrete Mathematics*, 308 (2008) 5570-5575.
5. J.H. Hatting and A.R. Plummer, Restrained bondage in graphs, *Discrete Mathematics*, 2007.

On connected k-domination in graphs

Karima Attalah and Mustapha Chellali*
LAMDA-RO Laboratory, Department of Mathematics
University of Blida
B.P. 270, Blida, Algeria
E-mail: k_attalah@yahoo.com; m_chellali@yahoo.com

Abstract

Let $G = (V(G), E(G))$ be a simple connected graph, and let k be a positive integer. A subset $D \subseteq V(G)$ is a connected k-dominating set of G, if its induced subgraph is connected and every vertex of $V(G) - D$ is adjacent to at least k vertices of D. The connected k-domination number $\gamma_k^c(G)$ is the minimum cardinality among the connected k-dominating sets of G. In this paper, we give some properties of connected graphs G with $\gamma_k^c(G) = n - 2$. Then we provide a complete characterization of connected cubic graphs G with $\gamma_2^c(G) = n - 2$ and connected 4-regular claw-free graphs with $\gamma_3^c(G) = n - 2$.

Keywords: k-domination, connected k-domination, regular graphs.

AMS Subject Classification: 05C69

1 Introduction

We consider finite, undirected and simple graph $G = (V(G), E(G))$ with vertex set $V(G)$ and edge set $E(G)$. The number of vertices $|V(G)|$ of a graph G is called the *order* and is denoted by $n = n(G)$. The *open neighborhood* $N(v) = N_G(v)$ of a vertex v consists of the vertices adjacent to v and $d_G(v) = |N(v)|$ is the *degree* of v. The *closed neighborhood* of a vertex v is defined by $N[v] = N_G[v] = N(v) \cup \{v\}$. If S is a subset of $V(G)$, then $N(S) = \cup_{x \in S} N(x)$, $N[S] = \cup_{x \in S} N[x]$ and the subgraph induced by S in G is denoted by $G[S]$. We may write $G - X$ instead of $G[V(G) - X]$ for any $X \subseteq V(G)$. By $\delta(G)$ and $\triangle(G)$, we denote the *minimum degree* and the *maximum degree* of the graph G, respectively.

We write K_n for the *complete graph* of order n, and $K_{s,t}$ for the *complete bipartite graph* with bipartition X, Y such that $|X| = s$ and $|Y| = t$. A *k-regular graph* or regular graph of degree k is a graph whose vertices have all the degree k. A 3-regular graph will be called a *cubic graph*. The *claw* is the star $K_{1,3}$. A graph G is *claw-free* if it does not have any

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

induced subgraph isomorphic to $K_{1,3}$. A *clique* of a graph G is a complete subgraph of G. A *cut vertex* (*bridge*, respectively) of a connected graph is a vertex (an edge, respectively) the removal of which would disconnect G.

In [2], Fink and Jacobson generalized the concept of dominating sets. Let k be a positive integer. A subset D of $V(G)$ is *k-dominating* if every vertex of $V(G) - D$ is adjacent to at least k vertices of D. Thus the 1-dominating set is a dominating set and so $\gamma_1(G) = \gamma(G)$. For more details on k-domination, see the recent survey of Chellali et al. [1].

A subset $D \subseteq V(G)$ is a *connected k-dominating set* of a connected graph G, if D is k- dominating set of G and if the induced subgraph $G[D]$ is connected. The *connected k-dominating number* $\gamma_k^c(G)$ is the minimum cardinality among the connected k-dominating sets of G. A connected k-dominating set of minimum cardinality of a connected graph G is called a $\gamma_k^c(G)$*-set*.

In [3], Volkmann characterized connected graphs G with $\gamma_k^c(G) = n$ for every integer $k \geq 2$. He also characterized all connected graphs G with $\gamma_k^c(G) = n - 1$ when $\delta(G) \geq k \geq 2$. The problem of characterizing all connected graphs G with $\gamma_k^c(G) = n - 2$ when $\delta(G) \geq k \geq 2$ remained open.

In this paper, we first give some properties of connected graphs G with $\gamma_k^c(G) = n - 2$. Then we provide a complete characterization of connected cubic graphs G such that $\gamma_2^c(G) = n - 2$ and connected 4-regular claw-free graphs with $\gamma_3^c(G) = n - 2$.

2 Properties of graphs G with $\gamma_k^c(G) = n - 2$

We mention that we often use in our proofs the fact that in every nontrivial connected graph G, there exist two vertices such that the removal of each one from G leaves the resulting graph connected.

Lemma 1 *Let $k \geq 2$ be an integer and G a connected graph such that $\gamma_k^c(G) = n - 2$. Then $\delta(G) \leq k + 1$.*

Proof. Let G be a connected graph such that $\gamma_k^c(G) = n - 2$ for some integer $k \geq 2$ and assume that $\delta(G) \geq k + 2$. Let D be a $\gamma_k^c(G)$-set and x any vertex of D such that $G[D - \{x\}]$ is connected. Clearly such a vertex exists and since $d_G(x) \geq k + 2$, x has at least k neighbors in $D - \{x\}$. Also, every vertex of $V - D$ still have at least k neighbors in $D - \{x\}$. It follows that $D - \{x\}$ is a connected k-dominating set of G of cardinality $(n - 3)$, a contradiction. □

Recall that an *independent set* of a graph G is a set S of vertices such that no edge of G has its two endvertices in S.

Lemma 2 *Let $k \geq 2$ be an integer and G a connected graph with $\delta(G) \geq k$ such that $\gamma_k^c(G) = n-2$. Then for every independent set S of cardinality at least three, $G-S$ is a disconnected graph.*

Proof. Let G be a connected graph with $\delta(G) \geq k \geq 2$ such that $\gamma_k^c(G) = n-2$. Let S be an independent set with $|S| \geq 3$. Since every vertex of S has at least k neighbors in $V-S$, the subgraph induced by $V-S$ is disconnected for otherwise $V-S$ would be a connected k-dominating set of G of size less than $n-2$. □

Lemma 3 *Let $k \geq 2$ be a positive integer and G a connected graph of order n with $\delta(G) = k+1$. If $\gamma_k^c(G) = n-2$, then G contains no bridge.*

Proof. Let G be a connected graph with $\delta(G) = k+1$ such that $\gamma_k^c(G) = n-2$ for some integer $k \geq 2$. Assume that G contains a bridge uv. Let G_u and G_v denote the two components resulting from the removal of uv, where $u \in V(G_u)$ and $v \in V(G_v)$. We further assume that uv is chosen so that, say G_u, has no bridge. Clearly since $\delta(G) = k+1$ and $k \geq 2$, each of G_u and G_v has order at least three. Now let w and w' be any two adjacent vertices in G_u different to u and let $v' \neq v$ be a vertex of G_v such that $G_v - \{v'\}$ is connected. If $G_u - \{w, w'\}$ is connected, then $V(G) - \{w, w', v'\}$ is a connected k-dominating set of G of cardinality $(n-3)$, a contradiction. Thus let us assume that $G_u - \{w, w'\}$ is not connected. Let G_u^i denote the ith component of $G_u - \{w, w'\}$. Note that each G_u^i has order at least two and since G_u is assumed without bridges, there are at least two edges between $\{w, w'\}$ and $V(G_u^i)$ for every i. We consider two cases.

Case 1. $i \geq 3$. Without loss of generality, let G_u^1 and G_u^2 be two components that do not contain u. Now let x be any vertex of G_u^1 such that $G_u^1 - \{x\}$ is connected. Likewise let y be a vertex of G_u^2 defined as x. Observe that $S = \{x, y, v'\}$ is an independent set and so by Lemma 2, $G-S$ is disconnected. It follows that x is the unique neighbor of w and w' in G_u^1. But since $\delta(G) = k+1 \geq 3$, G_u^1 has order at least three and so let $x' \neq x$ be a vertex of G_u^1 such that $G_u^1 - \{x'\}$ is connected. Likewise, let y' be a vertex of G_u^2 with $y' \neq y$ if y is the unique neighbor of w and w' in G_u^2 and $y' = y$ otherwise. Clearly now $\{x', y', v'\}$ is an independent set and $G - \{x', y', v'\}$ is connected which contradicts Lemma 2.

Case 2. $i = 2$. Thus u belongs to either G_u^1 or G_u^2, say G_u^1. Clearly one of w and w' must have at least one neighbor in each component. Assume that w is a such vertex. Let y be any vertex of G_u^2 such that $G_u^2 - \{y\}$ is connected. If w has another neighbor in G_u^2 different from y, then $V(G) - \{w', y, v'\}$ is a connected k-dominating set of G of cardinality $(n-3)$, a contradiction. Hence y is the unique neighbor of w in G_u^2. Since $\delta(G) = k+1 \geq 3$, G_u^2 has order at least three and so there is a vertex $y' \neq y$ in such that $G_u^2 - \{y'\}$ is connected. Therefore $V(G) - \{w', y', v'\}$ is a connected k-dominating set of G of cardinality $(n-3)$, a contradiction too. The proof of Lemma 3 is complete. □

Lemma 4 *Let $k \geq 2$ be an integer and G a connected graph with $\delta(G) = k+1$. If $\gamma_k^c(G) = n-2$, then for every pair of adjacent vertices x, y, $V(G) - \{x, y\}$ is a minimum connected k-dominating set of G.*

Proof. Let G be a connected graph with $\delta(G) = k+1$ such that $\gamma_k^c(G) = n-2$ for some integer $k \geq 2$. Let x, y be two adjacent vertices of G. Clearly since $\delta(G) = k+1$, $V(G) - \{x, y\}$ k-dominates G. Hence to show that $V(G) - \{x, y\}$ is a $\gamma_k^c(G)$-set, it suffices to show that $G' = G - \{x, y\}$ is connected. Thus assume to the contrary that G' is not connected and let C_i be the ith component of G'. Note that each C_i is nontrivial and there are at least two edges between $V(C_i)$ and $\{x, y\}$, otherwise we have a bridge which contradicts Lemma 3. Now let x_i be a vertex of C_i such that $C_i - \{x_i\}$ is connected. If the subgraph induced by the vertices of $(V(C_i) - \{x_i\}) \cup \{x, y\}$ is not connected, then x_i is the unique neighbor of x and y in C_i. But then there exists another vertex x_i' in C_i such that the subgraph G_i' induced by $(V(C_i) - \{x_i'\}) \cup \{x, y\}$ is connected. Thus, without loss of generality, we may assume that such a vertex x_i' exists in each component C_i. Now if G' has three components or more, then vertices x_i' form an independent set whose removal does not disconnect G, contradicting Lemma 2. Therefore, G' has exactly two components. Moreover, since xy is not a bridge, one of x and y, say x, has neighbors in both C_1 and C_2. If C_1 has order two, then obviously $k = 2$, $\delta(G) = 3$ and so $\{x, y\}$ 2-dominates $V(C_1)$, implying that $V(G) - (V(C_1) \cup \{x_2'\})$ is a connected 2-dominating set of G of size $n-3$, a contradiction. Hence we can assume that $|V(C_1)| \geq 3$. Now let z be a vertex of G_1' different from x and y. Recall that x_1' does not belong to G_1'. Now if $G_1' - \{z\}$ is connected, then $V(G)$ minus $\{z, x_1', x_2'\}$ is a connected k-dominating set of G, a contradiction. Hence $G_1' - \{z\}$ is not connected, and so z is a cut vertex of G_1'. Clearly each component of $G_1' - \{z\}$ is nontrivial, and one of them contains both x and y. In this case, let z' be any vertex in the component, say C^*, that does not contain x and y, for which $C^* - \{z'\}$ is connected. Note that if z' is the unique neighbor of z in C^*, then we can find another vertex z'' such that $C^* - \{z''\}$ is connected. So we may assume that z' is not the unique neighbor of z in C^*. Now clearly we have $V(G) - \{z', x_1', x_2'\}$ is a connected k-dominating set of G, a contradiction too. This achieves the proof of Lemma 4. □

Lemma 5 *Let $k \geq 2$ be an integer and G a connected graph of order n and minimum degree $\delta(G) = k+1$. If D is a $\gamma_k^c(G)$-set of size $n-2$ such that the subgraph induced by $N(V-D)$ is connected, then for every vertex $x \in D$, $N(x) \cap (V-D) \neq \emptyset$.*

Proof: Let G be a connected graph of order n and minimum degree $\delta(G) = k+1$ for some integer $k \geq 2$. Let D be a $\gamma_k^c(G)$-set with $|D| = n-2$, $A = N(V-D)$ and $B = D - A$. Assume that $G[A]$ is connected and $B \neq \emptyset$. If there is a vertex $x \in B$ such that $G[D - \{x\}]$ is connected, then $D - \{x\}$ is a connected k-dominating set of G smaller than D, a contradiction. Hence every vertex of B is a cut vertex in $G[D]$. It follows that some vertex of B, say y has no neighbor in A, for otherwise $G[D - \{y\}]$ is connected, contradicting the fact that y is a cut vertex in $G[D]$. Thus $N(y) \cap A = \emptyset$. Now it is clear that some component of $G[D - \{y\}]$ contains all A. Let C be a component of $G[D - \{y\}]$ that does not contain A. Thus every vertex of C belongs to B, that is, has no neighbor in $V - D$. Note that C is nontrivial. Let y' be a vertex of C such that $C - \{y'\}$ is connected. In the case that y' is the unique neighbor of y in C, then C has another vertex y'' such that $C - \{y''\}$ is connected. In this case we consider y'' instead of y'. Hence we may assume that y has an neighbor in C besides y'. It follows that $D - \{y'\}$ is a connected k-dominating set of G smaller than D, a contradiction. Therefore $B = \emptyset$ and we obtain the desired result. □

3 Cubic graphs with $\gamma_2^c(G) = n - 2$

In this section, we give a complete characterization of cubic graphs with $\gamma_k^c(G) = n - 2$, when $k = 2$. It well known that a cubic graph contains a bridge if and only if it contains a cut vertex.

Recall that in every nontrivial connected graph G, there exist two vertices such that the removal of each one from G leaves the resulting graph connected.

Theorem 6 *Let G be a connected cubic graph of order n. Then $\gamma_2^c(G) = n-2$ if and only if $G = K_4, K_{3,3}$ or G is the complement graph of a cycle C_6.*

Proof. It is easy to check that if $G = K_4, K_{3,3}$ or G is the complement graph of a cycle C_6, then $\gamma_2^c(G) = n - 2$.

Conversely, let G be a connected cubic graph such that $\gamma_2^c(G) = n-2$. We first assume that G has a vertex x whose neighborhood, say $\{x_1, x_2, x_3\}$, is an independent set. By Lemma 4, $V(G) - \{x, x_1\} = V'$ is a $\gamma_2^c(G)$-set. Let $A = \{x_2, x_3\}$ and $B = N(x_1) - \{x\} = \{y_1, y_2\}$. Clearly $A \cap B = \emptyset$ since $N(x)$ is independent. Let $V'' = V' - (A \cup B)$. We shall show that $V'' = \emptyset$. Suppose to the contrary that $V'' \neq \emptyset$ and let z be any vertex of V''. Then z is a cut vertex in $G[V']$, for otherwise $V(G) - \{x, x_1, z\}$ is a connected 2-dominating set of G of cardinality $(n-3)$, a contradiction. Note that each component of $G[V' - \{z\}]$ contains at least one vertex of $A \cup B$, for otherwise z is a cut vertex in G, implying that G has a bridge, a contradiction with Lemma 3. Consider the following two cases.

Case 1. $G[V' - \{z\}]$ contains three connected components C_1, C_2 and C_3. Clearly each C_i is nontrivial and z has exactly one neighbor in each C_i. Also, since each component contains at least one vertex of $A \cup B$, we may assume, without loss of generality, that $y_1 \in V(C_1)$, $x_2 \in V(C_2)$ and $x_1, y_2 \in V(C_3)$. Now, let $a \in V(C_1)$ such that $a = y_1$ if $y_1 z \notin E(G)$ and $a \in N(y_1) - \{z, x_1\}$ otherwise. Observe that $C_1 - \{a\}$ is connected, for otherwise a is a cut vertex in C_1 and so in G, a contradiction. Likewise let $b \in V(C_2)$ such that $b = x_2$ if $x_2 z \notin E(G)$ and $b = N(x_2) - \{z, x\}$ otherwise. Then $C_2 - \{b\}$ is also connected. In addition, let $c \in V(C_3)$ such that $cz \notin E(G)$ and $C_3 - \{c\}$ is connected. Now it is evident that $\{a, b, c\}$ is an independent set whose removal does not disconnect G, contradicting Lemma 2.

Case 2. $G[V' - \{z\}]$ contains two connected components C_1 and C_2. Clearly each C_i is nontrivial. Also, let us assume, without loss of generality, that $|N(z) \cap C_1| = 1$. Let w be a vertex of C_1 such that $C_1 - \{w\}$ is connected. Note that C_1 contains at least two such vertices w, one of them is not adjacent to z. We now examine the following situations depending on whether C_1 contains one, two or three vertices of $A \cup B$.

- $|V(C_1) \cap (A \cup B)| = 1$. W.l.o.g, let $\{x_2\} \subset V(C_1)$. Here we only suppose for w to be different from x_2. Clearly, then $V(G) - \{z, w, x_1\}$ is a connected 2-dominating set of G of cardinality $(n-3)$.

- $|V(C_1) \cap A| = 1$ and $|V(C_1) \cap B| = 1$. W.l.o.g, let $\{x_2, y_2\} \subset V(C_1)$. Suppose that

$wz \notin E(G)$ and let $w' \in V(C_2)$ such that $C_2 - \{w'\}$ is connected. Then $V(G) - \{w, w', z\}$ is a connected 2-dominating set of G of cardinality $(n-3)$.

- $A \subset V(C_1)$ and $B \subset V(C_2)$. Suppose that $wz \notin E(G)$ and let $w' \in V(C_2)$ such that $C_2 - \{w'\}$ is connected. Clearly $V(G) - \{w, w', x\}$ is a connected 2-dominating set of G of cardinality $(n-3)$.

- $|V(C_1) \cap (A \cup B)| = 3$. W.l.o.g, suppose that $\{x_3, y_1, y_2\} \subset V(C_1)$ and let $x_2' \in N(x_2) \cap V(C_2)$. Then according to the Lemma 4, $V(G) - \{x_2, x_2'\}$ is a $\gamma_c^2(G)$- set. Also observe that z is adjacent to at most one of x_2 and x_2', for otherwise x_2' would be a cut vertex in C_2 and so in G, which is excluded. Now it is evident that $V(G) - \{x_2, x_2', x_1\}$ is a connected 2-dominating set of G of cardinality $(n-3)$.

Clearly, each of the previous situations leads to a contradiction. Therefore we conclude that $V'' = \emptyset$, implying that G is isomorphic to $K_{3,3}$.

From now on we can assume that the neighborhood of every vertex of G contains at least two adjacent vertices, and so G is a claw free graph. Let x be a vertex of G with $N_G(x) = \{x_1, x_2, x_3\}$ and $x_1x_2 \in E(G)$. By Lemma 4, $V(G) - \{x, x_3\} = V'$ is a $\gamma_2^c(G)$-set. Let $V'' = V' - N_G(\{x, x_3\})$. We shall show that $V'' = \emptyset$. Assume to the contrary that $V'' \neq \emptyset$ and let z be any vertex of V''. Then z is a cut vertex in $G[V']$, for otherwise $V(G) - \{x, x_3, z\}$ is a connected 2-dominating set of G of cardinality $(n-3)$, a contradiction. Note that since $V'' \neq \emptyset$, we have $N_G(\{x, x_3\}) \neq \{x_1, x_2\}$. Also, all neighbors of x_1 and x_3 in V' do not belong to a same component of $G[V' - \{z\}]$, for otherwise z would be a cut vertex of G. On the other hand, since G is claw free, $G[V' - \{z\}]$ contains exactly two connected components C_1 and C_2. W.l.o.g, let $|N(z) \cap C_1| = 1$. We consider the following two cases:

-$N_G(x_3) \cap \{x_1, x_2\} = \emptyset$. Let $N_G(x_3) = \{x, y_1, y_2\}$. Then $y_1y_2 \in E(G)$ since G is a claw free. Suppose that $\{x_1, x_2\} \subset V(C_1)$, and let w be a vertex of C_1 such that $C_1 - \{w\}$ is connected and $wz \notin E(G)$. Let also $w' \in V(C_2)$ such that $C_2 - \{w'\}$ is connected. Then $V(G) - \{w, w', x\}$ is a connected 2-dominating set of G of cardinality $(n-3)$, a contradiction.

- $N_G(x_3) \cap \{x_1, x_2\} \neq \emptyset$. Since $V'' \neq \emptyset$, we have that $|N(x_3) \cap \{x_1, x_2\}| = 1$. Let $x_3x_2 \in E(G)$ and $N_G(x_3) = \{x, x_2, y\}$. Assume that y belongs to the component C_i, where $i = 1$ or 2. It follows that $\{x_1, x_2\} \subset V(C_{3-i})$. Note that if $y \in V(C_1)$, then $yz \notin E(G)$, for otherwise y would be a cut vertex in C_1 and so in G. Now let w' be a vertex in the component that contains y such that $w'z \in E(G)$ and $w' \neq y$. By Lemma 4, $V(G) - \{w', z\}$ is a $\gamma_2^c(G)$-set and hence $V(G) - \{w', z, x\}$ is a connected 2-dominating set of G of cardinality $(n-3)$, a contradiction.

According to the previous situations, we conclude that $V'' = \emptyset$. Since G has an even order $n = 4$ or $n = 6$, we deduce that G is isomorphic to K_4 or the complement of a cycle C_6, respectively. □

4 4-regular claw-free graphs with $\gamma_3^c(G) = n - 2$

In this section we consider connected 4-regular claw-free graphs, where we give a characterization of such graphs G with $\gamma_3^c(G) = n - 2$. Note that by Lemma 1, there are no connected 4-regular graphs such that $\gamma_2^c(G) = n - 2$.

Theorem 7 *Let G be a connected 4-regular claw-free graph of order n. Then $\gamma_3^c(G) = n-2$ if and only if G is isomorphic to K_5 or $K_{2,2,2}$.*

Proof. It is a simple matter to check that if $G \in \{K_5, K_{2,2,2}\}$, then $\gamma_3^c(G) = n - 2$. To prove the necessity, let G be a connected 4-regular claw-free graph such that $\gamma_3^c(G) = n-2$. We first assume that G contains a vertex whose open neighborhood induces a disconnected subgraph. Let x be such a vertex with $N_G(x) = \{x_1, x_2, x_3, x_4\}$. Then by Lemma 4, $V' = V(G) - \{x, x_4\}$ is a $\gamma_3^c(G)$-set. Let $V'' = V' - N_G(\{x, x_4\})$. We shall show that $V'' = \emptyset$. Suppose to the contrary that $V'' \neq \emptyset$ and let z be any vertex of V''. Clearly z is a cut vertex in $G[V']$, for otherwise $V(G) - \{x, x_4, z\}$ is a connected 3-dominating set of G of cardinality $(n-3)$, a contradiction. Also, since G is a claw-free graph, the removal of z in $G[V']$ provides two nontrivial connected components, say C_1 and C_2. Observe that the subgraph induced by $N_G(\{x, x_4\})$ is not connected for otherwise by Lemma 5, z would be adjacent to x or x_4, which is impossible. If all vertices of $N_G(\{x, x_4\})$ belong to a same component in $G[V'-\{z\}]$, say C_1, then let t be a vertex of C_2 such that $C_2-\{t\}$ is connected. Note that if z has a unique neighbor in C_2, then t is chosen so that $tz \notin E(G)$. Now it is clear that $V(G) - \{x, x_4, t\}$ is a connected 3-dominating set of G of cardinality $(n-3)$, a contradiction. Hence each of C_1 and C_2 contains at least one vertex of $N_G(\{x, x_4\})$. Since is a claw-free graph, we have two cases depending on whether $G[N(x)] = K_3 \cup K_1$ or $K_2 \cup K_2$.

Case 1. $G[N(x)] = K_3 \cup K_1$. There are two situations depending on whether x_4 belongs to K_3 or not. So let us assume that $\{x_1, x_2, x_3\}$ induces a K_3. In this case, let $N_G(x_4) = \{x, y_1, y_2, y_3\}$. Then $\{y_1, y_2, y_3\}$ induces a K_3, since G is claw-free. Without loss of generality, we can assume that $\{x_1, x_2, x_3\} \subset V(C_1)$ and hence $\{y_1, y_2, y_3\} \subset V(C_2)$. Now let $w \in V(C_1)$ and $w' \in V(C_2)$ such that $C_1 - \{w\}$ and $C_2 - \{w'\}$ are connected. Note that if z has a unique neighbor in C_1, then w is chosen so that $wz \notin E(G)$. Likewise for w' in C_2. Then $V(G) \setminus \{w, w', x\}$ is a connected 3-dominating set of G of size $(n-3)$, a contradiction.

Now suppose that $K_3 = \{x_2, x_3, x_4\}$ and let y_1 be the fourth neighbor of x_4. Since $x_2x_3 \in E(G)$, we may assume that $\{x_2, x_3\} \subset V(C_1)$ and so C_2 contains at least one of x_1 and y_1.

- $\{x_1, y_1\} \subset V(C_2)$. Suppose that z has at least two neighbors in C_1. Then $zx_1 \notin E(G)$ for otherwise the closed neighborhood of x_1 induces a claw. Likewise $zy_1 \notin E(G)$. In this case let w' be any neighbor of z in C_2. By Lemma 4, $V(G) - \{z, w'\}$ is a $\gamma_3^c(G)$-set and so $V(G)-\{z, w', x\}$ is a connected 3-dominating set of G of size less than $n-2$, a contradiction. Now suppose that z has exactly one neighbor in C_1. Clearly the neighborhood of z in C_2

induces a clique K_3. Let w' be any vertex of C_2 adjacent to z and let w be a vertex of C_1 such that $wz \notin E(G)$ and $C_1 - \{w\}$ is connected. Using Lemma 4, it is clear that $V(G) - \{z, w', w\}$ is a connected 3-dominating set of G of size less than $n-2$, a contradiction too.

- Now suppose $y_1 \in V(C_2)$ and $x_1 \in V(C_1)$. Note that if z has a unique neighbor in C_2, then such a vertex is different from y_1, for otherwise the closed neighborhood of y_1 induces a claw. So we can assume that z has a neighbor in C_2, say w such that $w \neq y_1$. By Lemma 4, $V(G) - \{z, w\}$ is a $\gamma_3^c(G)$-set and hence $V(G) - \{z, w, x\}$ is a connected 3-dominating set of G of size less than $n-2$, a contradiction. The remaining case $y_1 \in V(C_1)$ and $x_1 \in V(C_2)$ can be seen by using a similar argument to that used for the previous situation.

Case 2. $G[N(x)] = K_2 \cup K_2$. Without loss of generality, let $x_1x_2 \in E(G)$ and $x_3x_4 \in E(G)$. We also let y_1, y_2 be the third and fourth neighbor of x_4. It follows that $y_1y_2 \in E(G)$ for otherwise $\{x, x_4, y_1, y_2\}$ induces a claw. Also, without loss of generality, we can assume that $\{x_1, x_2\} \subset V(C_1)$. We consider the following situations.

- $\{y_1, y_2\} \subset V(C_2)$. Clearly x_3 belongs to either C_1 or C_2 and so let us choose the component that contains exactly two vertices of $N_G(\{x, x_4\})$. Note that such a component contains at least four vertices. Let now w be a neighbor of z in the selected component. By Lemma 4, $V(G) - \{z, w\} = S$ is a $\gamma_3^c(G)$-set. Now if C_1 is the selected component, then x_3 is in C_2 and so $S - \{x_4\}$ is a connected 3-dominating set of G of size $(n-3)$. If C_2 is the selected component, then x_3 is in C_1 and so $S - \{x\}$ is a connected 3-dominating set of G of size $(n-3)$. In each case we have a contradiction.

- $\{y_1, y_2\} \subset V(C_1)$. It follows that x_3 belongs to C_2. Note that if z has a unique neighbor in C_2, then such a vertex is different from x_3 for otherwise the closed neighborhood of x_3 induces a claw. So we can assume that z has a neighbor in C_2, say w, such that $w \neq x_3$. By Lemma 4, $V(G) - \{z, w\}$ is a $\gamma_3^c(G)$- set and therefore $V(G) \setminus \{z, w, x_4\}$ is a connected 3-dominating set of G of size $(n-3)$, a contradiction.

According to the previous cases we conclude that $V'' = \emptyset$. Using the fact that $G[N(x)]$ is not connected and up to isomorphism, one can see that the only 4-regular claw-free graph is the graph with 8 vertices $x, x_1, x_2, x_3, x_4, y_1, y_2, y_3$ such that each of $\{x_2, x_3, x_4\}$ and $\{y_1, y_2, y_3\}$ induces a clique K_3, x_1y_1, x_2y_2, x_3y_3 and $x_4y_i \in E(G)$ for every i. But then $V(G) \setminus \{y_1, y_2, x\}$ is a connected 3-dominating set of G of size $(n-3)$, a contradiction.

From now on, we can assume that the subgraph induced by the neighborhood of every vertex is connected. Let x be any vertex of G with $N_G(x) = \{x_1, x_2, x_3, x_4\}$. Recall that by Lemma 4, $V' = V(G) - \{x, x_4\}$ is a $\gamma_3^c(G)$-set. Clearly the set $V'' = V' - N_G(\{x, x_4\})$ is empty for otherwise every vertex of V'' will be a cut vertex in $G[V']$, contradicting the fact that the open neighborhood of every vertex induces a connected subgraph. Also $G[N(x)]$ contains a path P_4 not necessarily induced, say x_1-x_2-x_3-x_4. Let $\{y_1, y_2\} = N(x_4) - (\{x, x_3\})$. Suppose that $y_1, y_2 \notin N(x)$. Since G is claw-free, $y_1y_2 \in E(G)$. On the other hand, the fact $G[N(x_4)]$ is connected implies that one of y_1 and y_2 is adjacent to x_3, say $y_1x_3 \in E(G)$. It follows that x_1y_1, x_1y_2 and $x_2y_2 \in E(G)$. But then $\{x_1, x_2, x_3, x_4\}$ is a connected 3-dominating set of G of size $(n-3)$, a contradiction. Thus at least one of y_1 and y_2 belongs to $N(x)$, say y_2. If $y_2 = x_2$, then it is easy to see that G is not 4-regular. Thus $y_2 = x_1$, and so $N(y_1) = N(x)$. Therefore $G = K_{2,2,2}$. Finally if $y_1, y_2 \in N(x)$, then

$G = K_5$. □

References

[1] M. Chellali, O. Favaron, A. Hansberg and L. Volkmann, k-Domination and k-independence in graphs: a survey. *Graphs and Combinatorics*, 28 (2012) 1–55.

[2] J.F. Fink and M.S. Jacobson, n-domination in graphs. *Graph Theory with Applications to Algorithms and Computer.* John Wiley and sons, New York (1985) 283-300.

[3] L. Volkmann, Connected p-domination in graphs. Util. Math. 79, 81-90 (2009).

Ontologies et leurs applications

Mapping d'Ontologies dans un Environnement Distribué

Fouzia Boudries1, A. Kamel Tari1, Carlos Juiz2
1Faculté des Sciences et des Sciences de l'Ingénieur Département d'Informatique,
Laboratoire des Mathématiques Appliqués
Université de Bejaia
boudriesfouzia@yahoo.fr, atari@mail.cerist.dz
Université des iles Baléares Espagne
cjuizgarcia@gmail.com

Résumé. Le mapping d'ontologies apparait dans différents domaines d'applications et sa résolution est très complexe dû aux différentes manières de représenter les connaissances. Notre approche compare les entités les plus significatives des ontologies à savoir les concepts et considère l'aspect lexical, structurel et sémantique de ces derniers pour le calcul de similarité. Elle se base sur une ressource externe « WordNet» afin d'enrichir les concepts avec des informations supplémentaires. Par la suite, l'algorithme compare ces entités pour décider s'il y a des similarités contrairement aux techniques traditionnelles qui comparent la plupart des entités des ontologies une à une.

Nous avons implémenté notre approche sous Java et en utilisant les deux api JENA etJWI. Pour l'évaluation, nous avons utilisé le benchmark « Bibliographie» proposé par OAEI en 2010 qui est composé d'un ensemble de test. Le mapping dans chaque test se fait entre une ontologie de référence et une ontologie alternative de même domaine. Nous avons comparé nos résultats avec d'autres algorithmes qui ont été évalué avec le même jeu de données et la comparaison est faite par les deux mesures « la précision » et « le rappe l ».

Mots clés : Intégration d'information, Web sémantique, Ontologies, Mapping d'ontologies.

1 Introduction

Le mapping d'ontologies est un nouveau paradigme dans le web sémantique où les ontologies sont utilisées pour la représentation de l'information en utilisant un langage de description tel que OWL (Ontology Web Language. D'après [1]Une ontologie est une spécification formelle d'une conceptualisation partagée. Les différentes manières de représenter un domaine ont encouragé la multiplicité d'ontologies ce qui a posé le problème de leurs hétérogénéités [2]. Afin de réduire ces hétérogénéités et de permettre leurs interopérabilité, le mapping d'ontologies est une solution permettant de découvrir les relations sémantiques implicites entre les entités

de deux ontologies et de les combiner afin d'obtenir de nouvelles connaissances et ainsi assurer la coopération, la communication et le partaged'informations.

Le mapping d'ontologies apparait dans plusieurs domaines d'applications tels que l'intégrationd'information, la composition des services web etc. Il est considéré comme étant le point clé pour la recherche d'interopérabilité entre agents et services basés sur les ontologies.

Pour la recherche des mapping, différentes techniques ont été introduites dans la littérature. Une classification de ces techniques est proposée par Jérôme Euzenat[18] où la première classe est les techniques terminologiques qui se basent sur la comparaison des labels des entités à comparer, les techniques structurelles considérant la relation d'hiérarchie entre les concepts et la dernière classe est les techniques qui se basent sur les instances.

Dans cet article, nous présentons un algorithme de mapping distribué qui compare les entités les plus significatives à savoir les concepts de deux ontologies. La comparaison des concepts permet de réduire le nombre d'entités comparées dans l'objectif de gagner de l'efficacité. Pour augmenter la probabilité de trouver des correspondances entre les concepts, nous avons enrichis ces derniers avec des termes supplémentaires comme les synonymes et les hyperonymes.

Notre objectif est d'appliqué cet algorithme sur les environnements distribués qui s'exécute sur des architectures hétérogènes et sur la définition de service. Dans ces environnements, l'interaction et la communication requirent un temps de réponse minimum et un degré élevé de qualité et les techniques traditionnelles de mapping ne sont pas applicables dans cette situation.

2 Etat de l'art

Différents algorithmes et approches ont été développés pour le mapping d'ontologies où chacun traite la découverte différemment [4]. Cependant, la plupart de ces approches appliquent une recherche exhaustive et considèrent toutes les caractéristiques possibles des différentes entités des ontologies; ce qui engendre un temps d'exécution considérable et perdent ainsi en efficacité dans les environnements distribués.

Parmi les approches proposées pour régler ce problème, nous avons :

- **QOM** : approche proposée par Marc Ehrig et al en 2004[5].C'estun système semi automatique pour la génération des mapping entre deux ontologies décrites en OWL. Il était conçu pour achever une interopérabilité entre agents et services et considère le nombre de comparaisons à effectuer comme facteur influent sur la complexité du temps d'exécution. QOM

compare toutes les entités des ontologies (concepts, instances, propriétés) et pour gagner de l'efficacité, il réduit le nombre de comparaisons par l'application des heuristiques. Parmi les heuristiques : la sélection aléatoire d'un sous ensemble de mapping parmi un ensemble de candidats, la comparaison des entités dont leurs labels ou les labels de leurs voisinages est très proche et enfin la comparaison des entités de niveau haut de la hiérarchie pour décider d'explorer les niveaux bas. L'utilisation des heuristiques à un apport négatif sur la qualité des résultats car tout se fait d'une manière aléatoire. En conséquence, QOM permet un gain du temps dans la découverte mais il perde de la qualité ce qui n'est pas souhaitable car pour qu'une communication soit de qualité, il faut que le résultat soit de qualité.

- **QOMFDE** : approche proposée par Isaac et al en 2008[3]. Il considère en entrée deux ontologies OWL. Pour chacune de ces deux ontologies, il crée une nouvelle structure qui contient que les éléments les plus significatifs.Le choix de ces derniers se base sur le nombre des propriétés et de relations d'hiérarchie de ce concept dans l'ontologie. Une fois la structure est définie, vient l'enrichissement de chaque élément de cette structure avec un ensemble de termes comme les synonymes, les hyperonymes et instances et ceci se fait à partir de WordNet. Puis, le mapping se fait entre les deux structures en appliquant différentes mesures de similarité. Cet algorithme règle deux problèmes majeurs dans les environnements distribués et mobiles : le premier problème est la multiplicité de la représentation d'un domaine par la création d'une structure qui synthétise l'objectif et le degré de granularité des ontologies à comparer, le deuxième problème est la quantité de données à comparer par la réduction du nombre de comparaisons à effectuer. Cette solution change le processus de développement d'ontologies car la création de la nouvelle structure qui se fait pondant la conception consomme du temps et de l'espace mémoire.

3 Approche proposée

Nous avons proposée une approche pour le mapping qui s'inspire des deux algorithmes QOM et QOMFD. Elle admet en entrée deux ontologies OWL et afin de réduire le nombre de comparaisons et gagner en efficacité du temps d'exécution, nous avons choisi de comparer seulement les entités les plus significatives des ontologies comparées à savoir les concepts et nous avons considéré différents aspects de ces derniers à savoir l'aspect lexical, structurel et sémantique pour la découverte des similarités.

Notre approche s'appuie sur une ressource externe WordNet qui est un dictionnaire de langue Anglaise dont l'unité de base est le concept [6] où son sens de ce dernier est défini par l'ensemble de ses synonymes (synset) et les différentes relations sémantiques entre les synsets. L'utilisation de cette ressource permet d'enrichir

chaque concept par ses synonymes,de récupérer ses superclasses et ainsi d'augmenter la probabilité de coïncidence entre les concepts dans le processus de mapping. WordNet a un apport considérable pour le mapping en raison de son utilisation par un nombre important d'algorithmes de mapping tel que [3, 5,9]. WordNet a permet de déduire l'équivalence entre les concepts dont les labels (nom attribué à un concept) ne sont pas syntaxiquement similaires grâce à relation de synonymie (synsets) et d'appliquer les deux méthodes choisies (Jaccard similarité et Upward cotopic similarity) qui s'appliquent sur la même hiérarchie.

Une fois les concepts des deux ontologies sont extraits avec l'api Jena, notre approche multi agents s'exécute en trois étapes pour établir le mapping à savoir :

Etape 1 (Normalisation) : C'est l'étape de normalisation linguistique où les méthodesde NLP sont appliquées pour ramener chaque terme à une forme normalisée afin de réduire l'hétérogénéité lexicale et de faciliter l'étape de découverte de similarités basée sur les labels. Les étapes de la normalisation appliquées sont la suppression des numéros et l'élimination des mots vides.

Etape 2 (Mapping) **:** Une fois la phase de normalisation est achevée, nous procédons à la création de la matrice de concepts Mat (n, m) où n et m sont le nombre de concepts des ontologies O et O' respectivement, nous diffusons cette matrice aux agents en charge du mapping. Il existe trois types d'agent à avoir :

- **Agent 1 :** Il est en charge de découvrir les similarités lexicales entre concepts. La méthode appliquée est N-gram similarité. N-gram similarité [10] est une mesure normalisée et fait apparaitre clairement la similarité lexicale entre les termes. Dans le cas où les deux termes sont identiques ou l'un est une sous chaine del'autre elle renvoie de valeur très satisfaisante ce qui signifie que les deux termes sont équivalents. Généralement, dans les langues, comme la langue anglaise, l'ajout d'un mot à un autre réduit sa largeur. Par exemple,'reviewedarticle' et 'article' sont des concepts similaires et l'application de cette méthode renvoie des très bons résultats.

- **Agent 2 :** Il détecte la similarité entre concepts en exploitant la relationd'hiérarchie is-a entre concepts dans WordNet. La méthode appliquée est Upward cotopic similarity (UCS) qui une mesure normalisée.Cette méthode s'appuie sur la similarité de Jaccard et les parents des concepts dans la même hiérarchie [11]. Pour l'adapter au mapping d'ontologies, nous avons utilisé la ressource WordNet pour avoir les classes générales de concepts à comparer. La similarité avec cette méthode est donnée par le contenu informationnel et pas en termes d'arcs séparant les deux concepts. La similarité entre deux concepts c et c'avec UCS est donnée par la relation suivante :

$$\sigma(c, c') = \frac{UC(c,H) \cap UC(c',H)}{UC(c,H) \cup UC(c',H)} \qquad (1)$$

Où UC(c, H) est l'ensemble des superclasses de concepts c dans H qu'est WordNet. Le choix de cette méthode est motivé par le faite qu'elle considère

et compare tous les parents des concepts jusqu'à la racine et que la relation d'hyperonymie est une relation sémantique transitive très utilisée pour définir le sens des concepts et ne pas seulement les parents directs comme le font les autres méthodes. Donc, la similarité retournée par cette méthode est une similarité sémantique. Une étude comparative entre les méthodes basées sur la hiérarchie des concepts pour tirer la similarité a montrée qu'UCS donne des très bons résultats par rapport à d'autres [11].

- **Agent 3 :** La découverte de similarité au niveau de cet agent se base sur les synsets deWordNet. Chaque concept est renforcé avec le synset (ensemble de ses synonymes) qui lui correspond dans le dictionnaire puis la méthode de Jaccard est appliquée pour calculer la similarité entre ces derniers. La similarité de Jaccard est donnée par la formule suivante où elle renvoie la similarité entre deux concepts c et c' :

$$\sigma(\mathrm{c}, \mathrm{c}') = \frac{|\mathrm{synset}(\mathrm{c}) \cap \mathrm{synset}(c')|}{|\mathrm{synset}(\mathrm{c}) \cup \mathrm{synset}(c')|} \qquad (2)$$

Nous obtenons avec cette mesure un 1.0 lorsque les deux concepts à comparer appartiennent au même synset et 0 autrement. Par exemple, la similarité entre " person " et "individual" ou entre " water " et "H2O" est 1.0.

Etape 3(Extraction des mappings pertinents) : Dans cette dernière phase, les résultats obtenus par les différents agents sont récupérés et analysés pour établir les correspondances entre les concepts de l'ontologie source et l'ontologie cible. Pour extraire ces mapping, la technique utilisée est le seuil. D'abord, nous commençons à analyser les résultats obtenus par la méthode n-gram(agent 1) le fait que deux concepts sont similaires s'ils sont écrits de la même manière. Dans ce cas, le seuil choisi est "0.5" car dans le cas où un conceptest une sous chaine de l'autre la valeur minimal est 0.5. Dans le cas où la similarité entre deux concepts est égale ou supérieur à 0.5, directement une correspondance est établit et nous passons un autre couple de concepts. Dans le cas contraire où nous n'avons pas trouvé une correspondance entre les deux concepts, nous passons à l'analyse les résultats produit par l'agent (3). Si la valeur renvoyée est égale à 1 une correspondance est établit directement entre les deux concepts sinon nous procédons à l'analyse des résultats de l'agent 2. Dans ce cas, si la valeur dépasse un seuil de 0.5, nous allons établir la correspondance sinon ces deux concepts ne sont pas équivalents.

3 Architecture de l'approche

La Figure 1 illustre l'architecture de notre approche. Elle consiste en deux ontologies en entrée pour lesquelles nous voulons calculer leur mapping et les différents agents

chargés de calcul de similarité entre les concepts des deux ontologies et leurs interaction avec le dictionnaire WordNet ainsi que les phases d'extraction et d'écriture des correspondances trouvées entre concepts.

La première étape exécutée dans l'architecture est l'extraction de concepts des deux ontologies en entrée O et O'. L'étape suivante est la normalisation qui permet de mettre tous les concepts sous une représentation uniforme. Elle consiste à supprimer les mots vides tels que les numéros, les conjonctions etc. et de mettre tous les concepts en minuscule. Une fois l'étape de normalisation est achevée, nous diffusons la matrice des concepts aux différents agents chargés de mapping. Nous avons trois types agent : agent1 calcule la similarité lexicale entre les concepts par l'application de la méthode N-gram similarité. L'agent2 est l'agent3 sont en interaction avec le dictionnaire WordNet afin d'enrichir les concepts avec des informations supplémentaires. L'agent2 applique la méthode "Upward cotopic similarity" qui est une méthode qui considère tous les super-concepts des concepts dans le dictionnaire WordNet. L'agent2 enrichi chaque concept avec ses synonymes, puis applique la méthode Jaccard pour estimer la similarité.

La dernière étape est l'extraction des mapping pertinents. La méthode adoptée est le seuil car toutes les méthodes appliquées aux niveaux des agents sont des méthodes normalisées qui renvoient des valeurs appartenant à l'intervalle [0,1]. Donc pour extraire les mapping pertinents, nous comparons les valeurs renvoyées par les différentes méthodes avec un seuil que nous définissons.

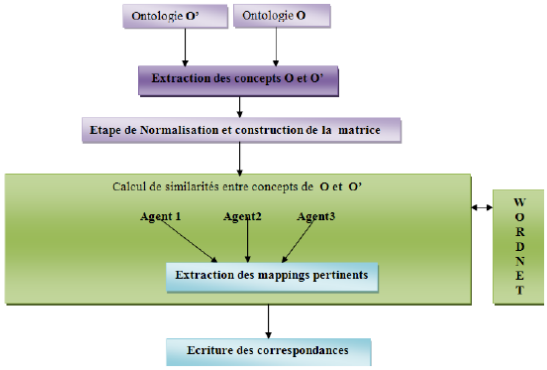


Fig.1. Architecture de l'approche proposée

4 Algorithme

L'algorithme implémenté considère en entrée deux ontologies O et O' de OWL,leurs URLset le dictionnaire WordNet.

Avec l'URL de chaque ontologie, nous créons un modèle avec l'Api Jena et à travers ce modèle, nous pouvons extraire les concepts de chacune des ces ontologies. Une fois que tous les concepts des ontologies sont extraits, nous allons les sauvegarder dans des vecteurs vect1 et vect2 pour les ontologies O etO' respectivement. Par la suite, pour chaque élément des deux vecteurs, nous appliquons une fonction de normalisation " Normaliser () " qui prend comme paramètre un string qui représente le concept. Une fois la normalisation est terminée, nous exécutons en parallèleles trois agents (agent1, agent2, agent3) qui se chargent de mapping et qui admettent les deux vecteurs vect1, vect2 comme données en entrée.

- L'agent1 applique la méthode lexicale Ngram () similarité sur le résultat du produit cartésien des deux vecteurs vect1 et vect2.
- L'agent2 applique la méthode simcotopic()toujours sur le produit cartésien des deux vecteurs vect1 et vect2 après l'enrichissement de chaque concepts avec ses super-concepts dans WordNet en appelant la fonction recuperer().
- L'agent3 applique la méthode Jaccard () toujours sur le produit cartésien des deux vecteurs vect1 et vect2 après l'enrichissement de chaque concepts avec son synset dans WordNet en appelant toujours la fonction recuperer().

A chaque fois un agent termine, nous récupérons ses résultats. Une fois tous les résultats des différents agents sont récupérés, nous les examinons pour établir les correspondances entre les différents concepts.Nous commençons par l'analyse des résultats obtenus par l'agent1, si une correspondance est déduite, nous passons à un autre couple de concepts sinon, nous vérifions pour l'agent2 pour vérifier est ce que nous pouvons établir une correspondance. Dans le cas échéant, nous passons aux résultats de l'agent3. Nous allons procéder de cette façon pour tous les couples de concepts.

Le code de l'algorithme proposé est le suivant :

Les entrées de l'algorithme :

```
O, O' : deux ontologies
URL1= chemin vers O
URL2=chemin vers O'
WordNet: Dictionnaire ;
Le traitement :
i=0 ;
Tant que (Trouver (classe, O)=vrai) faire
      Vect1 [i]=classe de O;
      i++ ;
fin Tant que ;
i=0 ;
Tant que (Trouver (classe, O')=vrai) faire
      Vect2 [i]=classe de O;
      i++ ;
fin Tant que ;
```

```
Pour i allant de 0 à taille de vect1 faire
      Normaliser (Vect1 [i]) ;
fin Pour ;
Pour i allant de 0 à taille de vect2 faire
      Normaliser (Vect2 [i]) ;
fin Pour ;
Créer la matrice des concepts Mat (n, m)
Diffusion de la matrice Mat(n,m) aux différents agents.
Agent1 :
Pour i allant de 0 à taille de vect1 faire
      Pour i allant de 0 à taille de vect2 faire
            Sim-Ngram[i][j]=Ngram(vect1[i],vect[j]) ;
      fin Pour ;
finPour ;


Agent2 :
Pour i allant de 0 à taille de vect1 faire
      Pour i allant de 0 à taille de vect2 faire
      SimJaccard[i][j]=Jaccard((recuperer(synset(vect1[i]
      ),Wordnet),recuperer(synset(vect2[i]),WordNet )) ;
fin Pour ;
fin Pour ;
Agent3 :
Pour i allant de 0 à taille de vect1 faire
Pour i allant de 0 à taille de vect2 faire
Simcotopic[i][j] = simcotopic((recuperer(super-
classes(vect1[i]);Wordnet), recuperer(super-
classes(vect2[i]);Wordnet)) ;
fin Pour ;
fin Pour ;
Récupérations des résultats des différents agents
Pour i allant de 0 à taille de vect1  faire
      Pour i allant de 0 à taille de vect2  faire
            Si (Sim- Ngram[i][j] >= 0,5) Alors
            Ecriture la correspondance entre Vect1 [i] et
      Vect2 [j]
            Sinon Si (Sim-Jaccard[i][j]=1.0)alors
                  Ecriture la correspondance entre Vect1
            [i] et Vect2 [j]
            Sinon Si (Sim - cotopic[i][j] >= 0,5) alors
            Ecriture la correspondance entre Vect1[i] et
Vect2[j] ;
            Finsi ;
Finsi ;
                  Finsi ;
fin Pour ;
fin Pour ;
```

5 Evaluation de l'approche

L'algorithme proposé couvre les étapes du processus de mapping. Ce qui revient à dire que sa complexité est donnée par le calcul de coût de chacune de ces étapes.

Notons par n le nombre des entités à comparer. Dans notre approche, nous comparons seulement les concepts (classes en OWL) des ontologies au lieu de comparer toutes les entités. Cela revient à dire que la valeur de n est petite par rapport au n considéré dans d'autres approches où ils comparent toutes les entités (concepts, propriétés, instances).

L'étape de choix des caractéristiques ne nécessite pas des transformations, ce qui engendre un coût nul. L'extraction des concepts à partir des deux ontologies requiert une complexité O(n) le fait que l'extraction d'une entité ou d'un ensemble d'entités est indépendant de la taille de l'ontologie. Pour l'étape de calcul de similarité, puisque les différents agents chargés de mapping s'exécutent en parallèle et appliquent une méthode sur le résultat du produit cartésien des concepts des deux ontologies, le coût de cette étape est le temps consommé par l'un des agents. Par exemple, l'agent 1 requiert un coût de O (n2) car il compare chaque concept de l'ontologie source avec tous les concepts de l'ontologie cible (même traitement pour les deux autres agents). L'étape d'agrégation des résultats ne s'applique pas dans notre cas et une seule itération est suffisante pour tirer les mapping, pour cela le coût de ces étapes est nul. Pour l'étape d'interprétation, son coût est n *(3*O(1)) qui est égale à O(n) car au pire des cas, nous devons parcourir tous les résultats des agents. En conséquence, la complexité de l'algorithme est : Complexité = 0 + O(n) + O (n2) + O(n) = O(n2).

La complexité obtenue par notre algorithme est une complexité quadratique c-à-d polynomiale et pour n pas top grand, les algorithmes polynomiaux sont encore efficaces.

L'évaluation qualitative et quantitative des systèmes de mapping peut se faire en calculant différentes mesures de performances. Les plus utilisées sont la précision et le rappel.

Pour l'évaluation de notre algorithme, nous avons travaillé sur le benchmark de domaine"Edition" qui est un ensemble d'ontologies du domaine de bibliographie proposé par AOEI[13].Nous avons utilisé ce benchmark vue la disponibilité de l'alignement de référence qui nous a permet de calculer la précision et le rappel des résultats obtenus ainsi la simulation du comportement de notre système avec le changement que les ontologies alternatives représentent par rapport à l'ontologie de référence. L'ontologie de référence est noté par (101) et les autres ontologies admettent comme noms des numéros de (102) jusqu'à (304). Dans ce qui suit, nous allons présenter les résultats d'exécution de l'algorithme proposé avec ce benchmark. Dans chaque test, le mapping se fait entre l'ontologie de référence (101) et une

ontologie alternative de même domaine. Nous avons exécuté l'algorithme sur 9 tests. Pour chaque test, la précision et le rappel de l'alignement produit sont calculés ainsi que le temps consommé. Le tableau 1 présente les résultats obtenus dans chaque test :

Table 1. Résultats du mapping des different tests

Mapping O1 et O2	Caractristiques d'O2	Precision	Rappel	Temps consommé (sec)
101 et 103	Généralisation de 101 en OWL lite	0,81	0,85	18
101 et 201	différentes conventions de nommage sont appliquées	0,77	0,62	20
101 et 205	Les labels des concepts de O(101) sont remplacés par leurs synonymes dans O(205) et Les commentaires sont supprimés	0,55	0,28	20
101 et 221	les sous classes des classes de O(101) sont supprimés dans O(221).	0,82	0,86	20
101 et 224	Pas d'instances dans O(224)	0,82	0,86	20
101 et 228	Les propriétés et les relations entre les objets sont complètement supprimées.	0,69	1,0	20
101 et 301	ontologie réelle appelée BibTeX où différentes conventions de nommage est utilisées	0,54	0, 74	14
101 et 232	Pas d'hiérarchie entre classes et absence d'instances	0,84	0,88	18
101 et 304	cas réel d'une ontologie de domaine de bibliographie où quelques classes sont des sous classes de O(101).	0 ,75	0,76	20

5.1 Discussion des résultats

L'algorithme que nous avons proposé obtient une qualité satisfaisante dans tous les différents tests en raison que les caractéristiques sur lesquelles nous nous sommes basés proviennent d'une ressource externe qui est WordNet. Pour cela les modifications apportées aux différentes ontologies n'affectent pas les résultats obtenus. Sauf que dans le test (101-205), nous avons obtenus des valeurs peu satisfaisantes (0,55 et 0,28) et sa peut se justifier par le fait que les synonymes des concepts utilisés dans l'ontologie 205 sont différents de ceux existants dans le dictionnaire WordNet. Pour cette raison, nous proposons aux développeurs d'ontologies de se basé sur la terminologie utilisée dans WordNet afin que son apport dans le processus de mapping soit considérable. Dans le test dont nous avons comparé l'ontologie 101 et l'ontologie 201 où les différentes conventions de nommages sont appliquées, l'étape de normalisation a joué un grand rôle pour obtenir des bons résultats. En résumant, l'algorithme proposé peut s'en passé des hétérogénéités que les ontologies de benchmark présentent vue les résultats obtenus.

5.2Comparaison avec d'autres approches

Pour pouvoir comparer notre algorithme avec d'autres algorithmes qui adressent le même problème "le mapping d'ontologies", nous avons utilisé un ensemble de tests issus de benchmark proposé en OAEI en 2010 de domaine d'Edition. Pour chaque test, nous avons calculé la précision et le rappel et sa par rapportà l'alignement de référence. Puis nous avons calculé la moyenne de chacune (précisionet rappel) afin de les comparer aux moyennes obtenus par d'autres algorithmes.

Les algorithmes que nous avons choisis pour la comparaison sont des algorithmes qui ont été déjà évalué par l'organisation OAEI en 2010[14] avec le même jeu de données et qu'ils présentent des manières différentes pour trouver l'alignement.Ces algorithmes sont : Falcon [15], Taxomap[16] , MapPso[17] et l'alignement de référence refAlign qu'est un alignement conçue par l'être humain et qui présente la meilleure précisionet rappel (1.0).

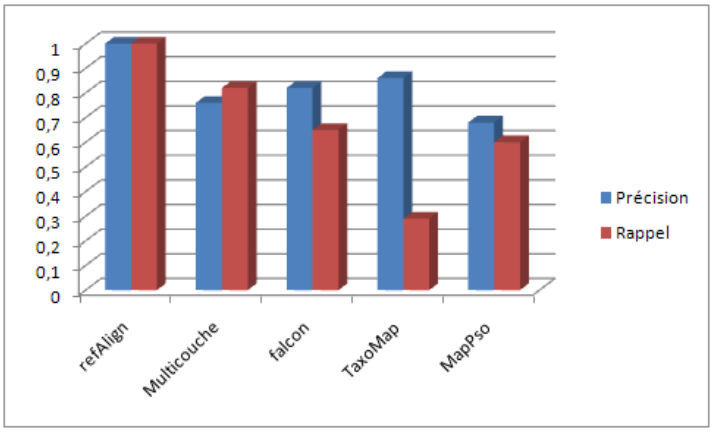


Fig.2. Comparaison entre les différents algorithmes

La Figure 2 présente une comparaison entre ces différents algorithmes en terme de précision et de rappel, où l'axe des ordonnées présente les valeurs des différentes précisions et rappels obtenues par les différents algorithmes, qui sont des valeurs appartenant à l'intervalle [0,1] et l'axe des abscisses qui présente les différents algorithmes à comparés. Les résultats des différents algorithmes sont obtenus à partir de [14]. A partir de cette figure, nous pouvons dire que notre algorithme obtient le meilleur rappel par rapport aux algorithmes comparés est sa car nous avons obtenu 82% de correspondances correctes par rapport à l'alignement de référence vue les différentes caractéristiques considérées dans notre approche (les labels des concepts, leurs superclasses dans WordNet et leurs synonymes) et la manière dont nous avons interprété nous résultats afin de tirer les correspondances. Premièrement, nous avons

donné plus d'importance à la méthode lexicale N-gram car deux entités qui s'écrivent de la même manière sont équivalentes. Si cette fonction, nous offre des valeurs satisfaisantes, nous établirons directement une correspondance sinon nous passons à la méthode Jaccard qui s'opère sur l'ensemble des synonymes de chacun des concepts pour remédier au problème des synonymes. Enfin, la dernière méthode UCS qui considère toutes les superclasses des concepts dans WordNet. Pour la précision, notre algorithme obtient une valeur satisfaisante mais pas trop loin des précisions obtenues par les autres algorithmes, le fait que ces derniers considèrent plus de caractéristiques (labels, commentaires, super classes, sous classes, propriétés. . .) et appliquent une variété de mesures de similarité. La précision que nous avons obtenue est plus grande que celle obtenu par MapPso, le fait qu'il a appliqué une seule méthode sur les labels attribués aux entités qu'est"String distance".

6 Conclusion

Dans ce papier, nous nous sommes intéressés au problème de mapping d'ontologies dans les environnements distribués, qui est une conséquence de problèmede l'interopérabilité entre les ontologies et la nature des applications où le mappingest une phase primordiale. Les environnements distribués s'exécutent sur des architectures et des données hétérogènes ce qui cause des difficultés dans lacommunication et la coopération.

Le mapping d'ontologies pose des vraies difficultés qui peuvent se résumer par l'hétérogénéité des ontologies à savoir l'hétérogénéité conceptuel, terminologique, syntaxique et sémantique, Leur nombre et leur taille qui ne cesse de croitre vue l'augmentation continue des sources d'informations sur le web ainsi que la nature distribué des applications et leurs besoins de communication et de partage et leurs exigences en qualité des mapping produit et en efficacité.

Afin de répondre aux exigences des environnements distribués et de régler le problème d'hétérogénéité des ontologies, nous avons proposé un algorithme distribué qui compare seulement les entités les plus significatives « concepts » au lieu de comparer toutes les entités présentent dans l'ontologie afin de réduire le nombre de comparaison et de gagner en efficacité. Pour le calcul des similarités, nous nous sommes basé sur l'aspect lexical, structurel et sémantique de concepts et nous avons utilisé le dictionnaire WordNet pour enrichir les concepts avec des termes supplémentaires pour augmenter la probabilité de trouver des mappings. L'exécution parallèle des différents agents permet aussi un gain de temps.

Dans notre évaluation expérimentale, nous avons montré que la nouvelle approche de mapping donne des résultats satisfaisants où nous avons obtenu une moyenne de précision de 0,76 et une moyenne de rappel de 0,82 et sa peut se justifier par les différents aspects sur lesquels nous nous sommes basés pour le calcul de similarité.En plus l'approche proposée peut s'en passé des hétérogénéités que les ontologies de

benchmark présentent vue que les caractéristiques des concepts proviennent d'un dictionnaire. Nous avons aussi la complexité obtenu qui est une complexité polynomiale de O (n2).

Références

1. W.N.Borst, 'Construction of engineering ontologies.PHD thèse, centre de communication etd'information',1997.
2. AnHai Doan, JayantMadhavan, Pedro Domingos,Alon Halevy, 'Learning toMap between Ontologies on the Semantic Web', Springer,2003.
3. Isaac Lera, Carlos Juiz, Ramon Puigjaner, 'Quick Ontology Mapping Algorithmfor distributed environments', 2008.
4. Yanniskalfoglou, Marco Schorlemmer, 'ontology Mapping: The State of the art', Advanced Knowledge Technologies, 2003.
5. Marc Ehrig, Steffen Staab, 'QOM - Quick Ontology Mapping', 2004,Springer.
6. George A. Miller, 'WordNet :A Lexical Database for English', 1995.
7. http ://www.w3.org/2004/OWL/.
8. http ://www.w3schools.com/RDF/
9. Von Dipl.-Wi.-Ing. Marc Ehrig , ' Ontology Alignment :Bridging the Semantic', 2005.
10. GrzegorzKondrak, 'N-gram similarity and distance. String Processing andInformation Retrieval', Springer, 2005.
11. B. Bagheri Hariri et al, ' A new Structural Similarity Measure for OntologyAlignment', 2006.
12. Mikalai Yatskevich1, Fausto Giunchiglia1, and Paolo Avesani2. 'A Large ScaleDataset for the Evaluation of Matching Systems', 2009.
13. Http: //oaei.ontologymatching.org/.
14. PavelShvaiko, JérômeEuzenat, 'Ontology Matching OM-2010', Proceedingsof the ISWC Workshop.2010.
15. Wei Hu, Yuzhong Qu. 'Falcon-AO: A practical ontology system', 2008.
16. F.Hamdi, C. Reynau, B. Safar. 'A Framework for Mapping Refinement Specification.
17. Jurgen Bock, Jan Hettenhausen. 'Ontology Alignment using Discrete ParticleSwarm Optimisation'.2010
18. JérômeEuzenat · PavelShvaiko, Ontology Matching, Springer, livre, 2007

Domain Ontology Learning from Texts:
Graph Theory Based Approach using Wikipedia

Khalida Ben Sidi Ahmed[1], Adil Toumouh[1], and Dominic Widdows[2]

[1] Department of Computer Science, Djillali Liabes University,
Sidi Bel Abbes, Algeria
send.to.khalida@gmail.com
toumouh@gmail.com
[2] Microsoft Bing
Washington, USA
dwiddows@microsoft.com

Abstract. Ontology Engineering is the backbone of the Semantic Web. However, the construction of formal ontologies is a tough exercise which requires time and heavy costs. Ontology Learning is thus a solution for this requirement. Since texts are massively available everywhere, making up of experts' knowledge and their know-how, it is of great value to capture the knowledge existing within such texts. Our approach is thus an original research work which answers the challenge of creating concepts' hierarchies from textual data taking advantage of the Wikipedia encyclopedia to achieve some good quality results. In addition, we present an integrated system for ontology learning that uses state-of-the-art extraction and graph-normalization techniques, and enables interactive human feedback, for evaluating and deploying the system.

Keywords : domain ontologies, ontology learning from texts, concepts' hierarchy, Wikipedia.

1 Introduction : Ontology Learning

Ontology engineering is a hard task which is time-consuming and quite expensive. The dificulty of designing manually such means stems mainly from the knowledge acquisition bottleneck; a commonly known issue. This problem is due to the fact that the needed knowledge is related to human know-how and te most important share of it is retained in their brains.

It should be noted that the main application of such a modern designed artifact, i.e. the ontology, is the semantic Web[1]. Furthermore, the success of it depends on the proliferation of ontologies, which requires speed and simplicity in engineering them [2]. The need for (semi) automatic domain ontological extraction has thus been rapidly felt by the research world. Ontology learning is then the research realm referred to.

Concepts' hierarchy building stands in the heart of ontology learning process. The construction of such structures is recently being supported by graph theory

[e.g. 3]. In fact, to support reasoning with a deterministic search-space, hierarchies with single-inheritance are preferable. This will be a central objective that our approach try to reach.

In fact, Wikipedia is recently showing a new potential as a lexical semantic resource [4]. When this collaboratively constructed resource is used to compute sematic relatedness[e.g. 5, 6] using its categories' system, this same system is also used to derive large scale taxonomies [e.g. 7] or even to achieve knowledge acquisition [e.g. 8]. Our approach capitalizes on the well organized Wikipedia articles to retrieve the most useful information at all, namely the definition of a concept. Three things seem to distinguish our work : i. the graph normalization, ii. the interactive evaluation technique and iii. the application to a particular domain (the oil industry).

First we will describe in Section 2 the ontology learning layer cake. In Section 3, we move straightforward to the explanation of our approach which will be followed by a corresponding evaluation in Section 4. Finally, Section 5 sheds the lights on some conclusions and research perspectives.

2 Ontology Learning Layer Cake

The process of extracting a domain ontology can be decomposed into a set of steps, summarized by [9] and commonly known as "ontology learning layer cake". The following page contains the figure which illustrates these steps.

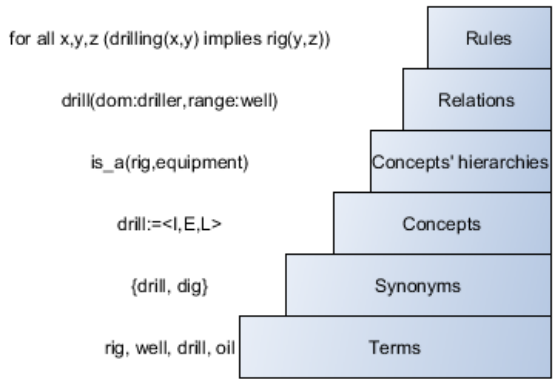


Fig. 1. Ontology learning layer cake (adapted from [9])

The first step of the ontology learning process is to extract the terms that are of great importance to describe a domain. A term is a basic semantic unit which can be simple or complex. Next, synonyms among the previous set of

terms should be extracted. This allows associate different words with the same concept whether in one language or in different languages. These two layers are called the lexical layers of the ontology learning cake. The third step is to determine which of the existing terms, those who are concepts. According to [9], a term can represent a concept if we can define: its intention (giving the definition, formal or otherwise, that encompasses all objects the concept describes), its extension (all the objects or instances of the given concept) and to report its lexical realizations (a set of synonyms in different languages).

The extraction of concepts hierarchies, our key concern, is to find the relationship 'is-a', ie classes and subclasses or hyperonyms. This phase is followed by the non-taxonomic relations' extraction which consists on seeking for any relationship that does not fit in a previously described taxonomic framework. The extraction of axioms is the final level of the learning process and it is argued to be the most difficult one. To date, few projects have attacked the discovery of axioms and rules from text.

3 Concepts' Hierarchy Building Approach

Our approach tackles primarily the construction of concepts' hierarchies from text documents. We will make a terminology extraction using a dedicated tool for this task which is TermoStat [10]. The initial terms will be the subjects of a definitions' investigation within Wikipedia. Adapting the idea of the lexico-syntactic patterns defined by [11] to our case, the hyperonyms of our terms will be learned. This process is iterative which comes to its end when an in advance predefined maximum number of iterations is reached. Our algorithm generates in parallel a graph which unfortunately contains cycles and its nodes may have more then one hyperonym. The hierarchy we promise to build is the transformation result of the graph to a forest focusing on the hierarchic structure of a taxonomy. The figure on the following page gives the overall idea of the proposed approach.

3.1 Preliminary Steps

In order to carry out our approach, we should first undergo the two lexical ontology learning's layers. The tool we used for the sake of retrieving the domain terminology is TermoStat. This web application was favored for determined reasons. In fact, TermoStat requires a corpus of textual data and, juxtaposing it to a generalized corpus such as BNC (British National Corpus), will give us a list of the domain terms that we need for the following step. Afterwards, we try to find out the synonyms among this list of candidate terms. The use of thesaurus.com as a tool in order to select synonyms was efficient. The third layer can be skipped in our context; concepts' hierarchies construction does not depend on the concepts' definitions. In other words, our algorithm needs mainly the candidate terms elected to be representative for the set of its synonyms (synset). The set of initial candidate terms is named C_0.

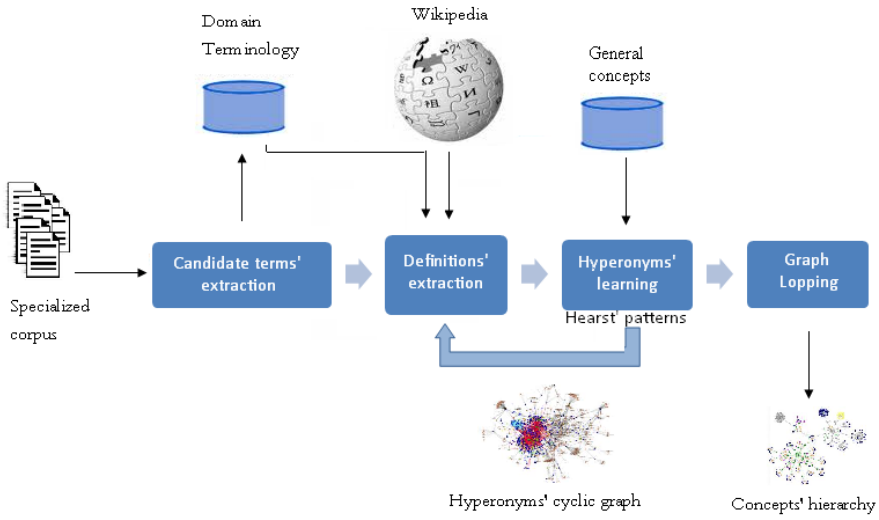


Fig. 2. Steps of the proposed approach

3.2 Concepts' Hierarchy

The approach we are proposing belongs to two research paradigms, namely concepts' hierarchies construction for ontology learning and secondly the use of Wikipedia for knowledge extraction. The achievement of our solution relies heavily on concepts from graphs' theory.

a. Hyperonyms' Learning using Wikipedia

At the beginning of our algorithm, we have the following input data:

- $G = (N, A)$ is an oriented graph such as N is the set of nodes and A is the set of arcs, $N = C_0$. Our objective is to extend the initial graph with new nodes and arcs; the former are the hyperonyms and the later are the subsumption links. The extension of C_i, i is the iteration index, is done by using the concepts' definitions extracted from Wikipedia.
- C_{gen} is a set of general concepts for which we will not look for hyperonyms. These elements are defined by the domain experts including for example object, element, human being, etc.

S1 For each $c_j \in C_i$, we check if $c_j \in C_{gen}$. If it is the case, this concept will be skipped. Else, we look for its definition in Wikipedia. The definition of a given term is always the first sentence of the paragraph before the TOC of the corresponding article. Three cases may occur:

1. The term exists in Wikipedia and its article is accessible. Then we pass to the following step.

2. The concept is so ambiguous that our inquiry leads to the Wikipedia disambiguation page. In this situation, we ignore the word.
3. Finally, the word for which we seek a hyperonym does not exist in the database of Wikipedia. Here again, we skip the element.

S2 For the definition of the given concept, we apply the principle of Hearst's patterns. We attempt to collect exhaustive listing of the key expressions we need. For instance, the definition may contain: is a, refers to, is a form of, consists of, etc. This procedure permits us to retrieve the hyperonym of the concept c_j. The new set of concepts is the input data for the following iteration.

S3 Add into the graph G the nodes corresponding to the hyperonyms and the arcs that link these nodes.

b. From Graph to Forest

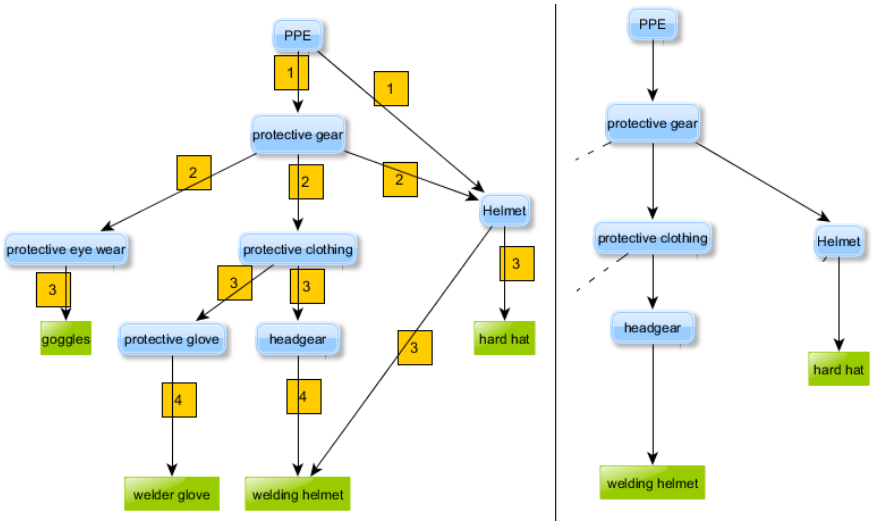


Fig. 3. From wells' drilling HSE graph to forest

The main idea which shapes the following stage shares a lot with [12]. In fact, the graph which results from the preceding step has two imperfections. The first one is that many concepts are connected to more then one hyperonym. In addition, the structure of the resulting graph is patently cyclic which does not concord with the definition of a hierarchy. An adequate treatment is paramount in order to clean up the graph from circuits as well as multiple subsumption links. Thus, we will obtain, at the end, a forest respecting the structure of a hierarchy.

The previous illustrative graph is a piece taken from the whole graph that we obtained during the evaluation of our approach. It represents a part of

drilling wells' HSE namely the PPE (Personal Protective Equipment). The green rectangles are the initial candidate concepts.
The resolution of the first raised imperfection implies obviously the resolution of the second one. Therefore, we will use the following solution:

1. Weigh the arcs so as to foster long roads within the graph. We will increment the value assigned to the arc the more we go in depth (it is already done in fig.3).
2. We adopt the Kruskal's algorithm[1956] which creates a maximal covering forest from a graph (fig.3). However, in our case we select heavy edges instead of lightweight ones in order to promote long paths.

Finally we have reached the aim we have planned.

4 Our Approach's Evaluation

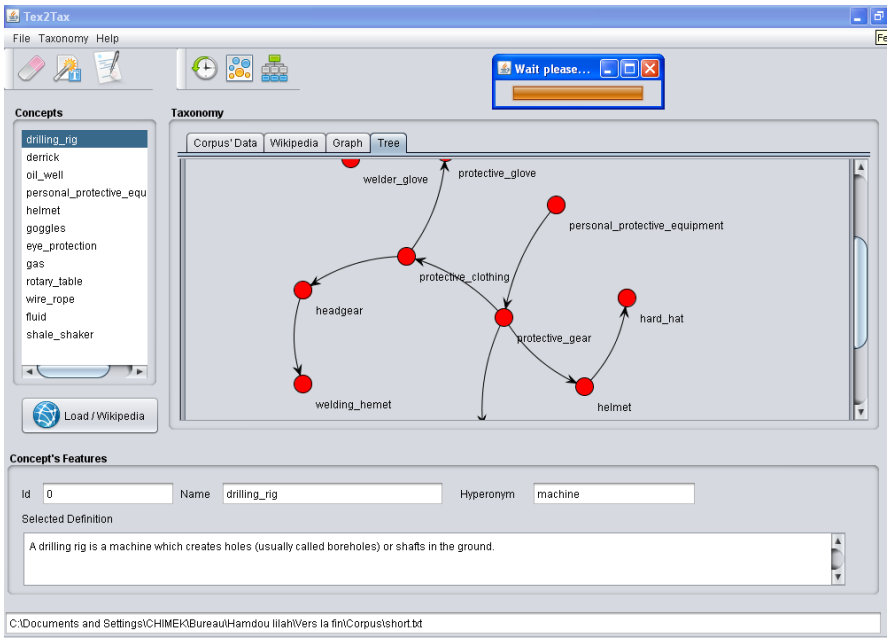


Fig. 4. Tex2Tax prototype's GUI

Our evaluation corpus is a set of texts that are collected in the Algerian/British/Norwegian joint venture Sonatrach / British Petroleum / Statoil. This specialized corpus deals with the field of wells' drilling HSE . Throughout our approach, interventions from the experts are inevitable.

Tex2Tax is the prototype we have developed using Java. Jsoup is the API which allows us to access online Wikipedia. The same result is reached if using JWPL with the encyclopedia's dump. JUNG is the API we have used for the management of our graphs. The preceding page's figure is the GUI of our prototype.

The terminology extraction phase and the synonyms retrieving have given a collection of 259 domain concepts. The final graph is formed by 516 nodes and 893 arcs. After having done the cleaning, the concepts' forest holds 323 nodes, among them 211 are initial candidate terms. The number of remaining arcs is of 322. In order to study the taxonomy structure we calculate the compression ratio for the nodes which is 0.63(323/516) and the one of the arcs which equals to 0.36(322/893).

In addition, with the help of domain experts, we have done a manual evaluation of the resulting concepts' hierarchy. The precision and the recall values are respectively:

$LP = 0.65(211/323)$

$LR = 0.37(211/566)$

The precision of our taxonomy is relatively low. This phenomenon is mainly due to the terms that do not exist in the database of Wikipedia. The graph's lopping is also responsible of some loss of nodes containing appropriate domain vocabulary.

5 Conclusion

Despite all the work which is done in the field of ontology learning, a lot of cooperation, many contributions and resources are needed to be able to really automate this process. Our approach is one of those few works that harness the collaboratively constructed resource namely Wikipedia. The results achieved and which are based on the exploitation of the idea of Hearst's lexico-syntactic patterns and the graphs' pruning is seen to be very promising. We intend to improve our work by addressing other issues such as enriching the research base by the Web, exploiting the categories' system of Wikipedia in order to attack higher levels of the ontology leaning process such as non-taxonomic relations. A further work suggestion we would make would be to try to resolve the disambiguation problem we face when we access disambiguation pages of Wikipedia. Finally, multi-lingual ontology learning is, in addition, an alive research area which is just timidly evoked.

Acknowledgement We are thankful to the Sonatrach / British Petroleum / Statoil joint venture's President and its Business Support Manager for giving us the approval to access the wells' drilling HSE corpus.

References

[1] A. Johannes Pretorius. Ontologies-Introduction and Overview. Vrije Universitiet Brussels 2004.

[2] IJCAI'2001 Workshop on Ontology Learning, Proceedings of the Second Workshop on Ontology Learning OL'2001, Seattle, USA, August 4, 2001 (Held in conjunction with the 17th International Conference on Arti

cial Intelligence IJCAI'2001). CEUR Workshop Proceedings, 2001

[3] R. Navigli, P. Velardi, S. Faralli. A Graph-based Algorithm for Inducing Lexical Taxonomies from Scratch. Proc. of the 22nd International Joint Conference on Arti

cial Intelligence, 2011

[4] Z. Zesch, C. Müller, and I. Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary . In Proceedings of the Conference on Language Resources and Evaluation (LREC). European Language Resources Association, 2008.

[5] S.P. Ponzetto and M. Strube. Knowledge Derived from Wikipedia for Computing Semantic Relatedness. Journal of Artificial Intelligence Research 30, 2007.

[6] M. Strube and S. Paolo Ponzetto. Wikirelate ! computing semantic relatedness using wikipedia. Proceedings of the National Conference on Arti

cial Intelligence (AAAI), 2006

[7]S. P. Ponzetto and M. Strube. Deriving a Large Scale Taxonomy from Wikipedia. AAAI '07, 2007

[8] V. Nastase et M. Strube. Decoding Wikipedia Categories for Knowledge Acquisition. AAAI '08, 2008.

[9] P. Buitelaar, P. Cimiano, B. Magnini. Ontology learning from text: An overview. ontology learning from text: Methods, evaluation and applications. Frontiers in Artificial Intelligence and Applications Series 123, 2005

[10] P. Drouin. Acquisition automatique des termes : l'utilisation des pivots lexicaux specialises, thse de doctorat, Montral : Universit de Montral, 2002.

[11] M. A. Hearst and H. Schutze. Customizing a lexicon to better suit a computational task. Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text, 1993.

[12] R. Navigli, P. Velardi, S. Faralli. A Graph-based Algorithm for Inducing Lexical Taxonomies from Scratch. Proc. of the 22nd International Joint Conference on Artificial Intelligence, 2011

Conceptualisation d'une Ontologie Floue

Asma Djellal[1], Zizette Boufaida[1]

[1] Laboratoire LIRE, Université Mentouri, Constantine, Algérie
{Asmadjellal, zboufaida}@gmail.com

Résumé. La construction des ontologies passe nécessairement par une étape de conceptualisation. C'est l'étape d'identification des connaissances contenues dans un corpus représentatif du domaine. Cette étape mérite plus d'attention car elle facilite la compréhension du domaine pour une meilleure représentation. Dans notre vie quotidienne, nous vivons dans un monde où les connaissances sont incomplètes, vagues généralement imprécises ou incertaines, que nous ne pourrons pas modéliser avec les logiques classiques. Nous nous intéressons dans notre travail à la représentation des connaissances imprécises et pour faciliter cette tâche, nous proposons dans cet article, un processus simple, clair et suffisamment complet pour la conceptualisation d'une ontologie floue. Ce processus est inspiré de la méthodologie générique de conceptualisation d'ontologies classiques METHONTOLOGY, enrichi par des notions de la logique floue afin de pouvoir représenter l'aspect flou des connaissances du domaine.

Mots-clés: Représentation de connaissances, Connaissances floues, Conceptualisation d'ontologies, Logique floue, Ontologies floues.

1 Introduction

Les ontologies sont à l'heure actuelle au centre de nombreuses applications de l'ingénierie des connaissances, en particulier le projet Web sémantique. Elles aident à concevoir le monde réel avec ses contraintes sémantiques [7]. Mais, ce monde comporte des imprécisions et des imperfections que nous ne pouvons pas concevoir en utilisant les ontologies classiques (i.e., précises). Les représentations des ontologies de manière formelle reposent généralement sur une description en logique classique (i.e., la logique du premier ordre). Or celle-ci montre ses limites pour tous les faits qui ne s'expriment pas avec des valeurs « vraies » ou « fausses ».

Des approches floues ont alors été proposées pour rendre plus souples les possibilités de représentation en affectant des poids aux différents liens. A titre d'exemple, nous dirons que « le patient a une fièvre moyennement forte » plutôt que «le patient a ou n'a pas de fièvre». Dans notre travail, nous nous sommes principalement intéressés au problème de la représentation des connaissances imprécises. Nous pensons que le moyen le plus approprié est de construire des ontologies dites « *floues* ». Tout comme les ontologies classiques, la construction des ontologies floues, passe forcement par l'étape de la conceptualisation. Cette dernière fait le sujet de cet article dont lequel nous proposons un processus détaillé et

suffisamment complet pour couvrir cette étape de conceptualisation. Ce processus admet en entrée un corpus représentatif du domaine et génère en sortie un ensemble de documents décrivant semi-formellement l'ontologie floue conceptuelle.

La suite de cet article est organisée comme suit. Dans la section 2 nous introduirons le problème de la représentation des connaissances imprécises en mettant l'accent sur l'impuissance des logiques de descriptions à exprimer le flou. Une brève introduction à la logique floue fera l'objet de la section 3. Nous présentons dans la section 4 notre contribution à savoir, un processus pour la conceptualisation d'ontologies floues. Dans la section 5, nous concluons ce travail et donnons quelques perspectives pour sa poursuite.

2 Les Ontologies et les Logiques de Descriptions

Les logiques de descriptions sont utilisées avec succès pour représenter les ontologies dans plusieurs domaines, en particulier le projet Web Sémantique. Cependant, leur représentation n'est pas une tâche facile du fait de la difficulté à trouver des définitions complètes, précises des concepts d'un domaine. En effet, la définition de Studer [10] qui stipule qu'*une ontologie est la spécification formelle et explicite d'une conceptualisation partagée*, suppose qu'une ontologie est un ensemble de connaissances définies explicitement et modélisées formellement dans un langage de représentation de connaissances. Or trouver des définitions complètes et précise des concepts du monde réel avec toutes ces imperfections et ces incomplétudes s'avèrent une tâche très difficile. Les logiques de descriptions constituent la base des formalismes de représentation des connaissances. Malheureusement, elles sont sévèrement limitées du fait de leur faiblesse à représenter notre incertitude sur le monde, en raison du manque de connaissances sur le monde réel d'une part, et par leur faiblesse à représenter les connaissances intrinsèquement imprécises d'autre part. En effet, il ya des concepts, comme la chaleur par exemple, pour lesquelles aucune définition exacte n'existe, et par conséquent, un fait comme « 35˚ Celsius est chaud», étant plutôt vrai ou faux, a une valeur de vérité entre vrai et faux [9].

Considérons cet exemple, dont lequel nous voulons construire une ontologie des fruits; il peut être demandé de représenter un concept comme: « Une pomme est un fruit de forme presque ronde, il peut être vert clair, jaune ou grenat foncé. Sa taille peut varier de petite à moyenne jusqu'à grande ». Nous pouvons remarquer qu'il est très difficile de représenter ce concept en logique de descriptions classique, du fait de l'utilisation de plusieurs termes flous, comme « *presque*», « *clair*», « *foncé*», « *petite* »... C'est dans ce genre de cas, où le concept à représenter n'a pas une définition précise, qu'il est préférable d'utiliser les logiques de descriptions floues.

La structure des logiques de descriptions classiques qui ne peut exprimer des faits qu'avec "vrai" ou "faux" limite leur champ d'action dans plusieurs applications de l'Intelligence Artificielle qui veulent imiter le raisonnement et l'esprit humain, c'est à dire des techniques qui s'appuient sur l'incertitude pour leur bon fonctionnement.

A cet effet, nous retrouvons dans la littérature, plusieurs propositions pour étendre les logiques de descriptions par des théories mathématiques qui traitent l'incertain et l'imprécis [6], [8]. Ceci a donné naissance aux *logiques de descriptions floues*.

3 La Logique Floue

La logique floue est conçue pour régler le problème de la représentation de l'incertain et l'imprécis. Elle permet la caractérisation des éléments de façon « graduelle ». Elle a été introduite par L.A. Zadeh à la fin des années 60 comme extension de la logique booléenne [9]. Cette logique ne consiste pas à être précis dans les propositions, mais au contraire répondre à des propositions vagues, nécessitant une certaine incertitude ou incomplétude (un flou). Par exemple, en logique classique, si nous posons la question « *Est ce qu'il fait chaud aujourd'hui?* », nous ne pouvons répondre que par vrai dans le cas où il fait chaud ou par faux dans le cas contraire. Par contre, avec la logique floue, on peut représenter les cas où il *fait très chaud, il fait chaud, il fait froid* et même *il fait fortement froid*…

Le raisonnement d'un être humain est souvent basé sur des connaissances floues. Pour résoudre les problèmes quotidiens, il utilise des connaissances dont il doute de leurs validités (incertaines) ou mal exprimées du fait de la complexité du problème posé (imprécis). En dépit de cela, il arrive souvent à résoudre ces problèmes complexes sans même avoir besoin de les modéliser. Selon [2], il est souvent intéressant de modéliser le comportement d'un opérateur humain face au système plutôt que de modéliser le système lui-même. Il est préférable aussi de décrire ce système par des quantificateurs globaux traduisant son état plutôt que par des valeurs numériques précises.

3.1 Notion d'Appartenance Partielle

Dans la théorie des ensembles classiques, deux situations peuvent être envisagés. Un élément peut ou ne pas appartenir à un ensemble. La notion d'ensemble classique ne permet pas de prendre en compte plusieurs situations rencontrées fréquemment dans notre vie quotidienne : Il sera très difficile d'affirmer qu'il fait chaud aujourd'hui car la chaleur est une notion progressive. Si pour une température de à 25°, nous disons qu'il fait chaud, est ce qu'il ne ferait pas chaud avec une température égale à 24.8° ?

La notion d'ensemble flou est justement née pour prendre en compte ce genre de situations. Dans [2] la théorie des ensembles flous est définie comme une théorie basée sur la notion d'appartenance partielle. Chaque élément appartient partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou (cf. Fig.1) ne sont pas « nets », mais « flous » ou «graduels ».

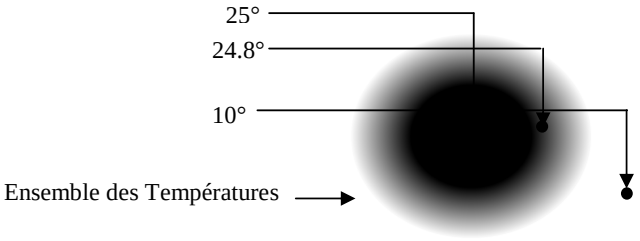


Fig.1. Ensemble flou et appartenance partielle.

Cet exemple montre que la température 25° appartient totalement à l'ensemble, donc c'est une température chaude. La température 24.8° appartient partiellement à l'ensemble, donc c'est une température moyennement chaude. Quand à 10°, elle n'appartient pas à l'ensemble, donc ce n'est pas une température chaude.

Dans ce qui suit, nous donnons une comparaison entre la logique classique et la logique floue. Nous considérons les ensembles « Il fait froid », « Il fait beau » et «Il fait chaud ». Dans Fig. 2, nous considérons ces ensembles comme des ensembles classiques.

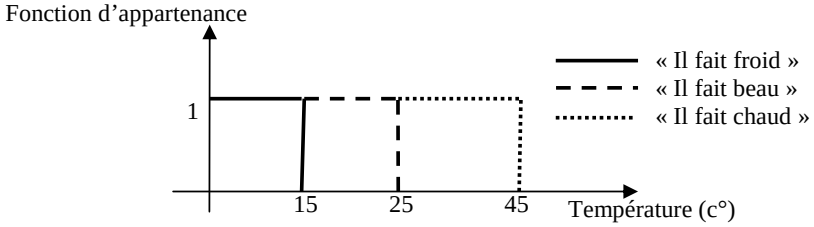


Fig.2. Représentation classique des trois ensembles

Considérons maintenant ces ensembles avec une représentation floue :

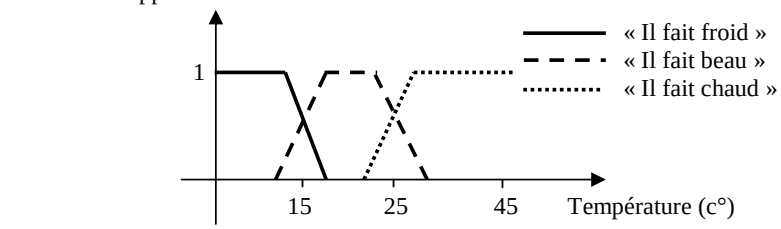


Fig.3. Représentation floue des trois ensembles.

Nous discernons que la logique classique ne peut prendre en compte que deux valeurs : le 0 et le 1. Ainsi il fait totalement *froid* puis brusquement *beau* et enfin *chaud*. Par contre, la deuxième représentation (cf. Fig. 3) montre le passage graduel entre les trois états (froid, beau et chaud).

Définition de la fonction d'appartenance : Soit $\mathcal{X}$ un ensemble, A un sous-ensemble flou de $\mathcal{X}$ défini par une fonction d'appartenance μ_A sur $\mathcal{X}$ qui peut prendre toutes les valeurs dans l'intervalle [0, 1]. La notion de sous-ensemble flou englobe celle de sous-ensemble classique pour laquelle μ_A est la fonction indicatrice $\mathcal{X}_A$. Un élément appartiendra à un sous-ensemble flou avec un degré d'appartenance. Il s'agira en fait de la valeur prise par la fonction d'appartenance du sous-ensemble flou au point considéré. La fonction d'appartenance μ_A est caractérisée par :

Support de A : $supp\ A=\{x\in\mathcal{X},\ \mu_A(x)\neq 0\}$

Hauteur de A : $h(A)= sup\ x\in X\ \mu A(x)$

Noyau de A : $noy(A)=\{x \in X, \mu A(x)=1\}$

A est plus spécifique que B ssi $noy(A) \subseteq noy(B)$ *et* $sup(A) \subseteq sup(B)$.

3.2 Les Opérateurs de la Logique Floue

Dans la logique floue, les opérations définies sont les mêmes qu'en logique classique. Soit l'ensemble $\mathcal{X} = \{x_1, x_2, \ldots..x_n\}$, sur lequel nous définissons le triplet (c, d, n) comme étant des opérateurs flous de base:

La conjonction:

$c : [0, 1]^n \rightarrow [0, 1]$

$(x_1, x_2 \ldots x_n) \rightarrow c(x_1, x_2 \ldots x_n) = x_1 \wedge x_2 \wedge \ldots \wedge x_n = \min(\mu(x_1), \mu(x_2), \ldots \mu(x_n))$

La disjonction:

$d : [0, 1]^n \rightarrow [0, 1]$

$(x_1, x_2 \ldots x_n) \rightarrow d(x_1, x_2 \ldots x_n) = x_1 \vee x_2 \vee \ldots \vee x_n = \max(\mu(x_1), \mu(x_2), \ldots \mu(x_n))$

La négation:

$n : [0, 1] \rightarrow [0, 1]$

$x \rightarrow n(x) = 1 - \mu(x)$

En utilisant ces opérateurs flous, nous pouvons définir d'autres opérateurs, comme l'implication (i) et l'équivalence floue (e) comme suit:

L'implication:

$i : [0, 1]^2 \rightarrow [0, 1]$

$(x_1, x_2) \rightarrow i(x_1, x_2) = x_1 \Rightarrow x_2 = d(n(x_1), x_2)$

L'équivalence:

$e: [0, 1]^2 \rightarrow [0, 1]$

$(x_1, x_2) \rightarrow e(x_1, x_2) = x_1 \Leftrightarrow x_2 = c(i(x_1, x_2), i(x_2, x_1))$

La table suivante représente les deux propositions a et b définies sur l'ensemble $\mathcal{X}$ comme suit :

a= « Il fait froid » est vrai à 0.8,

b= « il neige » est vrai à 0.6.

Leurs fonctions d'appartenances sont respectivement μ_a et μ_b.

Table 1. Représentation de la conjonction, la disjonction et la négation floues.

Opération Floues	Description	Exemple
La Conjonction : c (a, b) $c(a, b) \rightarrow \min(\mu(a), \mu(b))$	le degré de vérité de la proposition c(a,b) est le minimum des degrés de vérité des deux propositions a et b.	« Il fait froid ET il neige » s*era vraie à 0.6*
La Disjonction : d (a, b) $d(a,b) \rightarrow \max(\mu(a), \mu(b))$	le degré de vérité de la proposition d(a, b) est le maximum des degrés de vérité des deux propositions a et b.	« Il fait froid OU il neige » s*era vraie à 0.8*
La négation : n(a) $n(a) \rightarrow 1 - \mu(a)$	le degré de vérité de la proposition n(a) est égal à 1- le degré de vérité de a.	« Il ne fait pas froid » s*era vraie à 0.2*

Notons que ces définitions sont celles les plus communément utilisées mais parfois, pour certains cas, d'autres sont plus appropriées. Par exemple, la conjonction peut être définie par le produit des fonctions d'appartenance et la disjonction par la moyenne arithmétique des fonctions d'appartenance. Ces différentes techniques de calcul engendrent une énorme capacité d'adaptation des raisonnements flous.

4 Construction d'une Ontologie Floue

La construction des ontologies est une tâche d'une grande importance dans le projet Web sémantique. Nous trouvons dans la littérature de nombreuses méthodes pour leur construction. Cependant, l'imprécision des connaissances modélisées n'est pas prise en considération par ces méthodes, tandis que l'être humain manipule un volume considérable de connaissances imprécises ou incertaines du fait que la plupart des domaines d'application peuvent inclure des incertitudes et des imperfections. Pour cela, notre objectif est de proposer un processus permettant la conceptualisation d'ontologies, en prenant en compte l'aspect flou des connaissances. Les grandes étapes de ce processus sont inspirées de la méthodologie générique de conceptualisation d'ontologies classiques « METHONTOLOGY » [4], enrichi par des notions de la logique floue dans les différentes étapes du processus. Notamment, nous mettons en œuvre les composants flous de l'ontologie (concepts et relations floues).

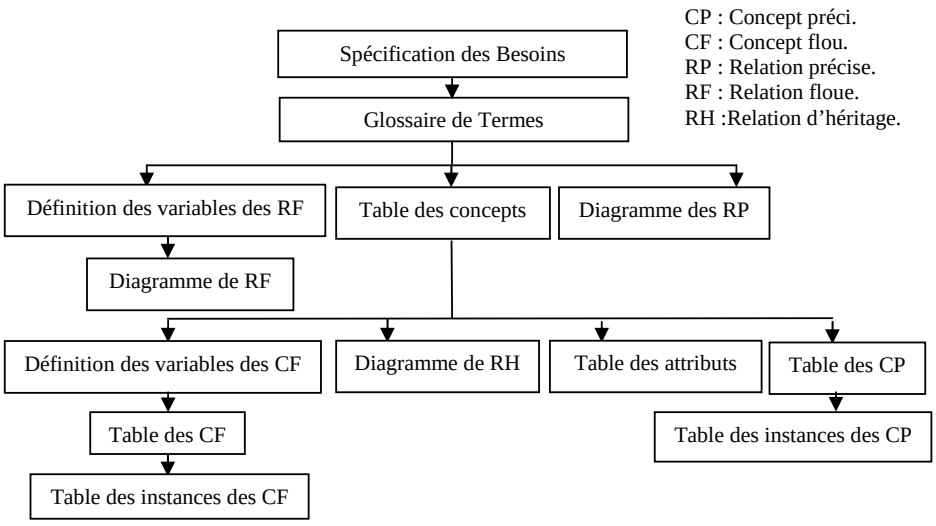


Fig. 4. Processus de conceptualisation d'ontologies floues.

4.1 Spécification des Besoins

C'est la phase initiale du processus. Elle vise la détection et la spécification des besoins qui permet notamment de circonscrire précisément le domaine de connaissances afin d'identifier les connaissances contenues dans le corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté d'un ingénieur de la connaissance. C'est dans cette phase du processus qu'il faut s'assurer que le domaine à modéliser nécessite la construction d'une ontologie floue.

4.2 Construction du Glossaire de Termes

Ce glossaire comprend les termes pertinents du domaine (concepts, instances, attributs, relation, etc). Pour chaque terme, nous joindrons une description en langage naturel.

Table 2. Glossaire de termes[1].

Terme	Description
Personne	Être humain qui a une conscience claire de lui-même et qui agit en conséquence
Voiture	Véhicule de transport monté sur roues
Possède	A quelque chose à sa disposition
……..	……..

4.3 Construction de la Table des Concepts

Cette table contient tous les concepts identifiés dans le glossaire de termes, pour chaque concept, il faut définir ses synonymes et ses attributs.

Table 3. Table des concepts.

Concept	Synonyme	Attribues
Personne	Individu	Nom
	Humain	Prénom
		Age

4.4 Construction de la Table des Concepts Précis (CP)

Parmi les concepts collectés dans la table des concepts, il ne faut garder que les concepts précis: un concept est dit précis si nous pouvons lui donner une définition claire et complète dans laquelle nous n'utilisons pas des propriétés floues, exemple : une voiture, une personne…Les experts du domaine peuvent ajouter de nouveaux concepts qui n'apparaissent pas dans la table de termes.

[1] Les descriptions sont tirées du dictionnaire « Le Robert pour tous ». Édition juin 1994

Cette table contient, pour chaque concept précis, ses synonymes, ses attributs et ses instances.

Table 4. Table des concepts précis (CP).

Concept Précis	Synonyme	Attribus	Instances
personne	Individu Humain	Nom Prénom Age	Mouhamed Ahmed

4.5 Définition des Variables et des Termes Linguistiques

Absente dans METHONTOLOGY, cette phase est très importante car elle constitue le point de départ de la définition des concepts flous. Dans cette phase, nous nous intéressons à la colonne « Attribut » de la table des concepts afin d'identifier les variables linguistiques du domaine (exemple : Age, Taille…). Chaque variable linguistique prend ses valeurs dans un ensemble de termes linguistiques. Pour chaque terme, il faut définir sa fonction d'appartenance.

Considérons la variable « Taille » par exemple, nous pouvons définir les termes suivants: « Taille-Petite, Taille-Moyenne et Taille-Grande ».

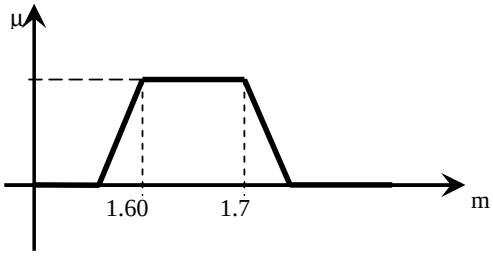


Fig.5 Fonction d'appartenance du terme linguistique « Taille-Moyenne ».

4.6 Construction de la Table des Concepts Flous (CF)

Parmi les concepts collectés dans la table des concepts, cette fois ci, il ne faut garder que les concepts flous: un concept est dit flou si nous ne pouvons pas lui donner une définition claire et précise, exemple : une voiture verte, une personne adulte, une personne de taille moyenne… Dans [5] un concept flou est décrit comme un concept défini sur la base d'une valeur particulière d'une variable linguistique relative a l'univers de discours. Ces variables représentent les propriétés floues du concept (exemple : L'âge, la taille, la dégradation de couleur…) et c'est avec ces derniers que nous pouvons représenter l'incertitude des concepts flous. Les termes linguistiques décelés dans la cinquième phase du processus deviennent des concepts flous comme Personne-TailleMoyenne.

Cette table contient, pour chaque concept flou, ses synonymes, ses attributs et ses instances. L'appartenance d'une instance à un concept flou n'est pas totale comme

dans le cas de l'appartenance à un concept précis. Une instance appartient partiellement à un concept flou, son degré d'appartenance est déterminé par la valeur prise par la fonction d'appartenance de l'instance au concept flou (modélisé comme un ensemble flou). Pour cela nous avons ajouté une colonne dans cette table indiquant le degré d'appartenance de l'instance au concept.

Les degrés d'appartenance sont calculés à l'aide de formules de calcul relatives aux types des fonctions d'appartenance. Considérant l'exemple du CF «Personne-TailleMoyenne », sa fonction d'appartenance est de type Trapézoïdale, donc les degrés d'appartenance à ce CF sont calculés à l'aide des formules suivantes:

- $\mu(x) = 1$ si $x \in [1.60, 1.75]$
- $\mu(x) = (1.60-x)/ (1.75 -1.60)$ si $x \in [1.50, 1.60[$
- $\mu(x) = (x-1.75)/(1.80-1.75)$ si $x \in]1.75, 1.80]$
- $\mu(x) = 0$ ailleurs

Table 5. Table des concepts flous (CF).

Concept Flou	Synonyme	Attributes	Instances	Degré d'appartenance
Personne-TailleMoyenne	Individu Humain	Nom Prénom Age	Ahmed	0.8

4.7 Construction du Diagramme des Relations d'héritage

Dans ce diagramme, les concepts du domaine sont organisés sous forme de taxonomie, Les relations d'héritage peuvent être entre CP comme concept générique et CF comme concept spécialisé. Nous ne trouvons jamais une relation d'héritage dont le concept générique est un CF et le concept spécialisé est un CP du fait qu'un concept spécialisé hérite toutes les propriétés du concept générique. Parmi ces propriétés, il peut y avoir des propriétés floues. Or un concept défini avec des propriétés floues ne peut être qu'un concept flou [5].

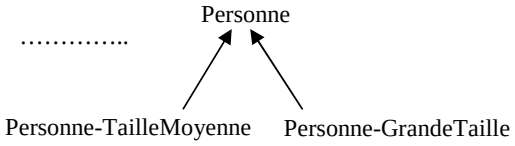


Fig.6. Diagramme de relations d'héritage

4.8 Construction de la Table des Attributs des Concepts

Cette table contient tous les attributs des concepts identifiés. Pour chaque attribut, il faut décrire son nom, le concept auquel il appartient, son type, sa cardinalité (nombre min ou max de valeurs) et son domaine de valeurs.

Table 6. Table des attributs

Attribut	Concept	Type	Cardinalités	Domaine de valeurs
Taille	Personne	Réel	1-1	[0.70, 1.90]
	Personne-TailleMoyenne	Réel	1-1	[1.50, 1.80]

4.9 Diagramme de Relations Précises (RP)

Un diagramme de relations précises est la représentation graphique des relations précises entre les concepts dans le diagramme de relations d'héritage [3]. Les losanges représentent les concepts, et les relations sont représentées par des rectangles et des arcs.
Exemple :

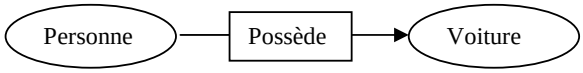


Fig.7. Diagramme de relations précises

Ce diagramme représente la relation « Possède » qui relie les deux concepts « Personne » comme domaine et « Voiture » comme Co-domaine.

4.10 Définition des Variables Linguistiques des Relations Floues

Comme dans la construction de la table des concepts flous, cette fois ci nous nous intéressons aux variables linguistiques représentant des relations. Pour chaque relation floue, il faut calculer les valeurs de sa variable linguistique et le type de sa fonction d'appartenance.

Exemple : La variable « Fait-Effort » peut prendre comme valeur « Effort-Moyen », « Effort-Faible » et « Grand-Effort ».

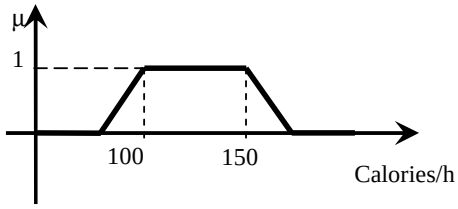


Fig.8. Fonction d'appartenance du terme linguistique « Effort-Moyen»

4.11 Diagramme de Relations Floues

Dans cette phase du processus, nous représentons les relations floues identifiées dans la phase précédente sous forme graphique comme dans le cas du diagramme des relations précises.

Exemple :

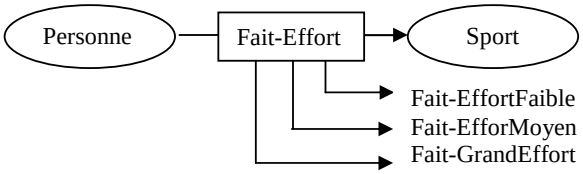


Fig.9. Diagramme de relations floues

Ce diagramme représente trois relations floues : « Fait- EffortFaible, Fait-EfforMoyen et Fait-GrandEffort» qui relie les deux concepts : « Personne » comme domaine et « Sport» comme Co-domaine.

4.12 Construction de la Table des Instances des CP

Cette table contient toutes les instances identifiées dans la colonne instance de la table des concepts précis avec leurs attributs évalués.

Table 7. Table des instances des CP

Instance	Concept	Attributs	Valeurs
Voiture-Mouhamed	Voiture	Marque	Renault-Clio
		Puissance	4 chevaux
		Année-Circulation	2004
		Energie	Gaz oil
		Nbr-place	5
		Num-immatriculation	05832-104-25

4.13 Construction de la Table des Instances des CF

Dans cette table, il faut ajouter une colonne indiquant le degré d'appartenance de l'individu aux concepts flous (l'appartenance d'une instance à un CF n'est pas sûre, elle est déterminé pas sa fonction d'appartenance).

Table 8. Table des instances des CF

Instance	Attributs	Valeurs	Concept	Degré d'appartenance
Ahmed	Nom	Slimani	Personne-TailleMoyenne	0.8
	Prénom	Ahmed		
	Age	32	Personne-PetiteTaille	0.2
	Taille	1.52		

5 Conclusion et Perspectives

Dans notre vie quotidienne, nous vivons dans un monde où les connaissances sont vagues généralement imprécises ou incertaines que nous ne pourrons pas modéliser avec la théorie des ensembles classiques et la logique binaire qui en résulte. C'est une logique où seulement deux situations peuvent être décrites formellement : l'appartenance ou la non appartenance d'un élément à un ensemble. Pour pallier à cette insuffisance, nous proposons dans cet article l'utilisation de la logique floue afin de représenter ce genre de connaissances.

Dans cet article, nous avons présenté un processus de conceptualisation d'ontologies floues inspiré de la méthodologie générique de conceptualisation d'ontologies classiques, enrichi par des notions de la logique floue afin de pouvoir représenter l'incertitude et l'incomplétude des différentes connaissances fréquentées quotidiennement. Ce processus permet la description des connaissances floues qui peuvent être des concepts ou des relations. Avec une telle description, la tâche de la représentation formelle de ces connaissances est devenue plus facile.

Dans la suite de notre travail, nous envisageons d'utiliser les logiques de descriptions floues afin de pouvoir formaliser l'ontologie floue conceptualisée dans cette étape.

Références

1. Baader, F, Horrcks, I, Sattlerl, U: Description Logics as Ontology Languages for the Semantic Web. Festschrift in honor of Jorg Siekmann, LNAI. Springer. (2003).
2. Chevri, F, Guély, F: La logique floue, Cahier technique n°191, Schneider Electric www.schneider-electric.com, rubrique maîtrise de l'électricité. (1998).
3. Hemam, M, Boufaida, Z: MVP-OWL: A Multi-Viewpoints Ontology Language for the Semantic Web. Int. Journal Reasoning- based intelligent Systems (IJRIS). Vol 3, N3/4, pp. 147--155. Inderscience Publishers (2011).
4. López M. F., Gómez-Pérez A, Pazos-Sierra A: Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. IEEE Intelligent Systems & their applications. (1999)
5. Maâlej , S, Ghorbel , H, Bahri , A, Bouaziz, R : Construction des composants ontologiques flous à partir de corpus de données sémantiques floues. Actes du XXVIII° congrès INFORSID, Marseille, (2010)
6. Mlynski, M, Zimmermann, H : An efficient method to represent and process imprecise knowledge. Applied Soft Computing 8 1050-1067, (2008)
7. Roberts, R: Introduction aux technologies du Web Sémantique. AIDAinformazioni,. http://www.aidainformazioni.it/pub/roberts122008.pdf. Italie, (2008)
8. Stoilos, G, Stamou, G : Handling Imprecise Knowledge with Fuzzy Description Logic. Proceedings of the 2006 International Workshop on Description Logics (DL2006), Windermere, Lake District, UK (2006)
9. Straccia, U: Reasoning within Fuzzy Description Logics. Journal of Artificial Intelligence Research 14, pp. 137--166 Submitted 9/00; published 4/01. (2001)
10. Studer, R, Benjamins, VR, Fensel, D: Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering. (1998).

Optimisation I

Calcul d'un Z-équilibre d'un jeu fini: Application à la résolution d'un problème CSP

K. Bouchama[1], M. S. Radjef[1], and L. Sais[2]

[1] LAMOS, Département de Recherche Opérationnelle
Université of Béjaia, Algérie
kahina.bouchama@gmail.com
radjefms@gmail.com

[2] CRIL - CNRS
Université de Lille Nord de France
Artois, France
sais@cril.fr

Abstract. La programmation par contraintes et la théorie des jeux constituent chacune un domaine de recherche très actif. Elles offrent des cadres de modélisation, d'analyse et de développement des outils pour la résolution de nombreuses applications dans des domaines variés tels que l'informatique, l'intelligence artificielle, et l'aide à la décision en général. Dans la dynamique de leurs développements, on recense quelques travaux novateurs ayant établi certains liens entre la théorie des jeux et les problèmes de satisfaction de contraintes.
Dans ce travail, nous avons établi l'équivalence entre le concept de solution pour un problème de satisfaction de contraintes (CSP) et la notion du Z-équilibre pour le jeu qui lui est associé. Par la suite, nous avons développé une procédure de recherche du Z-équilibre, en s'inspirant des approches par retour-arrière, connues pour la résolution des CSP.

Mots clés: Problèmes de satisfaction de contraintes (CSP), Théorie des Jeux, Z-équilibre.

1 Introduction

La programmation par contraintes est une discipline permettant de modéliser un grand nombre de problèmes de nature combinatoire largement présents dans divers domaines. Elle offre à l'utilisateur la possibilité de décrire son problème sous forme de contraintes à satisfaire, sa résolution étant prise en charge par un solveur de contraintes.

D'un autre coté, la théorie des jeux s'intéresse à l'étude des interactions entre agents. Elle peu servir de cadre pour la modélisation et la résolution de problèmes rencontrés dans divers domaines tels que le transport et la logistique, les télécommunications,...etc. Dans cette dynamique, on recense quelques travaux novateurs ayant établis certains liens entre la théorie des jeux et les problèmes de satisfaction de contraintes. En effet, F.Ricci [1] a proposé une représentation des CSP par un jeu noncoopératif à n joueurs. Il a prouvé que toute solution d'un CSP donné est aussi un équilibre de Nash pour le jeu associé. De plus, lorsque l'ensemble des solutions du CSP n'est pas vide, celui-ci

coincide avec l'ensemble des équilibres de Nash admissibles du jeu. Une autre modélisation envisageable est de représenter les CSP par des jeux coopératifs (Gosti et Bistarelli [8]) visant à résoudre un CSP distribué en passant par la résolution d'un jeu de nomination.En revanche, Vardi et Kolaitis [2] ont prouvé la possibilité de lier les notions de consistence pour un CSP représenté par un problème d'homomorphisme, et l'existence d'une solution pour les jeux existentiels k-cailloux. On peut éventuellement représenter un jeu par un problème de satisfaction de contraintes, dans le but de le résoudre. Bordeaux et Pajot ont adoptés cette approche pour calculer l'équilibre de Nash [5], de même pour Porter, Nudelman et Shoham [6]. Krzysztof, Rossi et Venable ont quant à eux, comparé les notions d'optimalité dans les jeux stratégiques et les softs-CSP [9]. Notons que la plupart des travaux effectués dans ce domaine, se sont surtout focalisés sur le concept d'équilibre de Nash, vu son importance majeur en théorie des jeux [4, 6, 5]. Dans ce travail, nous nous somme intéressés au concept du Z-équilibre, en prouvant son équivalence à la solution du CSP auquel le jeu en question est associé. Vu qu'il n'existe aucun algorithme dans la littérature nous permettant de le calculer, nous avons exploité cette équivalence, pour proposer une procédure nous permettant de le faire, en nous appuyant sur les outils de résolution des CSP.

La prochaine section de ce papier englobe quelques généralités sur la théorie des jeux et les problèmes CSP. La section 3 est dédiée à la représentation des CSP par un jeu. Le concept du Z-équilibre est défini dans la section 4, ainsi que les conditions de son existence et ses principales propriétés. La section 5 présente un algorithme développé pour le calcul du Z-équilibre. L'équivalence entre cet équilibre et la solution du CSP est prouvée dans la section 6. Nous appliquerons l'algorithme de calcul du Z-équilibre à la résolution d'un CSP dans la section 7, et donnerons les résultats obtenus par les tests effectués dans la section 8. Enfin, nous finirons par une conclusion.

2 Notations et définitions préliminaires

2.1 La théorie des jeux

La théorie des jeux propose un formalisme qui vise à analyser le comportement rationnel des décideurs (appelés joueurs) en situation d'interaction, sachant que de telles interactions peuvent s'étendre de la coopération au conflit. Chaque joueur est conscient que les résultats qu'il obtient en conséquence de ses actions, dépendent également des actions des autres joueurs (agents) [10].

Definition 1 (Jeu [11]) *Un* jeu *est une situation où des joueurs sont conduits à faire des choix stratégiques parmi un certain nombre d'actions possibles, et dans un cadre défini à l'avance constituant les* règles du jeu*, le résultat de ces choix forme une* issue du jeu*, à laquelle est associé un* gain *(ou payement) à chacun des participants.*

Caractériser un jeu revient à donner l'ensemble des joueurs, l'ensemble des choix possibles (ensemble des stratégies) pour chaque participant et la fonction qui donne leurs gains (fonction utilité) dans toutes les éventualités possibles [11].

Une hypothèse de base de la théorie des jeux est de considérer que tous les joueurs sont rationnels, c'est-à-dire qu'ils tentent d'atteindre la situation qui leur apporte le meilleur gain.

On appelle *utilité* d'un joueur, la mesure de chaque situation aux yeux de ce joueur. Autrement dit, l'utilité est une mesure subjective du contentement d'un joueur [11].

2.2 Les problèmes de satisfaction de contraintes (CSP)

Le concept CSP vise à représenter, sous forme de contraintes, les propriétés et les relations qui existent entre les objets manipulés. Ces contraintes peuvent être décrites de multiples façons (par une équation, une inéquation, un prédicat, une fonction booléenne, une énumération des combinaisons de valeurs autorisées,...etc). Elles traduisent l'autorisation ou l'interdiction d'une combinaison de valeurs.

La définition formelle, proposée par Montanari en 1974, est énoncée dans [12], comme suit:

Definition 2 *[13] Un problème de satisfaction de contraintes (CSP) est un problème* $(\mathcal{P})$ *caractérisé par un triplet* (X, D, C)*, où:*

$X = \{X_1, X_2, ..., X_n\}$ *est un ensemble de n variables.*
$D = \{D_1, D_2, ..., D_n\}$ *est un ensemble de domaines finis, où* D_i *est le domaine associé à la variable* X_i *représentant l'ensemble de ses valeurs possibles.*
$C = \{C_1, C_2, ..., C_m\}$ *est un ensemble de m contraintes, où la contrainte* C_i *est définie par un sous-ensemble de variables* $\{X_{i_1}, X_{i_2}, ..., X_{i_{n_i}}\} \subseteq X$.

La résolution d'un CSP consiste à affecter une valeur pour chaque variable X_i de façon que toutes les contraintes soient satisfaites.

On montrera dans la section suivante comment peut-on modéliser le CSP$(\mathcal{P})$ par un jeu noncoopératif.

3 Représentation d'un CSP sous forme d'un jeu fini

Associons à chaque variable X_i un joueur i. Ainsi, on aura autant de joueurs que de variables. Notons alors par $I = \{1, \ldots, n\}$ l'ensemble de ces joueurs.

L'ensemble S_i des stratégies pures du joueur $i \in I$ est identifié à l'ensemble D_i des valeurs possibles de la variable X_i, $i \in I$. Ainsi, $S_i = D_i, \quad i \in I$.

Notons par:

$R(i)$, l'ensemble des contraintes de C liées à la variable X_i, $i \in I$.
r, désigne une contrainte du CSP$(\mathcal{P})$ et $k(r)$, son arité.

$x = (x_1, \ldots, x_n) \in S = \prod_{i=1}^{n} S_i$, une instanciation complète des n-variables du CSP($\mathcal{P}$).
Soit la fonction indicatrice:

$$\chi_r(x_{j_1}, \ldots, x_{j_{k(r)}}) = \begin{cases} 1, \text{ si } (x_{j_1}, \ldots, x_{j_{k(r)}}) \in r, \\ 0, \text{ sinon.} \end{cases} \quad (1)$$

où $(x_{j_1}, \ldots, x_{j_{k(r)}}) \in r$ signifie que la contrainte r est vérifiée par l'instanciation $x = (x_1, \ldots, x_n)$ et $(x_{j_1}, \ldots, x_{j_{k(r)}})$ correspondent aux valeurs des variables intervenants dans la contrainte r.

Pour une instanciation $x = (x_1, \ldots, x_n) \in S$, on associe un paiement pour chaque joueur $i \in I$, défini par:

$$U_i(x_1, \ldots, x_n) = \sum_{r \in R(i)} k(r)\chi_r(x_{j_1}, \ldots, x_{j_{k(r)}}), \quad \forall i \in I, \quad (2)$$

On définit le jeu noncoopératif $\mathcal{G}(\mathcal{P})$ associé au problème de satisfaction de contraintes ($\mathcal{P}$) comme suit:

$$\mathcal{G}(\mathcal{P}) = < I, \{S_i\}_{i \in I}, \{U_i\}_{i \in I} >, \quad (3)$$

4 Le concept du Z-équilibre pour le jeu $\mathcal{G}(\mathcal{P})$

Le concept du Z-équilibre a été introduit par V.I. Zhukovski [14] pour les jeux différentiels.

Definition 3 (Z-Equilibre[14]) *Une issue $s^* \in S$ est un Z-équilibre du jeu (3), si:*

(a) s^* est un équilibre actif, ie $\forall i \in I$, $\forall s_i \in S_i$, $s_i \neq s_i^*$, il existe $t_{-i} \in S_{-i}$ telle que $U_i(s_i, t_{-i}) \leq U_i(s^*)$.
(b) $s^* \in S$ est un équilibre de Pareto, c-à-d il n'existe pas une autre issue $s \in S$ qui vérifie le système d'inégalités $U_i(s) \geq U_i(s^*)$, $\forall i \in I$, dont, au moins, une est stricte.

On distingue le Z-équilibre par les propriétés suivantes:

Propriétés 41 *[7] Soit $s^* = (s_i^*, s_{-i}^*) \in S$ un Z-équilibre du jeu (3).*

(a) La condition que s^* soit un équilibre actif garantit sa stabilité. En effet, si un joueur $i \in I$ opte pour une stratégie $s_i \in S_i$, $s_i \neq s_i^*$, le reste des joueurs peuvent toujours choisir une stratégie $t_{-i} \in S_{-i}$, telle que le gain obtenu par ce joueur dans la situation (s_i, t_{-i}) ne dépasse pas le gain qui lui est fourni par le Z-équilibre s^*.
(b) Le Z-équilibre est individuellement et collectivement rationnel.
(c) L'issue s^* étant un équilibre de Pareto, permet d'éviter le paradoxe de Tucker.

Le théorème suivant donne les conditions d'existence d'un Z-équilibre en stratégies pures dans un jeu fini sous forme normale.

Théorème 1 *Si pour tout joueur $i \in I$, l'ensemble de ses stratégies S_i est fini et non vide, alors le jeu (3) admet un Z-équilibre en stratégies pures.*

***Preuve* 1** *Pour un jeu fini, le gain de sécurité*

$$\alpha_i = \sup_{s_i \in S_i} \inf_{s_{-i} \in S_{-i}} U_i(s_i, s_{-i})$$

existe pour tout joueur $i \in I$.

Posons

$$A = \{s \in S, \quad U_i(s) \geq \alpha_i, \quad \forall i \in I\}.$$

Montrons que l'ensemble A est non vide. Notons par s_i^G la stratégie de sécurité du joueur $i \in I$, définie par la relation

$$\alpha_i = \sup_{s_i \in S_i} \inf_{s_{-i} \in S_{-i}} U_i(s_i, s_{-i}) = \inf_{s_{-i} \in S_{-i}} U_i(s_i^G, s_{-i}), \ i \in I.$$

Considérons l'issue du jeu où chaque joueur choisit sa stratégie de sécurité, ie $s^G = (s_1^G, \ldots, s_n^G) \in S$, avec $s_i^G \in S_i$, $\forall i \in I$. On a

$$U_i(s^G) = U_i(s_i^G, s_{-i}^G) \geq \inf_{s_{-i} \in S_{-i}} U_i(s_i^G, s_{-i}) = \sup_{s_i \in S_i} \inf_{s_{-i} \in S_{-i}} U_i(s_i, s_{-i}) = \alpha_i,$$

d'où

$$U_i(s^G) \geq \alpha_i, \quad \forall i \in I, \ et \qquad A \neq \emptyset.$$

Soit $\lambda = (\lambda_1, \ldots, \lambda_n)$ avec $\lambda_i \in]0, 1[$, $\forall i \in I$. Calculons

$$\sup_{s \in A} \sum_{i=1}^{n} \lambda_i U_i(s) = \sum_{i=1}^{n} \lambda_i U_i(s^*). \tag{4}$$

La borne sup *est atteinte en un point $s^* \in A$, puisque A est un ensemble fini.*

Montrons que s^* est un Z-équilibre:

Supposons que le joueur $i \in I$ change de stratégie, et choisit la stratégie $s_i \in S_i$, et notons par

$$t_{-i} \in \arg \inf_{v_{-i} \in S_{-i}} U_i(s_i, v_{-i})$$

une stratégie du reste des joueurs en réaction au changement de stratégie du joueur $i \in I$. On aura alors,

$$U_i(s_i, t_{-i}) = \inf_{v_{-i} \in S_{-i}} U_i(s_i, v_{-i}) \leq \sup_{s_i \in S_i} \inf_{v_{-i} \in S_{-i}} U_i(s_i, v_{-i}) = \alpha_i \leq U_i(s^*),$$

ce qui démontre que l'issue s^* est un équilibre actif.

Démontrons maintenant que $s^* \in S$ est un équilibre Pareto optimal. Supposons le contraire, i.e il existe une autre issue $\widetilde{s} \in S$ qui vérifie le système d'inégalités

$$U_i(\widetilde{s}) \geq U_i(s^*), \quad \forall i \in I,$$

dont, au moins, une est stricte. En multipliant chacune de ces inégalités par $\lambda_i \in]0,1[$, $i \in I$ et en faisant la somme, on déduit:

$$\sum_{i=1}^{n} \lambda_i U_i(\widetilde{s}) > \sum_{i=1}^{n} \lambda_i U_i(s^*),$$

ce qui contredit la relation (4) et démontre que s^* est Pareto optimal. Ayant prouvé que s^* est un équilibre actif, alors c'est aussi un Z-équilibre pour le jeu.

5 Algorithme de calcul du Z-équilibre du jeu (3)

Au cours de notre recherche bibliographique autour du concept du Z-équilibre, nous avons constaté que très peu d'auteurs se sont intéressés à cette notion. Cette minorité de travaux qui ont été effectués pour l'étude de cet équilibre ne proposent que sa définition, ses propriétés [15], ainsi que les conditions de son existence [16]. Jusqu'à présent, aucun algorithme n'a été proposé pour son calcul.

Par conséquent, nous nous somme inspirés de la preuve du théorème 1, établissant les conditions d'existence d'un Z-équilibre $s^* \in S$ pour le jeu $\mathcal{G}(\mathcal{P})$, afin de proposer un algorithme pour le calcul de cet équilibre.

Algorithme 1

Les étapes de cet algorithme consiste à :

(1) Calculer le gain de sécurité associé à chaque joueur $i \in I$, représentant le meilleur gain garanti possible, défini par

$$\alpha_i = \sup_{s_i \in S_i} \inf_{s_{-i} \in S_{-i}} U_i(s_i, s_{-i}), \ \forall i \in I.$$

(2) Déterminer l'ensemble

$$A = \{s \in S, \quad U_i(s) \geq \alpha_i, \quad \forall i \in I\},$$

qui représente l'ensemble des issues du jeu (3) fournissant à chacun des n joueurs un gain au moins égale à leurs gain de sécurité.

(3) Générer aléatoirement un vecteur $\lambda = (\lambda_1, \ldots, \lambda_n)$, $\lambda_i \in]0,1[$, $\forall i \in I$;

(4) Calculer $s^* \in S$ tel que

$$\sum_{i=1}^{n} \lambda_i U_i(s^*) = \sup_{s \in A} \sum_{i=1}^{n} \lambda_i U_i(s).$$

s^* représente le Z-équilibre recherché.

6 Equivalence entre la solution d'un CSP($\mathcal{P}$) et le Z-équilibre du jeu associé $\mathcal{G}(\mathcal{P})$

Sous l'hypothèse de la non vacuité de l'ensemble des solutions du CSP($\mathcal{P}$), l'ensemble des solutions de ($\mathcal{P}$) coincide avec l'ensemble des Z-équilibres du jeu $\mathcal{G}(\mathcal{P})$ associé :

Proposition 1 *Toute solution d'un CSP($\mathcal{P}$) est un Z-équilibre pour le jeu $\mathcal{G}(\mathcal{P})$ qui lui est associé.*

***Preuve* 2** *Soient $s^*=(s_1^*,\ldots,s_n^*)=(s_i^*,s_{-i}^*)\in S$ une solution au CSP($\mathcal{P}$) et $\mathcal{G}(\mathcal{P})$ le jeu qui lui est associé, défini par la relation* (3).

Par définition de la fonction utilité (2)*, on déduit que*

$$U_i(s^*)=\max_{s\in S}U_i(s)\geq U_i(s),\quad \forall s=(s_i,s_{-i})\in S,\quad \forall i\in I. \tag{5}$$

Par conséquent, pour tout $s_i\in S_i$, $s_i\neq s_i^*$ et pour tout $t_{-i}\in S_{-i}$, on aura

$$U_i(s^*)\geq U_i(s_i,t_{-i}),\quad \forall s_i\in S_i,\quad \forall i\in I, \tag{6}$$

d'où s^* est un équilibre actif du jeu $\mathcal{G}(\mathcal{P})$.

s^* est une solution du CSP($\mathcal{P}$), alors

$$U_i(s)\leq U_i(s^*),\quad \forall s\in S,\quad \forall i\in I. \tag{7}$$

Supposons maintenant que s^* n'est pas un équilibre de Pareto, i.e il existe $\widetilde{s}\in S$ telle que

$$U_i(\widetilde{s})\geq U_i(s^*);\quad \forall i\in I, \tag{8}$$

$$et\quad \exists j\in I/\ \ U_j(\widetilde{s})>U_j(s^*). \tag{9}$$

La relation (9) est en contradiction avec la relation (7). Par conséquent, s^* est un équilibre optimal de Pareto. Ayant déjà montré que s^* est un équilibre actif, on peut conclure que s^* est un Z-équilibre pour le jeu $\mathcal{G}(\mathcal{P})$.

Proposition 2 *Supposons que l'ensemble des solutions du CSP($\mathcal{P}$) n'est pas vide. Alors, tout Z-équilibre du jeu $\mathcal{G}(\mathcal{P})$, défini par la relation* (3)*, associé au CSP($\mathcal{P}$) est une solution du CSP ($\mathcal{P}$).*

***Preuve* 3** *Soit $s^*\in S$ un Z-équilibre du jeu $\mathcal{G}(\mathcal{P})$ associé à un CSP($\mathcal{P}$) et supposons que s^* n'est pas solution à ce CSP. Comme l'ensemble des solutions du CSP($\mathcal{P}$) n'est pas vide, alors posons $\tilde{s}\in S$ une de ses solutions. $\tilde{s}$ vérifie la relation* (5)*, i.e*

$$U_i(\tilde{s})=\max_{s\in S}U_i(s),\qquad \forall i\in I. \tag{10}$$

La supposition que s^* n'est pas solution du CSP($\mathcal{P}$), signifie que s^* ne vérifie pas un certain nombre de ses contraintes. Notons par $\overline{R}(s^*)$ l'ensemble des contraintes non vérifiées par s^* et considérons l'ensemble $\overline{S}(s^*)$ des variables intervenant dans les contraintes $r \in \overline{R}(s^*)$. Soit $i \in \overline{S}(s^*)$. On a:

$$\begin{aligned} U_i(\tilde{s}) = U_i(\tilde{s}_1, \ldots, \tilde{s}_n) &= \sum_{r \in R(i)} k(r)\chi_r(\tilde{s}_{j_1}, \ldots, \tilde{s}_{j_{k(r)}}) \\ &= \sum_{r \in R(i)\setminus\overline{R}(s^*)} k(r)\chi_r(\tilde{s}_{j_1}, \ldots, \tilde{s}_{j_{k(r)}}) + \sum_{r \in \overline{R}(s^*)\cap R(i)} k(r)\chi_r(\tilde{s}_{j_1}, \ldots, \tilde{s}_{j_{k(r)}}) \\ &> \sum_{r \in R(i)\setminus\overline{R}(s^*)} k(r)\chi_r(s^*_{j_1}, \ldots, s^*_{j_{k(r)}}) + \sum_{r \in \overline{R}(s^*)} k(r)\overbrace{\chi_r(s^*_{j_1}, \ldots, s^*_{j_{k(r)}})}^{=0} = U_i(s^*). \end{aligned}$$

Ainsi,

$$\forall j \in \overline{S}(s^*), \quad U_j(s^*) < U_j(\tilde{s}) = \max_{s \in S} U_j(s) \tag{11}$$

D'autre part, puisque s^* est un Z-équilibre, alors il est de Pareto, i.e pour $\tilde{s} \in S$ on aura soit

$$U_i(s^*) = U_i(\tilde{s}), \qquad \forall i \in I; \tag{12}$$

soit il existe un $k \in I$ tel que

$$U_k(\tilde{s}) < U_k(s^*). \tag{13}$$

La relation (12) est en contradiction avec (11). De même, la relation (13) ne peut avoir lieu. En effet, si $k \in \overline{R}(s^*)$, alors la relation (13) serait en contradiction avec la relation (11), puisque elle conduirait à l'absurdité suivante:

$$\max_{s \in S} U_k(s) \overbrace{=}^{(11)} U_k(\tilde{s}) < U_k(s^*) \leq \max_{s \in S} U_k(s).$$

D'où le résultat.

7 Application de l'algorithme 1 pour la résolution d'un CSP($\mathcal{P}$).

La possibilité de représenter le CSP($\mathcal{P}$) par le jeu noncoopératif $\mathcal{G}(\mathcal{P})$ et l'équivalence établie entre le Z-équilibre de ce jeu et la solution de ce CSP, nous a inspirée pour développer une procédure permettant de résoudre un problème de satisfaction de contraintes en passant par le calcul du Z-équilibre du jeu associé. Ceci, en intégrant l'algorithme 1 et en s'inspirant des approches par retour arrière connues pour la résolution des CSP.

Les étapes détaillées de cette procédure sont les suivantes:

Procédure de recherche du Z-équilibre

Entrées: CSP($\mathcal{P}$)=(X,D,C); $\lambda = (\lambda_1, \ldots, \lambda_n)$, $\sum_{i=1}^{n} \lambda_i = 1$, $\lambda_i \in]0,1[$.

Sorties: $x^* = (x_1^*, \ldots, x_n^*)$ représente le Z-équilibre recherché pour le jeu $\mathcal{G}(\mathcal{P})$ associé au CSP($\mathcal{P}$), qui correspond à une solution au CSP($\mathcal{P}$) en entrée.

Début

(1) $k \leftarrow 1$, $S \leftarrow \emptyset$, $G \leftarrow \emptyset$, $A \leftarrow \emptyset$.
(l'ensemble S sauvegarde les instanciations parcourues, et l'ensemble G sauvegarde les gains associés).

(2) Instancier les variables $x_i^{(k)}$ à tour de rôle, jusqu'à l'obtention d'une instanciation complète $x^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})$.

Si $(x^{(k)} \notin S)$ **alors**
$S \leftarrow S \cup \{x^{(k)}\}$, aller à l'étape (3).
sinon
faire une retour-arrière dans l'arborescence pour changer $x^{(k)}$.
fin.

(3) Évaluer l'instanciation $x^{(k)}$ avec la fonction

$$U_i(x^{(k)}) = \sum_{r \in R(i)} k(r) \chi_r(x_{j_1}^{(k)}, \ldots, x_{j_{k(r)}}^{(k)})$$

précédemment définie par la relation (2).
Si (toutes les branches de l'arborescence sont coupées) **alors**
Aller à l'étape (4).
sinon
$k \leftarrow k + 1$, retourner à l'étape (2).
fin.

(4) Parcourir l'ensemble G, pour déterminer le gain de sécurité α_i pour chaque joueur i, tel que

$$\alpha_i = \sup_{x_i \in X_i} \inf_{x_{-i} \in X_{-i}} U_i(x_1, \ldots, x_n).$$

(5) **Pour** j de 1 à k **faire**
Si $(U_i(x^{(j)}) \geq \alpha_i, \forall i \in I)$ **alors**
$A \leftarrow A \cup \{x^{(j)}\}$.
Évaluer la fonction: $f_j(x^{(j)}, \lambda) = \sum_{i=1}^{n} \lambda_i U_i(x^{(j)})$.
fin.
fin pour.

(6) **Si** $(A = \emptyset)$ **alors**
le jeu $\mathcal{G}(\mathcal{P})$ n'admet pas de Z-équilibre
sinon
Poser $j^* = \arg(\max_{j \in \{1,\ldots,k\}} f_j(x^{(j)}, \lambda))$.
$x^* \leftarrow x^{(j^*)}$.
fin.

(7) Retourner x^*.

Fin de la procédure.

---

8 Tests et résultats obtenus avec la procédure proposée

La procédure proposé à été implémenté puis testé sur deux problèmes classiques de satisfaction de contraintes:

(a) ***Le problème de k-coloration d'un graphe:*** ayant un graphe et un ensemble de k couleurs, le problème consiste à affecter une couleur pour chaque noeud du graphe, de façon à ce que deux noeuds adjacents n'aient pas la même couleur.

(b) ***Le problème des N reines:*** ce problème consiste à placer N reines sur un échiquier à N lignes et N colonnes, de manière à ce qu'aucune reine ne soit en prise avec une autre. Deux reines sont en prise, si elles se trouvent sur une même diagonale, une même ligne ou une même colonne de l'échiquier.

Les résultats obtenus pour le problème de k-coloration sont résumés dans la figure suivante:

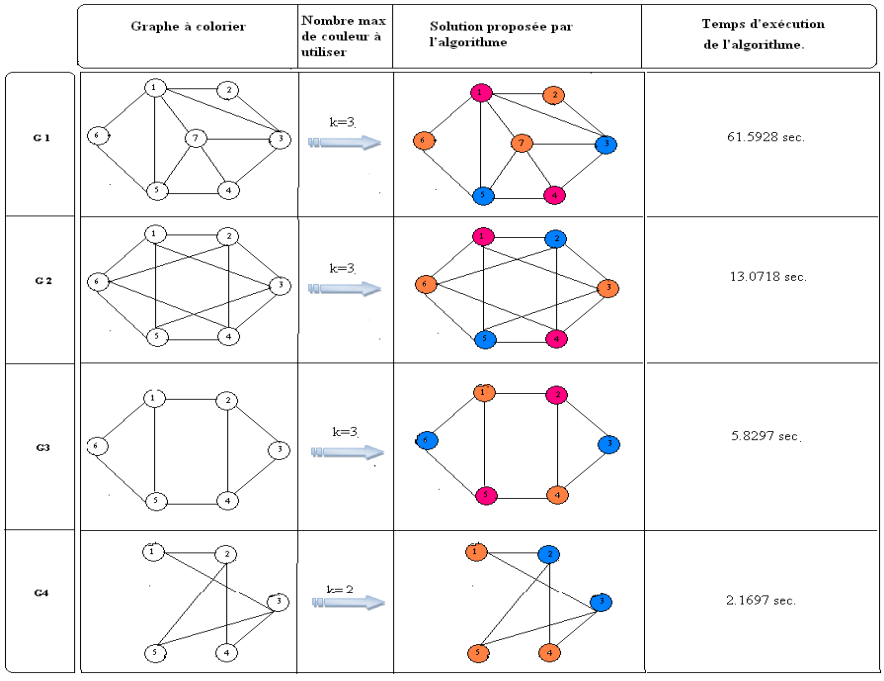


Fig. 1. Résultats obtenus pour le problème de k-coloration d'un graphe.

Le modèle CSP($\mathcal{P}$) du problème de k-coloration du graphe $G4$ est le suivant:

- $X = \{X_1, X_2, X_3, X_4, X_5\}$, où X_i représente la couleur affectée au noeud i du graphe $G4$.
- $D_i = \{\text{orange } (O), \text{Bleu } (B)\}$, $i = \overline{1,5}$.
- $C = \{X_1 \neq X_2, X_1 \neq X_3, X_2 \neq X_4, X_2 \neq X_5, X_3 \neq X_4, X_3 \neq X_5\}$.

L'instanciation $s^* = (O, B, B, O, O)$ vérifie toutes les contraintes de l'ensemble C, d'où s^* est une solution au CSP($\mathcal{P}$).

Considérons maintenant le jeu $\mathcal{G}(\mathcal{P})$ associé au CSP($\mathcal{P}$), défini par:

- l'ensemble des joueurs: $I = \{1, 2, 3, 4, 5\}$;
- l'ensemble des stratégies: $S_i = \{\text{orange } (O), \text{Bleu } (B)\}$ de chaque joueur $i \in I$.

Une stratégie d'un joueur i représente une couleur que ce joueur peut affecter au noeud i.

Toutes les contraintes du CSP($\mathcal{P}$) sont binaires, alors $k(r) = 2$, $\forall r \in C$. Par conséquent, la fonction utilité définie par la relation (2) pour un joueur $i \in I$, s'écrit:

$$U_i(x_1, \ldots, x_8) = \sum_{r \in R(i)} 2\chi_r(x_i, x_j), \tag{14}$$

où

$$\chi_r(x_i, x_j) = \begin{cases} 1, \text{ si } (x_i, x_j) \in r, \\ 0, \text{ sinon.} \end{cases} \tag{15}$$

A partir de l'ensemble des contraintes C, on a:

$$R(1) = 2 \Rightarrow \max_{s \in S} U_1(s) = 2; \qquad R(2) = 3 \Rightarrow \max_{s \in S} U_2(s) = 3;$$
$$R(3) = 3 \Rightarrow \max_{s \in S} U_3(s) = 3; \qquad R(4) = 2 \Rightarrow \max_{s \in S} U_4(s) = 2);$$
$$R(5) = 2 \Rightarrow \max_{s \in S} U_5(s) = 2.$$

Pour l'issue $s^* = (O, B, B, O, O)$, on a $\chi_r(x_i, x_j) = 1$, $\forall r \in R(i)$, $\forall i \in I$, d'où $U_i(s^*) = \max_{s \in S} U_i(s)$, $\forall i \in I \Leftrightarrow U(s^*) = (2, 3, 3, 2, 2)$.

De là, il s'en suit que cette solution représente un équilibre de Nash, puisqu'aucun joueur ne peut améliorer son utilité en changeant seul sa stratégie, c'est aussi un équilibre Pareto-optimal, du moment qu'aucun joueur ne peut améliorer son gain sans détériorer celui d'un autre joueur. De plus, l'équilibre de Nash est un équilibre actif, ce qui confirme que cette solution est effectivement un Z-équilibre pour le jeu $\mathcal{G}(\mathcal{P})$.

Pour le problème des n-reines, nous avons obtenu les résultats suivants:

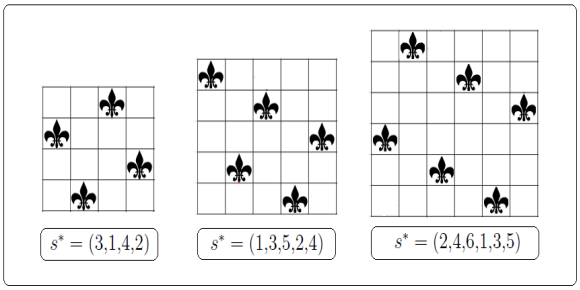


Fig. 2. Résultats obtenus pour le problème des n-reines, avec $n = 4$, $n = 5$ et $n = 6$.

Nombre de reines à positionner (n)	4	5	6
Taille de l'espace de recherche	256	3125	46656
Temps d'exécution (sec)	0.9375	108.1913	7h29mn54s

L'analyse de ce tableau révèle que le temps d'exécution de cette procédure s'accroît considérablement en fonction du nombre de reines à positionner. En effet, il suffit que ce nombre dépasse 5 reines pour que le temps d'exécution devienne trop important.

Cette grande complexité est justifiée par le fait que cette procédure parcourt tout l'ensemble des instanciations complètes du CSP en entrée, ce qui n'est pas avantageux par rapport à sa résolution, ce qui nous amène à réfléchir à la possibilité que la solution obtenue possède certaines caractéristiques par rapport au problème CSP initial.

9 Conclusion

Dans ce travail, nous avons formulé une représentation des problèmes de satisfaction de contraintes par un jeu noncoopératif à n joueurs. Nous avons établi la relation entre les solutions du CSP et le Z-équilibre du jeu associé. Enfin, nous avons proposé un algorithme pour le calcul du Z-équilibre, puis nous l'avons appliquée à la résolution d'un problème de satisfaction de contraintes, en proposant une procédure intégrant les étapes de cet algorithme, et s'inspirant des approches par retour-arrière, utilisées pour la résolution des CSP. Les tests effectués sur cette procédure révèlent que les résultats obtenus vérifient toutes les contraintes du CSP initial, et sont effectivement des Z-équilibre pour le jeu associé. Cependant, elle présente une importante complexité en terme de temps de calcul, d'où la nécessité d'apporter quelques améliorations et d'étudier les caractéristiques des solutions fournies par cette procédure par rapport au problème initial du CSP.

References

1. F. Ricci (1990). Equilibrium Theory and Constraint Networks. International Conference on Game Theory, Florence, Italy.
2. P.G. Kolaitis and M.Y. Vardi (2000). A game theoretic approach to constraint satisfaction. American Association for Artificial Intelligence Proceedings (AAAI'2000), 175-181.
3. R. A. Krzysztof and F. Rossi and K. B. Venable (2008). Comparing the notions of optimality in strategic games and soft constraints. Annals of Mathematics and Artificial Intelligence, 52: 25-54.
4. M.Jiang (2007). Finding pure nash equilibrium of graphical game via constraints satisfaction approach. LNCS, 4614: 483-494.
5. L. Bordeaux and B. Pajot (2005). Computing equilibria using interval constraints. Joint ERCIM/CoLogNet International Workshop on Constraint Solving and Constraint Logic Programming Proceedings (CSCLP'2004), LNCS, 3419: 157-171.
6. R. Porter and E. Nudelman and Y. Shoham (2008). Simple search methods for finding a Nash equilibrium. Games and Economic Behaviour 64: 642-662.
7. A. Ferhat and M. S. Radjef (2008). Existence Conditions of a Zm-Equilibrium for Multicriteria Games. 13-th International Symposium on Dynamic Games and Applications, Wroclaw, Poland.
8. G. Gosti and S. Bistarelli (2009). Solving CSP with naming games. LNCS, 5655: 16-32.
9. R.Apt. Krzysztof and F.Rossi and K.B. Venable (2008). A comparaison of the notion of optimality in soft constraints and graphical games. LNAI, 5129: 1-16.
10. P. Borm and H. Hamers and R. Hendrickx (2001). Operations Research Games: A Survey. Société de la Statistique et d'Investigation Opérante, 9: 139-216.
11. M.S. Radjef (2010). Cours sur la théorie des jeux et l'optimisation multicritère, Université de Béjaia.
12. L. Paris (2007). Approches pour les problèmes SAT et CSP: ensembles strong backdoor, voisinage consistant et forme normale généralisée. Thèse de doctorat, Université de Provence.
13. F. Rossi and P.V. Beek and T. Walsh (2006). Handbook of Constraint Programming. Elsevier.
14. E.M. Vaisbord and V.I. Zhukovskii (1988). Introduction to Multi-Player Differential Games and Their Applications. Gordon et Breach Science Publishers.
15. S. Gaidov (1993). Z-equilibria in many-player stochastic differential games. Archivum Mathematicum (BRNO), 29: 123-133.
16. V. I. Zhukovskii and A. A. Tchikry (1994). Linear Quadratic differential games. Naoukova Doumka.
17. D. Fudenberg and J. Tirole (1993). Game theory. The MIT Press.

Generalized Fritz-John optimality in nonlinear programming in the presence of nonlinear equality and inequality constraints

Hachem Slimani[1] and Mohammed Said Radjef [2]

Laboratory of Modeling and Optimization of Systems (LAMOS)
Computer Science Department [1], Operational Research Department [2]
University of Bejaia, 06000 Bejaia, Algeria,
haslimani@gmail.com [1], radjefms@gmail.com [2]

Abstract. In this paper, we study Fritz-John type optimality conditions for constrained nonlinear programming in which nonlinear equality and inequality constraints are together present. We introduce a generalized Fritz-John condition which is necessary and sufficient for a feasible point to be an optimal solution under weak invexity with respect to different $(\eta_i)_i$. Moreover, it is shown with examples that the generalized Fritz-John condition combining with the invexity with respect to different $(\eta_i)_i$ are especially easy in application and useful in the sense of sufficient optimality conditions. Furthermore, some results from literature can be derived as particular cases from the obtained results.
Keywords: Nonlinear programming; Weak pseudo-invexity; Generalized Fritz-John condition; Equality and inequality constraints; Optimality.

1 Introduction

Optimality criteria in mathematical programming are important both theoretically as well as computationally and can be formulated in various different forms. In general, optimality criteria serve as a basis to develop computational procedures. The best-known necessary optimality criterion for a constrained mathematical programming problem is due to Karush [18] and Kuhn-Tucker [21]. However, the F. John criterion [17], known in the literature under the complete name Fritz-John criterion, is in a sense more general. From the Fritz-John criterion, the Karush-Kuhn-Tucker one is obtained by adding an assumption which is to impose a suitable constraint-qualification [22] on the constraints of the optimization problem. Moreover, the Fritz-John criterion itself can be used to derive a form of constraint qualification for the Karush-Kuhn-Tucker criterion [8, 23]. Thus, in case of necessary optimality criteria, the only restriction on a constrained program is that the constraints should satisfy certain qualification but for sufficient optimality criteria and duality results to hold, the objective and constraints functions are required to satisfy some convexity or generalized convexity requirements, see, for example, Bazaraa et al. [3] and Mangasarian [22].

Several classes of functions have been defined for the purpose of weakening the hypothesis of convexity in mathematical programming. Hanson [14] introduced the concept of invexity for the differentiable functions, generalizing the difference $(x - x_0)$ in the definition of convex function to any function $\eta(x, x_0)$. He proved that if, in a mathematical programming problem, instead of the convexity assumption, the objective and constraint functions are invex with respect to the same vector function η, then both the sufficiency of Karush-Kuhn-Tucker conditions and weak and strong Wolfe duality still hold. Further, Ben Israel and Mond [5] considered a class of functions called pre-invex and also showed that the class of invex functions is equivalent to the class of functions whose stationary points are global minima, see also Craven and Glover [10]. Hanson and Mond [15] introduced two other classes of functions called type I and type II functions for the scalar optimization problem, which were further generalized to pseudo-type I and quasi-type I by Rueda and Hanson [30] and sufficient optimality conditions are obtained involving these functions. Further properties and applications of invexity for some more general problems were studied by Antczak [1, 2], Bector et al. [4], Bhatia and Sharma [6], Craven [9], Fulga and Preda [12], Jeyakumar and Mond [16], Kaul and Kaur [19], Kaul et al. [20], Martin [24], Pini and Singh [29], Osuna-Gomez et al. [28], Mishra et al. [26], Soleimani-damaneh and Sarabi [37], Stancu-Minasian [38], Suneja et al. [40] and others.

However, one major difficulty in this extension of convexity is that invex problems require a same function $\eta(x, x_0)$ for the objective and constraint functions. This requirement turns out to be a major restriction in applications. In [33], a constrained nonlinear programming is considered and KT-invex, weakly KT-pseudo-invex and type I problems with respect to different $(\eta_i)_i$ are defined (each function occurring in the studied problem is considered with respect to its own function η_i instead of a same function η). A new Karush-Kuhn-Tucker type necessary condition is introduced for nonlinear programming problems and duality results are obtained, for Wolfe and Mond-Weir type dual programs, under generalized invexity assumptions. In [34], the invexity with respect to different $(\eta_i)_i$ is used in the nondifferentiable case. Fritz-John type necessary, Karush-Kuhn-Tucker type necessary and sufficient optimality conditions and duality results are obtained for nondifferentiable multiobjective programming. See also [35, 36].

In parallel to all these developments and advances of the invexity and its extensions in theory, some applications in practice begin to take place. Recently, Dinuzzo et al. [11] have obtained some kernel function in Machine Learning which is not quasi-convex (and hence also neither convex nor pseudoconvex) but it is invex. Nickisch and Seeger [27] have studied a multiple kernel learning problem and have used the invexity to deal with the optimisation which is non convex.

In this paper, we study Fritz-John type optimality for constrained nonlinear programming by considering the case with inequality and equality constraints together. New necessary and sufficient optimality conditions for a feasible point to be an optimal solution are obtained under weak invexity with respect to different $(\eta_i)_i$. We prove that the obtained sufficient optimality conditions are easy

and useful in application especially for some nonconvex nonlinear programming programs for which many results in the literature are not applicable. For illustration, some examples and particular cases of the obtained results will be given.

2 Preliminaries and definitions

Invex functions were introduced to optimization theory by Hanson [14], and called by Craven [9], as a very broad generalization of convex functions.

Definition 1. *[14] Let D be a nonempty open set of $\mathbb{R}^n$ and $\eta : D \times D \to \mathbb{R}^n$ be a vector function. A function $f : D \to \mathbb{R}$ is said to be (def) at $x_0 \in D$ with respect to η, if the function f is differentiable at x_0 and for each $x \in D$, (cond) holds.*

(i) def: invex,
cond:

$$f(x) - f(x_0) \geq [\nabla f(x_0)]^t \eta(x, x_0). \tag{1}$$

(ii) def: pseudo-invex,
cond:

$$[\nabla f(x_0)]^t \eta(x, x_0) \geq 0 \Rightarrow f(x) - f(x_0) \geq 0. \tag{2}$$

(iii) def: quasi-invex,
cond:

$$f(x) - f(x_0) \leq 0 \Rightarrow [\nabla f(x_0)]^t \eta(x, x_0) \leq 0. \tag{3}$$

If the inequality in (1) (resp. second (implied) inequality in (3)) is strict ($x \neq x_0$), we say that f is strictly invex (resp. strictly quasi-invex) at x_0 with respect to η. f is said to be (strictly) invex (resp. pseudo-invex or (strictly) quasi-invex) on D with respect to η, if f is (strictly) invex (resp. pseudo-invex or (strictly) quasi-invex) at each $x_0 \in D$ with respect to the same η.

Remark 1. When the function $\eta(x, x_0) = x - x_0$, the definition of (strict) invexity (resp. pseudo-invexity and quasi-invexity) reduces to the definition of (strict) convexity (resp. pseudo-convexity and quasi-convexity).

Craven and Glover [10] and Ben-Israel and Mond [5] stated that the class of invex functions are all those functions whose stationary points are global minima. Moreover, Ben-Israel and Mond [5] proved that the classes of invex and pseudo-invex functions coincide and every function f, with $\nabla f \not\equiv 0$, is invex.

Proposition 1. *[5] Any differentiable function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ at a point $x_0 \in D$, with $\nabla f(x_0) \neq 0$, is invex at x_0 with respect to $\eta(x, x_0) = [f(x) - f(x_0)] \frac{[\nabla f(x_0)]}{[\nabla f(x_0)]^t [\nabla f(x_0)]}$, $\forall\ x \in D$.*

Now, we give others η for which a given scalar function is pseudo-invex.

Proposition 2. *Any differentiable function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ at a point $x_0 \in D$, with $\nabla f(x_0) \neq 0$, is pseudo-invex at x_0 with respect to $\eta(x, x_0) = [f(x) - f(x_0)][\nabla f(x_0)]$, $\forall\ x \in D$ or $\eta(x, x_0) = [f(x) - f(x_0)]t(x_0)$, $\forall\ x \in D$ where $t(x_0) \in \mathbb{R}^n$ with $t_i(x_0) = \begin{cases} 1, & \text{if } \frac{\partial f}{\partial x_i}(x_0) \geq 0, \\ -1, & \text{otherwise,} \end{cases}$ for all $i = 1, ..., n$.*

Example 1. The function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x) = x_2^2 - 3x_1$ is pseudo-invex on $\mathbb{R}^2$ with respect to $\eta(x, \tilde{x}) = (x_2^2 - 3x_1 - \tilde{x}_2^2 + 3\tilde{x}_1)(-3, 2\tilde{x}_2)^t$. Furthermore, f is pseudo-invex on $\mathbb{R} \times \mathbb{R}_+$ with respect to $\eta_1(x, \tilde{x}) = (x_2^2 - 3x_1 - \tilde{x}_2^2 + 3\tilde{x}_1)(-1, 1)^t$ and it is pseudo invex on $\mathbb{R} \times (\mathbb{R}_- \backslash \{0\})$ with respect to $\eta_2(x, \tilde{x}) = (x_2^2 - 3x_1 - \tilde{x}_2^2 + 3\tilde{x}_1)(-1, -1)^t$.

In Slimani and Radjef [33], a new concept of weak KT-pseudo-invexity is introduced and duality results have been obtained for a constrained nonlinear programming. Now, we define a concept of weak pseudo-invexity for scalar functions given as follows.

Definition 2. *Let D be a nonempty open set of $\mathbb{R}^n$ and $\eta : D \times D \to \mathbb{R}^n$ be a vector function. A function $f : D \to \mathbb{R}$ is said to be weakly pseudo-invex at $x_0 \in D$ with respect to η, if the function f is differentiable at x_0 and for each $x \in D$:*

$$f(x) - f(x_0) < 0 \Rightarrow \exists\ \bar{x} \in D,\ [\nabla f(x_0)]^t \eta(\bar{x}, x_0) < 0. \tag{4}$$

f is said to be weakly pseudo-invex on D with respect to η, if f is weakly pseudo-invex at each $x_0 \in D$ with respect to the same η.

Remark 2. In the definition 2, if $\bar{x} = x$, we obtain the pseudo-invexity of scalar function given in the definition 1.

If a function f is pseudo-invex at x_0 with respect to η, then it is weakly pseudo-invex at x_0 with respect to the same η (take $\bar{x} = x$). However, if f is weakly pseudo-invex at x_0 with respect to η, then f may not be pseudo-invex at x_0 with respect to the same η but it will be pseudo-invex at x_0 with respect to $\tilde{\eta}$ with $\tilde{\eta}(x, x_0) = \eta(\bar{x}, x_0)$, $\forall\ x \in D$. Note that also, we can use the proposition 2 to obtain a function $\hat{\eta}$ for which f to be pseudo-invex (and then weakly pseudo-invex) at x_0. Thus the classes of pseudo-invex functions and weakly pseudo-invex functions coincide.

Example 2. • The function $f_1 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_1(x) = -x_1^2 - x_2$ is weakly pseudo-invex at $x_0 = (1, 0)$ with respect to $\eta_1(x, x_0) = (x_0 - x) \in \mathbb{R}^2$ (take $\bar{x} = [(f_1(x) - f_1(x_0), f_1(x) - f_1(x_0))^t + x_0] \in \mathbb{R}^2$). But f_1 is not pseudo-invex at x_0 with respect to the same η_1 because for $x = (2, -1)$, $f_1(x) - f_1(x_0) < 0$ and $[\nabla f_1(x_0)]^t \eta_1(x, x_0) > 0$. However, f_1 is pseudo-invex at x_0 with respect to $\tilde{\eta}_1(x, x_0) = (f_1(x_0) - f_1(x), f_1(x_0) - f_1(x))^t$. Note that if we use the proposition 2, we obtain the same $\tilde{\eta}_1$ for which f_1 is pseudo-invex (and then weakly pseudo-invex) at x_0.
• The function $f_2 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_2(x) = x_1 + \sin x_2$ is weakly pseudo-invex at $x_0 = (\frac{\pi}{6}, \frac{\pi}{3})$ with respect to $\eta_2(x, x_0) = x - x_0 \in \mathbb{R}^2$ (take

$\bar{x} = (f_2(x) - f_2(x_0), f_2(x) - f_2(x_0))^t \in \mathbb{R}^2$). But f_2 is not pseudo-invex at x_0 with respect to the same η_2 because for $x = (\frac{\pi}{3}, 0)$, $f_2(x) - f_2(x_0) < 0$ and $[\nabla f_2(x_0)]^t \eta_2(x, x_0) = 0$. However, f_2 is pseudo-invex at x_0 with respect to $\tilde{\eta}_2(x, x_0) = (f_2(x) - f_2(x_0), f_2(x) - f_2(x_0))^t - x_0$. Note that if we use the proposition 2, we obtain $\hat{\eta}_2(x, x_0) = (f_2(x) - f_2(x_0))(1, 1)^t$ for which f_2 is pseudo-invex (and then weakly pseudo-invex) at x_0.

Consider the following constrained nonlinear programming problem in the presence of nonlinear equality and inequality constraints (NP):

$$(NP) \quad \begin{array}{l} \text{Minimize } f(x), \\ \text{subject to } g_j(x) \leq 0,\ j = 1, ..., k, \\ \qquad\qquad h_i(x) = 0,\ i = 1, ..., m, \end{array}$$

where $f, g_j, h_i : D \to \mathbb{R},\ j = 1, ..., k,\ i = 1, ..., m,\ D$ is an open set of $\mathbb{R}^n$; $X = \{x \in D : g_j(x) \leq 0,\ j = 1, ..., k,\ h_i(x) = 0,\ i = 1, ..., m\}$ is the set of feasible solutions for (NP). For $x_0 \in D$, we denote $J(x_0) = \{j \in \{1, ..., k\} : g_j(x_0) = 0\}$, $J_0 = |J(x_0)|$ is the cardinal of the set $J(x_0)$, g_J is the semi-vector of g composed of the active constraints at the point x_0.

Now, before establishing optimality conditions for (NP), we give the following simple propositions that we will use.

Proposition 3. *Let S be a nonempty subset of $\mathbb{R}^n$. If a function $\varphi : S \to]-\infty, 0]$ is strictly quasi-invex at $x_0 \in S$ with respect to $\theta : S \times S \to \mathbb{R}^n$ and $\varphi(x_0) = 0$, then $[\nabla\varphi(x_0)]^t\theta(x, x_0) < 0,\ \forall\ x \in S$.*

Proof. For $x \in S$, we have $\varphi(x) \leq 0 = \varphi(x_0)$, which by strict quasi-invexity of φ at x_0 with respect to θ implies $[\nabla\varphi(x_0)]^t\theta(x, x_0) < 0$.

Corollary 1. *Let $x_0 \in X$ be a feasible solution of (NP). For each $j \in J(x_0)$ (resp. $i = 1, ..., m$), if g_j (resp. h_i) is strictly quasi-invex at x_0 with respect to θ_j(resp. ϕ_i) : $X \times X \to \mathbb{R}^n$, then $[\nabla g_j(x_0)]^t\theta_j(x, x_0) < 0,\ \forall\ x \in X$ (resp. $[\nabla h_i(x_0)]^t\phi_i(x, x_0) < 0,\ \forall\ x \in X$).*

Proposition 4. *Let x_0 be a feasible solution of (NP). For each $j \in \{1, ..., k\}$, if $\nabla g_j(x_0) \neq 0$ and the components of $\theta_j : X \times X \to \mathbb{R}^n$ are defined by $\theta_j^l(x, x_0) = \begin{cases} g_j(x) - \delta, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0) \geq 0, \\ -g_j(x) + \delta, & \text{otherwise,} \end{cases}$ for all $l = 1, ..., n$ with $\delta \in \mathbb{R},\ \delta > 0$, then $[\nabla g_j(x_0)]^t\theta_j(x, x_0) < 0,\ \forall\ x \in X$.*

Proof. We have $[\nabla g_j(x_0)]^t\theta_j(x, x_0) = \sum_{l=1}^{n} \frac{\partial g_j}{\partial x_l}(x_0) s_l^j(x_0)[g_j(x) - \delta] < 0,\ \forall\ x \in X$, $\forall\ j \in \{1, ..., k\}$ with $s_l^j(x_0) = \begin{cases} 1, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0) \geq 0, \\ -1, & \text{otherwise,} \end{cases}$ for all $l = 1, ..., n$.

Remark 3. The proposition 4 is also true for the constraint functions $h_i,\ i = 1, ..., m$.

3 Optimality conditions

In this section, we establish Fritz-John type necessary and sufficient optimality conditions for a feasible point to be an optimal solution of (NP) under weak pseudo-invexity. The obtained results treat equalities as equalities and do not transform them to inequalities. Thus the equalities and inequalities are handled easily together.

In the following theorem, we establish a Fritz-John type necessary condition for (NP).

Theorem 1. *(Fritz-John type necessary optimality condition) Suppose that*

(i) x_0 *is a local or global solution for (NP);*
(ii) the functions $f,\ g_j,\ j \in J(x_0),\ h_i,\ i = 1, ..., m$ *are differentiable at* x_0.

Then there exist vector functions $\eta : D \times D \to \mathbb{R}^n,\ \theta_j : D \times D \to \mathbb{R}^n,\ j \in J(x_0),\ \phi_i : D \times D \to \mathbb{R}^n,\ i = 1, ..., m$ $(\eta \not\equiv 0,\ \theta_j \not\equiv 0,\ \forall\ j \in J(x_0),\ \phi_i \not\equiv 0,\ \forall\ i = 1, ..., m)$ *and a vector* $(\mu, \lambda, \delta) \in \mathbb{R}_+^{1+J_0+m},\ (\mu, \lambda, \delta) \neq 0$ *such that* $(x_0, \mu, \lambda, \delta, \eta, (\theta_j)_j, (\phi_i)_i)$ *satisfies the following generalized Fritz-John condition*

$$\mu[\nabla f(x_0)]^t \eta(x, x_0) + \sum_{j \in J(x_0)} \lambda_j [\nabla g_j(x_0)]^t \theta_j(x, x_0) + \sum_{i=1}^{m} \delta_i [\nabla h_i(x_0)]^t \phi_i(x, x_0) \geq 0,\ \forall\ x \in D. \tag{5}$$

Proof. Suppose that x_0 is a local solution of (NP). Then there exists, a neighborhood of x_0, $v(x_0) \subset X$ such that for all $x \in v(x_0),\ f(x) - f(x_0) \geq 0$. Thus, it suffices to take $\eta,\ \theta_j,\ j \in J(x_0),\ \phi_i,\ i = 1, ..., m,\ \mu,\ \lambda_j,\ j \in J(x_0)$ and $\delta_i,\ i = 1, ..., m$ as follows:

- $$\eta(x, x_0) = \begin{cases} [f(x) - f(x_0)]t(x_0), & \text{if } x \in v(x_0), \\ t(x_0), & \text{if } x \in D \setminus v(x_0), \end{cases} \tag{6}$$

  with $t(x_0) \in \mathbb{R}^n$ and $t_l(x_0) = \begin{cases} 1, & \text{if } \frac{\partial f}{\partial x_l}(x_0) \geq 0, \\ -1, \text{ otherwise}, \end{cases}$ for all $l = 1, ..., n$;
- $$\theta_j(x, x_0) = \begin{cases} -g_j(x) s^j(x_0), & \text{if } x \in X, \\ s^j(x_0), & \text{if } x \in D \setminus X, \end{cases} \tag{7}$$

  with $s^j(x_0) \in \mathbb{R}^n$ and $s_l^j(x_0) = \begin{cases} 1, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0) \geq 0, \\ -1, \text{ otherwise}, \end{cases}$ for all $l = 1, ..., n$;
- $$\phi_i(x, x_0) = q^i(x_0),\ x \in D, \tag{8}$$

  with $q^i(x_0) \in \mathbb{R}^n$ and $q_l^i(x_0) = \begin{cases} 1, & \text{if } \frac{\partial h_i}{\partial x_l}(x_0) \geq 0, \\ -1, \text{ otherwise}, \end{cases}$ for all $l = 1, ..., n$;
- $\mu = 1$;
- $\lambda_j = \frac{1}{J_0}$, for all $j \in J(x_0)$;

- $\delta_i = \frac{1}{m}$, for all $i = 1, ..., m$.

If x_0 is a global solution of (NP), we replace $v(x_0)$ by X in the relation (6). Note that, if $\nabla f(x_0) \neq 0,\ \nabla g_j(x_0) \neq 0,\ j \in J(x_0)$ and $\nabla h_i(x_0) \neq 0,\ i = 1, ..., m$, we can replace in the relation (6) $t(x_0)$ by $\nabla f(x_0)$, in the relation (7) $s^j(x_0)$ by $\nabla g_j(x_0)$ and in the relation (8) $q^i(x_0)$ by $\nabla h_i(x_0)$.

Remark 4. If we redefine the feasible set of (NP) as: $X = \{x \in X^0 \subset D : g_j(x) \leq 0,\ j = 1, ..., k,\ h_i(x) = 0,\ i = 1, ..., m\}$ such that X^0 is a convex set with a nonempty interior (X^0 is not necessary open) and h have continuous first partial derivatives at x_0 and, if furthermore, $\theta_j(x, x_0) = \phi_i(x, x_0) = \eta(x, x_0) = x - x_0,\ \forall\ j \in J(x_0),\ \forall\ i = 1, ..., m$, then the theorem 1 reduce to the minimum-principle necessary optimality theorem of Mangasarian [22, p.168].

Using the generalized Fritz-John condition (5), we prove the following sufficient optimality conditions for a feasible solution to be optimal for (NP) under weak pseudo-invexity.

Theorem 2. *Let $x_0 \in X$ and suppose that:*

(i) f is weakly pseudo-invex at x_0 with respect to $\eta : X \times X \to \mathbb{R}^n$;
(ii) g_J is differentiable at x_0 and for all $j \in J(x_0)$, there exists a function $\theta_j : X \times X \to \mathbb{R}^n$ such that $[\nabla g_j(x_0)]^t \theta_j(x, x_0) < 0,\ \forall\ x \in X$;
(iii) h is differentiable at x_0 and for all $i = 1, ..., m$, there exists a function $\phi_i : X \times X \to \mathbb{R}^n$ such that $[\nabla h_i(x_0)]^t \phi_i(x, x_0) < 0,\ \forall\ x \in X$.

If there exists a vector $(\mu, \lambda, \delta) \in \mathbb{R}_+^{1+J_0+m},\ (\mu, \lambda, \delta) \neq 0$ such that the generalized Fritz-John condition (5) is satisfied for all $x \in X$, ie:

$$\mu[\nabla f(x_0)]^t \eta(x, x_0) + \sum_{j \in J(x_0)} \lambda_j [\nabla g_j(x_0)]^t \theta_j(x, x_0) + \sum_{i=1}^{m} \delta_i [\nabla h_i(x_0)]^t \phi_i(x, x_0) \geq 0,\ \forall\ x \in X, \tag{9}$$

then the point x_0 is an optimal solution of (NP).

Proof. Let us suppose that x_0 is not an optimal solution of (NP). Then there exists a feasible point $x \in X$ such that $f(x) - f(x_0) < 0$.
Since f is weakly pseudo-invex at x_0 with respect to η, it follows that

$$\exists\ \bar{x} \in X,\ [\nabla f(x_0)]^t \eta(\bar{x}, x_0) < 0. \tag{10}$$

By hypothesis, we have

$$[\nabla g_j(x_0)]^t \theta_j(\bar{x}, x_0) < 0,\ \forall\ j \in J(x_0), \tag{11}$$

and

$$[\nabla h_i(x_0)]^t \phi_i(\bar{x}, x_0) < 0,\ \forall\ i = 1, ..., m. \tag{12}$$

As $(\mu, \lambda, \delta) \geq 0,\ (\mu, \lambda, \delta) \neq 0$ and from (10), (11) and (12), it follows that

$$\mu[\nabla f(x_0)]^t\eta(\bar{x},x_0)+\sum_{j\in J(x_0)}\lambda_j[\nabla g_j(x_0)]^t\theta_j(\bar{x},x_0)+\sum_{i=1}^{m}\delta_i[\nabla h_i(x_0)]^t\phi_i(\bar{x},x_0)<0,$$

which contradicts (9), and therefore, x_0 is an optimal solution of (NP).

Remark 5. Note that we have not used any alternative theorem to prove the Fritz-John type necessary and sufficient optimality conditions (theorems 1 and 2), unlike to the usual procedure used in the literature where alternative theorems (Gordan, Motzkin, etc.) are used to prove Fritz-John type necessary and sufficient optimality conditions for nonlinear and multiobjective programming problems, see for example Bazaraa et al. [3], Bhatt and Misra [7], Mangasarian [22], Mangasarian and Fromovitz [23], Skarpness and Sposito [32] and Still and Streng [39].

Although the classical Fritz-John necessary optimality condition is more reduced than the generalized Fritz-John necessary optimality condition, this latter, combining with the invexity with respect to different $(\eta_i)_i$, has its usefulness in the sufficient optimality conditions. For illustration, in the following example, we consider a feasible point x_0 which is not a Karush-Kuhn-Tucker stationary point of problem and hence all the sufficient optimality conditions using this concept are not applicable to conclude on its optimality. Furthermore, we show that there exists no function η for which the objective and constraint functions are both (generalized) invex. Thus, we can not also use the sufficient optimality conditions using the (generalized) invexity with respect to a same η (in particular for $\eta(x,x_0)=x-x_0$) and the classical Fritz-John conditions. Therefore, we appeal to the invexity with respect to different $(\eta_i)_i$, the generalized Fritz-John condition and by using the theorem 2, we conclude on optimality of the point x_0.

Example 3. We consider the following nonlinear programming problem

$$\begin{array}{rl} \text{Minimize } & f(x)=-x_1,\\ \text{subject to } & g_1(x)=x_1^3-x_2\leq 0,\\ & g_2(x)=x_2\leq 0,\\ & h(x)=x_2(x_1+2)=0 \end{array} \tag{13}$$

where $f,h:\mathbb{R}^2\to\mathbb{R}$ and $g=(g_1,g_2):\mathbb{R}^2\to\mathbb{R}^2$. The set of feasible solutions of problem is $X=\{x=(x_1,x_2)\in\mathbb{R}^2: x_1^3-x_2\leq 0,\ x_2\leq 0 \text{ and } x_2(x_1+2)=0\}$. For this problem, we have $x_0=(0,0)\in X$ is not a Karush-Kuhn-Tucker stationary point of problem (13), because the condition of Karush-Kuhn-Tucker at x_0 takes a form $\nabla f(x_0)+\lambda_1\nabla g_1(x_0)+\lambda_2\nabla g_2(x_0)+\delta\nabla h(x_0)=(-1,-\lambda_1+\lambda_2+2\delta)\neq(0,0),\ \forall\ (\lambda_1,\lambda_2,\delta)\geq 0$. Thus, all the sufficient optimality conditions using this concept, for example from [3, 7, 22, 31], are not applicable.
Moreover, we have x_0 satisfies the classical Fritz-John conditions (theorem 4.3.2 [3]) but it is not difficult to prove that there exists no a function $\eta:X\times X\to\mathbb{R}^2$ for which the functions g_1 and g_2 are both (strictly) (pseudo)-invex at x_0, also g is not (quasi)convex at x_0 and h is not (quasi)concave and not affine at x_0 (take

$x = (-2, -1) \in X$). Furthermore, it is not difficult to prove that there exist no an ε-neighborhood $N_\varepsilon(x_0)$ of x_0, $\varepsilon > 0$, and a function $\eta : X \times X \to \mathbb{R}^2$ for which the functions g_1 and g_2 are both strictly (pseudo)-invex at x_0 over $X \cap N_\varepsilon(x_0)$ (take $x = (-\alpha, 0) \in X \cap N_\varepsilon(x_0)$ with $0 < \alpha < \varepsilon$).
Hence, the (local) sufficient optimality conditions using the (generalized) invexity with respect to a same η (in particular for $\eta(x, x_0) = x - x_0$) with the classical Fritz-John conditions are not applicable, for example the theorem 4.3.6 of Bazaraa et al. [3, p.203], the theorem 3 of Bhatt and Misra [7], the theorem 11.1.1 of Mangasarian [22, p.162], the theorem 2.2 of Singh [31] and the theorem 2.1 of Skarpness and Sposito [32]. Besides, even the sufficient optimality conditions of Martinez and Svaiter [25] (theorems 3.1 and 4.1), that use the approximate gradient projection (AGP) property which is strictly stronger than and implies the classical Fritz-John optimality conditions, are not applicable. On the other hand, since the vectors $\lambda_0 \nabla f(x_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\lambda_1 \nabla g_1(x_0) = \begin{pmatrix} 0 \\ -\lambda_1 \end{pmatrix}$, $\lambda_2 \nabla g_2(x_0) = \begin{pmatrix} 0 \\ \lambda_2 \end{pmatrix}$ and $\nabla h(x_0) = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ do not span $\mathbb{R}^2$, then the theorem 9.7 of Güler [13, p.217], where the original version is due to Fritz-John [17], is also not applicable (Note that $\lambda_i,\ i = 0, 1, 2$ are the scalar multipliers satisfying the classical Fritz-John condition). Furthermore, the first and second order primal (resp. dual) sufficient optimality conditions of theorem 3.2 (resp. theorem 3.4) of Still and Streng [39] are not applicable, because $\bar{C}^0_{x_0} \neq \{0\}$ ($0 \notin int\bar{\mathcal{D}}^0_{x_0}$) and for any nonzero $\tilde{x} \in \bar{C}^0_{x_0}$ we have $\bar{C}^0_{x_0,\tilde{x}} \neq \varnothing$ and $\tilde{x}^t[\mu\nabla^2 f(x) + \lambda_1\nabla^2 g_1(x) + \lambda_2\nabla^2 g_2(x) + \delta\nabla^2 h(x)]\tilde{x} = 0$, $\forall\ \mu, \lambda_1, \lambda_2, \delta \in \mathbb{R}$. Also, the theorems 3.6 and 3.7 of Still and Streng [39] are not applicable, because the (second order) Mangasarian-Fromovitz constraint qualification [23, 39] is not satisfied at x_0.
However, by using the invexity with respect to different $(\eta_i)_i$ and the generalized Fritz-John condition (9), we obtain

- f is pseudo-invex at x_0 with respect to $\eta(x, x_0) = (x_1, -x_1)$ using the proposition 2;
- g is differentiable at x_0 , g_1 and g_2 are active constraints at x_0 and using the proposition 4, for $\theta_1(x, x_0) = (x_1^3 - x_2 - 1, -x_1^3 + x_2 + 1)$ and $\theta_2(x, x_0) = (x_2 - 1, x_2 - 1)$, we obtain that $[\nabla g_j(x_0)]^t\theta_j(x, x_0) < 0,\ \forall\ x \in X,\ \forall\ j \in J(x_0) = \{1, 2\}$.
- h is differentiable at x_0, and using the proposition 4, for $\phi(x, x_0) = (x_2(x_1 + 2) - 1, x_2(x_1 + 2) - 1)$, we obtain that $[\nabla h(x_0)]^t\phi(x, x_0) < 0,\ \forall\ x \in X$.

The generalized Fritz-John condition (9) at x_0 for $\mu = 1$ and $\lambda_1 = \lambda_2 = \delta = 0$ takes the form $\mu[\nabla f(x_0)]^t\eta(x, x_0) = -x_1 \geq 0,\ \forall\ x \in X$. It follows that, by theorem 2, x_0 is an optimal solution for the given nonlinear programming problem.

Particular cases and discussion of theorem 2

In what follows, we give some particular cases which can be derived from theorem 2.

(a) According to the proposition 3, we can replace in the theorem 2 the hypotheses (ii) and (iii) by "$\forall\ j \in J(x_0),\ g_j$ is strictly quasi-invex at x_0 with respect to $\theta_j : X \times X \to \mathbb{R}^n$" and "$\forall\ i = 1, ..., m,\ h_i$ is strictly quasi-invex at x_0 with respect to $\phi_i : X \times X \to \mathbb{R}^n$" respectively.

(b) If the function f is pseudo-invex, the functions $g_j,\ j \in J(x_0)$ and $h_i,\ (-h_i),\ i = 1, ..., m$ are quasi-invex at x_0 with respect to the same function $\eta(x, x_0) = x - x_0$ and if $\mu = 1$, then the theorem 2 reduces to a result which is in the form of sufficient optimality theorem 11.1.1 of Mangasarian [22, p.162], with $\delta \geq 0$. Note that, in this case, we do not need to require that $-h_i, i = 1, ..., m$ are quasi-invex at x_0 with respect to η as in [22] and the proof of this result, without replacing the equality constraint $h(x) = 0$ by the equivalent inequality restrictions $h(x) \leq 0$ and $-h(x) \leq 0$, follows the same lines as in theorem 2 by using the proposition 3 for quasi-invexity.

(c) If the function f is pseudo-invex, the functions $g_j,\ j \in J(x_0)$ and $h_i,\ i = 1, ..., m$ are strictly pseudo-invex at x_0 with respect to the same function $\eta(x, x_0) = x - x_0$ and if we replace "$[\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0)](x - x_0) \geq 0,\ \forall\ x \in X$" by "$\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0) = 0$", then the theorem 2 reduces to the sufficient optimality theorem 2.1 of Skarpness and Sposito [32].

(d) If the function f is invex, the functions $g_j,\ j \in J(x_0)$ and $h_i,\ i = 1, ..., m$ are strictly invex at x_0 with respect to the same function $\eta(x, x_0) = x - x_0$ and if we replace "$[\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0)](x - x_0) \geq 0,\ \forall\ x \in X$" by "$\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0) = 0$", then the theorem 2 reduces to the sufficient optimality theorem 3 of Bhatt and Misra [7].

(e) If there exists an ε-neighborhood $N_\varepsilon(x_0)$ of $x_0,\ \varepsilon > 0$ such that "f is pseudo-invex and the functions $g_j,\ j \in J(x_0)$ are strictly pseudo-invex" at x_0 over $\tilde{X} \cap N_\varepsilon(x_0)$ with respect to the same function $\eta(x, x_0) = x - x_0$ ($\tilde{X} = \{x \in D : g_j(x) \leq 0,\ j \in J(x_0),\ h_i(x) = 0,\ i = 1, ..., m\}$), $h_i,\ i = 1, ..., m$ are affine and $\nabla h_i(x_0),\ i = 1, ..., m$ are linearly independent and if we replace "$[\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0)](x - x_0) \geq 0,\ \forall\ x \in X$" by "$\mu \nabla f(x_0) + \lambda \nabla g_J(x_0) + \delta \nabla h(x_0) = 0$", then the theorem 2 reduces to a result which is in the form of the local sufficient optimality theorem 4.3.6 of Bazaraa et al. [3, p.203], with $\delta \geq 0$.

4 Conclusion

In this paper, we have studied Fritz-John type optimality conditions for constrained nonlinear programming in the presence of nonlinear equality and inequality constraints. New necessary and sufficient conditions for a feasible point to be an optimal solution are obtained under weak invexity with respect to different $\eta,\ (\theta_j)_j$ and $(\phi_i)_i$ and without using of any alternative theorem unlike the usual procedure. We have established simple propositions which helped us to construct easily these different functions ($\eta,\ (\theta_j)_j$ and $(\phi_i)_i$) to verify the optimality of a feasible point (example 3). Moreover, it is illustrated with examples that the obtained sufficient optimality conditions allow to prove easily

for nonconvex programming that a feasible point is optimal, when many results in the literature, including the Karush-Kuhn-Tucker optimality conditions and the Fritz-John ones combining with the (generalized) invexity with respect to the same η, are not applicable. In this way, the sufficient optimality conditions obtained in this paper generalize and extend previously known results in this area.

References

1. Antczak, T.: A class of B-(p,r)-invex functions and mathematical programming. J. Math. Anal. Appl. **286**, 187–206 (2003).
2. Antczak, T.: r-preinvexity and r-invexity in mathemetical programming. Comp. Math. with Appl. **50**, 551–566 (2005).
3. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. Wiley, New York, Third Edition (2006).
4. Bector, C.R., Suneja, S.K., Lalitha, C.S.: Generalized B-vex functions and generalized B-vex programming. J. Optim. Theory Appl. **76**, 561–576 (1993).
5. Ben-Israel, A., Mond, B.: What is invexity ?. J. Austral. Math. Soc. Ser. **B 28**, 1–9 (1986).
6. Bhatia, D., Sharma, A.: New-invexity type conditions with applications to constrained dynamic games. Eur. J. Oper. Res. **148**, 48–55 2003).
7. Bhatt, S.K., Misra, S.K.: Sufficient optimality criteria in non-linear programming in the presence of convex equality and inequality constraints. Zeitschrift für Operations Research **19**: 101–105. Physica-Verlag, Würzburg (1975).
8. Cottle, R.W.: A theorem of Fritz John in mathematical programming. RAND Memorandum RM-3858-PR (1963).
9. Craven, B.D.: Invex functions and constrained local minima. Bull. Austral. Math. Soc. **24**, 357–366 (1981).
10. Craven, B.D., Glover, B.M.: Invex functions and duality. J. Austral. Math. Soc. Ser. **A 39**, 1–20 (1985).
11. Dinuzzo, F., Ong, C.S., Gehler, P., Pillonetto, G.: Learning output kernels with block coordinate descent. Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA (2011).
12. Fulga, C., Preda, V.: Nonlinear programming with E-preinvex and local E-preinvex functions. Eur. J. Oper. Res. **192**, 737–743 (2009).
13. Güler, O.: Foundations of Optimization. Graduate Texts in Mathematics 258, DOI 10.1007/978-0-387-68407-9, Springer Science + Business Media, LLC (2010).
14. Hanson, M.A.: On sufficiency of the Kuhn-Tuker conditions. J. Math. Anal. Appl. **80**, 445–550 (1981).
15. Hanson, M.A., Mond, B.: Necessary and sufficient conditions in constrained optimization. Math. Program. **37**, 51–58 (1987).
16. Jeyakumar, V., Mond, B.: On generalized convex mathematical programming. J. Austral. Math. Soc. Ser. **B 34**, 43–53 (1992).
17. John, F.: Extremum problems with inequalities as side conditions. In: Friedrichs, K.O., Neugebauer, O.E., Stoker, J.J. (eds.) Studies and Essays, Courant Anniversary Volume, Wiley (Interscience), New York, pp. 187–204 (1948).
18. Karush, W.: Minima of functions of several variables with inequalities as side conditions. Dissertation, Department of Mathematics, University of Chicago (1939).

19. Kaul, R.N., Kaur, S.: Optimality criteria in nonlinear programming involving nonconvex functions. J. Math. Anal. Appl. **105**, 104–112 (1985).
20. Kaul, R.N., Suneja, S.K., Srivastava, M.K.: Optimality criteria and duality in multiple-objective optimization involving generalized invexity. J. Optim. Theory Appl. **80 (3)**, 465–482 (1994).
21. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (J. Neyman, ed.), Univ. of Calif. Press, Berkeley, Calif., pp. 481–492 (1951).
22. Mangasarian, O.L.: Nonlinear Programming. McGraw–Hill, New York (1969).
23. Mangasarian, O.L., Fromovitz, S.: The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. J. Math. Anal. Appl. **17**, 37–47 (1967).
24. D.H. Martin. The essence of invexity. J. Optim. Theory Appl. **47**, 65–76 (1985).
25. Martinez, J.M., Svaiter, B.F.: A practical optimality condition without constraint qualifications for nonlinear programming. J. Optim. Theory Appl. **118 (1)**, 117–133 (2003).
26. Mishra, S.K., Wang, S.Y., Lai, K.K.: On non-smooth α-invex functions and vector variational-like inequality. Opt. Lett. **02**, 91–98 (2008).
27. Nickisch, H., Seeger, M.: Multiple kernel learning: a unifying probabilistic viewpoint. arXiv:1103.0897v2 [stat.ML] 30 March (2011).
28. Osuna-Gomez, R., Beato-Morero, A., Rufian-Lizana, A.: Generalized convexity in multiobjective programming. J. Math. Anal. Appl. **233**, 205–220 (1999).
29. Pini, R., Singh, C.: A survey of recent [1985-1995] advances in generalized convexity with applications to duality theory and optimality conditions. Optim. **39 (4)**, 311–360 (1997).
30. Rueda, N.G., Hanson, M.A.: Optimality criteria in mathematical programming involving generalized invexity. J. Math. Anal. Appl. **130**, 375–385 (1988).
31. Singh, C.: Sufficient optimality criteria in nonlinear programming for generalized equality-inequality constraints. J. Optim. Theory Appl. **22 (4)**, 631–635 (1977).
32. Skarpness, B., Sposito, V.A.: A modified Fritz John optimality criterion. J. Optim. Theory Appl. **31 (1)**, 113–115 (1980).
33. Slimani, H., Radjef, M.S.: Duality for nonlinear programming under generalized Kuhn-Tucker condition. Journal of Optimization: Theory, Methods and Applications (IJOTMA) **1 (1)**, 75–86 (2009).
34. Slimani, H., Radjef, M.S.: Nondifferentiable multiobjective programming under generalized d_I-invexity. Eur. J. Oper. Res. **202**, 32-41 (2010).
35. Slimani, H., Radjef, M.S.: Multiobjective Programming under Generalized Invexity: Optimality, Duality, Applications. LAP Lambert Academic Publishing AG & Co. KG, Saarbrücken, Germany (2010).
36. Slimani, H., Radjef, M.S.: Fonctions invexes généralisées et optimisation vectorielle: Optimalité, Caractérisations, Dualité et Applications. Editions Universitaires Européennes GmbH & Co. KG, Saarbrücken, Germany (2011).
37. Soleimani-damaneh, M., Sarabi, M.E.: Sufficient conditions for nonsmooth r-invexity. Numer. Funct. Anal. Optim. **29**, 674–686 (2008).
38. Stancu-Minasian, I.M.: Optimality and duality in nonlinear programming involving semilocally B-preinvex and related functions. Eur. J. Oper. Res. **173**, 47–58 (2006).
39. Still, G.,Streng, M.: Optimality conditions in smooth nonlinear programming. J. Optim. Theory Appl. **90**, 483–515 (1996).
40. Suneja, S.K., Khurana, S., Vani: Generalized nonsmooth invexity over cones in vector optimization. Eur. J. Oper. Res. **186**, 28–40 (2008).

A conflict hypergraph to identify facets for the binary knapsack problem

Chafia Boughani and Méziane Aïder

LAID3, Faculty of Mathematics, USTHB,
BP 32 El Alia, 16111 Algiers, Algeria
chaboughani@gmail.com}
{m-aider@usthb.dz
http://www.usthb.dz/perso/math/maider

Abstract. This paper deals with a simultaneous lifting of a set of variables into a cover inequality for the binary knapsack problem using conflict hypergraphs. Our aim is to generate facets for the knapsack polytope using the structure of the conflict hypergraph. For this, we present sufficient conditions for the generated lifted cover inequalities to define facets for the knapsack polytope. We then define a new class of hard knapsack problems and develop a quadratic time algorithm to generate facets for the binary knapsack polytope.

Keywords: Simultaneous lifting, Hard knapsack problem, Polyhedral approach, Conflict hypergraphs.

1 Introduction

This paper deals with the 0-1 knapsack problem defined as follows:

$$\max\{p^T x | w^T x \leq b, x \in \{0,1\}^n\} \tag{1}$$

where $p \in \mathbb{Z}_+^n$, $w \in \mathbb{Z}_+^n$ and $b \in \mathbb{Z}_+$. This problem is NP-Hard in the weak sense since it can be solved in a pseudo-polynomial time by dynamic programming [3]. Because of its numerous applications, this problem has been intensively studied both theoretically and computationally (see for example [12] and [15]).

The $0-1$ knapsack polytope $\mathcal{P}$ is the convex hull of the set $\mathbb{P}$ of all the feasible solutions of (1), i.e. $\mathcal{P} = conv(\mathbb{P})$ where:

$$\mathbb{P} = \left\{ x \in \{0,1\}^n, \sum_{j=1}^{n} w_j x_j \leq b \right\} \tag{2}$$

By the assumption $w_j \leq b,\ j = 1, \ldots, n$, on the weights of the items, $\mathcal{P}$ is of full dimension ($dim(\mathcal{P}) = n$) since it contains the set of the $(n+1)$ affinely independent vectors $\{e^1, \ldots, e^n, 0\}$, where e^j is the j^{th} unit vector (the vector whose the j^{th} component is 1 and all the others are zero) that belong to $\mathbb{P}$.

Let $P \subseteq \mathbb{R}^n$ be a polyhedron. The inequality $\pi^T x \leq \pi_0$, with $\pi \in \mathbb{R}^n$ and $\pi_0 \in \mathbb{R}$, is valid for P if it is satisfied by every point in P. The polyhedron $F = \{x \in P \mid \pi^T x = \pi_0\}$, if it is not empty, is called a face of P. The dimension of a polyhedron P is the maximum number of its affinely independent points minus 1. If $dim(F) = dim(P) - 1$ then F is a facet and the corresponding valid inequality is said to be a facet defining, or simply a facet of, P.

The notion of cover is extensively used when studying the $0-1$ knapsack polytope $\mathcal{P}$ (see for instance [1], [9] and [17]). A set $C \subseteq \{1, 2, \ldots, n\}$ is called a cover if

$$\sum_{j \in C} w_j > b.$$

A cover C is minimal, if for each $k \in C$, $\sum_{j \in C \setminus \{k\}} w_j \leq b$.

Observe that every minimal cover induces a valid inequality of the form:

$$\sum_{j \in C} x_j \leq |C| - 1, \tag{3}$$

called a cover inequality.

If a cover C is minimal, the corresponding cover inequality (3) is said to be a minimal cover inequality. It is a facet of a lower dimensional knapsack polytope $\mathcal{P}_C = conv(\mathbb{P}_C)$, where $\mathbb{P}_C = \{x \in \{0,1\}^{|C|} : {w_C}^T x \leq b\}$, with w_C consisting in the components of w that correspond to the elements of C. Nevertheless, this inequality can generate a facet of $\mathcal{P}$.

The notion of lifting was introduced by Gomory [6]. It consists to increase the dimension of a valid inequality by adding variables with the greatest coefficient that maintains the validity of the obtained inequality. Recently, some works, based upon lifting over superadditive functions, have been done on sequence independent lifting (see [8], [17] and [19]).

In 1978, Zemel [18] introduced the simultaneous lifting for binary integer variables into a cover inequality. This method allows the generation of many inequalities but this requires solving an exponential number of integer programs. In 2005, Hooker and Easton [4] improved the Zemel's technic by developing a linear time method to simultaneously lift variables into cover inequalities.

Numerous authors have used conflict graphs to create both valid and facets defining inequalities. Hypergraph structures have also been used to identify various valid inequalities. A hypergraph $H = (V, A)$ is defined by a set of vertices $V = \{1, \ldots, n\}$ and a set of edges $A = \{E_1, \ldots, E_m\}$, where $E_i \subseteq V$ for all $i = 1, \ldots, m$. The definitions of a subhypergraph and an induced subhypergraph are similar to the classical graph theory definitions.

In this work, we define a conflict hypergraph that allows a simultaneous lifting of a set of variables into a cover inequality.

The paper is organized as follows. In section 2, we introduce the conflict hypergraph that allows us to simultaneously lift a set of variables into a cover

inequality. We also give sufficient conditions that the generated lifted cover inequalities must verify to define facets. In section 3, we develop a quadratic time algorithm that generates facet defining inequalities by a simultaneous lifting of a set of variables into a cover inequality using the given sufficient conditions. In section 4, we introduce a new class of hard knapsack problems.

2 Conflict hypergraph

We suppose that the elements of the set $N = \{1, 2, \ldots, n\}$ of the indices of the variables x_j in (1) are arranged in a decreasing order of the w coefficients, $w_1 \geq w_2 \geq \ldots \geq w_n$, and that the set $E_0 = \{1, 2, \ldots, |E_0|\}$ has at least three elements ($|E_0| \geq 3$) and is a minimal cover for the knapsack polytope $\mathcal{P}$.

This means that $\sum_{j \in E_0} w_j \geq b$ and $\sum_{j \in E_0 \setminus \{k\}} w_j \leq b$ for all $k \in E_0$.

We also suppose that the set $F_0 = N \setminus E_0 = \{|E_0| + 1, \ldots, n\}$ has at least two elements ($n - |E_0| \geq 2$) and forms a cover.

Let us define the conflict hypergraph $H_c = (V_c, A_c)$ that depicts the binary knapsack constraint $\sum_{i \in N} w_j x_j \leq b$, as follows:

the vertex set is $V_c = N = \{1, \ldots, n\}$, where each vertex $j \in V_c$ is associated to a variable x_j. The edge set A_c contains the edge $E_0 = \{1, \ldots, |E_0|\}$ and the edges E_k, $k = 1, \ldots, p$ defined sequentially in the following way:

- **Step 1:** For $k = 1$, $E_1 = \{3, \ldots, |E_0|, n - q^{(1)}, \ldots, n\}$, where $1 < q^{(1)} < n - |E_0|$ and $\dfrac{1}{q^{(1)}} \leq \dfrac{|E_0| - 2}{n - |E_0| - 1}$.
- **Step 2:** For $1 < k < p$, $E_k = \{3, \ldots, |E_0|, n - q^{(k-1)} - 1 - q^{(k)}, \ldots, n - q^{(k-1)} - 1\}$, where $1 < q^{(k)} < q^{(k-1)}$.
- **Step 3:** For $k = p$, $E_p = \{3, \ldots, |E_0| + 2\}$, where $q^{(p)} = 1$

Note that the edge E_k, for $k \geq 1$, consists in the vertices in $E_0 \setminus \{1, 2\}$ and the $q^{(k)} + 1$ "largest free vertices" of F_0 (those with the largest labels that do not belong to any E_l, for $l < k$) and $q^{(k)} + 1$ is defined to be the minimum number of vertices of F_0 to be added to $E_0 \setminus \{1, 2\}$ that make the set E_k a minimal cover for the knapsack polytope $\mathcal{P}$.

Observe that the hypergraph H_c is an hyperstar with the set $\{3, \ldots, |E_0|\}$ as a center (a center of a hyperstar consists in the vertices belonging to all the edges) and the set $F_0 \cup \{1, 2\}$ as a perimeter (a perimeter of a hyperstar consists in the vertices belonging to only one edge). It allows to generate facets for the knapsack polytope which are simultaneously lifted cover inequalities of the form:

$$\sum_{j \in E_0} x_j + \frac{1}{q^{(1)}} \sum_{j \in N \setminus E_0} x_j \leq |E_0| - 1 \tag{4}$$

Theorem 1. *Given a binary knapsack problem with the corresponding conflict hypergraph $H_c = (V_c, A_c)$, the inequality of the form (4) is a facet of $\mathcal{P}$ if the following conditions are satisfied:*

1. *The set $S_1 = \{2, \ldots, |E_0|, n\}$ is a minimal cover.*
2. *The set $S_2 = \{\beta, \ldots, |E_0|, n-(\beta-2)q^{(1)}-1, \ldots, n\}$, where $3 \leq \beta \leq |E_0|$, is not a cover.*
3. *The set $S_3 = \{3, \ldots, |E_0|, |E_0|+1, n-q^{(1)}+1, \ldots, n\}$ is not a cover.*

Proof. Let us show that if the three conditions 1., 2. and 3. are satisfied, then the inequality (4) defines a facet. For this, we use the direct technic of facets proof, i.e. by showing that the inequality is valid and that it defines a face of $\mathbb{P}$ of dimension $dim(\mathbb{P}) - 1$.

First, to prove by contradiction that the inequality (4) is valid, assume that it is not. Thus, there exists $x^* \in \mathbb{P}$ such that:

$$\sum_{j \in E_0} x_j^* + \frac{1}{q^{(1)}} \sum_{j \in N \setminus E_0} x_j^* > |E_0| - 1 \tag{5}$$

Set $T_1 = \{i \in E_0 \mid x_i^* = 1\}$, $|T_1| = p_1^*$, $T_2 = \{i \in N \setminus E_0 \mid x_i^* = 1\}$ and $|T_2| = p_2^*$. Since E_0 is a cover, we necessarily have $p_1^* \leq |E_0| - 1$.

We distinguish the following three cases:

Case 1, $p_1^* = 0$ and $p_2^* \neq 0$: from the inequality (5) we have $\frac{1}{q^{(1)}} p_2^* > |E_0| - 1$. Since $\frac{1}{q^{(1)}} \leq \frac{|E_0|-2}{n-|E_0|-1}$ and $N \setminus E_0$ is a cover, we then have $p_2^* \leq n - |E_0| - 1$. It follows that $\frac{1}{q^{(1)}} p_2^* \leq |E_0| - 2$, a contradiction.

Case 2, $p_2^* = 0$ and $p_1^* \neq 0$: the inequality (4) reduces to the cover inequality

$$\sum_{j \in E_0} x_j \leq |E_0| - 1 \tag{6}$$

induced by the cover E_0 and the inequality (5) reduces to the inequality

$$\sum_{j \in E_0} x_j > |E_0| - 1, \tag{7}$$

a contradiction because the cover inequality is valid.

Case 3, $p_1^* \geq 1$ and $p_2^* \geq 1$: the inequality (5) reduces to

$$p_1^* + \frac{1}{q^{(1)}} p_2^* > |E_0| - 1 \Leftrightarrow \frac{1}{q^{(1)}} > \frac{|E_0| - 1 - p_1^*}{p_2^*}$$

But we have $\min\left(\frac{|E_0|-1-p_1^*}{p_2^*}\right) = \frac{|E_0|-2}{n-|E_0|-1}$, so $\frac{1}{q^{(1)}} > \frac{|E_0|-2}{n-|E_0|-1}$.

Thus contradiction because $\frac{1}{q^{(1)}} \leq \frac{|E_0|-2}{n-|E_0|-1}$.

So the inequality of the form (4) is valid.

▷ Second, we shall prove that the inequality (4) dominates the inequalities of the form

$$\sum_{j\in E_0} x_j + \alpha \sum_{j\in N\setminus E_0} x_j \leq |E_0|-1 \qquad \alpha = \frac{|E_0|-1-p_1^*}{p_2^*} \tag{8}$$

By contradiction, assume that the inequality (8) dominates the inequality (4), i.e. $\exists\alpha$ such that $\alpha < \frac{1}{q^{(1)}}$. We distinguish two cases:

Case 1, $\alpha = 0 < \frac{1}{q^{(1)}}$: so we necessarily have $p_1^* = |E_0|-1$ and $p_2^* \geq 1$, which means that there exists a set $S_1^{'} = \{2,\ldots,|E_0|, n-p_2^*,\ldots,n\}$ that forms a minimal cover. Nevertheless, we have

$$S_1^{'} = \{2,\ldots,|E_0|,n\} \cup \{n-p_2^*,\ldots,n-1\} = S_1 \cup \{n-p_2^*,\ldots,n-1\}$$

As S_1 is a minimal cover it follows that $S_1^{'}$ is not a minimal cover, a contradiction.

Case 2, $0 < \alpha = \frac{\phi}{p_2^*} < \frac{1}{q^{(1)}}$, with

$$\phi = |E_0|-1-p_1^* = |E_0|-1-(|E_0|-(\beta-1)) = \beta-2$$

where $3 \leq \beta \leq |E_0|$. We distinguish two cases:

1. **Case 2.1:** $\phi > 1$, so we obtain $\phi = \beta-2 > 1 \Rightarrow \beta > 3$. We have supposed that
$$\alpha < \frac{1}{q^{(1)}} \Leftrightarrow \frac{\phi}{p_2^*} < \frac{1}{q^{(1)}} \Leftrightarrow \frac{\beta-2}{p_2^*} < \frac{1}{q^{(1)}}$$
$$\Leftrightarrow p_2^* > (\beta-2)q^{(1)} \Leftrightarrow p_2^* \geq (\beta-2)q^{(1)}+1$$
Which mean that the set $S_2^{'} = \{\beta,\ldots,|E_0|,n-p_2^*,\ldots,n\}$ is a minimal cover. We have for $p_2^* = (\beta-2)q^{(1)}+1$ the set $S_2^{'} = \{\beta,\ldots,|E_0|,n-(\beta-2)q^{(1)}-1,\ldots,n\} = S_2$ is also a minimal cover, contradiction because S_2 is not a cover.
2. **Case 2.2:** $\phi = 1$, so we obtain $\phi = \beta-2 = 1 \Rightarrow \beta = 3$. We have supposed that
$$\alpha < \frac{1}{q^{(1)}} \Leftrightarrow \frac{\phi}{p_2^*} < \frac{1}{q^{(1)}} \Leftrightarrow \frac{1}{p_2^*} < \frac{1}{q^{(1)}} \Leftrightarrow p_2^* > q^{(1)} \Leftrightarrow p_2^* \geq q^{(1)}+1$$

Which mean that the set $E_1^{'} = \{3, \ldots, |E_0|, n - p_2^*, \ldots, n\}$ is a minimal cover. As $p_2^* \geq q^{(1)} + 1$ it follows that $E_1 \subset E_1^{'}$, contradiction because E_1 is a minimal cover which mean that $E_1^{'}$ can not be a minimal cover.

▷ Third, we shall exhibit the n affinely independent points which are determined as follows:

- The set E_0 forms a minimal cover and so the points $\sum_{i \in E_0 \setminus \{h\}} e_i$, for each $h \in E_0$, are feasible and meet the inequality (4) at equality. We obtain $|E_0|$ points.
- From the conditions 1. and 2. and by the construction of the edge E_1, the inequalities of the form (8) are dominated by the inequality (4). Therefore, and since the set N is sorted, the points:

$$\sum_{i \in E_0 \cap E_1} e_i + \sum_{i \in (F_0 \cap E_1) \setminus \{k\}} e_i$$

where $k \in \{n - q^{(1)}, \ldots, n\}$, are feasible and meet the inequality (4) at equality. We obtain $|F_0 \cap E_1|$ points.
- From the condition 3, the set S_3 is not a cover. So the point:

$$\sum_{i \in E_0 \cap E_1} e_i + e_{|E_0|+1} + \sum_{i \in (F_0 \cap E_1) \setminus Y} e_i \qquad Y = \{n - q^{(1)}, n - q^{(1)} + 1\}$$

is feasible. Since the set N is sorted, we obtain the $|N \setminus (E_0 \cup (F_0 \cap E_1))|$ feasible points that meet the inequality (4) at equality:

$$\sum_{i \in E_0 \cap E_1} e_i + e_l + \sum_{i \in (F_0 \cap E_1) \setminus Y} e_i \qquad Y = \{n - q^{(1)}, n - q^{(1)} + 1\}$$

where $l \in N \setminus (E_0 \cup (F_0 \cap E_1))$.

We obtain $|E_0| + |F_0 \cap E_1| + |N \setminus (E_0 \cup (F_0 \cap E_1))| = n$ feasible points.

Clearly, these points are affinely independent. ■

Example 1. Let us consider the following knapsack constraint:

$$30x_1 + 30x_2 + 25x_3 + 25x_4 + 22x_5 + 21x_6 + 19x_7 + 15x_8 + 15x_9 + 14x_{10} + 14x_{11} + 13x_{12} + 13x_{13} \leq 92$$

We can define the conflict hypergraph $H_c = (V_c, A_c)$ that depicts this knapsack constraint, with $V_c = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$, $A_c = \{E_0, E_1, E_2, E_3\}$, $E_0 = \{1, 2, 3, 4\}$, $E_1 = \{3, 4, 10, 11, 12, 13\}$, $E_2 = \{3, 4, 7, 8, 9\}$, $E_3 = \{3, 4, 5, 6\}$ and $F_0 = \{5, 6, 7, 8, 9, 10, 11, 12, 13\}$. It allows us to generate the following facet:

$$\sum_{j \in E_0} x_j + \frac{1}{3} \sum_{j \in N \setminus E_0} x_j \leq |E_0| - 1$$

$$\Leftrightarrow x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13}) \leq 3$$

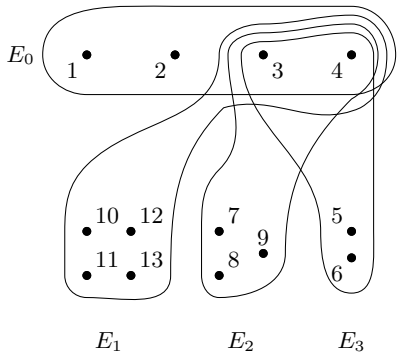


Fig. 1. The Hypergraph $H_c = (V_c, A_c)$

3 A quadratic time algorithm for simultaneously lifting a set of variables to a cover inequality

Theorem 1 can be used to derive a quadratic time algorithm to generate simultaneously lifted cover inequalities of the form (4). The main variables used in Algorithm 1 are the following:

- ✓ α : The lifting coefficient
- ✓ W_α : The sum of the coefficients w_j with $j \in E_0 \setminus \{1, 2\}$
- ✓ q : The maximum number of elements that we can take from the set E_3 to be lifted
- ✓ $sum1$: The sum of the coefficients w_j with $j \in E_3$
- ✓ $sum2$: The sum of the coefficients w_j with $j \in S_1$
- ✓ $sum3$: The sum of the coefficients w_j with $j \in S_2$
- ✓ $sum4$: The sum of the coefficients w_j with $j \in S_3$

The running time of Algorithm 1 is $O(n^2)$. Therefore, Algorithm 1 is quadratic.

Algorithm 1 Algorithm for simultaneously lifting a set of variables

Require: The sets N, E_0 and b.
Ensure: Inequalities of the form (4).

if $N \neq \emptyset$ **then**
  $W_\alpha \leftarrow 0$
end if
for $k \leftarrow 2$ **to** $|E_0|$ **do**
  $W_\alpha \leftarrow W_\alpha + w_i$
end for
$i \leftarrow |N|$
$sum1 \leftarrow W_\alpha$
while $sum1 \leq b$ **do**
  $sum1 \leq b$
  $sum1 \leftarrow sum1 + w_i$
  $i \leftarrow i - 1$
end while
$q \leftarrow |N| - i - 1$
$sum2 \leftarrow 0$
for $k \leftarrow 2$ **to** $(|E_0| + 1)$ **do**
  $sum2 \leftarrow sum2 + w_i$
end for
$W_\beta \leftarrow W_\alpha$
for $j \leftarrow 3$ **to** $|E_0|$ **do**
  $W_\beta \leftarrow W_\beta - w_j$
  $sum3 \leftarrow W_\beta$
  for $i \leftarrow (|N| - (j-2) \times q - 1)$ **to** $|N|$ **do**
    $sum3 \leftarrow sum3 + w_i$
  end for
end for
$sum4 \leftarrow sum1 + w_{|E_0|+1} - w_{|N|-q} - w_{|N|-q+1}$
if $(sum2 > b)$ **and** $(sum3 \leq b)$ **and** $(sum4 \leq b)$ **then**
  $\alpha \leftarrow \frac{1}{q}$
  $N \leftarrow N \setminus \{w_{|N|-q}, \ldots, w_{|N|}\}$
  $SimultaneousLifting(N, E_0, b)$
else
  $N \leftarrow \emptyset$
end if

4 A new class of hard knapsack problems

Despite the hardness of the knapsack problem, most solvers can quickly solve most of its instances that requires exponentially many branches to solve when using recursive algorithms. However, Chvàtal [2] provided the following class of

knapsack instances:

$$\begin{cases} \max \sum_{j=1}^{n} w_j x_j \\ \sum_{j=1}^{n} w_j x_j \leq \left\lfloor \frac{1}{2} \sum_{j=1}^{n} w_j \right\rfloor \\ x_j = 0, 1 \ (j = 1, \ldots, n) \end{cases} \tag{9}$$

where the weights $w_j, j = 1, \ldots, n$ are randomly chosen integers between 1 and $10^{\frac{n}{2}}$.

Theorem 2. *[2] If $w_1, w_2, \ldots, w_n$ have the following four properties:*

P1: $\sum_{i \in I} w_i \leq \frac{1}{2} \sum_{j=1}^{n} w_j$ *whenever* $I \subset \{1, \ldots, n\}$ *and* $|I| \leq \frac{n}{10}$.

P2: Every integer greater than one divides fewer than $\frac{9n}{10}$ *of the integers* w_j.

P3: $\sum_{i \in I} w_i \neq \sum_{j \in J} w_j$ *whenever* $I, J \subset \{1, \ldots, n\}$ *and* $I \neq J$.

P4: There is no set $I \subset \{1, \ldots, n\}$ *such that* $\sum_{i \in I} w_i = \left\lfloor \frac{1}{2} \sum_{j=1}^{n} w_j \right\rfloor$

then every recursive algorithm creates at least $2^{\frac{n}{10}}$ *partial vectors in the process of solving (9).*

Later, Hunsaker and Tovey [10] strengthen this result to show that there exist knapsack instances that require exponentially many branches even if all of the sequence dependent lifted covers are added to the formulation.

Gu et al. [7] considered the use of simple lifted cover inequalities on knapsack problems. They showed that a branch and cut procedure using simple lifted cover inequalities requires an exponential number of nodes for the following set of knapsack instances, parameterized by the scalar n and the vectors δ and ξ:

$$\begin{cases} \max \sum_{j=1}^{12r} (2\theta - \xi_j) x_j + \sum_{j=12r+1}^{20r} (3\theta - \xi_j) x_j \\ \sum_{j=1}^{12r} (2 \cdot 2^r - \delta_j) x_j + \sum_{j=12r+1}^{20r} (3 \cdot 2^r - \delta_j) x_j \leq 6r \cdot 2^r \\ x \in \{0, 1\}^{20r} \end{cases} \tag{10}$$

where $r \geq 10$, $\theta = (60r \cdot 2^r)^{20r+1}$, $\delta_j \in \{0, 1, \ldots, \lfloor \frac{2^{r-1}}{3r} \rfloor\}$ for all $1 \leq j \leq 20r$, and $\xi_j \in \{0, 1, \ldots, 2^r\}$ for all $1 \leq j \leq 20r$

Based on these results, we introduce the following class of knapsack problems :

$$\begin{cases} \max \displaystyle\sum_{j=1}^{12r}(2\cdot 2^r-\delta_j)x_j+\sum_{j=12r+1}^{20r}(3\cdot 2^r-\delta_j)x_j \\ \displaystyle\sum_{j=1}^{12r}(2\cdot 2^r-\delta_j)x_j+\sum_{j=12r+1}^{20r}(3\cdot 2^r-\delta_j)x_j \leq \left\lfloor \alpha\big(48r\cdot 2^r-\sum_{j=1}^{20r}\delta_j\big)\right\rfloor \\ x\in\{0,1\}^{20r} \end{cases} \tag{11}$$

where $r \geq 10$, $\alpha \in]0,1[$ and $\delta_j \in \{0,1,\ldots,\lfloor\frac{2^{r-1}}{3r}\rfloor\}$ for all $1 \leq j \leq 20r$.

5 Conclusions and future research

In this paper we have defined a conflict hypergraph allowing a simultaneous lifting of a set of variables into a cover inequality. The conflict hypergraph is an hyperstar. Some theoretical results provide sufficient conditions that the lifted cover inequalities must verify to define facets. These theoretical results allow us to develop a quadratic time algorithm that generates facet defining inequalities. We have introduced a new class of hard knapsack problems.

It is very interesting to do computational researches to strengthen our theoretical results.

References

[1] E. Balas. Facets of the knapsack polytope. Mathematical Programming, 8: 146–164, 1975.

[2] V. Chvàtal. Hard knapsack problems. Operations Research, 28 (6): 1402–1412, 1980.

[3] G. Dantzig. Discrete variable extremum problems. Operations Research, 5: 266–277, 1957.

[4] T. Easton and K. Hooker. Simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. Discrete Optimization, Special Issue: In Memory of George B. Dantzig, 5(2) May 2008, 254-261.

[5] T. Easton and K. Hooker. Hypergraphs and integer programming polytopes. PhD thesis, Kansas State University, 2005.

[6] R. E. Gomory. Some polyhedra related to combinatorial problems. Linear Algebra and its Applications, 2: 451–588, 1969.

[7] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs : complexity. Mathematecal Programming, 85(3): 439–467, 1999.

[8] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. Journal of Combinatorial Optimization, 4: 109–129, 2000.

[9] P. L. Hammer, E. L. Johnson, and U. N. Peled. Facets of regular 0-1 polytopes. Mathematical Programming, 8: 179–206, 1975.

[10] B. Hunsaker, C. Tovey. Simple lifted cover inequalities and hard knapsack problems. Discrete Optimization, 2 (3): 219–228, 2005.

[11] K. Kaparis and A.N. Letchford. Separation algorithms for 0-1 knapsack polytopes. Mathematical Programming, 124 (1-2): 69–91, 2010.

[12] S. Martello and P. Toth. Knapsack Problems : Algorithms and Computer Implementations. John Wiley & Sons, Chichester, 1990.

[13] G.L Nemhauser and L.A Wolsey. Integer and Combinatorial Optimization. John Wiley & Sons, New York, 1999.

[14] S. Pahwa. The theory of simultaneous lifting: constellations in conflict hypergraphs. MS Dissertation, Kansas State University, 2009.

[15] H. M. Salkin and C. A. De Kluyver. The knapsack problem : a survey. Naval Research Logistics Quarterly, 22(1): 127–144, 2006.

[16] K. Sharma. Simultaneously lifting sets of varibales in binary knapsack problems. MS Dissertation, Kansas State University, 2009.

[17] L.A. Wolsey. Valid inequalities and superadditivity for 0-1 integer programs. Mathematic of Operations Research, 2(1): 66–77, February 1977.

[18] E. Zemel. Lifting the facets of 0-1 polytopes. Mathematical Programming, 15: 268–277, 1978.

[19] B. Zenga and J. P. P. Richard. A polyhedral study on 01 knapsack problems with disjoint cardinality constraints: Strong valid inequalities by sequence-independent lifting. Discrete Optimization, 8: 259–276, 2011.

Sur la convergence d'une méthode projective de type Karmarkar pour la programmation linéaire

Djamel Benterki*, Mousaab Bouafia*

Abstract

Dans ce travail, on s'intéresse aux performances de l'algorithme projectif de Karmarkar pour la programmation linéaire. En se basant sur les travaux de Scherijver, nous proposons un pas de déplacement meilleur que celui de Scherijver permettant une amélioration modérée du comportement de l'algorithme. On montre par la suite, que l'algorithme converge après $\frac{n}{1-\log(2)+(\frac{nr^2}{10})} \ln\left(\frac{c^t e_n}{\varepsilon}\right)$ itérations.

Mots clés : Programmation linéaire, Méthode de points intérieurs, Fonction potentiel.

1 Introduction

L'algorithme de Karmarkar pour résoudre les problèmes d'optimisation linéaire [2] est le premier algorithme réellement efficace qui résout ces problèmes en un temps polynomial. La méthode des ellipsoïdes fonctionne aussi en temps polynomial mais est inefficace en pratique. En utilisant une fonction potentiel, Karmarkar montre que son algorithme converge après un nombre $\mathbf{o(nq + n \log n)}$ itérations pour un pas de déplacement $\alpha_k = \frac{1}{4}$. Depuis,

*LMFN, Laboratoire de Mathématiques Fondamentales et Numériques, Département de Mathématiques, Faculté des Sciences, Université Ferhat Abbas, Sétif, Algérie.

plusieurs chercheurs s'intéressent à cet algorithme dans le but d'améliorer au mieux son comportement numérique. Les deux éléments de base visés sont la direction de déplacement qui domine le côut du calcul à chaque itération et le pas de déplacement qui joue un rôle important dans la vitesse de convergence. En effet, en modifiant la forme analytique de la fonction potentiel, Padberg [3] a pu réduire le nombre des itérations à $\mathbf{o(nq)}$ pour $\alpha_p = \frac{1}{2}$. De son côté Scherijver [8], en gardant la même fonction potentiel de Karmarkar a pu montrer que l'algorithme converge après $\frac{n}{1-\log(2)} \log\left(\frac{c^t e_n}{\varepsilon}\right)$ itérations pour $\alpha_s = \frac{1}{1+nr}$.

Notre travail rentre dans ce cadre, où en s'inspirant des travaux de Karmarkar [2] et Sherijver [8], nous proposons un nouveau pas de déplacement permettant d'améliorer le comportement numérique de l'algorithme de Karmarkar.

Le papier est organisé comme suit :

Dans la section 2, on présente le programme linéaire traité par Karmarkar. La section 3, représente l'objectif principal de notre travail. Dans un premier temps, on propose un nouveau pas de déplacement puis on montre que ce pas est meilleur que celui de Scherijver. D'autre part, on donne un résultat de convergence polynomiale amélioré par rapport à celui de Scherijver. Enfin, on termine par des tests numériques et une conclusion dans la section 4.

2 Problème linéaire traité par Karmarkar

Dans son article [2], Karmarkar traite le problème linéaire sous la forme réduite suivante :

$$(PK) \begin{cases} \min c^t x = 0 \\ Ax = 0 \\ x \in S_n = \left\{ x \in \mathbb{R}^n_+ \ , \ e_n^t x = 1 \right\}, \end{cases}$$

où $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ est une matrice de plein rang ($rang\,(A) = m < n$) et $e_n = (1, 1, ..., 1)^t \in \mathbb{R}^n$.

Le vecteur $x^0 = \frac{e_n}{n} \in S_n$, est le centre du simplexe S_n.

2.1 Transformation de Karmarkar sur S_n

Cette transformation est définie à chaque itération par :

$$\begin{aligned} T_k : S_n &\longrightarrow S_n \\ x \mapsto y = \quad & T_k(x) = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x} \end{aligned}$$

tel que : $D_k = diag\left(x^k\right)$

De même, la transformation T_k est inversible et on a :

$$x = T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}$$

Le problème transformé de (PK) par la transformation T_k est le problème linéaire :

$$(PKT)\begin{cases} \min\left(D_k c\right)^t y \\ AD_k y = 0 \\ y \in S_n = \left\{ y \in \mathbb{R}_+^n : \sum_{i=1}^{n} y_i = 1 \right\}, \end{cases}$$

qui peut s'écrire aussi sous la forme :

$$(PKT)\begin{cases} \min\left(D_k c\right)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y \geq 0 \end{cases}$$

Le problème transformé (PKT) est aussi mis sous la forme réduite de Karmarkar. De même cette transformation permet de ramener à chaque itération, l'itéré x^k au centre du simplexe S_n, i.e., $T_k\left(x^k\right) = \frac{e_n}{n}$.

Avant d'appliquer les conditions d'optimalité, Karmarkar [2] relaxe le problème (PKT) en remplaçant la condition $y \geq 0$ par la sphère $s\left(\frac{e_n}{n}, \alpha r\right)$, (on montre facilement dans [3] que si $y \in s\left(\frac{e_n}{n}, \alpha r\right)$ alors $y \geq 0$) avec $r = \frac{1}{\sqrt{n(n-1)}}$ et $0 < \alpha < 1$.

Le problème (PKT) devient alors :

$$(PKT)_r \begin{cases} \min (D_k c)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \left\| y - \frac{e_n}{n} \right\|^2 \leq (\alpha r\ )^2. \end{cases}$$

En utilisant les conditions d'optimalité, la solution optimale du problème $(PKT)_r$ est donnée par :

$$y = \frac{e_n}{n} - \alpha r d_k$$

où $d_k = \frac{P_k}{\|P_k\|}$; tel que P_k est la projection du vecteur $D_k c$ sur la noyau de la matrice des contraintes $\begin{bmatrix} AD_k \\ e_n^t \end{bmatrix}$.

En revenant par la transformation inverse T_k^{-1}, on obtient une solution réalisable x^k pour le problème initial (PK) telque $x^k = T_k^{-1}\left(y^k\right) = \frac{D_k y^k}{e_n^t D_k y^k}$.

2.2 Convergence de l'algorithme de Karmarkar

Pour étudier la convergence polynomiale de l'algorithme, Karmarkar a utilisé (dans le cas $z^* = 0$) la fonction potentiel suivante :

$$f(x) = n \log c^t x - \sum_{i=1}^{n} \log x_i$$

définie sur :

$$D_x = \left\{ x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1 \right\}.$$

Theorème 2.1 *(Théorème de convergence de Karmarkar) [2]*

Si $0 < \alpha_k \leq \frac{1}{4}$ *, alors en partant de* $x^0 = \frac{e_n}{n}$*, après* $\boldsymbol{o(nq + n \log n)}$ *itérations, l'algorithme trouve un point réalisable* x^* *tel que:*
(1) $c^t x^* = 0$
ou
(2) $\frac{c^t x^*}{c^t x^0} \leq 2^{-q}$ *où* q *est une précision fixée.*

3 Amélioration de la convergence de l'algorithme de Karmarkar

Dans cette section, nous proposons un nouveau pas de déplacement α_m meilleur que celui introduit par Scherijver, permettant d'accélérer la convergence de l'algorithme de Karmarkar. Avant cela, nous présentons des lemmes ci-dessous pour faciliter au lecteur la compréhension des propos que nous avons établis.

Lemme 1 *[7] Soit ψ_n une fonction définie par :*

$$\psi_n :]-1, +\infty[^n \ \to \mathbb{R}_+$$

$$x \ \mapsto \ \psi_n(x) = e_n^t x - \sum_{i=1}^{n} \log(1 + x_i)$$

Alors pour $\|x\| < 1$ *on a :* $\psi_1(-\|x\|) \geq \psi_n(-x)$.

Rappelons que la fonction potentiel de Karmarkar est définie par :

$$f(x) = n \log c^t x - \sum_{i=1}^{n} \log x_i$$

sur l'ensemble :

$$D_x = \left\{ x \in \mathbb{R}^n : x > 0, \ Ax = 0, \ e_n^t x = 1 \right\}.$$

Lemme 2 *[7] Si $x \in D_x$, alors :*

$$c^t x \leq \exp\left(\frac{f(x)}{n}\right)$$

Plus généralement, Keraghel [3] a montré le lemme suivant :

Lemme 3 *[3] Soit x^k le $k^{ième}$ itéré de l'algorithme de Karmarkar, alors :*

$$\frac{c^t x^k}{c^t x^0} \leq \left(\exp\left[f(x^k) - f(x^0)\right]\right)^{\frac{1}{n}},$$

où $x^0 = \frac{e_n}{n}$.

Lemme 4 *[7] A chaque itération k, la fonction potentiel diminue de la quantité suivante :*

$$f(x^{k+1}) - f(x^k) = -\Delta,$$

telle que :

$$x^{k+1} = T_k^{-1}\left(y^k\right) \text{ et } \Delta = n \log \frac{c^t D_k e_n}{c^t D_k y^k} + \sum_{i=1}^{n} \log y_i^k$$

Lemme 5 *[7] Pour* $0 < \alpha < \frac{1}{nr}$ *on a :*

$$\Delta \geq \alpha n^2 r^2 + n\psi_1\left(-\alpha \frac{r}{R}\right) - \psi_1(-\alpha nr)$$

Lemme 6 *[1]Pour* $\alpha = \alpha_m = \frac{5+nr^2}{5+(5+nr^2)nr}$ *on a :*

a) $\psi_1\left(-\alpha_m nr^2\right) > \frac{1}{5}\alpha_m n^2 r^4$.

b) $\Delta \geq \psi_1\left(\frac{1}{5}nr\left(5+nr^2\right)\right)$

3.1 Amélioration du pas de déplacement

Rappelons que pour $\alpha_s = \frac{1}{1+nr}$ (n taille du problème (PL)), Scherijver [8] a montré que l'algorithme converge après $\frac{n}{1-\log(2)} \log\left(\frac{c^t e_n}{\varepsilon}\right)$ itérations.

Dans cette partie, en se basant sur les lemmes précédents, on donne dans le lemme suivant un nouveau pas de déplacement $1 > \alpha_m > \alpha_s$ qui permet de réduire le nombre d'itérations nécessaire pour la convergence de l'algorithme de Karmarkar.

Lemme 7 *Si* $\alpha_m = \frac{5+nr^2}{5+(5+nr^2)nr}$, *l'algorithme de Karmarkar converge après* $\frac{n}{1-\log(2)+(\frac{nr^2}{10})} \ln\left(\frac{c^t e_n}{\varepsilon}\right)$ *itérations.*

Preuve :

D'après le lemme **(6)** b) on a **:**

$$\Delta > \psi_1\left(\frac{1}{5}nr\left(5+nr^2\right)\right) = \psi_1\left(nr + \frac{nr}{5}nr^2\right) > \psi_1\left(1 + \frac{1}{5}nr^2\right).$$

Comme $nr > 1$ et ψ_1 est croissante, alors

$$\Delta > \psi_1\left(1 + \frac{1}{5}nr^2\right), \tag{3.1}$$

D'autre part

$$\psi_1\left(1 + \frac{1}{5}nr^2\right) - \psi_1(1) > \frac{1}{2}\left(\frac{1}{5}nr^2\right) = \frac{nr^2}{10} \text{ car } \log(1+x) \leq x,$$

donc d'après (3.1) :

$$\Delta > \psi_1(1) + \frac{nr^2}{10}, \quad \forall k \in \mathbb{N}, \tag{3.2}$$

et du lemme (5) on a :

$$f(x^k) - f(x^{k-1}) = -\Delta,$$

d'où de (3.2) on a :

$$f(x^i) - f(x^{i-1}) < -\left(\psi_1(1) + \frac{nr^2}{10}\right),$$

en conséquence :

$$\sum_{i=1}^{k}(f(x^i) - f(x^{i-1})) < -\sum_{i=1}^{k}\left(\psi_1(1) + \frac{nr^2}{10}\right),$$

ce qui donne :

$$f(x^k) - f(x^0) < -k\left(\psi_1(1) + \frac{nr^2}{10}\right), \quad x^0 = \frac{e_n}{n},$$

donc :

$$f(x^k) < -k\left(\psi_1(1) + \frac{nr^2}{10}\right) + f(x^0),$$

comme :

$$f(x^0) = f(\frac{e_n}{n}) = n\log c^t\frac{e_n}{n} \; - \; \sum_{i=1}^{n}\log\frac{1}{n} = n\log c^t e_n,$$

on a :

$$f(x^k) < -k\left(\psi_1(1) + \frac{nr^2}{10}\right) + n\log c^t e_n,$$

ou encore :

$$\frac{f(x^k)}{n} < \frac{-k\left(\psi_1(1) + \frac{nr^2}{10}\right) + n\log c^t e_n}{n}, \tag{3.3}$$

d'après le lemme (2) et (3.3), on a :

$$c^t x^k < \exp\left(\frac{f(x^k)}{n}\right) < \exp\left(\frac{-k\left(\psi_1(1) + \frac{nr^2}{10}\right) + n\log c^t e_n}{n}\right).$$

Rappelons que l'algorithme converge lorsque $c^t x^k$ soit inférieur à ε ($\varepsilon > 0$ assez petit). Il suffit donc de chercher k qui vérifie :

$$\exp\left(\frac{-k\left(\psi_1(1) + \frac{nr^2}{10}\right) + n\log c^t e_n}{n}\right) < \varepsilon, \tag{3.4}$$

ce qui revient à chercher k vérifiant :

$$\left(\frac{-k\left(\psi_1(1) + \frac{nr^2}{10}\right) + n\log c^t e_n}{n}\right) < \log\varepsilon, \tag{3.5}$$

comme $\psi_1(1) = 1 - \log 2$ l'inégalité (3.5) se traduit par :

$$k > \frac{n}{1 - \log 2 + \frac{nr^2}{10}} \log\frac{c^t e_n}{\varepsilon}, \tag{3.6}$$

finalement $c^t x^k \leq \varepsilon$ lorsque k vérifie l'inégalité (3.6), d'où le résultat.

4 Experimentations numériques

On présente dans cette partie des tests numériques comparatifs effectués sur differents exemples de programme linéaire pris de la litterature [4]. Nous avons testé l'algorithme de Ye-Lustig (cas où z^* est inconnue) en utilisant

différement deux alternatives, une fois le pas α_s de (Scherijver) et une autre fois le pas α_m de (M. Bouafia & D. Benterki), la précision ε est prise entre 10^{-4} et 10^{-6}.

Dans chaque cas, on note par k Le nombre d'itérations nécessaire pour l'optimalité.

4.0.1 Tableau comparatif

Exemple	(m, n)	**Nbr itérations** α_s **Scherijver**	**Nbr itérations** α_m **Bouafia & Benterki**
1	$(3, 4)$	$k = 18$	$k = 17$
2	$(3, 5)$	$k = 26$	$k = 25$
3	$(4, 7)$	$k = 36$	$k = 35$
4	$(5, 9)$	$k = 47$	$k = 46$
5	$(5, 11)$	$k = 46$	$k = 46$
6	$(6, 12)$	$k = 48$	$k = 47$
7	$(16, 26)$	$k = 71$	$k = 70$

Conclusion

L'étude faite sur les performances de l'algorithme de Karmarkar, montre que l'introduction du nouveau pas α_m a introduit une amélioration de la convergence polynomiale et a aboutit à une réduction du nombre des itérations.

Notons que lorsque la dimension du problème devient importante, le pas α_m ainsi que celui de Scherijver α_s convergent vers $\frac{1}{2}$ lorsque n tend vers l'infini. Il serait probablement intéressant d'insister beaucoup plus sur la forme analytique de la fonction potentiel toute en conservant ses bonnes propriétés dans le but de ramener le pas de déplacement au delà de 1 et obtenir de meilleurs résultats.

References

[1] M. Bouafia, Sur les performances numériques d'une variante de l'algorithme de Karmarkar, Mémoire de Magister, Université Ferhat Abbas, Sétif, (2011).

[2] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4, 373–395 (1984).

[3] A. Keraghel, *Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar,* Dissertation thesis , Université Joseph Fourier, Grenoble, France (1989).

[4] A. Keraghel, D. Benterki, *Sur les performances de l'algorithme de Karmarkar pour la programmation linéaire,* Revue Roumaine des sciences techniques-Mécanique appliquées, Tome 46, n°.1, 87–96 (2001).

[5] I.J. Lustig, *A pratical approach to Karmarkar's algorithm*, Technical report sol 85-5 Systems optimizatioin laboratory; dep of operations res. Stanford. univ; Stanford California 94305 (1985).

[6] R.Naseri , A.Valinejad ,*An extended variant of Karmarkar's interior point algorithm* , Applied Mathematics and computation 184, 737–742 (2007).

[7] C. Roos,T. Terlaki, J. Vial, *Optimization Theory and Algorithm for Linear Programming Optimization* , Princton University, (2001).

[8] A. Scherijver, *Theory of linear and integer programming*, John Wiely & Sons, New York (1986).

IA

Construction et Génération d'un Graphe AoA en Tenant Compte des Contraintes Temporelles

[1]Nasser Eddine Mouhoub, Samir Akrouf, [3]Abdelhamid Benhocine,

[1] [1]Département d'informatique, Université de Bordj Bou Arréridj, 34265 Algérie
mouhoub_n@yahoo.fr
[1]Département d'informatique, Université de Bordj Bou Arréridj, 34265 Algérie
samir.akrouf@gmail.com
[3]Département Systèmes d'information, Université d'El Qassim, Arabie Saoudite
abdelhamid-benhocine@yahoo.fr

Abstract. Le problème d'ordonnancement consiste à organiser dans le temps un ensemble d'activités, de façon à satisfaire un ensemble de contraintes et optimiser le résultat. La contrainte temporelle touche au problème central de l'ordonnancement et le modifie, donc il n'aura plus ses caractéristiques. Notre travail consiste, pour résoudre ce problème, à modéliser par les graphes ce type de contraintes et le ramener à un problème classique d'ordonnancement ; ensuite appliquer une nouvelle technique de transformation d'un graphe AoN (Activities on Nodes) qui est unique avec un nombre important d'arcs et non préféré par les praticiens de gestion de projet vers un graphe AoA (Activities on Arcs) avec moins d'arcs et plus souhaitable. Le papier contient des concepts sur les graphes adjoints et un exemple illustratif de la méthode proposée.

Keywords: Graphe AoA, Graphe AoN, Graphe adjoint de graphe, Contrainte temporelle, Ordonnancement de projet.

1 Introduction

Dans les problèmes d'ordonnancement de projet, le suivi opérationnel des tâches est très important. Le chef de projet dresse le planning en utilisant, entre autre la modélisation par les graphes.

Le dessin du graphe AoN (Activities on Nodes) appelé également graphe de potentiels ou graphe français est facile à cause de son unicité malgré le nombre important d'arcs qu'il génère. Par contre le graphe AoA (Activities on Arcs) appelé également graphe PERT ou graphe américain est plus difficile à cause des tâches fictives qu'il génère. Or, les praticiens préfèrent travailler avec le graphe AoA parce qu'il est simple à lire puisque chaque tâche est représentée par un arc.

Les spécialistes qui insistent sur l'utilisation du graphe AoA présentent un certain nombre d'arguments qui justifient leur choix. Selon Fink et al. [1], le graphe AoA est plus concis. En outre, Hendrickson et al. [2] expliquent qu'il est à proximité du célèbre diagramme de Gantt qui est utilisé jusqu'à présent, et selon Yuval et Sadeh [3], la structure du graphe AoA est beaucoup plus adaptée à certaines techniques d'analyse et d'optimisation des formulations. Dans ce papier, on s'intéresse à trouver

une méthode qui permet de passer d'un graphe facile (graphe AoN) vers le graphe AoA qui sera correct et qui respectera la table d'ordonnancement tout en tenant compte des contraintes d'antériorité et des contraintes temporelles également. Cette méthode sera une ébauche pour la construction d'un algorithme qui réalisera le passage d'un graphe AoN vers le graphe AoA en tenant compte des contraintes temporelles.

2 L'ordonnancement de projet

Les contraintes auxquelles sont soumises les diverses tâches qui concourent à la réalisation d'un projet sont de divers types. On distingue les contraintes potentielles, disjonctives et cumulatives. Les contraintes potentielles sont de deux sortes:

- Les *contraintes d'antériorité* selon lesquelles une tâche j ne peut commencer avant qu'une tâche i ne soit terminée, par exemple, la construction des piliers suit les fondations.
- Les *contraintes temporelles* impliquant qu'une tâche donnée i ne peut débuter avant une date imposée, ou qu'elle peut s'achever après une date imposée.

Le problème d'ordonnancement avec des contraintes potentielles uniquement est appelé *problème central de l'ordonnancement* ou *ordonnancement de projet*.

2.1 Le graphe AoN

Les tâches sont symbolisées par des sommets auxquels on donne le même code, 2 sommets u et v sont reliés par un arc de u vers v si et seulement si la tâche u précède la tâche v.

On porte en suite sur chaque arc incident extérieurement à un sommet u la durée de la tâche u correspondante, de sorte que les valeurs sur les arcs de même extrémité initiale soient égales [4].

Fig. 1. La tâche u précède la tâche v dans le graphe AoN.

2.2 Le graphe AoA

Une tâche est représentée par un arc auquel on donne le même code, deux arcs u et v sont dessinés de telle sorte que T(u) = I(v) (voir Fig. 2 (b)) si et seulement si la tâche u précède la tâche v. T(u) et I(v) correspondent aux extrémités finale et initiale respectivement de l'arc u. Ces extrémités initiale et terminale d'un arc sont respectivement *les événements début de tâche et fin de tâche,* elles sont appelées étape (voir Fig. 2 (a)). Les durées t_i sont portées sur les arcs correspondants.

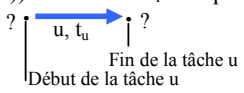


Fig. 2. (a).

Fig. 2. (b).

Fig. 2. La tâche u précède la tâche v dans le graphe AoA.

Pour dessiner le graphe AoA, on balaye la table d'ordonnancement selon la colonne des codes et la tâche correspondant à la ligne de balayage en cours est rajoutée en tenant compte des antériorités.

2.2.1 La tâche fictive dans le graphe AoA

Le graphe AoA (Fig. 3. (a)) du sous tableau des antériorités (Tab. 1.) est faux ; pour y remédier on introduit une tâche supplémentaire de durée 0 et qui n'influe pas sur la durée totale du projet. Cette tâche est appelée *tâche fictive*. On modifie alors le dessin de la Fig. 3. (a) et le problème est résolu (voir Fig. 3. (b)).

L'introduction des tâches fictives permet de solutionner certaines situations et de lever les ambiguïtés. Elles ne mettent en jeu aucun moyen matériel ou financier dans la gestion de projet.

Code	Anté.
c	a, b
d	b

Tab.1 Un sous-tableau des antériorités de c et d.

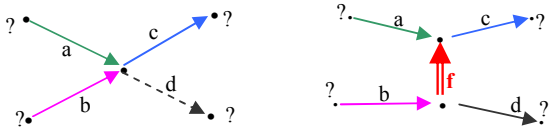


Fig. 3. (a) et (b) Problème de représentation dans le graphe AoA avec introduction de la tâche fictive f dans le nouveau graphe AoA.

3 Le graphe adjoint de graphe

Soit G=(X, U) un graphe orienté simple ou multiple. On construit à partir de G un graphe ou line graphe noté L(G), appelé graphe adjoint (ou graphe représentatif des arcs) de G comme suit [5]:

- Les sommets de L (G) sont en correspondance biunivoque avec les arcs de G. Pour des raisons de simplicité, on donne le même nom aux arcs de G et aux sommets correspondants de L (G).
- 2 sommets u et v de L (G) sont reliés par un arc de u vers v si et seulement si les arcs u et v de G sont tels que l'extrémité terminale de u coïncide avec l'extrémité initiale de v c.à.d. T(u)= I(v). Soit le graphe G suivant (Fig. 4.):

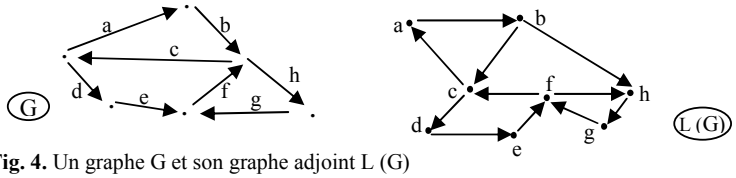


Fig. 4. Un graphe G et son graphe adjoint L (G)

Cette représentation est une des notions de dualité qui est bien connue en théorie des graphes.

Par définition, tout graphe G admet un graphe adjoint L (G) unique. Par contre, deux graphes non isomorphes peuvent avoir le même graphe adjoint.

On pose le problème inverse suivant:

Etant donné un graphe H, est-il le graphe adjoint d'un graphe? Autrement dit, existe-t-il un graphe G tel que L (G) est isomorphe à H, où H = L (G)?

Avant de répondre à cette question, nous définissons la notion de la configuration 'Z'.

G admet une *configuration* 'Z' qui est un sous graphe de G (voir fig. 5.) si G contient 4 sommets a, b, c et d tels que si (a, c), (b, c) et (b, d) sont des arcs de G, alors (a, d) *n'est pas un arc* de G. Donnons le nom de *barre* de 'Z' à l'arc (b, c).

La configuration 'Z' apparaît lorsque 2 sommets ont des successeurs communs et des successeurs non communs ou par symétrie lorsque 2 sommets ont des prédécesseurs communs et des prédécesseurs non communs [6].

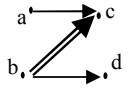


Fig. 5. La configuration « Z ».

3.1 Théorème

H est le graphe adjoint d'un graphe G sans boucles si et seulement si :
- H ne contient aucune configuration 'Z'
- **L**es arcs de H peuvent être partitionnés en bipartis complets $B_i = (X_i, Y_i)$, i=1,…, m, tels que $X_i \bigcap X_j = \phi$ et $Y_i \bigcap Y_j = \phi$, $\forall i \neq j$.

Les bipartis B_i de H sont alors en bijection avec les sommets notés aussi B_i qui ne sont ni sources ni puits, deux sommets B_i et B_j de G étant reliés par un arc de B_i vers B_j si et seulement si les bipartis complets B_i et B_j de H sont tels que $Y_i \bigcap X_j = \phi$.

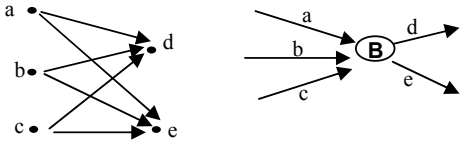


Fig. 6. Un biparti complet B de H l'étoile de G

- Toute paire de sommets ayant des successeurs communs ont tous leurs successeurs communs.
- Toute paire de sommets ayant des prédécesseurs communs ont tous leurs prédécesseurs communs.

Pour plus de détails sur ce théorème, le lecteur peut se référer à [6].

Ainsi H n'est le graphe adjoint d'aucun graphe si et seulement si il existe une paire de sommets ayant des successeurs communs et des successeurs non communs ou des prédécesseurs communs et des prédécesseurs non communs. En d'autres termes H n'est le graphe adjoint d'aucun graphe s'il contient une ou plusieurs configurations 'Z' [7].

On se pose alors le problème de savoir comment transformer H pour en faire un nouveau graphe qui est le graphe adjoint de G.

4 Passage du graphe AoN au graphe AoA

A cause de la facilité de dessin du graphe AoN, on doit se concentrer sur l'étude de la possibilité de transformer le graphe des potentiels (nombre d'arcs important) au graphe AoA (nombre d'arcs réduit) [8].

On se pose alors le problème de savoir comment transformer le graphe adjoint H (qui est le graphe AoN) pour obtenir le nouveau graphe G qui est le graphe AoA.

Le problème qui se pose : est ce que H contient des configurations 'Z' ou non ? S'il ne contient pas des 'Z' il est alors adjoint d'un graphe G, sa transformation est alors immédiate. Mais s'il contient des 'Z' on est amené à éliminer la barre de chaque 'Z' en préservant naturellement les contraintes de succession uniquement. Etudions chaque cas à part :

4.1 Cas où le graphe AoN est un graphe adjoint

Dans ce cas le graphe AoN ne contient pas de configurations 'Z'. On Construit le graphe AoA à partir du graphe AoN dans le cas où celui-ci est un graphe adjoint. En vertu des résultats du paragraphe 3.1, on procède comme suit :

On partitionne les arcs du graphe des potentiels en bipartis complets $B_i = (X_i, Y_i)$. Dans le graphe AoA que l'on veut construire, chaque B_i est représenté par un sommet encore noté B_i et sera le centre de l'étoile tel qu'il est montré dans la Fig. 6.

4.2 Cas où le graphe AoN n'est pas un graphe adjoint

Dans ce cas le graphe AoN contient des configurations 'Z'. La construction du graphe AoA est cependant plus complexe car il n'admet pas de partition des arcs en bipartis complets. C'est dans ce cas où l'on doit le modifier afin de le transformer en graphe adjoint en préservant les contraintes d'antériorités.

Supposons que les tâches $a_1,\ldots,a_m$ précèdent les tâches $b_1,\ldots b_n$. Dans le graphe AoN, ces contraintes d'antériorité sont représentées par un biparti complet. Dans le graphe AoA, elles sont représentées par une étoile.

Code	Antériorités
b_1	$a_1, \ldots, a_m$
.	.
b_n	$a_1\ldots., a_m$

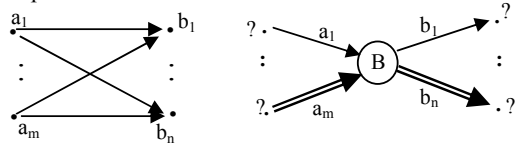


Fig. 7. Le sous-tableau des antériorités de $b_1,\ldots..b_n$, le biparti complet $B = (\{a_1,\ldots, a_m\}, \{b_1,\ldots.., b_m\})$ dans le graphe AoN et le sommet B du graphe AoA correspondant au biparti B.

Revenons maintenant au problème de la tâche fictive dans le graphe AoA. Si on a par exemple 4 tâches a, b, c et d avec les contraintes d'antériorité suivantes : a et b précèdent c, mais d est précédée par b uniquement.

Dans le graphe AoN, il n'y a aucun problème pour la représentation de ces tâches. Elle est faite comme dans la Fig. 7. Or, pour le passage du graphe AoN (qui est considéré comme le graphe adjoint H), on est obligé d'éliminer toutes les

configurations ‘Z’. On introduit alors, dans le graphe AoN une tâche fictive f dans tout ‘Z’ comme dans la Fig. 8.

Fig 8. La configuration « Z » et sa transformation dans le graphe AoN avec la partition des arcs en bipartis complets.

L’introduction des tâches fictives vise donc à *éliminer* toutes les configurations ‘ Z ‘ du graphe AoN, les contraintes restant inchangées. Il faut rappeler que les tâches fictives ne sont nullement nécessaires dans le graphe des potentiels mais ne sont introduites que pour construire le graphe AoA.

On pose alors le problème de la recherche des ‘ Z ‘ dans le graphe AoN, c’est-à-dire des sommets ayant des successeurs communs et des successeurs non communs ou des sommets ayant des prédécesseurs communs et des prédécesseurs non communs [9].

5 Les contraintes temporelles dans l’ordonnancement de projet

La contrainte temporelle (appelée contrainte de durée) est une contrainte de temps alloué. Elle est issue d’impératif de gestion tel que la disponibilité des approvisionnements ou le délai de livraison, etc.

Elle précise l’intervalle de temps (ou le semi-intervalle) pendant lequel il est possible de réaliser une tâche. Ces contraintes sont souvent dues aux disponibilités des intervenants (ressources humaines) : l’entreprise qui réalise la charpente ne peut intervenir qu’entre le 15 juin et le 31 août [1].

La contrainte temporelle touche au problème central de l’ordonnancement et le modifie. Il n’a plus les caractéristiques du problème central. Le problème donc, consiste à chercher un moyen ou une technique qui permet de normaliser la situation et le ramener au problème central.

Dans ce qui suit, nous allons proposer une méthode originale qui nous permet de modéliser les contraintes temporelles et les inclure dans le problème central de l’ordonnancement.

On peut classer les contraintes temporelles les plus importantes en six types et en ajoutant la contrainte d’antériorité elles deviennent sept, qui sont :

(C1) Une tâche A commence t unités de temps avant le début des travaux.

(C2) Une tâche A ne peut commencer que t unités de temps après le début des travaux.

(C3) La tâche B doit commencer t unités de temps après la fin de la tâche A.

(C4) La tâche B commence une fraction de temps a/b après le début de la tâche A ($a<b$).

(C5) La tâche B doit commencer un temps t après le début de la tâche A ($t<t_A$).

(C6) La tâche A doit commencer avant la date t.

(C7) La tâche B doit suivre immédiatement la tâche A.

5.1 Modélisation des contraintes temporelles

Dans le problème central et particulièrement dans le graphe AoN, les arcs incidents extérieurement d'un sommet (c'est-à-dire d'une tâche) ont la même valeur.

La présence des contraintes temporelles dans le graphe AoN bafoue cette propriété, ce qui rend la résolution du problème central impossible. Le calcul des dates et la recherche du chemin critique… deviennent impossibles également.

La modélisation par les graphes peut solutionner ce type de problème. Nous présenterons dans ce qui suit une technique nouvelle qui permet de prendre en charge ce genre de contraintes.

5.1.1 Les contraintes temporelles dans le graphe AoN

La figure suivante (Fig. 9.) donne la représentation unique de ces contraintes dans le graphe AoN. On voit bien que les valeurs sur les arcs incidents extérieurement à un sommet sont différentes (voir l'exemple de la figure 11. (a)). Là, on sort du problème central.

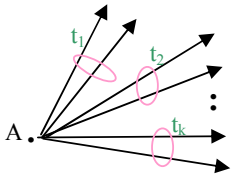
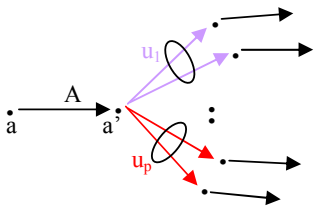


Fig. 9. Tout type de contrainte temporelle dans le graphe AoN.

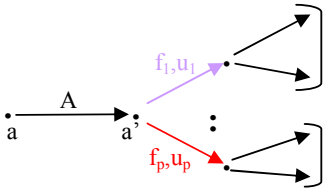
5.1.2 Les contraintes temporelles dans le graphe AoA

Pour dessiner ces contraintes dans le graphe AoA, il convient de les étudier en groupe de deux. Il importe également d'introduire de nouvelles tâches et d'en subdiviser certaines.

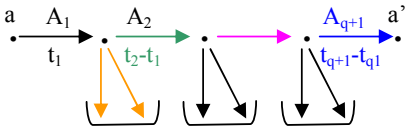
Ensembles, ces contraintes sont représentées comme l'indique la Fig. 11., les t_i et les u_i étant des durées qu'on suppose telles que $t_1<t_2<\ldots<t_k$ et $u_1<u_2<\ldots<u_p$. u_i étant des tâches fictives.



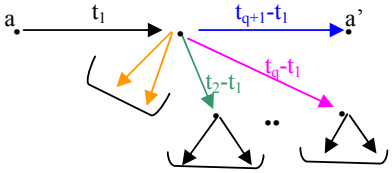
(a). Une représentation des contraintes de type (C2) et (C3) dans le graphe AoA.



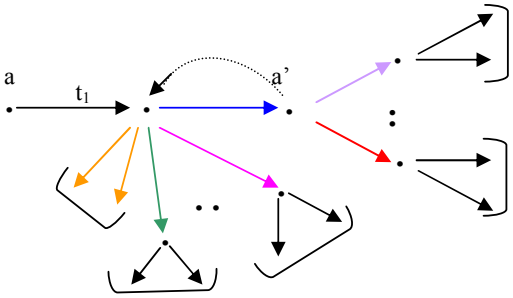
(b). Une autre représentation des contraintes de type (C2) et (C3 dans le graphe AoA avec moins de tâches que (b).



(c). Une représentation des contraintes de type (C4) et (C5) dans le graphe AoA.



(d). Une autre représentation des contraintes de type (C4) et (C5) dans le graphe AoA avec autant de tâches que (c).



(e). Tout type de contraintes dans graphe PERT en combinant (c) et (d).

Fig. 10. Représentation des contraintes temporelles dans le graphe AoA.

La Fig. 10. (a). montre que dans le graphe AoA, chaque tâche qui vient après la tâche A, a sa propre tâche fictive u_i. Cette représentation est médiocre puisque le nombre de tâches fictives risque d'être très important, ce qui encombre le graphe.

Une représentation meilleure (Fig. 10. (b).) regroupe, en une seule tâche fictive, plusieurs tâches fictives des tâches réelles (après A) et qui ont la même valeur.

Signalons que la tâche fictive dans ce contexte**,** n'a pas une durée nulle**.** Elle est introduite pour solutionner ce problème et introduire les contraintes de durée dans le problème central.

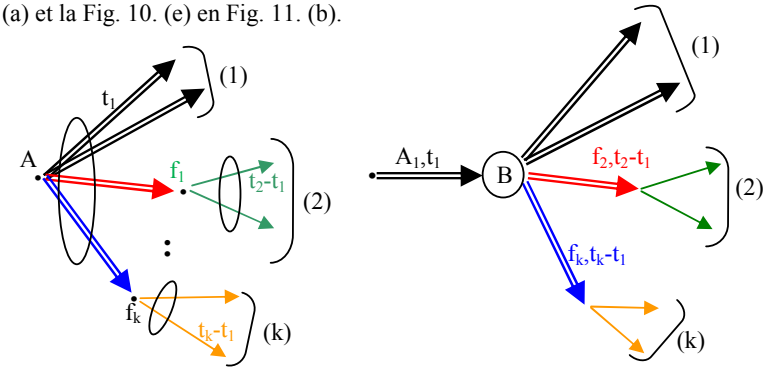
Pour les contraintes de type (C4) et (C5), on remarque que les deux débutent après un certain temps du début de la tâche A. la représentation dans le graphe AoA implique la segmentation de A en plusieurs sous tâches dans le cas général ($A = A_1 + A_2 + \ldots + A_{q+1}$). Deux modélisations de ces contraintes sont possibles (Fig. 10 (c). et 10. (d).). Nous remarquons que la représentation de la Fig. 10. (d) est plus commode.

Enfin, nous pouvons combiner les Fig. 10. (b) et 10. (d) tout en gardant l'idée de minimisation des tâches fictives.

En conclusion, pour arriver à la Fig. 10. (e) il faut modifier, dans le graphe AoN, les arcs incidents extérieurement d'un sommet et qui n'ont pas la même valeur, par l'introduction de tâches fictives de durée 0 afin de pouvoir partitionner le graphe AoN en bipartis complets.

Toutes ces combinaisons précédentes nous conduisent aux dernières transformations dans les deux graphes AoN et AoA respectivement (Fig. 11.) :

La correspondance entre les représentations des contraintes temporelles dans le graphe AoN et dans le graphe AoA étant notre objectif, on modifie la Fig. 9. en Fig. 11. (a) et la Fig. 10. (e) en Fig. 11. (b).



(a) Modification dans le graphe AoN. **(b)** Correspondance dans le graphe AoA.

Fig. 11. Représentation des différentes contraintes temporelles dans les graphes AoN et AoA.

L'introduction des tâches f_2,…, f_k de durées t_2-t_1, …, t_k-t_1 a pour avantage de donner la même valeur aux arcs de même extrémité initiale dans le graphe AoN. Il n'y a alors aucune difficulté à vérifier que les arcs du graphe (Fig. 11. (a)) sont partitionnés en bipartis complets et qu'il est le graphe adjoint du graphe en (Fig. 11. (b)).
A titre d'exemple, soit A une tâche de durée 5 unités de temps. Supposons que:

A précède B,

B_1 et B_2 ne peuvent commencer qu'une unité de temps après le début de la tâche A,

B_3 et B_4 ne peuvent commencer que 4 unités de temps après le début de A

B_5 ne peut commencer que lorsque A est terminé au ¾,

B_6 et B_7 ne peuvent commencer que 6 unités de temps après la fin de A.

Dans le graphe AoN, dessinons les arcs sortants du sommet A :

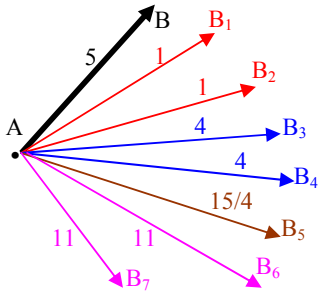


Fig. 11. (a) Aucune modification dans le graphe AoN.

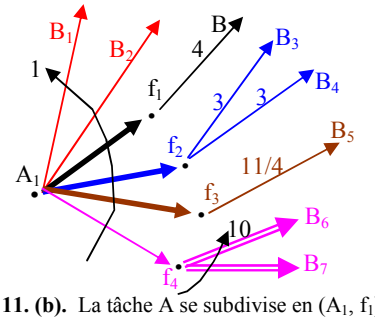


Fig. 11. (b). La tâche A se subdivise en (A_1, f_1) dans le graphe AoN. Les arcs de même extrémité initiale ont la même valeur.

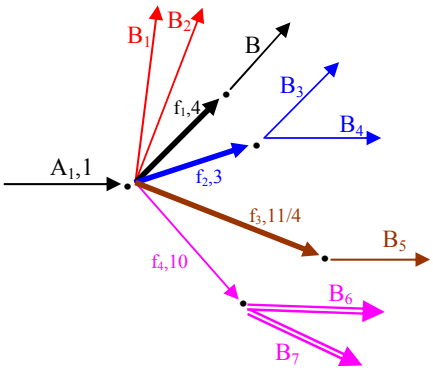


Figure 11. (c). Correspondance dans le graphe AoA.

Pour illustrer ce qu'on a vu depuis le début de ce papier et en vue de construire le graphe AoA à partir du graphe AoN en tenant compte des contraintes temporelles, on considére l'exemple suivant :

6 Exemple

Le tableau Tab. 2. ci-contre donne les contraintes d'antériorités. Les contraintes temporelles sont :

- B ne peut commencer que 3 unités de temps après le début des travaux.
- C ne peut commencer que 7 unités de temps après le début des travaux.
- E commence lorsque C est exécuté aux ¾
- G commence 4 unités de temps après la fin de E.

Code s	Durée	Antériorités
A	6	-
B	5	-
C	8	-
D	4	A
E	3	A, B, C
F	9	C
G	2	D, E
H	8	E, F
I	5	E, F

Tab. 2. Les contraintes d'antériorités.

Les graphes de la Fig. 12. (a, b, c, d) montrent les modifications dans le graphe AoN, ensuite la construction du graphe AoA:

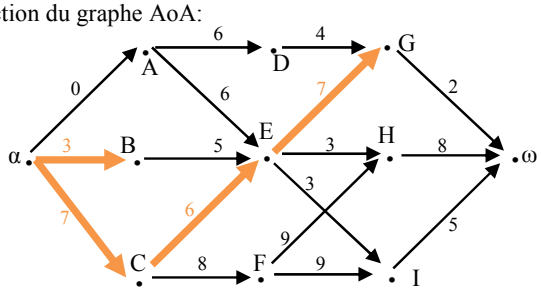


Fig. 12. (a) Le graphe AoN initial de la table d'ordonnancement Tab. 2. (Les arcs en gras représentent les contraintes temporelles).

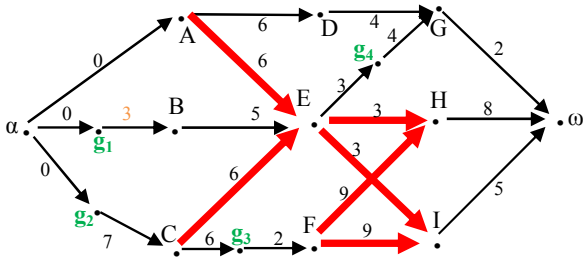


Fig. 12. (b) Le graphe AoN dont les arcs de même extrémité initiale ont la même valeur. Les tâches fictives g_i proviennent des contraintes temporelles: Les tâches α, C, E sont subdivisées deux tâches chacune. Les barres des 'Z' sont en gras.

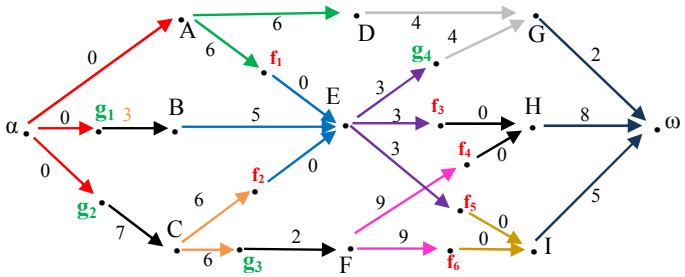


Fig. 12. (c) Le graphe AoN sans aucune configuration 'Z' et dont les sommets sont réorganisés en niveaux. On vérifie que les arcs peuvent être partitionnés en bipartis complets.

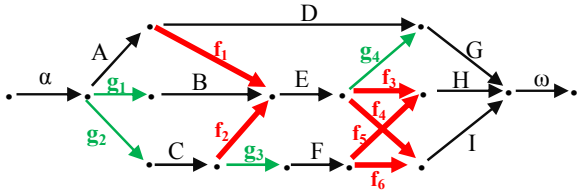


Fig. 12. (d) Le graphe AoA de la table Tab. 2. Les durées des tâches non reprises (les f_i « en gras » de durées zéro).

7 Conclusion

Ce travail vient d'introduire les graphes adjoints dans les problèmes d'ordonnancement de projet avec ou sans la présence des 'Z' dans le graphe AoN et

ceci pour la construction du graphe AoA. Il a également pris en charge la modélisation des contraintes temporelles qui permet de les inclure dans le problème central où la résolution devient plus facile c. à. d. le calcul des dates au plus tôt, au plus tard, les marges libres, le chemin critique, etc. qui devient possible par l'application de l'algorithme de Bellman.

Ce travail ouvre la voie à des perspectives, tel que la recherche du graphe PERT minimal en termes de tâches fictives qui est un problème NP-difficile ou en terme de sommets. L'ordonnancement de projet à moyens limités peut être envisagé en utilisant la modélisation par les graphes.

References

1. Fink, G.: Recherche opérationnelle et réseaux, Lavoisier, Paris, (2002).
2. Henderckson, C., Project Management for Construction, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA l52l3, Version 2.2, (2008).
3. Cohen, Y., Sadeh, A.: A New Approach for Constructing and Generating AoA Networks, Journal of computer science, 1-1, (2007). *http://www.scientificjournals.org/journals2007/articles/1049.htm*
4. Esquirol, P. & P. Lopez, P.: l'ordonnancement, ECONOMICA, Paris, France, ISBN 2-7178-3798-1, (1999).
5. Mouhoub, N. E., Belouadah, H. & Boubetra A.: Algorithme de construction d'un graphe Pert à partir d'un graphe des potentiels donné, STUDIA UNIV. BABES BOLYAI, INFORMATICA, Volume LI, Number 2, (2006).
6. C. Heuchenne, Sur une certaine correspondance entre graphes, Bull. Soc. Roy. Sci. Liege, 743-753, 33(1964).
7. Mouhoub, N. E., Benhocine, A. & Belouadah, H.: A new method for constructing a minimal PERT network, (APM) Applied Mathematical modelling, Elsevier ISSN: 0307904X, Vol. 35, Issue: 9, 4575-4588, (2011).
8. J. E. Kelley, Critical path planning and scheduling-Mathematical basis, Operations research, vol. 9, n° 3, p. 296-320, (1961).
9. Mouhoub, N. E., Akrouf, S.: Approche optimisée dans les problèmes d'ordonnancement de projet, ANL'07, Actes du colloque international sur l'analyse non linéaire, Sétif university, Algeria, (2007).

A new semantics for logic programming capturing the stable model semantics

Belaïd Benhamou and Pierre Siegel

Aix-Marseille Université
Laboratoire des Sciences de l'Information et des Systèmes (LSIS)
Centre de Mathématiques et d'Informatique
39, rue Joliot Curie - 13453 Marseille cedex 13, France
{Belaid.Benhamou;siegel}@cmi.univ-mrs.fr

Abstract. Answer set programming is a well studied framework in logic programming. Many research works had been done in order to define a semantics for logic programs. Most of these semantics are iterated fixed point semantics. The main idea is the canonical model approach which is a declarative semantics for logic programs that can be defined by selecting for each program one of its canonical models. The notion of canonical models of a logic program is what it is called the stable models. The stable models of a logic program are in a certain sense the minimal Herbrand models of its "reduct" programs. Here we introduce a new semantics for logic programs that is different from the known fixed point semantics. In our approach, logic programs are expressed as CNF formulas (sets of clauses) of a propositional logic for which we define a notion of extension. We prove in this semantics, that each consistent CNF formula admits at least an extension and for each given stable model of a logic program there exists an extension of its corresponding CNF formula which logically entails it. On the other hand, we show that some of the extensions do not entail any stable model, in this case, we define a simple condition which allows to recognize such extensions. These extensions could be very important, but are not captured by the stable models semantics. Our approach, extends the stable model semantics in this sens. Following the new semantics, we give a full characterization of the stable models of a logic program by means of the extensions of its CNF encoding verifying a simple condition, and provide a procedure which can be used to compute such extensions from which we deduce the stable models of the given logic program.

1 Introduction

The work we propose here investigates the Answer Set Programming (ASP) framework which can be considered as a sub-framework of the default logic [12]. One of the main questions in ASP, is to define a semantics to logic programs. A logic program π 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 π 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 [1] and the stable model for the answer set semantics [7]. It is well know that each answer set for a logic program is a model of its completion, but the converse, is not always true. Fages in [6] showed that both semantics are equivalent for free loop logic programs that are also 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 [5]. On the other hand Fangzhen Lin and Yutin Zhao proposed in [3] 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.

Almost all of the known semantics for logic programs are declarative fixed point semantics. Here, we introduce a new semantics for logic programs which is different from the previous ones. More precisely, we investigate a theoretical study of a new semantics for general logic programs. This framework could be seen as a particular case of the default logic [12] or the hypothesis logic [14, 13] which give a full characterization of the default logic. In our approach, a ground logic program with negations as failure is completely expressed as a set of propositional logic clauses (a CNF formula) for which we define a notion of extension that generalizes and captures the answer set semantics. In our semantics, we prove that each consistent CNF formula admits at least an extension and the answer sets of the logic program match with a well identified sub-set of the set of extensions of its corresponding CNF encoding. That is, there is a bijection with the answer sets of a logic program and a sub-set of extensions of its CNF encoding satisfying a well defined condition that is very easy to verify. More precisely, we prove that for each given answer set of the logic program there exists an extension of its corresponding CNF formula which entails it. On the other hand, we show that some of the extensions do not entail any stable model and in this case, we give a simple condition which allows to recognize such extensions. A very interesting and intriguing fact here, is the extensions that do not match with answer sets. These extensions could be very important, but are not captured by the answer set semantics. For instance, we can find logic programs which do not contain any answer set because only of one or few of its rules, but the other rules could infer some important sets that we could miss. We give then a full characterization of the answer sets of a logic program by means of the extensions of its CNF encoding verifying a simple condition. Our approach is in this sens, an extension of the stable model semantics. Following

this new semantics, we give a new procedure which can be used to compute the set of extensions from which can be computed the answer sets of the given logic program. This procedure can be implemented as a slightly modified SAT solver which performs on the CNF encoding of the logic program and which uses a sub-set of variables as a strong backdoor [16] (denoted by STB). This, backdoor defines the complexity of the procedure and could be used to enhance its efficiency.

The rest of the paper is structured as follow: in section 2 we introduce the necessary background on answer set programming. We define the new semantics for logic programs and show its relationship with the answer set semantics in Section 3. In section 4 we define a SAT based method that can be used to compute the extensions from which the answer sets of the logic program can be deduced. We give a conclusion in Section 5.

2 Background

We summarize in this section some background on the answer set programming framework. There exist several variants of logic programs depending on the fact if they include classic negations, and negation as failure or not. We assume here that the logic programs are ground[1] (without variables). The most known classes are the following:

- *Positive logic programs:* a positive logic program π is a set of rules of the form $r : L_0 \leftarrow L_1, \ldots, L_m$, $(m \geq 0)$ where L_i $(0 \leq i \leq m)$ is an atom. In other words a positive program is a logic program that does not contain classic negations and negations as failure. It can be seen as a Prolog program without negations as failure or a set of Horn clauses. Here, we have $head(r) = L_0$ and $body(r) = L_1, \ldots, L_m$. The meaning of the previous rule is "if we can prove the body $L_1, L_2, \ldots, L_m$ then we will conclude the head L_0". Given a set of atoms A, we say that a rule r is applicable (active) in A if $body(r) \subseteq A$. A set of atoms A is closed with respect to a logic program π if only if for each rule $r \in \pi$, if $body(r) \subseteq A$, then $head(r) \in A$. A positive logic program π contains one canonical Herbrand model which is the single minimal Herbrand model that we denote by $CM(\pi)$. The minimal Herbrand model of π is the least set of atoms that is closed with respect to π. Formally, for a given logic program π and a set of atoms A the operator $T_\pi(A) = \{head(r)/r \in \pi, body(r) \subseteq A\}$ compute all the atoms that can be deduced from A by using the rules of π. Now let us define the suite $T_\pi^0 = T_\pi(\emptyset)$, $T_\pi^{k+1} = T_\pi(T_\pi^k), \forall k \geq 0$. The Minimal Herbrand model is the least fixed point of T_π, that is $CM(\pi) = \bigcup_{k \geq 0} T_\pi^k$. Thus, the minimal Herbrand model $CM(\pi)$ includes all the atoms that can be deduced from π. This model is exactly the minimal model expressed by a Prolog program formed by the rules of π or the minimal model of the Horn clauses.

[1] Each logic program π 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.

– *General logic programs:* A general logic program π 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* used for logic programs 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 any 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 π with respect to a given set A of atoms is the positive program π^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 [7] 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 π if and only if A is identical to the minimal Herbrand model of π^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 interpreted to false.

– *Extended logic programs:* An extended logic program π 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 forall $(0 \leq i \leq n)$ are literals (atoms L_i or their negations $\neg L_i$). The semantics of the extended logic programs [8] is an extension of the stable models semantics defined for general logic programs [7]. First consider an extended program π that do not contain negations as failure (m=n, in each rule of π). The answer set of π is the smallest set of literals A such that for each rule $L_0 \leftarrow L_1, \ldots, L_m$ of π, if $L_1, \ldots, L_m$ are in A, then L_0 will be in A, and if two complementary literals are in A, then A should contain all the literals in the language of π. We will denote this answer set by $A = \alpha(\pi)$. It is easy to see that there is only one single model $\alpha(\pi)$, and if π does not contain negations, then the model $\alpha(\pi)$ is identical the minimal Herbrand model of π. Now let π be any extended program and Lit the set of literals of its language. For any subset A of literals, the reduct of the program π with respect to A is the program π^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. The resulting program π^A does not contain *not*, and the subset A is an answer set for π if and only if it is identical to the answer set of π^A. That is $A = \alpha(\pi^A)$. Here the semantics is different from the one of general programs, since there is classic negation, it is not based on the wold closed assumption.

3 The negation as failure expressed in propositional logic and the notion of extension

Our semantics follows a logic approach that concerns a propositional logic language L where we distinguish two types of propositional variables: a subset of classical variables $V = \{L_i : L_i \in L\}$ and another subset $nV = \{notL_i : notL_i \in L\}$. For each variable in $L_i \in V$ there exists a corresponding variable $notL_i \in nV$. The total set of variables of L is $V \cup nV$. We use the symbol $notL_i$ to express the negations as failure used in logic programs. At this level both kind of variables L_i and $notL_i$ are independent. That is, there is no semantics relation defined between both of them.

Since the introduction of Prolog in 1972 and its negation as failure, the purpose of logic programming is to give a relationship between these two kinds of variables. Particularly, the defaults logic [12], or the hypothesis logic [14, 13] represent two ways to formalize the relationship. Generally, all the non-monotonic logics work in this direction.

Here, we introduce a semantics which gives the relationship between both kind of variables in the framework of answer set programming. This is expressed by adding to the propositional logic language L the set of negative clauses ME=$\{(\neg L_i \vee \neg notL_i) : L_i \in V\}$ which is equivalent to all of the sets of formulas $\{\neg(L_i \wedge notL_i) : L_i \in V\}$, $\{(L_i \rightarrow \neg notL_i) : L_i \in V\}$ and $\{(notL_i \rightarrow \neg L_i) : L_i \in V\}$. This is a pseudo axiom scheme which should be applied only for the variables L_i of V. It is important to see that the generated set ME of negative clauses expresses only the mutual exclusion between each literal $L_i \in V$ and its correspondent literal $notL_i \in nV$, but does not establish the equivalence between $\neg L_i$ and $notL_i$.

Given a set of formulas F expressed in this propositional language L, and a sub-set S of nV, we define an extension of $(F \cup ME, S)$ as the theory obtained from $F \cup ME$ by adding a maximal number of literals $notL_i$ of S to $F \cup ME$ such that the resulting theory remains consistent. In other word, an extension of $(F \cup ME, S)$ is a maximal consistent theory with respect to inclusion of literals $notL_i$ of S. Formally:

Definition 1. *Given a set of formulas F of a the language L, a sub-set S of nV and a sub-set S' of S, an extension of $(F \cup ME, S)$ is a set $E = (F \cup ME) \cup S'$ such that the following conditions hold:*

1. *E is consistent.*
2. *$\forall notL_i \in S - S', E \cup \{notL_i\}$ is inconsistent*

Example 1. Let $F = \{(notb \wedge c) \rightarrow a, a \rightarrow b, notd \rightarrow c, a\}$ be a set of formulas of the language L, $ME = \{\neg a \vee \neg nota, \neg b \vee \neg notb, \neg c \vee \neg notc, \neg d \vee \neg notd\}$ and $S = \{notb, notd\}$ a sub-set of variables of nV. The pair $(F \cup ME, S)$ admits a single extension $E = (F \cup ME) \cup \{notd\}$.

Example 2. Let $F = \{(notb \wedge c) \rightarrow a, a \rightarrow b, notd \rightarrow c\}$ be a set of formulas of the language L, $ME = \{\neg a \vee \neg nota, \neg b \vee \neg notb, \neg c \vee \neg notc, \neg d \vee \neg notd\}$, and

$S = \{notb, notd\}$ *a sub-set of variables of* nV*. The pair* $(F \cup ME, S)$ *admits two extensions* $E = (F \cup ME) \cup \{notd\}$ *and* $E = (F \cup ME) \cup \{notb\}$*.*

For each consistent set of formulas of L including the clauses of ME, we can prove that it admits at least an extension forall sub-set $S \subset nV$.

Proposition 1. *Let* F *be a set of formulas of the language* L *and* S *a sub-set of* nV*. If* $(F \cup ME)$ *is consistent, then there exists at least an extension of* $(F \cup ME, S)$ *forall sub-set* $S \subset nV$*.*

Proof. Let 's S be a subset of nV. We have to study to cases: If S is finite, then the proof is trivial. Indeed, since $(E \cup ME)$ is consistent by the hypothesis, then by adding successively the literals $notL_i$ to $(E \cup ME)$ we shall reach the maximal consistent set $(E \cup ME) \cup S'$ where $S' \subset S$. If S is not finite, the proof can be achieved by using both the compactness theorem and the Zorn Lemma (it is not given here for space reason).

Proposition 2. *If* F *is a set of clauses that contain at least a positive literal of* V *and do not include any positive literal* $notL_i$ *of* nV*, then the set of clauses* $F \cup ME$ *is consistent.*

Proof. The interpretation which includes both all the positive literals $L_i \in V$ and all the literals $\neg notL_i \in nV$ is a trivial model.

3.1 A new semantics for logic programs

To show our theoretical results on logic programming, we focus our study on the general logic program class where a program π is expressed by 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 $(0 \leq i \leq n)$ is an atom and where we define the set of literals $STB = \{notL_i : notL_i \in \pi\} \subset nV$ as the strong backdoor [16] of π. In our approach, each rule r of π is expressed by the propositional logic formula (clause) $c : L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \ldots \neg notL_n$. We add to this clause, the set of mutual exclusion clauses $ME = \{(\neg L_i \vee \neg notL_i) : L_i \in V\}$.

A general logic program $\pi = \{r : L_0 \leftarrow L_1, L_2, \ldots, L_m, notL_{m+1}, \ldots, notL_n\}$, $(0 \leq m < n)$, is then expressed by the set of propositional clauses which represents its CNF encoding in the propositional language L:

$$L(\pi) = \{\bigcup_{r \in \pi}(L_0 \vee \neg L_1 \vee, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \ldots \neg notL_n)$$
$$\bigcup_{L_i \in V}(\neg L_i \vee \neg notL_i)\}$$

Given the encoding $L(\pi)$ of the logic program π expressed in this propositional language, and the strong backdoor sub-set $STB \subset nV$, we will focus in the sequel on the extensions of $(L(\pi), STB)$. An extension of $(L(\pi), STB)$ is the theory obtained from $L(\pi)$ by adding a maximal number of literals $notL_i \in STB$ to $L(\pi)$ such that the resulting theory remains consistent. Following Definition 1 we can deduce the following particular case definition for extension of logic programs :

Definition 2. *Given the logical encoding $L(\pi)$ of a logic program π, its strong backdoor STB, and a sub-set $S' \subset STB$, then $E = L(\pi) \cup S'$ is an extension of $(L(\pi), STB)$ if the following conditions hold:*

1. *E is consistent.*
2. *$\forall notL_i \in STB - S', E \cup \{notL_i\}$ is inconsistent*

Example 3. Now consider the formulas of the set F given in Example 1 as rules of a general logic program π. We give below both π and its logic encoding $L(\pi) = L(\pi) - Rules \cup L(\pi) - ME$:

π :	$L(\pi)$-Rules:	$L(\pi)$-ME:
$a \leftarrow c, notb$	$a \vee \neg c \vee \neg notb$	$\neg a \vee \neg nota$
$b \leftarrow a$	$b \vee \neg a$	$\neg b \vee \neg notb$
$c \leftarrow notd$	$c \vee \neg notd$	$\neg c \vee \neg notc$
$a \leftarrow$	a	$\neg d \vee \neg notd$

We can see that the strong backdoor set is $STB = \{notb, notd\}$. The pair $(L(\pi), STB)$ has a single extension $E = L(\pi) \cup \{notd\}$

Example 4. To express the set of formulas F of Example 2 as a general logic program, we take the logic program π of Example 3 and drop the rule $a \leftarrow$. We obtain the logic program $\pi' = \pi - \{a \leftarrow\}$ whose CNF logic encoding is $L(\pi') = L(\pi) - \{a\}$ and whose STB is the same the one of π. There is two extensions for $(L(\pi'), STB)$: $E_1 = L(\pi') \cup \{notd\}$ and $E_2 = L(\pi) \cup \{notb\}$

Theorem 1. *If X is a stable model (an answer set) of a logic program π, then there exist an extension E of $(L(\pi), STB)$ such that $X = \{L_i \in V : E \models L_i\}$ and which verify the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$.*

Proof. Let's give the set $E = L(\pi) \cup \{notL_i / L_i \notin X\}$. To prove the theorem we shall show that E is an extension (1 and 4 below), $X = \{L_i : E \models L_i\}$ (2 and 3) and E verifies the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ (5).

1. E is consistent: it is sufficient to show that the interpretation $I = \{\neg L_i, notL_i / L_i \notin X\} \cup \{L_i, \neg notL_i / L_i \in X\}$ is a model of E. As $\{notL_i / L_i \notin X\} \subset I$, it is sufficient to prove that I is a model of $L(\pi)$. We have to prove that each clause $(\neg L_i \vee \neg notL_i)$ of ME and each clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \cdots \vee \neg notL_i \cdots \vee \neg notL_n)$ of $L(\pi)$ corresponding to a rule in π is satisfied by I. It is trivial to see that each clause $(\neg L_i \vee \neg notL_i)$ of ME is satisfied by I. Indeed, even $L_i \in X$, and in this case we have $\neg notL_i \in I$ by definition of I. Thus, the clause is satisfied by I. Now, if $L_i \notin X$, then by definition of I we have $\neg L_i \in I$, thus I satisfies the clause $(\neg L_i \vee \neg notL_i)$. Therefore, $(\neg L_i \vee \neg notL_i)$ is satisfied by I in both cases. Now, to show that each clause of the form $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \cdots \vee \neg notL_i \cdots \vee \neg notL_n)$ is satisfied by I, we have to study two cases. If the sub-clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m)$ matches with a rule of the reduct π^X, then X satisfies $L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m$ since X is an answer set of π, thus I satisfies the clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m)$ since $X \subset I$. It results that the clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \cdots \vee \neg notL_i \cdots \vee \neg notL_n)$ is also satisfied by I. In the other case the sub-clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m)$ does not match with a rule of π^X. In this case, there exists an expression $notL_i$ in the negative body of the rule of π corresponding to the clause $(L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee$

$\neg notL_{m+1} \cdots \vee \neg notL_i \cdots \vee \neg notL_n$) of $L(\pi)$ such that $L_i \in X$. Thus, $\neg notL_i \in I$ by definition of I and the last clause is then satisfied by I. Therefore, I is a model of both $L(\pi)$ and E. We conclude that E is consistent.

2. $L_i \in X \Rightarrow E \models L_i$**:** by considering the set X we obtain the reduct π^X by first deleting in π each rule $r : L_0 \leftarrow L_1, L_2, \ldots, L_m, notL_{m+1}, \ldots, notL_n$ $(0 \leq m < n)$ having a literal $notL_i$ in its negative body such that $L_i \in X$. In this case the corresponding clause $c : L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \ldots \neg notL_n$ of the set E is not deleted. Secondly we suppress in the other rules of π each occurrence of $notL_i$ such that $L_i \notin X$, and in this case it follows from the definition of E that $notL_i \in E$, then we suppress the literals $\neg notL_i$ in the corresponding clauses of E. We deduce then that the reduct π^X is include in the set E^X obtained from E by applying unit resolution on the literals of the form $notL_i$ $(E \models E^X)$. As X is a stable model of π, then by definition, L_i belongs to the minimal Herbrand model of π^X. It follows that $\forall L_i \in X, \pi^X \models L_i$. As $\pi^X \subset E^X$, then $E^X \models \pi^X$ and then $E \models \pi^X$. Since the inference is monotonic, we deduce that $\forall L_i \in X, E \models L_i$.
3. $E \models L_i \Rightarrow L_i \in X$**:** it is equivalent to show that $L_i \notin X \Rightarrow E \nvDash L_i$. From $L_i \notin X$ we deduce by the definition of E that $notL_i \in E$, thus by application of unit resolution to the clause $(\neg L_i \vee \neg notL_i)$ generated by the new pseudo axiom we deduce that $E \models \neg L_i$. Therefore we conclude that $E \nvDash L_i$ since E is consistent (shown in 1).
4. E is maximal consistent with respect to STB: Let the set $E' = E \cup \{notL_i\}$, such that $notL_i \in STB$, but $notL_i \notin E$. As $notL_i \notin E$, by the definition of E, we deduce that $L_i \in X$. Now by applying the property that we have shown in 2), we get $E \models L_i$. Thus, $E' \models L_i$. By applying unit resolution on the clause $(\neg L_i \vee \neg notL_i)$ generated by the pseudo axiom, we deduce that $E' \models \neg notL_i$ and then E' is not consistent since it should infer both $notL_i$ and its complement $\neg notL_i$. We conclude that E is maximal consistent with respect to STB.
5. $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$: it is equivalent to prove that $(\forall L_i \in V, E \nvDash L_i \Rightarrow E \nvDash \neg notL_i)$. Suppose that $E \nvDash L_i$, as $X = \{L_i : E \models L_i\}$, we deduce that $L_i \notin X$, thus $notL_i \in E$ by the definition of E. As E is consistent, we conclude that $E \nvDash notL_i$.

Example 5. Take the logic program π of Example 3 and its CFN encoding $L(\pi)$. π has one stable model $X = \{a, b, c\}$ which matches with the single extension $E = L(\pi) \cup \{notd\}$ of $(L(\pi), STB)$. After the assignment of the STB variables, we use unit resolution to show that $E \models \{a, b, c, \neg d, \neg nota, \neg notb, \neg notc\}$. We can then see that E verifies the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ and $X = \{L_i \in V : E \models L_i\} = \{a, b, c\}$.

Remark 1. Since $L(\pi)$ is a consistent set of horn clauses (Proposition 2), all the literals $L_i \in V$ that could be deduced form an extension E of $(L(\pi), STB)$ are inferred by unit resolution. In particular, the set of positive literals $L_i \in X$ which represent a stable model of π when the extension E verifies the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ are inferred from E by unit resolution.

For instance, the literals of the stable model $X = \{a, b, c\}$ of the logic program π of Example 3 are inferred from the extension $E = L(\pi) \cup \{notd\}$ of $(L(\pi), STB)$ by unit resolution.

Remark 2. By using Proposition 1, we deduce that in our semantics there always exists an extension of $(L(\pi), STB)$. But, there is not always a stable model for the program

π. It could exists then some extensions (extra-extensions) of $(L(\pi), STB)$ which do not entail any stable model of π. These extensions E are exactly those that do not satisfy the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$. Such extensions could be very important, but are not captured by the stable models semantics. Our approach, extends the stable model semantics in this sens.

For instance, in Example 4, there is two extensions for $(L(\pi'), STB)$: $E_1 = L(\pi') \cup \{notd\}$ and $E_2 = L(\pi) \cup \{notb\}$, but there is no stable model for the corresponding logic program π'. By using unit resolution we can show that $E_1 \models \{c, \neg d, \neg notb, \neg notc\}$ and $E_2 \models \{\neg b, \neg a, \neg c, \neg notd\}$. We can then remark that both extensions do not verify the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ of Theorem 1, since E_1 entails $\neg notb$ but does not entail b and E_2 entails $\neg notd$ but does not entail d. These are two extra-extensions which do not entail any stable model of the logic program. We can consider in this case that $(L(\pi), STB)$ does not have any classical extension which could match with any answer set of π, and then the stable semantics is captured. However, these extra-extensions could infer some informations that can not be inferred in the stable models semantics. For instance, we can remark that the atom c is inferred by E_1, but it can not be inferred in the stable model semantics, since there is no answer set for the logic program π. This is an intriguing fact, and one can think that it could be important in some cases, to consider the information inferred by the extra-extensions and extend then the stable model semantics.

Below, we give two school examples (Example 6, Example 7) which are well-known in the default logic community. We will see that our semantics is valid for both them.

Example 6. Consider the program $\pi = \{a \leftarrow nota\}$ formed by a single rule. This program has no answer set. Its corresponding CNF encoding is $L(\pi) = \{a \vee \neg nota, \neg a \vee \neg nota\}$ and its strong backdoor set is $STB = \{nota\}$. The pair $(L(\pi), STB)$ has an extra-extension $E = L(\pi)$. By applying resolution on the two clauses of $L(\pi)$ we can deduce the mono-literal resolvent $\neg nota$. Thus, $E \models \neg nota$. But, $E \not\models a$. E does not verify the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ of theorem 1, then it does not entail any stable model of π. We can consider in this case that $(L(\pi), STB)$ does not have a classical extension which matches with any answer set of π. Our semantics captures the stable models semantics for this example.

Example 7. Now take the logic program given in Example 6 to which we add the rule $a \leftarrow$. We obtain the logic program $\pi = \{a \leftarrow nota, a \leftarrow\}$. The program π has a single stable model $X = \{a\}$. Its corresponding CNF encoding is $L(\pi) = \{a \vee \neg nota, \neg a \vee \neg nota, a\}$ and its backdoor set is $STB = \{nota\}$. The pair $(L(\pi), STB)$ has an extension $E = L(\pi)$, such that $E \models \neg nota$ and $E \models a$. E verifies the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$, and entails the single stable model $X = \{a\}$ of π. Our semantics captures the stable models semantics for this example too.

Now we show that under the condition cited in Remark 2, each extension of $(L(\pi), STB)$ shall entails a stable model of π.

Theorem 2. *If E is an extension of $(L(\pi), STB)$, such that the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$ holds, then $X = \{L_i : E \models L_i\}$ is an answer set of π.*

Proof. . Let E be an extension of $(L(\pi), STB)$. That is, there exists $S' \subset STB$ such that $E = L(\pi) \cup S'$. As E is maximal consistent with respect to STB, we can then deduce that $E \models \neg notL_i$, forall $notL_i \in STB - S'$. Thus, $E \models L(\pi) \cup S' \cup \{\neg notL_i : notL_i \in STB - S\}$. On the other hand the set $X = \{L_i : E \models L_i\}$ is a minimal model

of E. By using the pseudo axiom and the condition given in the theorem, we can show that $X = \{L_i : E \models L_i\} = \{L_i : E \models \neg notL_i\}$. To show that X is a stable model of π, we shall show that X is a minimal Herbrand model of π^X. For doing that, we shall study two cases: if $notL_i \in E$, thus we suppress each occurrence of the literal $\neg notL_i$ in each clause $c : (L_0 \vee \neg L_1 \vee \neg L_2, \ldots, \neg \vee L_m \vee \neg notL_{m+1} \ldots, \neg notL_i, \ldots, \neg notL_n)$ of $L(\pi)$. By using the clause $(\neg L_i \vee \neg notL_i)$ of ME, we deduce that $E \models \neg L_i$, thus $E \not\models L_i$ since E is consistent. We deduce that $L_i \notin X$, and in this case, to obtain the reduct π^X, we suppress each occurrence of $notL_i$ in the body of the rule $r : L_0 \leftarrow L_1, L_2, \ldots, L_m, notL_{m+1}, \ldots, notL_i, \ldots, notL_n$ of π corresponding to the clause c. The second case is when $notL_i \notin E$, in this case we have $E \models \neg notL_i$ and each clause c of $L(\pi)$ having the literal $\neg notL_i$ is subsumed in E. By using the given condition of the theorem on the extension E, we deduce that $E \models L_i$. Thus $L_i \in X$, and in this case, to obtain the reduct π^X, the corresponding rule r in π of the clause c is suppressed. We conclude that the simplification of E by the literals of the STB induce a reduct π^X which is a logical consequence of E. That is $E \models \pi^X$. As X is minimal model of E, it follows that X is a minimal model of π^X. Thus by definition of a stable model, we conclude that X is a stable model of π.

Now, we give two other examples of the default logic school (community). We will prove that our semantics is valid for both them too.

Example 8. Let's consider the logic program π formed by the two following rules:

$$\pi = \{a \leftarrow notb, b \leftarrow nota\}$$

Its CNF encoding is the set of clauses $L(\pi) = Rules \cup ME$:

$$Rules = \{a \vee \neg notb, b \vee \neg nota\}$$

$$ME = \{\neg a \vee \neg nota, \neg b \vee \neg notb\}$$

Its backdoor set is $STB = \{nota, notb\}$. The pair $(L(\pi), STB)$ has two extensions $E_1 = L(\pi) \cup \{nota\}$ and $E_2 = L(\pi) \cup \{notb\}$. By unit resolution we can deduce that $E_1 \models \{b, \neg a, \neg notb\}$ and $E_2 \models \{a, \neg b, \neg nota\}$. Both extensions verify the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$. Thus, we obtain from E_1 resp. E_2 the sets $X_1 = \{b\}$ resp. $X_1 = \{a\}$ which are the two stable models of the program π. Thus, the stable model semantics is captured.

Example 9. Now consider the logic program π formed by the rules :

$$\pi = \{a \leftarrow notb, b \leftarrow notc, c \leftarrow nota\}$$

Its CNF encoding is the set of clauses $L(\pi) = Rules \cup ME$:

$$Rules = \{a \vee \neg notb, b \vee \neg notc, c \vee \neg nota\}$$

$$ME = \{\neg a \vee \neg nota, \neg b \vee \neg notb, \neg c \vee \neg notc\}$$

,

The backdoor set is $STB = \{nota, notb, notc\}$. The pair $(L(\pi), STB)$ has three extensions $E_1 = L(\pi) \cup \{nota\}$, $E_2 = L(\pi) \cup \{notb\}$ and $E_3 = L(\pi) \cup \{notc\}$. We can deduce by unit resolution that $E_1 \models \{\neg a, c, \neg notc, \neg notb\}$, $E_2 \models \{\neg b, a, \neg nota, \neg notc\}$ and $E_3 \models \{\neg c, b, \neg notb, \neg nota\}$. We can see that all of the extensions do not verify the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$. These are extra-extensions which do

not entail any answer set of π. We conclude that π has no answer set and the stable model semantics is captured. On the other hand, all of these extra-extensions infer some atoms ($E_1 \models c$, $E_2 \models a$,$E_3 \models b$) that can not be inferred by the stable model semantics since π has no stable model. It will be important to see if it is possible to extend to stable model semantics by using these extra-extensions.

We showed then by both Theorem 1 and Theorem 2 that the answer sets of logic program π are in bijection with a subset of extensions of $(L(\pi), STB)$ and we have characterized the extra-extensions by a simple condition. We give in the next section a method which can be used to compute such extensions from which the answer set of logic program can be deduced.

4 A procedure for extension or answer set computing based on the new semantics

It is well known that each stable model of a program π is a models of its completion $comp(\pi)$, but the converse is in general not true. Fages [6] showed that if the program π is tight (without loops) then its set of stable models (answer sets) is identical to the set of models of its completions $comp(\pi)$ [1]. If the completion of a such tight program is translated into a set of clauses Γ then a SAT solver can be used on Γ as a black box to generate the stable models of π. Lin and Zhao [3] showed for non-tight programs that the models of their completions which are not answer sets can be eliminated by adding to the completion what they called *loops formulas*. Their solver ASSAT is based on this technique and had been shown to outperform ASP state-of-the-art systems like Smodels [11, 15] and DLV [4] on several problem instances. Nevertheless, the solver ASSAT 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 π is a model of its completion $comp(\pi)$, Guinchiglia et al. in [9] do not use SAT solvers as black boxes, but implemented a method which is based on the DLL [2] 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 [10] and has the following advantages: it performs the search on $comp(\pi)$ without introducing any extra variables except those used by the clause transformation of $comp(\pi)$, deals with tight and not tight programs, and works in a polynomial space.

4.1 The method

Our method is also based on SAT solvers, but it is different since it is based on the new semantics that we described in the previous section. Given a general logic program π, our method performs on its logical representation $L(\pi)$ rather on its completion $comp(\pi)$. The clauses of $L(\pi)$ are all Horn clauses build on the two sets of variables V and nV. Our method considers the set of variable $STB \subset nV$ appearing in the bodies of the rules forming the program π as a strong backdoor (STB) [16] on which it performs its search to obtain the extensions of $(L(\pi), STB)$ from which we can deduce the answer sets of the program π. That is, the method computes in the enumeration phase a maximal model with respect to the backdoor set (STB) variables which guarantees to have by construction an extension for each consistent partial interpretation of the

backdoor. The idea, here is to assign first, the value true to a maximal number of literals $notL_i \in STB$, and backtrack only on the nodes that propagate negative literals of the backdoor. In this way, we generate only maximal consistent interpretations. We give below the fundamental outlines of the method.

Proposition 3. *Let π be a program for which the logical encoding is $L(\pi)$ and where initially $E = L(\pi)$. If all the variables of the STB are assigned (by a DLL procedure) and the unit resolution reductions (subsumption of clauses and suppression of variables) is applied on $L(\pi)$ then the following assertions hold:*

1. *The set of clauses E resulting from the simplification of $L(\pi)$ is formed by the union of a subset of unit clauses C_1 and a subset of non-unit Horn clauses C_2 without STB variables. The sets C_1 and C_2 do not have common variables (they are variable independent).*
2. *The assignment to false of all the remain variables $\{notL_i\}$ and $\{L_i\}$ which are not already fixed in C_1 does not affect the answer sets of the program π, and leads to a minimal model of E that could be a candidate for answer set checking for the program π.*
3. *After assigning all the STB variables $\{notL_i\}$ and all the unit resolution propagations are done, the candidate for answer set checking is completely defined by the set of positive atoms $\{L_i\}$ fixed in C_1. It is an answer set of π iff the extension E of $(L(\pi), STB)$ verifies the condition $(\forall L_i \in V, E \models \neg notL_i \Rightarrow E \models L_i)$.*

Proof.
- Assertion 1: since $L(\pi)$ is a horn clause set, then by assigning the variables of the backdoor STB and after doing all the simplification by unit resolution, the encoding $L(\pi)$ will be reduced to a union of a subset of unit clauses deduced by unit resolution, and a subset of horn clauses. The fact that C_1 and C_2 do not have common variable is trivial, since $L(\pi)$ is simplified by unit resolution and subsumption.
- Assertion 2: Since all the variables $\{notL_i\}$ of the bodies of the rules of π are assigned, and all the unit clause propagation are done, then all the atoms that could be involved in the eventual answer set are fixed. We can then complete the model by assigning to false the remain variables to obtain a kind of a minimal model which will be a candidate for answer set checking.
- Assertion 3: this had been shown in the proof of Theorem 2.

Based on the previous proposition, we give below the main steps of the new method for extensions and answer sets computing:

1. Set $E = L(\pi)$.
2. First assign the variables of the STB by using a DLL enumeration method where all the unit resolution propagations are done at each choice node of the search tree on all the variables. The strategy, here is to assign the value true in prior to the free STB variables and when a failure is detected the method backtracks only on the nodes that had propagated at least a negative literals of the STB. In this way, we get a maximal consistent partial interpretation I_{STB} of the strong backdoor variable STB which extend E $(E = E \cup I_{STB}))$.
3. Since E is an extension of $(L(\pi), STB)$ by construction in step 1, assign all the remain variables (the ones that are not fixed yet by the current interpretation) to false in order to get a kind of a minimal model of E.

4. Take the extension E of $(L(\pi), STB)$. If the condition $(\forall L_i \in V : E \models \neg notL_i \Rightarrow E \models L_i)$ holds, then the restriction to positive atoms in E forms an answer set of the program π.

The method performs a DLL enumeration on the backdoor variables and at each step it verifies that the partial extension is consistent. When all the variables of the backdoor STB are assigned and all the unit resolution propagations are done without detection of conflicts, then the remain variables are assigned implicitly to false. After that, the procedure guarantees that the set E build is an extension. That is, E is consistent and maximal in term of inclusion of variables of the form $\{notL_i\}$. Besides, if the condition $(\forall L_i \in V : E \models \neg notL_i \Rightarrow E \models L_i)$ holds, then the set of positive atoms of the extension E forms an answer set of the original logic program π.

4.2 Complexity

If n is the number of the variables of the logic encoding $L(\pi)$ of the given program π, and if among them k are STB variables, and m is the number of clauses of $L(\pi)$, then the main step of the method (Step 2) can be achieved in $O(nm2^k)$ in the worst case. The steps 3 and 4 can be performed in $O(n-k)$ and in $O(k)$ respectively. Therefore, the asymptotic complexity of the algorithm is $O(nm2^k)$ with $k \leq n$.

Our method has the advantage to perform on the CNF encoding $L(\pi)$ whose size is equal to $size(\pi) + 2n$. It can be used for tight and non-tight programs and is able to compute all the answer sets of the program π. This method can be implemented by a slightly modified DLL procedure which performs an enumeration on the STB variables and maintains a unit resolution saturation process at each node of the search. This is followed by a simple and basic answer set verification step (step 4) after the assignment of the STB variables.

This work is totally theoretical, but we aim to implement in future a prototype of this method with several improvements and compare its performances to the state-of-art ASP methods.

5 Conclusion

We defined in this theoretical work a new semantics for logic program where the rules of the program are expressed in term of propositional clauses called its logic encoding. We defined a notion of extension for this logic encoding which under some conditions captures the answer sets of the original logic program. This semantics is different from the well known fixed point ones. Our approach is based on the notion of extensions that we defined for logic programs. We showed that each consistent logic program has at least an extension and the stable models of a logic program could be inferred from a sub-set of these extensions. The notion of extensions seems to be more general than the answer sets since some of them are not captured by the stable model semantics. Based on this new semantics, we defined a procedure that can be used to compute the set of extensions which capture the answer sets of the initial logic program. This procedure can be implemented as a slightly modified DLL procedure which shall perform an enumeration on a the restricted set of variables (the STB variables) on which depends its complexity.

As a future work, we are first interested in implementing this new methods and compare its performances to the ones of existing methods for answer sets computing.

Another interesting direction is to extend this work to other frameworks like extended logic programs, disjunctive logic programs and to more general frameworks like the default logic.

References

1. Clark, K.: Negation as failure. In: Logic and data bases. pp. 293–322. In Herve Gallaire and Jack Minker, editors (1978)
2. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem proving. JACM, 5(7)
3. E.Erdem, Y.Zhao: Assat: Computing answer sets of a logic program by sat solver. In: AAAI-02 (2002)
4. Eiter, T.: Th kr sysem dlv:progress report, comparisons and benchmarks. In: KR (1978)
5. Erdem, E., Lifschitz, V.: Tight logicprograms. Thoery and Practice of Logic Programming 3, 499–518 (2003)
6. Fages, F.: Consistency ofclark's completion and existence of stable models. Thoery and Practice of Logic Programming 1, 51–60 (1994)
7. 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)
8. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9, 365–385 (1991)
9. Giunchiglia, E., Lierler, Y., Mratea, M.: Sat-based answer set programming. In: 19th National Conference on Artificial Intelligence,July 25-29, San Jose, California. AAAI (2004)
10. 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)
11. Niemela, I.: Logic programs with stable models semantics as a constraint programming paradigm. Anals of mathematics and Artificial Intelligence 25, 241–273 (1999)
12. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
13. Schwind, C., Siegl, P.: A modal logic for hypothesis theory. Workshop on Nonstandard Queries and Answers 21 (1/2), 89–101 (1994)
14. Siegl, P., Schwind, C.: Hypothesis theory for nonmonotonic reasoning. In: Workshop on Nonstandard Queries and Answers (Toulouse, July 1991)
15. Simons, P.: Extending and implementing the stable model semantics. In doctoral dessertation pp. 305–316 (2000)
16. Williams, R., Gomes, C.P., Selman, B.: Backdoors to typical case complexity. In: IJCAI. pp. 1173–1178 (2003)

Une approche de résolution à population pour le problème du CDO carré

Lebbah Fatima Zohra, Lebbah Yahia**

Université d'Oran Es-Senia, Lab. LITIO, B.P. 1524 EL-M'Naouar, 31000 Oran, Algerie
fz_lebbah@yahoo.fr, ylebbah@gmail.com

Résumé. Ce papier décrit une approche de résolution locale à population pour appréhender la version quadratique du problème CDO de gestion de portefeuilles bancaires. Le problème CDO carré provient de l'ingénierie des finances, qui consiste à affecter aux portefeuilles des actifs (ou des titres) qui doivent être variés, permettant ainsi un compromis entre la maximisation du gain et la minimisation du risque de perte. Ce problème est formalisable en termes d'un programme quadratique en nombres entiers, dont les approches de résolution exacte proposées dans la littérature se sont avérées peu performantes quand on augmente le nombre d'actifs et le nombre de portefeuilles. L'échec des méthodes exactes est à l'origine de notre recours à une méthode approchée, à savoir la méthode à population GWW (Go With the Winner). Nous proposons une fonction de voisinage et une fonction objectif dédiées à ce problème. Nous avons réalisé une mise en oeuvre sous l'environnement INCOP, afin de montrer l'intérêt de notre approche sur des instances non triviales du problème.

Mots-clés: CDO carré, recherche locale, GWW, IDWalk.

1 Introduction

Les méthodes de recherche locale ont connu un véritable succès en résolution de problèmes combinatoires de grande taille dans l'industrie et dans divers domaines (transport, ordonnancement, emploi du temps, etc.). La raison qui est derrière cet engouement sur les méthodes locales est l'incapacité des méthodes exactes à appréhender les problèmes de grandes tailles. Le problème CDO carré abordé dans ce papier est issu de l'ingénierie des finances et plus précisément le marché des crédits. Le problème CDO carré modélise une problématique critique dans la gestion équitable des portefeuilles des clients dans une banque, à savoir éviter à ce que des portefeuilles soient profitables au détriment du reste des portefeuilles. Etant donné un ensemble de portefeuilles appartenant à des clients, l'institution bancaire (ou financière) est en charge d'affecter à ces portefeuilles un ensemble d'actifs d'une façon fiable, c'est-à-dire en minimisant le risque de perte, et en maximisant le gain. Mettre les meilleurs actifs dans un sous-ensemble de portefeuilles exposerait le reste des portefeuilles aux actifs risqués, donc une perte assurée. Par conséquent, la solution consisterait donc à trouver une solution de compromis et d'équilibre entre les affectations des actifs aux différents portefeuilles. En d'autres

** Ce travail est supporté par le programme TASSILI 11MDU839 (France, Algérie).

termes, il est nécessaire de partager entre tous les portefeuilles, et d'une façon équitable, les actifs gagnants, et les actifs risqués. Différents modèles de gestion de portefeuilles ont été proposés dans la littérature, comme par exemple le modèle CDO carré, noté CDO^2. Chaque portefeuille P est composé de m tranches CDO_i de valeurs N_i. Au niveau de chaque tranche i, P est exposée à une perte Att_i, dite dans le jargon financier coupon d'attachement, à condition que la valeur du portefeuille $V(P)$ soit comprise entre Att_i et une grandeur Det_i dite coupon de détachement. La valeur d'un portefeuille $V(P)$ est donnée par la formule $V(P) = \sum_{i=1..m} N_i \times V(CDO_i, Att_i, Det_i)$, où $V(CDO_i, Att_i, Det_i) = (V(P) - Att_i)$ si $Att_i \leq V(P) \leq Det_i$.

Bien évidemment, la fiabilité du modèle CDO est pleinement conditionnée par les valeurs des coupons d'attachement et de détachement. Ces valeurs sont intimement liées au comportement des actifs, et doivent provenir d'une expertise de l'historique de ces actifs : un travail pointu qui relève de la finance et des statistiques, et qui ne rentrent pas dans les ambitions du présent papier. Cependant, une fois ces valeurs proposées, reste la problématique complexe de l'affectation des tranches des actifs aux portefeuilles des clients : une problématique fortement combinatoire. Plusieurs modèles d'affectation des portefeuilles existent dans la littérature. Nous utilisons dans ce papier le modèle noté PD (Portfolio Design) donné par Flenner et al. [3] qui ont proposé une démarche de résolution qui relève des méthodes exactes. Flener et al. ont signalé la nature fortement combinatoire de la recherche de la solution optimale du problème, qui n'a pu être donnée que sur des instances petites des CDO^2. Cependant, ils ont donné une borne inférieure de la valeur optimale associée à ce problème, tout en notant que cette borne est d'une faible qualité (par rapport à la borne optimale) sur les grandes instances. Ils ont aussi proposé une stratégie de décomposition du problème afin de ramener dans certains cas les grandes instances du problème à de petites instances en exploitant une forme de symétrie.

Justement, en raison de la forte combinatoire existante dans la problème de l'affectation des portefeuilles, nous proposons dans ce papier un travail préliminaire de l'exploitation d'une méthode de recherche locale à population, la méthode GWW. Nous proposerons tout particulièrement une fonction de voisinage qui tiendrait compte de la sémantique bien particulière des contraintes de CDO^2. La mise en oeuvre a été faite sous l'environnement INCOP.

Nous avons adopté la méthode GWW, car elle intègre la stratégie puissante qui consiste à lancer une méthode locale à partir de plusieurs configurations différentes, tout en faisant coopérer intelligemment ces différents lancements. De plus la recherche locale utilisée par GWW est un simple paramètre; permettant ainsi de pouvoir expérimenter facilement la panoplie des recherches locales existantes dans la littérature. Nous avons aussi comme objectif de montrer que le problème CDO carré, peut être abordé avec une méthode à population utilisant une recherche locale simple. Notre choix ne s'est pas porté sur les autres méthodes de recherche locale (e.g., algorithmes génétiques, colonies de fourmis, etc.), car ces méthodes exigent un paramétrage minutieux, souvent complexe, qui va à l'encontre de l'objectif de ce papier, à savoir montrer que le problème CDO carré peut être résolu avec une recherche locale assez simple, en obtenant des performances qui concurrencent la méthode exacte [3].

Le papier est organisé comme suit. Nous commençons par détailler le modèle (PD) issu de CDO^2. Puis dans la section qui suit, nous rappelons la méthode locale GWW. Par la suite, nous exposerons la fonction de voisinage et la fonction d'évaluation dédiées au problème (PD). Après, nous exposerons les résultats expérimentaux de notre démarche. La dernière section conclut ce papier en dressant les perspectives immédiates de notre démarche.

2 Le modèle PD d'affectation des portefeuilles CDO^2

PD (Portfolio Design) [3] est un modèle qui décrit les relations existantes entre les différentes tranches du CDO^2. On considère, dans notre cas, que le portefeuille doit contenir v tranches et chaque tranche comporte r actifs. Au niveau de chaque tranche, les actifs doivent être choisis de telle sorte que le nombre des actifs partagés par toute paire de tranches ne doit pas dépasser λ actifs. Ici on parle du chevauchement ou d'overlapping. Cependant les valeurs des coupons d'attachement et de détachement à l'origine de ce problème n'ont pas été indiquées, sans pour autant nuire à la compréhension de la formulation donnée. Le raisonnement sur les valeurs de ces tranches ne rentre pas dans les ambitions du modèle donné.

L'univers des actifs est de la taille $250 \leq b \leq 500$. Un portefeuille typique est contraint par $4 \leq v \leq 25$ tranches, dont chacune est de taille $r \approx 100$. La définition 1 suivante, donné par [3], dresse formellement le modèle d'affectation PD issu de CDO^2.

Definition 1 (Portfolio Design : PD).

Soit $V = \{1, ..., v\}$ un ensemble de v éléments appelés tranches. Soit $B = \{1, ..., b\}$. Un modèle de portefeuille (PD) $\langle v, b, r, \lambda \rangle$, est composé de v ensembles $V_1, ..., V_v$, dits tranches (co-blocks), dans chaque élément (actif) appartient à un sous-ensemble de B contenant r éléments. Appelée condition d'équilibre, chaque paire de tranches présente au maximum λ éléments en commun. On désigne λ l'overlap maximum.

Soit l'ensemble B_j des actifs appartenant au block d'indice j dont chaque actif apparait dans la sous-tranche V_i : $B_j = \{i \in V \mid j \in V_i\}$. Les B_j sont appelés actifs (blocks) et sont tous des sous-ensembles de B, dont les nombres d'éléments sont arbitraires. Formellement, on a :

$$(PD) \begin{cases} \min \lambda \\ \forall j \in B : B_j \subseteq V \\ \forall i \in V : V_i \subseteq B \\ \forall j \in B :\mid B_j \mid \leq v \\ \forall i \in V :\mid V_i \mid = r \\ \forall i \neq j \in V :\mid V_i \cap V_j \mid \leq \lambda \end{cases} \tag{1}$$

La forme matricielle de PD est donnée par :

$$PD\,\langle v, b, r, \lambda \rangle \begin{cases} \min(\lambda) \\ \forall i \in 1..v, \sum_{j \in 1..b} M_{ij} = r \\ \forall i \in 1..v, \forall k > i, \sum_{j=1}^{b} M_{ij} M_{kj} \leq \lambda \\ \forall i \in 1..v, \forall j \in 1..b, M_{ij} \in \{0, 1\} \end{cases} \tag{2}$$

avec

v le nombre de tranches,
b le nombre d'actifs,
r le nombre de titres à prendre de chaque tranche,
λ l'overlapp (nombre de titres partégés par toute paire de tranches).

Donc le $PD\,\langle v, b, r, \lambda\rangle$ est le problème à résoudre, sachant que v, b et r sont connus et il faut trouver le λ, la valeur minimale à calculer pour le PD. Par conséquent, notre problème est un problème d'optimisation avec contraintes (Constrained Optimization Problem).

Dans [3], Flener et al. ont réussi à calculer analytiquement une borne inférieure de λ, donnée par l'inéquation (3).

$$\lambda \geq \frac{\lceil \frac{rv}{b} \rceil^2 mod(rv, b) + \lfloor \frac{rv}{b} \rfloor^2 (b - mod(rv, b)) - rv}{v(v-1)} \tag{3}$$

On remplace, par la suite λ par cette borne dans la dernière contrainte du modèle mathématique (2), et on élimine la fonction objectif. Par conséquent, le problème à résoudre devient un problème de satisfaction de contraintes. Flener et al. ont montré que cette borne est de piètre qualité quand la taille est grande.

En fait, Flener et al. ont constaté une ressemblance entre le problème (PD) et un autre problème pleinement étudié en optimisation combinatoire, le problème $(BIBD)$ (Balanced Incomplete Block Designs) qui est structuré comme suit :

$$(BIBD) \quad \begin{cases} \forall j \in 1..b, B_j \subseteq V \\ \forall i \in 1..v, V_i \subseteq B \\ \forall j \in 1..b, \mid B_j \mid = k \\ \forall i \in 1..v, \mid V_i \mid = r \\ \forall i \neq j \in 1..v, \mid V_i \cap V_j \mid = \lambda \end{cases} \tag{4}$$

La résolution du problème $(BIBD)$ en programmation par contraintes, est très performante en exploitant ses différentes symétries. La plupart de ces symétries disparaissent dans le contexte de (PD), rendant la résolution de (PD) nettement plus complexe que $(BIBD)$. Justement, Flener et al. [3] ont proposé une heuristique qui, dans certaines conditions, permettrait de ramener la résolution de (PD) en $(BIBD)$. Une deuxième heuristique propose de ramener la résolution d'un (PD) avec de grandes dimensions en un multiple (PD) de petites dimensions. Pour plus de détails, nous renvoyons le lecteur à la référence [3].

3 Description de l'algorithme GWW

L'algorithme 1, GWW [1], est une méthode à population, qui explore différentes configurations en même temps dans un espace de recherche en gérant une population de particules (solutions). Au début les particules sont distribuées aléatoirement et un seuil est fixé au-dessus de la plus mauvaise solution. A chaque itération on diminue la valeur

Algorithm 1 $GWW - LS$

Input: B nombre de particules; S longueur de la marche; T paramètre seuil; WP paramètre de la marche;

Chaque particule est aléatoirement placée sur une configuration et rangée dans le tableau $Particules$
$Seuil \leftarrow$ la particule de moindre coût dans $Particules$
while True **do**
  $Seuil \leftarrow$ baisser le seuil $Seuil$
  if $Meilleure(Particules) > Seuil$ **then**
    Retourner $Meilleure(Particules)$
  else
    Redistribuer $Particules$ en plaçant chaque particule ayant un coût supérieur à $Seuil$ sur une particule vivante choisie aléatoirement
    for particule p dans $Particules$ **do**
      for $i \in 1..S$ **do**
        déplacer la particule p dans le meilleur de ses voisins en exploitant une recherche locale de paramètre WP
      end for
    end for
  end if
end while

du seuil. Les particules dont les valeurs sont supérieures au seuil sont dites des particules mortes, sinon elles sont dites vivantes. Après la diminution du seuil, les particules mortes sont redistribuées sur les particules vivantes.

Dans [4], la marche aléatoire a été remplacée par une méthode locale (e.g., la méthode taboue, le recuit simulé, ...). L'algorithme 1, ainsi que d'autres méthodes incomplètes ont été implémentés dans la bibliothèque INCOP [2,5]. Dans notre mise en oeuvre, nous avons utilisé au lieu de la marche aléatoire, l'algorithme IDWalk [6].

4 Fonction de voisinage et fonction d'évaluation

Soit le problème d'optimisation (PD) décrit dans la section 2. Une configuration de (PD) d'une particule est donnée par la matrice $M[1..v][1..b]$, où $\forall i \in 1..v$, $j \in 1..b$, $M[i][j] \in \{0, 1\}$. Nous utilisons le vecteur $val\_conflict$ qui comptabilise le taux de conflits au niveau des variables. Nous associons donc une valeur dite de conflit $val\_conflict_{i,j}$ à chaque variable $M_{i,j}$. Le vecteur $val\_conflict$ est défini comme suit :

$$val\_conflict_{i,j} = \sum_{k \in 1..m} \begin{cases} +1 \text{ si } M_{i,j} \in vars(C_k), C_k \text{ est de la forme "} f_k(M) = 0", \\ \quad val(f_k(M)) < 0, M_{i,j} = 0. \\ -1 \text{ si } M_{i,j} \in vars(C_k), C_k \text{ est de la forme "} f_k(M) = 0", \\ \quad val(f_k(M)) > 0, M_{i,j} = 1. \\ -1 \text{ si } M_{i,j} \in vars(C_k), \text{,} C_k \text{ est de la forme "} f_k(M) \leq 0", \\ \quad val(f_k(M)) > 0, M_{i,j} = 1. \\ 0 \quad \text{sinon.} \end{cases}$$

où m est le nombre de contraintes du système (PD), $vars(C_k)$ est l'ensemble des variables participant dans la contrainte C_k, $val(f_k(M))$ est la valeur de la fonction f_k pour la configuration M.

Etant donnée une configuration $M[1..v][1..b]$, nous repérons la variable $Var_{Conflict}$ qui a le plus grand conflit. Cette variable $Var_{Conflict}$ est par la suite changée : si elle avait la valeur 0, elle prend la valeur 1, sinon 0. La matrice M de la configuration est changée en opérant la modification que nous venons de décrire sur la variable $Var_{Conflict}$.

Par conséquent, la fonction de voisinage d'une solution est une autre solution ayant les mêmes valeurs que la solution entrée où nous changeons seulement la variable ayant le plus grand conflit comme décrit ci-haut.

La fonction d'évaluation adoptée comptabilise simplement le nombre de contraintes non satisfaites.

5 Expérimentations

Nos expérimentations se sont portées sur le problème (PD) où λ est fixe. Nous donnons ci-dessous les résultats en temps d'exécution et en nombre de solutions obtenus sur des instances de problèmes pris de [3]. La machine utilisée est à base d'un processeur Intel Core 2 Duo, avec 2 GO de ram, sous Linux 32 bits. Les expérimentations ont été réalisées en intégrant dans INCOP notre fonction de voisinage et notre fonction d'évaluation décrites ci-haut. Les instances pour lesquelles on ne dispose pas du temps d'exécution de l'approche de Flener et al. sont signalées avec un tiret $-$.

L'expérimentation est effectuée sur le problème CDO carré décrit dans la Section 2, en utilisant la borne inférieure proposée par Flenet et al. En effet, il s'est avéré [3] que cette borne est très éloignée de la valeur optimale sur les instances de grandes tailles. Cependant, notre objectif est justement de montrer qu'avec une recherche locale simple à population, on peut concurrencer une méthode exacte performante. La perspective immédiate de ce travail est d'appréhender le problème d'optimisation sans fournir une quelconque borne.

La méthode utilisée pour le type de marche dans GWW est l'algorithme IDWalk [6]. Cette dernière est choisie, pour la simplicité qu'elle offre au niveau de la maîtrise de ses paramètres. Le résultat des expérimentations en invoquant GWW(IDWalk) à la recherche d'une seule solution, est donné dans le tableau ci-dessous.

$PD < v, b, r, \lambda >$	GWW(IDWalk) T(sec)	Flener [3] T(sec)
$PD < 7, 7, 3, 1 >$	0.002	0.00
$PD < 10, 8, 3, 2 >$	0.000	-
$PD < 10, 20, 1, 0 >$	0.017	-
$PD < 10, 32, 9, 2 >$	0.272	-
$PD < 10, 37, 14, 6 >$	0.068	0.63
$PD < 9, 37, 12, 3 >$	0.512	24.42
$PD < 9, 24, 8, 3 >$	0.006	1.99
$PD < 10, 38, 10, 2 >$	0.105	4.95
$PD < 10, 25, 8, 2 >$	0.420	0.09
$PD < 10, 31, 9, 15 >$	0.008	152.22

Nous remarquons que nos temps d'exécutions sont très proches, et même meilleurs pour quelques instances, de ceux de la démarche proposée par Flenet et al. Ce résultat est très encourageant, car la fonction de voisinage adoptée est plutôt simple. Nous pensons qu'il existe un terrain intéressant d'amélioration de la fonction de voisinage afin de pouvoir appréhender les grandes instances de (PD).

6 Conclusion et perspectives

Dans ce papier, nous avons détaillé le problème d'optimisation (PD) issu de l'ingénierie des finances. Ce problème est un modèle provenant de la gestion des portefeuilles CDO^2. La forte combinatoire de ce problème est à l'origine de l'échec des méthodes exactes à le résoudre, notamment en raison de l'absence de symétries.

Nous avons proposé dans ce papier l'exploitation d'une recherche locale à population GWW, en adoptant une fonction de voisinage simple qui consiste à basculer la variable la plus en conflit. Malgré la simplicité de cette fonction, nos premières expérimentations se sont avérées fructueuses sur des instances non triviales du problème (PD). Les deux perspectives immédiates de notre travail consistent en : (1) améliorer la fonction de voisinage afin d'appréhender les grandes instances de (PD), (2) revoir le schéma courant de notre recherche locale afin d'appréhender le problème initial de CDO carré sans utiliser une borne.

Remerciements

Nous remercions les relecteurs qui nous ont permis d'améliorer la clarté de plusieurs points de cet article.

References

1. Vazinari U. Aldous D. Go with the winners algorithms. *IEEE Symposium on Foundations of Computer Science*, pages 492–501, 1994.

2. Neuveu B. *Techniques de résolution des problèmes de satisfaction de contraintes*. PhD thesis, Université de Nice-Sophia Antipolis, 2005.
3. Reyna P.L Silverstsson O. Flener P., Pearson J. Design of financial cdo squared transactions using constraint programming. *Constraints*, pages 12:179–205, 2007.
4. Trombettoni G. Neuveu B. Incop : Une bibliothèque c++ de méthodes incomplètes pour l'optimisation combinatoire. *INRIA*, 2004.
5. Trombettoni G. Neuveu B. Hybridation de gww avec la recherche locale. *9èmes journées Nationales sur le Résolution Pratique de problèmes NP-Complets, JNPC'03, Amiens, France*, pages 277–292, 2007.
6. Bertrand Neveu, Gilles Trombettoni, and Fred Glover. Id walk: A candidate list strategy with a simple diversification device. In Mark Wallace, editor, *CP*, volume 3258 of *Lecture Notes in Computer Science*, pages 423–437. Springer, 2004.

Adaptive Holonic Multi-agent Product Driven Manufacturing Control with Genetic Algorithm-Based Simulation-Optimization

Mehdi Gaham[1] , Brahim Bouzouia[1], Noura achour[2],

[1] Laboratoire Systèmes robotisés de production
Centre de Développent des Technologies Avancées
Baba Hasen Alger, Algerie 16303
[2] Laboratoire de Robotique Parallélisme Electro-énergétique
USTHB, BP32 el alia, bab ezzaouar,
Alger, Algerie
Corresponding author: mgaham@cdta.dz

Abstract. The work presented in this paper investigates the combination of agent-based technology and simulation-optimisation approaches for real time adaptive dynamic scheduling and control of real world stochastic manufacturing systems. Mainly, within the context of holonic multi-agent product-driven manufacturing control, a scheduling rules-based genetic algorithm simulation-optimisation dynamic scheduling approach is proposed. Main motivation of the developed hybrid intelligent systems framework is the realization of effective and efficient intelligent product driven agent-based distributed dynamic scheduling and control strategy, that enhances manufacturing system reactivity, flexibility and fault tolerance, as well as maintains behavioural stability and optimality**.**

Keywords: Agent-Based *Manufacturing, Product-driven Control, Holonic Systems, Simulation-Optimization, Dynamic Scheduling, Genetic Algorithms.*

1 Introduction

Induced by market globalization and pressure, poor demand visibility, shorter product life cycles and adoption of new consumer-driven business practices such as mass customisation, are only some of the innumerable factors that makes flexibility and agility the key competitiveness issues of nowadays manufacturing enterprise. However, due to their centralized or hierarchically organized planning, scheduling and control structures, legacy manufacturing control frameworks respond weakly to theses challenges and behaves poorly when the system is subject to internal or external unforeseen disturbances. So, distributed or "*hiterarchical*" control paradigms, aiming to bring manufacturing enterprise more competitiveness by enhancing their ability to dexterously react to customer orders and changing production environments have been proposed [1]. Distributed manufacturing control means that each system's component representation in the control software is designed as autonomous, processing unit, which have its one goal and responsibility

and which interact with other components for constructing overall manufacturing system dynamic behavior. With the aid of appropriate coordination strategy, decisional distribution enhances system modularity, flexibility, fault tolerance, as well as system adaptability and reconfiguration ability. Because of multi-agents technology cope efficiently with such complex distributed systems, distributed manufacturing approaches are usually put into practice using the agent paradigm [2] [3] [4] [5].

Within manufacturing dynamic scheduling and control context, bidding based approaches inspired from the well known contract net protocol pioneered multi-agent local decision-making methods employed [6]. However these approaches and some of their variants are increasingly criticized for their inability to cope efficiently with the complexity of real world manufacturing dynamic scheduling and control problems [7]. Actually, efficiency of dynamic scheduling-related real time local decision-making process employed represent a vital concern for manufacturing organisations evolving within a dynamic and unpredictable environment, and although they enhances agility, adaptability and reconfiguration ability of the manufacturing system, agent-based dynamic allocation approaches still incarnate immaturity facing these concerns and penalize seriously industrial adoption of this emerging paradigm. Motivating an important number of research works, hybrid hierarchic/hiterarchic multi-agent decisional structures have been also investigated for the improvement of agent-based manufacturing approaches [8]. As a core line of investigation, Holonic Manufacturing systems (HMSs) represents a major declination of distributed manufacturing dynamic scheduling and control structures where manufacturing system components such as machines, products, AGVs, etc., features autonomy and cooperation. Mainly, HMSs focus on decisional efficiency and are characterized by a hybrid decisional structure that combine the desirable characteristics of hierarchic and hiterarchic control frameworks, which are behavioural optimality of the former and flexible strategy of the later. Abundantly documented PROSA and ADACORE architecture exemplifies application of the holonic concept to manufacturing control [9] [10]. Another relevant characteristic of HMSs is that they promote the full integration of the manufacturing products or parts as computational control entities within the manufacturing distributed decisional system. The product becomes an active decisional and communicative entity capable of participating in or making decisions relevant to its fabrication. Within this context, association of physical product and its informational counterpart is realized by Radio Frequency Identification (RFID)-based product identification technology.

Intelligent product driven manufacturing control emerge as a promising declination of multi-agent HMSs, and is actually defined by Pannequin [11] as a specialization of holonic agent-based distributed control paradigm where agent technology brings forward new fundamental insights on decentralized coordination and auto-organization, enabling new manufacturing decision-making policies and on-the-fly reconfiguration capabilities and infotronic technologies address the issue of synchronization between physical objects and their informational representation.

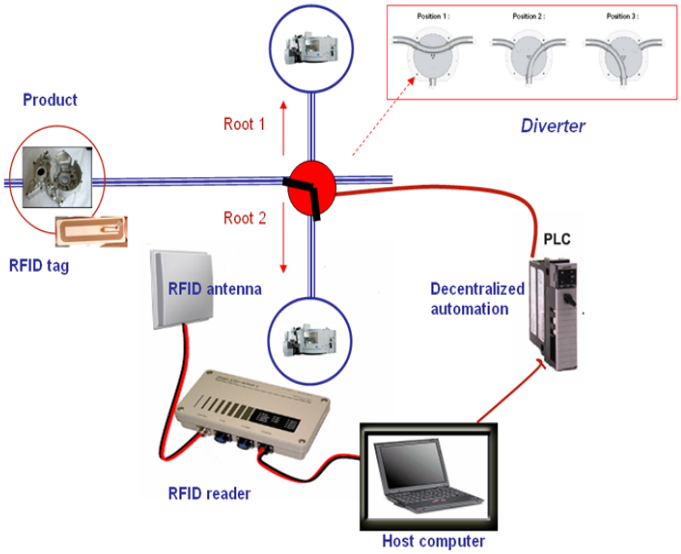


Fig. 1. Product Driven manufacturing Control technological issues related to industrial implementation.

Focusing on decisional efficiency, the presently reported research proposes an innovative adaptive manufacturing dynamic scheduling and control framework that explores the challenging combination of mains capabilities of this emerging control paradigm and simulation-optimization approaches. Relevant fundamental contribution of the proposed approach to multi-agent product driven control is that it exploits a genetic algorithm simulation-optimization approach to dynamically select the most appropriate local decision policies to be used by the agentified manufacturing system components. Exactly, it addresses products and machines agents' local decisional efficiency issues by adapting dynamically their behaviour to the fluctuations of the manufacturing system state. Maintaining the correct balance between hierarchic and hiterarchic behaviour, the proposed hybrid decisional framework is mostly intended to introduce some level of optimization capabilities within product-driven manufacturing control paradigm by dynamically tuning the used local operational policies.

This paper is organized in the following way. In section 2, the adopted multi-agent control architecture is succinctly presented. Section 3 is dedicated to the presentation of the genetic algorithm simulation-optimization adaptive scheduling approach designed. Prototype implementation and result discussion are presented in section 4. And finally section 5 concludes the paper.

2 Multi-agent product-driven control system presentations

Manufacturing facilities organized in flexible job shop production type are main focus of the proposed multi agent-based control architecture. According to specific implementation issues related to project development perspective, the architecture is separated into two distinct and independent parts: high level decisional and low level system emulation part. Each agent in the multi-agent emulation part is a representation of manufacturing component that can be either, a physical resource, an RFID reader or a product. The sited agents are mainly intended to emulate the operational and informational activities of the manufacturing facility. Informational activities such as contract net-based real time information gathering are also encapsulated within this part. Within decisional part, machine and product agents counterparts are implemented as an independent agent (Decision Machine (***D_Machine***) and Decision Product Agents (***D_Product***)) that encapsulate decisional capabilities of product and machine agent.

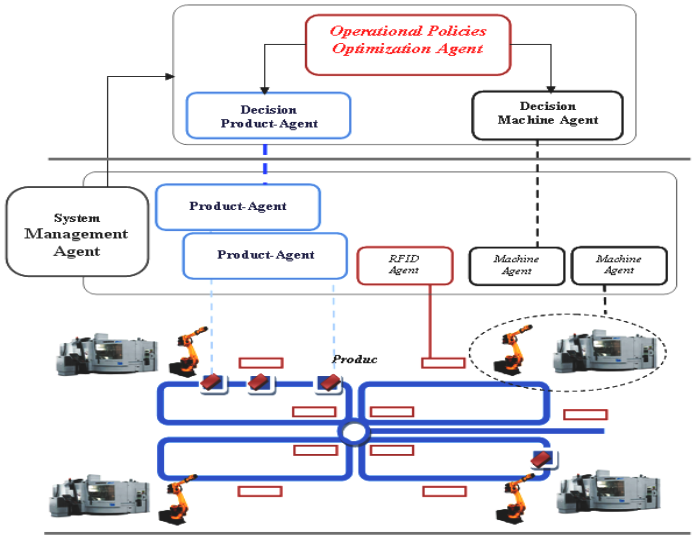


Fig. 2. Product-driven manufacturing control multi-agent architecture.

According to overall system design, common manufacturing machines selection rules and job dispatching rules are used by ***D_Product*** and ***D_Machine*** agents as local decision policies. According to machines related real time information, machines selection rules are heuristics used by ***D_Product*** agent for the selection of the next processing resource to be visited by product agent. Heuristic job dispatching rules are used by ***D_Machine*** agents to select from waiting products witch one to process next. As an integral component of the decisional part of the multi-agent

architecture, Operational Policies Optimization Agent (***OPOA_Agent***) is responsible of the online tuning of the decisional capabilities of ***D_Product*** and ***D_Machine*** agents. Rather than uses a fixed decisional policy all along manufacturing system operational horizon, an optimized set of decision rules is assigned to decisional agents at each occurrence of a perturbation that can affect system' stability. The architecture is represented in Figure 1.

3 Scheduling rules-based simulation-optimization

Although they plays a significant role within practical manufacturing dynamic scheduling context, one of the commonly identified shortcomings of dispatching rule is that their relative performance depends upon the system attributes, and no single rule is dominant across all possible manufacturing system states. Addressing this issue, simulation is usually used to assess empirically the performance of various dispatching rules and for the determination of the best rule to use according to manufacturing system configuration. However, theses approaches do not propose a clear optimization strategy that can guarantee the calculation of an optimal set of rules. By providing a unified integrated framework, simulation-based optimization or simply "simulation-optimization" approaches overcome this limitation. Indeed, as an increasingly investigated research topic online scheduling rules-based simulation-optimization has been this last years identified as offering a real efficiency perspective to practical manufacturing dynamic scheduling approaches. Within this context, optimization is used to orchestrate the simulation of a sequence of system configurations (each configuration corresponds to particular settings of the decision variables) so that a system configuration is eventually obtained that provides an optimal or near optimal solution [12]. Decisions variables correspond to the set of machine selection and jobs dispatching rules, and simulation is carried out by a simulation model that reproduces the stochastic behaviour of the modelled system. Hence scheduling rules-based simulation-optimization is a well suited adaptive dynamic scheduling approach as it makes possible a real time tuning of used local operational policies according to the manufacturing system state and it have been successfully applied to a number of real industrial operational management problem. Recent work by Yang [13], adopting a Genetic Algorithm-simulation approach to solving multi-attributes combinatorial dispatching decision (MACD) problem in a flow shop with a multiples processors (FSMP) environment, illustrate the effectiveness and the efficiency of that kind of methodology compared to several common industrial practices. But as stressed by the authors, although the MACD decision is effective for a practical application, it adds complexity to the shop-floor control problem and its implementation requires supports for a sophisticated shop-floor control system that can perform the dispatching algorithms and control. This shortcoming is well addressed by the product-driven multi-agent integration and technological framework adopted in this research.

System's online simulation-optimization capabilities are encapsulated within ***OPOA_Agent***. Triggered by System Management Agent (***SM_Agent***) at each

occurrence of an internal or an external disturbance (Product arrival, machine breakdown), ***OPOA_Agent*** uses simulation-optimization (Optimization and Simulation modules) for the calculation of the new set of decisional policies (scheduling rules) to dispatch to decisional agents of the system (via the communication module) and used as dynamic scheduling operational policies. Optimization calculation is carried out using a Genetic Algorithm (GAs) (Figure 2).

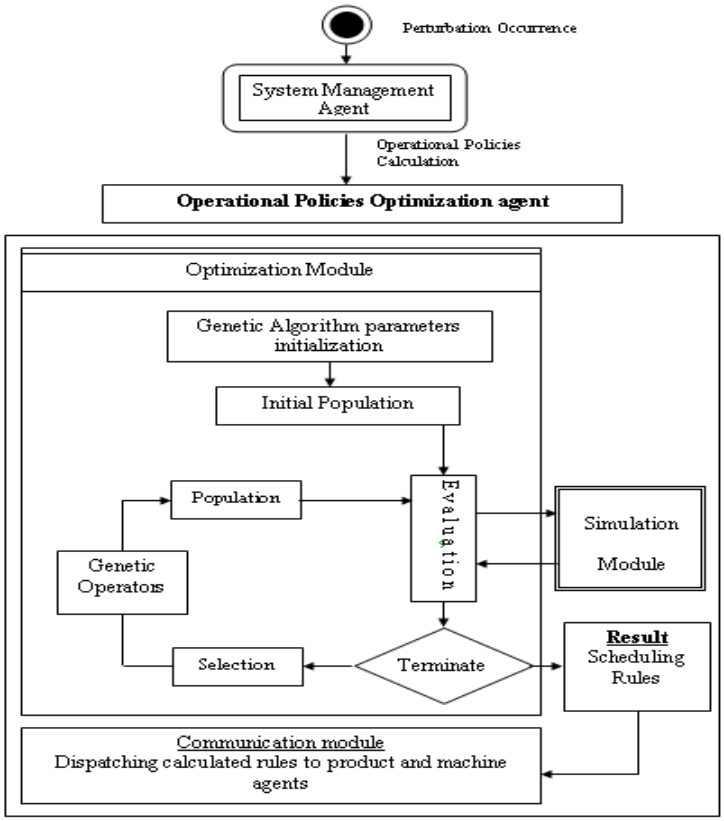


Fig. 3. Internal architecture of Operational Policies Optimization Agent.

In GAs, individuals within a population reproduce according to their fitness in an environment (optimization space). Using stochastic recombination operators the population of individuals combines to perform an efficient domain-independent search strategy. During each generation, a new population of individuals is created from the old one via application of genetic operators (crossover, mutation), and evaluated as solutions to a given problem (the environment). Due to selective pressure (Selection operator), the population adapts to the environment over the generations, evolving better solutions. Our approach uses a real coded genetic

algorithm (an individual codes a sequence of scheduling rules equals to the number of machines and products in the system) combined to classical crossover and mutation and selection operators.

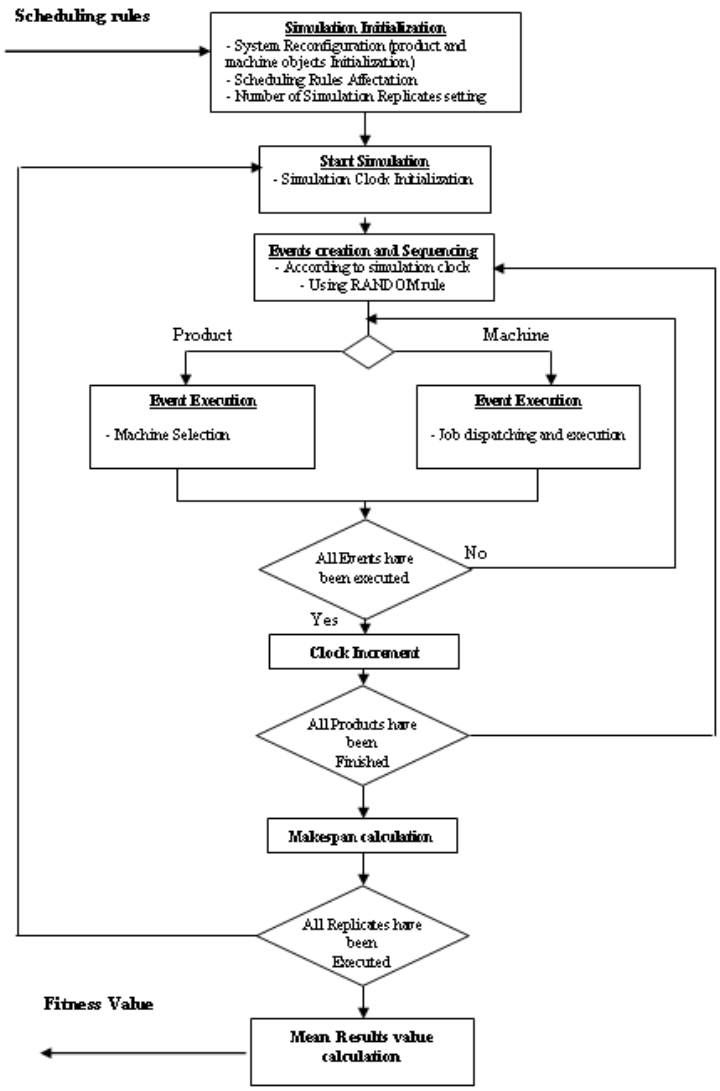


Fig. 4. Flowchart of The Object oriented discreet event simulation model developed.

Evaluation of each individual is carried out using a predefined number of simulation replicates that assesses the performances of the set of scheduling rules according to the stochastic nature of the multi-agent manufacturing control framework adopted. In charge of the simulation execution, the independent simulation module is implemented as an object oriented discrete-time simulation framework. For each simulation replicates, the simulation framework guided by simulation clock evolution constructs a schedule using the manufacturing system real time status and the set of machine selection and job dispatching rules. Stochastic transportation times and randomized synchronization of decisional time conflicts are integrated within schedule construction and evaluation for each simulation replicate. Schedule evaluation is done using the Makespan criterion. Figure 3 illustrates the functional operation of the simulation module.

4 Prototype Development and Approach validation

As depicted in figure 4, Open source NetBeans IDE and JADE *(Java Agent Development Environment*), have been used for the development of the prototype emulation and control system. JADE is an open source platform [14] that provides basic middleware-layer functionalities which simplify the realization of distributed applications that exploit the software agent abstraction [15].

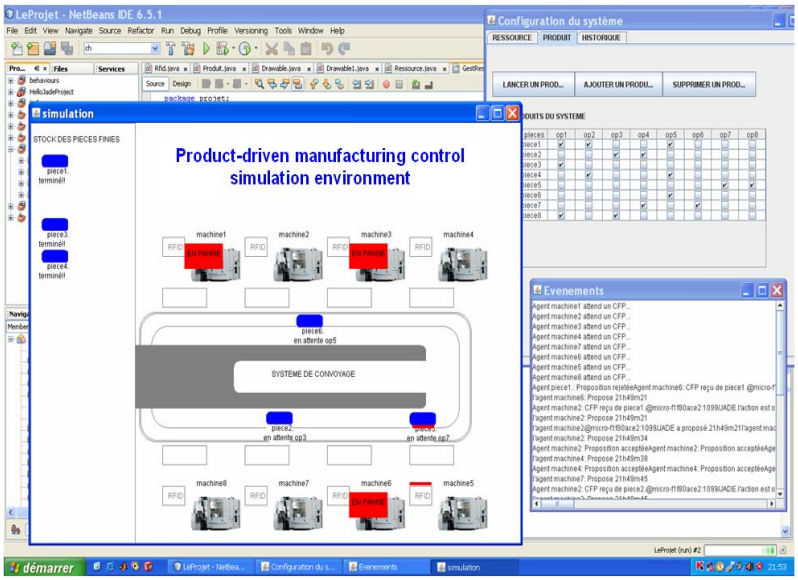


Fig. 5. The realized prototype with its main interfaces.

A multi-agent system based on Jade has the following features: fully distributed, compliant with the FIPA specifications, efficient transport of asynchronous messages, implements both white pages and yellow pages, simple, yet effective, agent life-cycle management, supports agent mobility, provides an agent subscription mechanism, supports ontology's and content languages, provides a library of interaction protocols. JADE also provide a runtime environment and a set of graphical tools to support programmers when debugging and monitoring applications.

In order to evaluate the pertinence of the dynamic scheduling approach in term of solution quality and computational effort, tests have been carried out using an instance problem of 8 machines and 15 products. Different machine selection and job dispatching rules have been used. Table 1 summarizes the set of rule.

Table 1. The set of scheduling and dispatching rules used as local decisional policies by the product and machine agent.

Scheduling rule Number	Scheduling rule	Description
	Product :	
Pr1	LRW	Least remaining work
Pr2	LRPT	Least Remaining processing time
Pr3	LPT	Longest processing time
Pr4	SPT	Smallest processing time
	Machine :	
Mr1	SPT	Smallest processing time
Mr2	LPT	Longest processing time
Mr3	EDD	Earlier due date
Mr4	LRW	Least remaining work
Mr5	FIFO	First in first out
Mr6	LIFO	Last in first out
Mr7	RANDO	Random selection

Figure 6 illustrate the evolution of the fitness value for the test problem. Genetic algorithm parameters have been respectively set to 20, 10 for population size and number of generation. Number of replicate has been empirically assessed and has been set to 10.

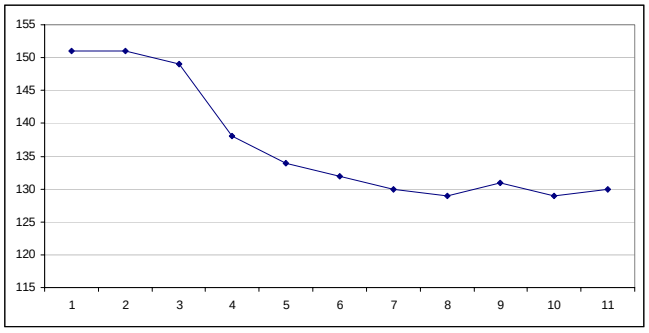


Fig. 6. Genetic Algorithm Evolution

The conducted experiments showed the effectiveness of the approach for the resolution of the test problem both in term of quality of the solution and computational time. Thus, genetic algorithm simulation-optimization seems to be a well suited approach for the online determination of dynamic scheduling operational local policies, but it still highly sensitive to the number of replicates. In fact, the dynamic nature of the simulation-based optimization problem makes this parameter a critical one. Figure 6 show the influence of this parameter for the conducted test problem. In the figure, the dotted curve corresponds to the evolution of solution for a number of replicates equals to 1. It can be seen that the variance of the values is clearly superior to the values variance of the other curve that correspond to a number of replicates equals to 20.

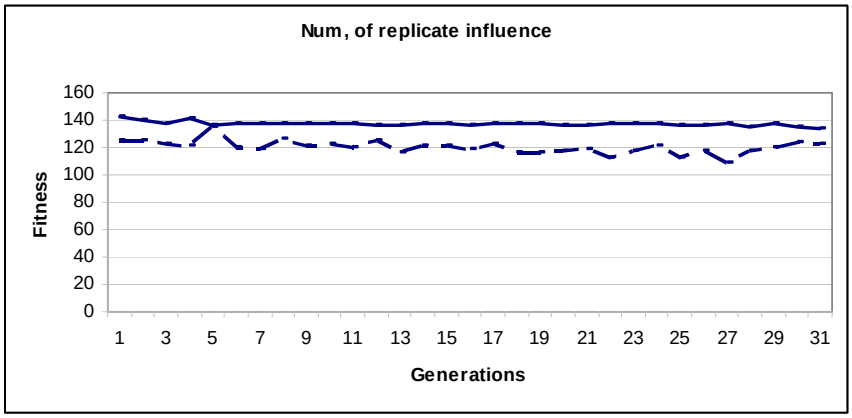


Fig. 7. Influence of replicates number parameter.

4.1 Computational results

The computational tests are conducted using a flexible job shop benchmark problems (Brandimarte [14]). Six test problems are chosen. As the problems are static ones with no transportation time defined and with no due date, the system has been modified according to those facts. The computational tests compare the genetic algorithm based simulation-optimization approach with some of most relevant combination of scheduling and dispatching rules. To handle stochastic nature of the system, ten realization are conducted for each rules combination and average Makespan (completion time for all the tasks) are given as indicator. Genetic algorithm parameters are as follow :

Population= 100.
Generation= 50.
Crossover probability= 0.9.
Mutation probability= 0.02.
Number of simulation replicates = 10.

Table 2. Comparative results of the simulation approach and rules combination.

	Number of products	Number of machines	**Best Rules Value**	Rule Combination	Brandimart Values	**GA values**
Mk 01	10	6	**54**	Pr 1+ Mr 1	42	**47**
Mk 02	10	6	**42**	Pr 1+ Mr 1	32	**38**
Mk 03	15	8	**234**	Pr 1+ Mr 5	211	**222**
Mk 04	15	8	**81**	Pr 1+ Mr 1	81	**77**
Mk 05	15	4	**197**	Pr 1+ Mr 5	186	**188**
Mk 07	20	5	**217**	Pr 3+ Mr 1	157	**168**

The conducted experiments validate the proposed approach in term of computational efficiency. In fact, for that specific set of benchmark problem the approach is superior to simple combination of scheduling and dispatching rules. The approach also give a very interesting results compared to those of Brandimarte, for example for the of Mk04 problem.

5 Conclusions

This paper investigates an innovative hybrid framework combining holonic product-driven manufacturing control and scheduling rules-based genetic algorithm simulation-optimization approaches for real time adaptive dynamic scheduling of real world stochastic manufacturing systems. Both design and implementation issues related to the adopted multi-agent system have been presented, and as a core component of the overall framework, scheduling rules-based genetic algorithm simulation-optimization approach has been described and evaluated. The applicability and effectiveness of the proposed hybrid framework has been demonstrated by the developed prototype and the conducted and succinctly presented tests. Future research direction will be probably the investigation of a more formal approach for the determination of algorithm parameters, particularly simulation replicates number parameter.

6 References

1. Duffie N. A., Piper R. S., (1987) Non-hierarchical control of a flexible manufacturing cell, Robotics & Computer-Integrated Manufacturing, vol. 3, no. 2, pp.175-179.
2. V. Marik, , J. Lazansky, “Industrial applications of agent technologies,” Control Engineering Practice., Vol. 15(11), pp. 1364-1380, 2007.
3. W. Shen, D.H. Norrie, “Agent-based systems for intelligent manufacturing: a state-of-the-art survey,” KAIS Vol. 1(2), pp. 129–156, 1999.

4. W. Shen, Q. Hao, H.J. Yoon, D.H. Norrie, “Applications of agent-based systems in intelligent manufacturing: An updated review,” Advanced Engineering Informatics, Vol. 20, pp. 415–431, 2006.
5. Deen, S.M. (Editor) (2003) Agent-based manufacturing Advances in the holonic approach, Springer, ISBN 3-540-44069-0.
6. Parunak H.V.D., (1987) Manufacturing experience with the contract net, in: M.N. Huhns (Ed.), Distributed Artificial Intelligence, Pitman, pp. 285–310.
7. G. Maione and D. Naso, (2003) A Genetic Approach for Adaptive Multiagent Control in Heterarchical Manufacturing Systems, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 33, NO. 5.
8. Heragu S.S., Graves R.J., Kim B., Onge A., (2002) Intelligent Agent Based Framework for Manufacturing Systems Control, IEEE transactions on systems, man, and cybernetics - PART A: Systems and humans, Vol. 32, no. 5, SEP. 2002, pp. 560-572.
9. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. Peeters, P. (1998) Reference architecture for holonic manufacturing systems: Prosa. Computers in Industry, 37(3): pp. 255–274.
10. P. Leitão, A. W. Colombo, F. Restivo: A formal specification approach for holonic control systems: the ADACOR case. IJMTM 8(1/2/3): 37-57 (2006).
11. R. Pannequin, G. Morel, A. Thomas, “The performance of product-driven manufacturing control: An emulation-based benchmarking study”, Computers in Industry, In Press, DOI: 10.1016/j.compind.2008.12.007.
12. Averill M. Law, Michael G. McComas (2000), SIMULATION-BASED OPTIMIZATION, Proceedings of the 2000 Winter Simulation Conference.
13. T. Yang, Y. Kuo, C. Cho, (2007), A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem, European Journal of Operational Research V. 176, pp. 1859–1873.
14. P. Brandimarte, (1993, Routing and scheduling in a flexible job shop by tabu search, Ann Oper Res V. 41, pp. 157–183.

Optimisation II

Ordonnancement avec graphe de concordance

Bendraouche Mohamed[1] et Boudhar Mourad[2]

[1] Faculté des Sciences, Université Saad Dahleb de Blida,
Route de Soumaa, BP 270, Blida, Algérie
mbendraouche@yahoo.fr
[2] Laboratoire LAID3, Faculté de Mathematiques, USTHB ,
BP 32 El-Alia, Bab-Ezzouar, Alger, Algérie
mboudhar@yahoo.fr

Résumé Dans ce travail, nous considérons le problème d'ordonnancement de tâches en mode non préemptif, sur des machines identiques. Chaque tâche a une durée de traitement et une date de disponibilité et l'objectif est de minimiser la date de fin de traitement de l'ordonnancement. Nous supposons que seules certaines tâches spécifiques, peuvent être ordonnancées simultanément sur deux machines différentes. Ces contraintes sont représentées par un graphe appelé graphe de concordance. Ce problème est NP-difficile. Nous proposons une formulation mathématique sous forme d'un programme linéaire en variables bivalentes et réelles et nous donnons une nouvelle application. Ensuite nous établissons un résultat d'inapproximabilité pour le problème général et un résultat polynomial dans le cas de deux machines.

Mots clés. ordonnancement, machines identiques, graphe de concordance, complexité, algorithme absolu.

1 Introduction

Dans ce travail, nous considérons le problème d'ordonnancement d'un ensemble de n tâches, $V = \{J_1, J_2\ , .., J_n\}$ sur m machines parallèles identiques. On suppose qu'à tout instant une machine n'est allouée qu'à au plus une tâche et chaque tâche n'est traitée que par au plus une machine. Chaque tâche J_i $(i = 1, 2, ..., n)$ possède une durée de traitement p_i et une date de disponibilité r_i. Le mode de traitement des tâches est non-préemptif, c'est-à-dire l'interruption des tâches n'est pas autorisée.

On suppose qu'il existe une relation de concordance entre les tâches : deux tâches sont concordantes (ou en concordance) si leurs intervalles de traitement peuvent s'intersecter, c'est-à-dire si elles peuvent être traitées simultanément sur deux machines différentes. Cette relation est donnée par un graphe $G = (V, E)$ où V est l'ensemble des tâches et une paire de tâches est dans E si et seulement si ces deux tâches sont concordantes. Ce graphe est appelé graphe de concordance. Les contraintes induites par la relation de concordance, ou d'une manière équivalente par le graphe de concordance sont appelées contraintes de concordance.

Un ordonnancement de tâches est dit réalisable, s'il respecte les contraintes de concordance, les durées de traitement et les dates de disponibilité des tâches. Le problème consiste alors à trouver un ordonnancement réalisable qui minimise la date de fin de traitement de l'ensemble des tâches (makespan).

Dans l'article [1] de M. Bendraouche et M. Boudhar intitulé "Scheduling jobs on identical machines with agreement graph", nous avons utilisé le terme "agreement graph" qui est le synonyme du terme "graphe de concordance". Afin de faciliter la description et la classification des problèmes étudiés, on adoptera le formalisme de Graham et al. [2] à trois champs $\alpha|\beta|\gamma$.

Notre problème général est noté : $P|AgreeG = (V, E), r_i|C_{max}$. Ce problème est NP-difficile puisqu'il contient le problème classique $P||C_{max}$.

2 Motivation et applications

Notre problème peut être rencontré dans les problèmes d'ordonnancement sous contraintes de ressources non-partageables. Parmi les applications on cite celles de Baker et Coffman [3] présentées dans l'équilibrage de charge pour les calculs parallèles, d'autres sont mentionnées dans le contrôle de la circulation aux intersections, allocation des fréquences dans les réseaux cellulaires et la gestion des sessions dans les réseaux locaux (voir [4]). Motivés par un problème d'allocation d'opérations à des processeurs Bodlaender et Jansen [5] ont établi une série de résultats sur ce problème.

Nous proposons une nouvelle application de ce problème définie comme suit :

2.1 Planning des examens (Université)

Dans une Université (système LMD), n examens $T_1, \ldots, T_n$ doivent se dérouler à la fin du semestre. Chaque examen doit être pris par un ensemble d'étudiants. Les durées des examens sont de 1 heure ou de 2 heures. Les examens se déroulent dans des salles, dont le nombre est égal à m. Aussi nous supposons que les classes ont une grande capacité de sorte que n'importe quel examen peut être déroulé dans n'importe quelle salle. Le but est de minimiser la durée totale de l'ensemble des examens.

Modélisation :
Les examens sont représentés par des tâches $(T_1, T_2, ..., T_n)$ tandis que les salles sont représentées par des machines $(M_1, M_2, ..., M_m)$.
Le graphe de concordance $G = (V, E)$ est tel que $V = \{T_1, T_2, ...T_n\}$, $(T_i\ , T_j\ ) \in E \iff$ il n'existe aucun étudiant qui passe les deux examens T_i et T_j à la fois. Ce problème s'écrit alors $Pm|AgreeG = (V, E), p_i \in \{1, 2\}|C_{\max}$.

Une autre application due à F. Gardi [6] est donnée par :

2.2 Planification d'horaires de travail

Soient $T_1, T_2, ..., T_n$ n tâches telles que chaque tâche T_i doit être exécutée dans un intervalle de temps $[a_i, b_i]$. La règlementation impose pas plus de m tâches par employer. Sachant que les tâches affectées à un employer ne doivent pas se chevaucher, trouver un planning pour l'entreprise utilisant un nombre minimum d'employés.

Modélisation :
A chaque tâche T_i, $i = \overline{1,n}$, du problème de planification d'horaires, on fait correspondre une tâche T'_i, du problème avec graphe de concordance, de durée 1. Le graphe de concordance $G = (V, E)$ est tel que $V = \{T'_1, T'_2, ...T'_n\}$ et (T'_i, T'_j) $\in E \iff$ les deux tâches T_i et T_j ne se chevauchent pas, c'est-à-dire que leurs intervalles d'exécution ne s'intersectent pas. Ce nouveau problème s'écrit alors $Pm|AgreeG = (V, E), p_i = 1|C_{\max}$ et le nombre minimum d'employer est égal à la valeur optimale du $Cmax$.

3 Formulation mathématique du problème

Considérons le problème $P|AgreeG = (V, E), r_i|C_{max}$. Soient $p_1, \ldots, p_n$ et $r_1, r_2, ..., r_n$ les durées de traitement et les dates de disponibilité des tâches $J_1, J_2, ..., J_n$ respectivement. Soient $M_1, M_2, ..., M_m$ les machines traitant ces tâches. Pour la modélisation du problème général nous avons utilisé la variable C_{max} et trois types de variables qui sont définies comme suit : pour toute tâche J_i, on associe la variable t_i qui représente la date de début de traitement de la tache J_i ($i = \overline{1,n}$). Le deuxième type de variables sont les variables bivalentes X_{ik} définies par :

$$X_{ik} = \begin{cases} 1 \text{ si la tâche } J_i \text{ est ordonnancée sur la machine } Pk \\ 0 \text{ sinon} \end{cases}$$

pour tout couple (i, k) tel que $i = \overline{1,n}$, $k = \overline{1,m}$. Ces deux types de variables nous permettent de définir le premier type de contraintes : $\sum_{k=1}^{m} X_{ik} = 1$

Ces contraintes expriment le fait que chaque tâche doit être affectée à une seule machine.

Le troisième type de variables sont les variables bivalentes Z_{ij} telles que :

$$Z_{ij} = \begin{cases} 1 \text{ si } t_i \leq t_j \\ 0 \text{ sinon} \end{cases}$$

pour tout couple (i, j) tel que $i \neq j$, $i, j = \overline{1,n}$. Soit M une trés grande valeur positive, qu'on peut estimer, par exemple, à $\sum_{i=1}^{n} p_i$. Les variables Z_{ij} peuvent être décrites par les contraintes suivantes :

$Z_{ij} + Z_{ji} = 1$ pour $i < j$; $i, j = \overline{1,n}$

$t_j - t_i \leq M\ Z_{ij}$ pour $i \neq j;\ i,j = \overline{1,n}$

$Z_{ij} \in \{0,1\}$ pour $i,j = \overline{1,n}$

Les contraintes de non chevauchement : expriment le fait qu'une machine ne peut traiter plus d'une tâche à la fois. Autrement, dit si deux tâches J_i et J_j sont affectées à la même machine P_k et que J_i commence avant J_j, alors $t_j \geq t_i + p_i$. Ceci est équivalent à dire que si $X_{ik} = 1$ et $X_{jk} = 1$, alors $t_j \geq t_i + p_i$.

Ces contraintes de non chevauchement s'écrivent :
$t_i + p_i - t_j \leq M\ (1 - Z_{ij} + 2 - X_{ik} - X_{jk})\ i \neq j$ et $i,j = \overline{1,n}$ et $k = \overline{1,m}$.

Les contraintes respectant les dates de disponibilité des tâches sont : $t_i \geq r_i$.

Les contraintes de concordance sont définies par :
$t_j - t_i \geq p_i$ ou $t_i - t_j \geq p_j$ pour toute paire de tâches J_i, J_j telle que $\{J_i, J_j\} \notin E$. Si nous considérons la matrice d'adjacence du graphe G : $a = (a_{ij})$, alors ces contraintes sont équivalentes au système de contraintes $t_j - t_i \geq (1 - a_{ij})p_i$ ou $t_i - t_j \geq (1 - a_{ij})p_j$ pour $i \neq j;\ i,j = \overline{1,n}$. Ces contraintes disjonctives peuvent être écrites sous forme conjonctive par :

$$\begin{cases} t_j - t_i \geq (1 - a_{ij})p_i + M(Z_{ij} - 1)\ i \neq j;\ i,j = \overline{1,n} \\ t_i - t_j \geq (1 - a_{ij})p_j - MZ_{ij} \qquad i \neq j;\ i,j = \overline{1,n} \end{cases}$$

Les contraintes définissant le makespan sont modélisées comme suit :
$t_i + p_i \leq C_{max}$, $i = \overline{1,n}$.

Autres contraintes sur les variables sont :
$t_i \geq 0$, $i = \overline{1,n}$; $X_{ik} \in \{0,1\}$ pour $i = \overline{1,n}$, $k = \overline{1,m}$

Notre problème général est modélisé comme ci-dessous.

$$\begin{cases} MinZ = C_{max} & & \\ \sum_{k=1}^{m} X_{ik} = 1 & i = \overline{1,n} & (1) \\ Z_{ij} + Z_{ji} = 1 & i < j;\ i,j = \overline{1,n} & (2) \\ t_j - t_i \leq M\ Z_{ij} & i \neq j;\ i,j = \overline{1,n} & (3) \\ t_i + p_i - t_j \leq M\ (3 - Z_{ij} - X_{ik} - X_{jk}) & (i \neq j);\ i,j = \overline{1,n};\ k = \overline{1,m} & (4) \\ t_i \geq r_i & i = \overline{1,n} & (5) \\ t_j - t_i \geq (1 - a_{ij})p_i + M(Z_{ij} - 1) & i,j = \overline{1,n} & (6) \\ t_i - t_j \geq (1 - a_{ij})p_j - MZ_{ij} & i,j = \overline{1,n} & (7) \\ t_i + p_i \leq C_{max} & i = \overline{1,n} & (8) \\ t_i \geq 0 & i = \overline{1,n} & (9) \\ Z_{ij} \in \{0,1\} & (i \neq j);\ i,j = \overline{1,n} & (10) \\ X_{ik} \in \{0,1\} & i = \overline{1,n};\ k = \overline{1,m} & (11) \\ C_{max} \geq 0 & & (12) \end{cases}$$

4 Etat de l'art

Dans la littérature, M. Bendraouche et M. Boudhar [1] ont étudié ce problème sous le "Scheduling jobs on identical machines with agreement graph". Notons que la donnée du graphe de concordance est équivalente à la donnée du complémentaire de ce graphe.
G. Even et al. [7] ont considéré une version équivalente à notre problème en utilisant le complémentaire du graphe de concordance, qu'ils l'ont appelé graphe de conflit. Dans la littérature cette version est connue sous le nom : "Scheduling With Conflicts (S.W.C).

Dans le cas où le nombre de machines m est égal à 2 et que les durées de traitement $p_i \in \{1, 2\}$, les même auteurs de [7] ont montré que le problème S.W.C est polynomial et ont proposé un algorithme polynomial pour sa résolution, basé sur la détermination d'un couplage maximum dans un graphe auxiliaire bien approprié.

Vu que l'opération de passage d'un graphe à son complémentaire peut être effectuée polynomiallement, nous déduisons que le problème $P2|AgreeG = (V, E)$, $p_i \in \{1, 2\}|C_{max}$ est polynomial.

Lorsque les durées de traitement $p_i \in \{1, 2, 3, 4\}$, les auteurs de [7] ont établi que ce problème est NP-difficile.

Quant au cas de deux machines mais avec $p_i \in \{1, 2, 3\}$, les mêmes auteurs ont proposé, dans la même référence ce problème comme étant un problème ouvert. Ce problème ouvert est équivalent au problème $P2|AgreeG = (V, E), p_i \in \{1, 2, 3\}|C_{max}$.
Récemment les auteurs M. Bendraouche et M. Boudhar [1] ont résolu ce problème ouvert et ont montré qu'il est NP-difficile. Ces mêmes auteurs de [1] ont montré que le problème $P2|AgreeG = (S_1, S_2; E), p_i \in \{1, 2\}, r_i \in \{0, r\}|C_{max}$ est NP-difficile (c'est le cas d'un graphe de concordance biparti arbitraire).

4.1 Ordonnancement avec exclusion mutuelle (M.E.S) : cas $p_i = 1$ et $r_i = 0$

Dans le cas des durées de traitement unitaires et dates de disponibilités nulles, notre problème est noté $P|AgreeG = (V, E), p_i = 1|C_{max}$.

En termes de la théorie des graphes, ce problème est équivalent au problème de la partition minimum du graphe de concordance, en cliques chacune de taille inférieure ou égale à m.

D'autre part dans la littérature, ce problème est équivalent au problème d'ordonnancement avec exclusion mutuelle en anglais : Mutual Exclusion Scheduling (M.E.S). Ce dernier a été introduit par Baker et Coffman [3] et est défini comme suit : n tâches avec des durées de traitement unitaires doivent être exécutées sur m processeurs identiques en un minimum de temps, sous contraintes que certaines tâches ne peuvent pas être exécutées simultanément parcequ'elles partagent une même ressource. De telles tâches sont dites en conflit.

Vu l'importance de leurs applications industrielles, plusieurs variantes du problème M.E.S ont été considérées dans la littérature. Le lecteur intéressé à ce

genre de problèmes peut consulter l'article de Krarup et De Werra [8] , recueil récent de Blazewicz et al. [9], Baker et Coffman [3], Jansen [10] et Gardi [6], [11]. Beaucoups de résultats sur le problème M.E.S on été cités dans la référence [8].

4.2 Ordonnancement sur une machine à traitement par batchs : cas $p_i = 1$ et r_i arbitraires

Dans [12] M. Boudhar et G. Finke ont étudié le problème d'ordonnancement suivant : n tâches indépendantes $J_1, T_2, ..., T_n$ doivent être exécutées sur une machine à traitement par batch $B1$ sachant qu'il existe une relation de compatibilité entre les tâches, où deux taches sont compatibles si elles peuvent être traitées simultanément dans un même batch. Cette relation est représentée par un graphe $G = (V, E)$ appelé graphe de compatibilité, où V représente l'ensemble des tâches et une paire de tâches est dans E si et seulement si ces dernières sont compatibles. Chaque tâche T_i a une durée de traitement p_i et une date de disponibilité r_i respectivement. La durée de traitement d'un batch est égal au maximum des durées de traitement des tâches appartenant à celui ci. Toutes les tâches d'un même batch commencent leur traitement à une même date et terminent à une même date. Il est aussi supposé que la capacité de la machine à traitement par batch peut être finie (elle peut traiter b tâches à la fois) ou infinie (elle peut traiter un nombre illimité de tâches à la fois). L'interruption des tâches n'est pas autorisée. Le problème consiste à chercher une partition $B_1, B_2, ..., B_q$ de l'ensemble des tâches, en batchs, ainsi que leurs dates de début d'exécution telles que $|B_j| \leq b$ pour $j = 1, 2, ..., q$ et minimisant la date de fin de traitement de l'ensemble des batchs (le makespan). Ce problème est noté : $B1|G = (V, E), b, r_i|C_{max}$ où G représente le graphe de compatibilité.

Notons que dans cette notation b est arbitraire, par contre si b est une constante égale à k, le problème est noté : $B1|G = (V, E), b = k, r_i|C_{max}$. Quand les dates de disponibilités sont nulles, le problème est noté $B1|G = (V, E), b|C_{max}$. Lorsque les durées de traitement des tâches sont unitaires, le problème est noté $B1|G = (V, E), b, r_i, p_i = 1|C_{max}$.

Dans ce qui suit nous montrons le lien qui existe entre le problème d'ordonnancement sur une machine à traitement par batchs et notre problème.

Proposition 1. *Les problèmes $Pm|AgreeG = (V, E), r_i, p_i = 1|C_{max}$ et $B1|G = (V, E), b = m, r_i, p_i = 1|C_{max}$ sont équivalents.*

Preuve. Soient (P) et (P') les problèmes $Pm/AgreeG = (V, E), r_i, p_i = 1/C_{max}$ et $B1|G = (V, E), b = m, r_i, p_i = 1|C_{max}$ respectivement.

Supposons qu'il existe un ordonnancement réalisable σ pour le problème (P). A tout instant t tel qu'il existe au moins une tâche ordonnancée (par rapport à σ) à t, on associe le batch $B_t = \{$tâches ordonnancées par σ à l'instant $t\}$. Les batchs ainsi définis forment une partition de l'ensemble des tâches. En ordonnançant chacun de ces batchs B_t à l'instant t, on obtient un ordonnancement réalisable σ' pour le problème (P') avec un makespan $C_{max}(\sigma') = C_{max}(\sigma)$. Comme illustration voir figure 1 , qui représente les ordonnancements σ et σ' dans le cas de

deux machines, tel qu'ils exitent deux tâches J_t^1 et J_t^2 ordonnancées à l'instant t, par rapport à l'ordonnancement σ.

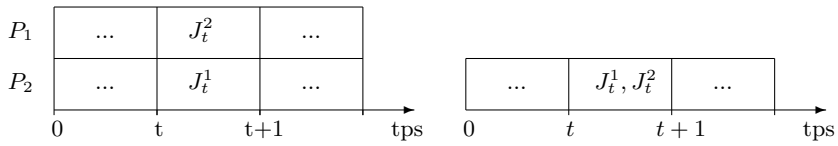


Figure 1. Les ordonnancements σ et σ', $B_t = \{J_t^1, J_t^2\}$

Réciproquement, supposons que (P') admet un ordonnancement réalisable σ'. Cet ordonnancement est défini par une partition des tâches en batchs $B_1, ..., B_q$ vérifiant $|B_j| \leq m$ pour tout $j = 1, 2, ..., q$, avec leurs dates d'exécution $t_1, t_2, ..., t_q$ respectivement. La valeur de la fonction objectif en σ' est $C_{max}(\sigma')$. A partir de σ', on construit un ordonnancement réalisable σ pour le problème (P) comme suit : pour tout j $(j \in \{1, 2, ..., q\})$, on ordonnance les tâches de B_j à l'instant t_j. L'ordonnancement σ ainsi obtenu est réalisable du fait que σ' l'est aussi. D'autre part il est clair que la valeur de la fonction objectif pour (P), en σ est $C_{max}(\sigma) = C_{max}(\sigma')$. ■

Lorsque le nombre de machines m est variable, c'est-à-dire lorsque m fait partie des données, avec le même argument on montre que les problèmes problèmes $P|AgreeG = (V, E), r_i, p_i = 1|C_{max}$ et $B1|G = (V, E), m, r_i, p_i = 1|C_{max}$ sont polynomialement équivalents. En particulier, pour le cas des durées de traitement unitaires, on en déduit que les trois problèmes $P|AgreeG = (V, E), p_i = 1|C_{max}$, $B1|G = (V, E), m, p_i = 1|C_{max}$ et le problème M.E.S, sont deux à deux polynomialement équivalents.

5 Inéxistence d'un algorithme absolu

Un algorithme fournissant une approximation absolue (ou algorithme absolu) est une heuristique donnant, pour toute instance du problème, une solution approchée dont l'écart par rapport à la solution optimale est borné par une constante. Dans le résultat suivant, on montre qu'il n'existe pas d'algorithmes absolus pour le problème général traité à moins que $P = NP$.

Théorème 1. *Si $P \neq NP$ alors il n'existe pas d'algorithme polynomial fournissant une approximation absolue pour le problème $P|AgreeG = (V, E), r_i|C_{max}$.*

Preuve. On procède par contradiction en utilisant un changement d'échelle. Supposons dans le cas contraire qu'un tel algorithme, disant A existe, fournissant une approximation absolue d'ordre K c-à-d $|A(I) - OPT(I)| \leq K$ pour toute instance I. Soit I une instance du problème général, représentée par la donnée d'un ensemble de n tâches $J_1, J_2, ...J_n$ avec leurs durées de traitement respectives $p_1, p_2, ..., p_n$ et le graphe de concordance G. Nous construisons une nouvelle

instance I' pour le problème comme suit : $J'_1, J'_2, ..., J'_n$ sont n tâches en correspondance avec les tâches $J_1, J_2, ..., J_n$ de sorte que le graphe de concordance G' est isomorphe au graphe G. Pour tout $i = 1, 2, ..., n$, la durée de traitement de la tâche J'_i est $p'_i = (K+1)p_i$.

Dans la suite, on montre qu'à tout ordonnancement réalisable σ de l'instance I, correspond un ordonnancement réalisable σ' de l'instance I' et vice-versa tel que $C_{\max}(\sigma') = (K+1)C_{\max}(\sigma)$ et en particulier $OPT(I') = (K+1)OPT(I)$.

En effet, soit σ un ordonnancement réalisable pour I et soient t_i et c_i les dates de début et fin de traitement respectivement pour toute tâche J_i, par rapport à σ. On construit l'ordonnancement σ' de I' comme suit : toute tâche J'_i est traitée à la date $t'_i = (K+1)t_i$ sur la même machine sur laquelle la tâche J_i a été traitée. Montrons que σ' est un ordonnancement réalisable pour I'. Pour cela , soit $\{J'_i, J'_j\}$ une paire de tâches qui se chevauchent sur un certain intervalle, par rapport à l'ordonnancement σ'. Ceci est équivalent à dire que $t'_j < c'_i$ et $t'_i < c'_j$, où t'_k et c'_k représentent la date de début et de fin de traitement respectivement de la tâche J'_k par rapport à σ'. En divisant par $K+1$, il en résulte que leurs tâches correspondantes respectives J_i et J_j de l'instance I vérifient aussi les relations $t_j < c_i$ et $t_i < c_j$. Ceci implique que les tâches J_i et J_j se chevauchent par rapport à l'ordonnancement σ. D'après la réalisabilité de σ, les tâches J_i et J_j sont alors concordantes pour l'instance I. Ceci veut dire que ces dernières sont adjacentes dans G. Comme G' est isomorphe à G, on en déduit que les tâches J'_i et J'_j sont adjacentes dans G', donc concordantes par rapport à l'instance I' et par conséquent l'ordonnancement σ' est réalisable.

Quant à la relation $C_{\max}(\sigma') = (K+1)C_{\max}(\sigma)$, soit J'_l une tâche de G' telle que $C_{\max}(\sigma') = c'_l$. Soit J_l la tâche correspondante à J'_l. On a : $c'_l = t'_l + p'_l = (K+1)t_l + (K+1)p_l = (K+1)c_l$. On affirme que $C_{\max}(\sigma) = c_l$. En effet, il est clair que $C_{\max}(\sigma) \geq c_l$ mais si $C_{\max}(\sigma) > c_l$, alors la tâche J_k de G dont la date de fin de traitement égale à $C_{\max}(\sigma)$ vérifiera $c_k > c_l$, donnant $c'_k > c'_l$ et ceci contredit le fait que $C_{\max}(\sigma') = c'_l$. Il en résulte que $C_{\max}(\sigma) = c_l$ et par conséquent $C_{\max}(\sigma') = (K+1)C_{\max}(\sigma)$.

D'autre part on peut facilement montrer, comme on a fait précédemment qu'un ordonnancement réalisable σ est optimal pour l'instance I si et seulement si l'ordonnancement correspondant σ' pour l'instance I' est optimal et que $OPT(I') = (K+1)OPT(I)$.

Réciproquement, on peut montrer de la même manière qu'à tout ordonnancement réalisable σ' de l'instance I' correspond un ordonnancement réalisable σ de I satisfaisant les relations $C_{\max}(\sigma') = (K+1)C_{\max}(\sigma)$ et $OPT(I') = (K+1)OPT(I)$.

Maintenant, on exécute l'algorithme A pour l'instance I' et soient σ'_A l'ordonnancement réalisable obtenu et σ_A l'ordonnancement réalisable correspondant pour l'instance I. Clairement on a $|A(I') - OPT(I')| \leq K$, mais $A(I') = C_{\max}(\sigma'_A) = (K+1)C_{max}(\sigma'_A)$ donc $|(K+1)C_{max}(\sigma_A) - (K+1)OPT(I)| \leq K$, ce qui implique que $|C_{max}(\sigma_A) - OPT(I)| \leq \frac{K}{K+1} < 1$.

Comme $C_{max}(\sigma_A)$ et $OPT(I)$ sont des entiers, on a alors $C_{max}(\sigma_A) = OPT(I)$. Il s'en suit que σ_A est un ordonnancement réalisable optimal pour

l'instance I et donc le problème général peut être résolu polynomiallement par l'algorithme A, contredisant la supposition que $P \neq NP$. ■

Dans la section suivante nous considérons le cas lorsque le graphe de concordance est un graphe biparti complet avec des durées de traitement unitaires et dates de disponibilités arbitraires.

6 Graphe biparti complet avec $m = 2$, $p_i = 1$, r_i arbitraires

Dans cette section nous considérons le problème pour $m = 2$, dans lequel le graphe de concordance est un graphe biparti complet et que les dates de disponibilités sont arbitraires. On montre que ce problème est polynomial et on propose un algorithme polynomial pour sa résolution. Le graphe de concordance est noté par $G = (S_1; S_2, E)$, où S_1, S_2 sont deux stables formant une partition de l'ensemble des sommets de G. Soit $S_1 = \{J_1, J_2, ..., J_{|S_1|}\}$ et $S_2 = \{J_{|S_1|+1}, J_{|S_1|+2}, ..., J_{|S_1|+|S_2|}\}$. Soient $r_1, r_2, ..., r_{|S_1|}$ les dates de disponibilités des tâches de S_1 respectivement et celles de S_2 sont $r_{|S_1|+1}, r_{|S_1|+2}, ..., r_{|S_1|+|S_2|}$. Considérons l'algorithme suivant dont le principe est comme suit : pour chacun des stables S_1 et S_2 on construit un ordonnancement optimal pour le problème $1|r_i, p_i = 1|C_{max}$ et puis on combine les deux ordonnancements optimaux.

Algorithme 1
Début

1. arranger les tâches de S_1 selon l'ordre croissant de leur dates de disponibilités, disant $J_1, J_2, ..., J_{|S_1|}$
2. $t_1 := r_1$
3. pour $i := 2$ to $|S_1|$ mettre $t_i := max(r_i, t_{i-1} + 1)$
4. pour tout i ($i = 1$ to $|S_1|$) ordonnancer la tâche J_i à la date t_i sur $P1$
5. arranger les tâches de S_2 selon l'ordre croissant de leur dates de disponibilités, soient $J_{|S_1|+1}, J_{|S_1|+2}, ..., J_{|S_1|+|S_2|}$
6. $t_{|S_1|+1} := r_{|S_1|+1}$
7. pour $j := |S_1| + 2$ to $|S_1| + |S_2|$ mettre $t_j := max(r_j, t_{j-1} + 1)$
8. pour tout j ($j = |S_1| + 1$ à $|S_1| + |S_2|$) ordonnancer la tâche J_j à la date t_j sur $P2$

Fin.

Théorème 2. *L'algorithme 1 résout le problème $P2|AgreeG = (S_1 x S_2, E), r_i, p_i = 1|C_{max}$ polynomiallement en $O(nlogn)$ avec les conditions $m \geq n$, graphe de concordance arbitraire, durées de traitement unitaires et dates de disponibilités arbitraires.*

Preuve. Montrons qu'il existe une solution optimale dans laquelle chaque tâche J_i of S_1 est ordonnancée sur $P1$, à l'instant t_i obtenu par l'algorithme 1. Supposons que, dans une solution optimale il existe une tâche J_i de S_1 ordonnancée

après la date t_i. Trois cas possibles peuvent avoir lieu. Dans le premier cas aucune tâche n'est ordonnancée à la date t_i, dans ce cas J_i est translatée et ordonnancée à l'instant t_i sur $P1$. Dans le deuxième cas, il existe une tâche J_k de S_1 ordonnancée à t_i et une tâche J_j de S_2 ordonnancée à l'instant t_i. Dans ce cas, on permute les tâches J_i et J_k, et si J_i n'est pas sur $P1$, on la permute avec J_j. Le troisième cas est tel qu'il existe seulement une tâche, disant J_l de S_1 ordonnancée à t_i. Dans ce cas on permute les tâches J_i et J_l, et si J_i n'est pas sur $P1$, on l'ordonnance sur $P1$. En appliquant cet argument au plus $|S_1|$ fois, on obtient un ordonnancement dans lequel chaque tâche J_i de S_1 est ordonnancée sur $P1$, à l'instant t_i. Notons que le makespan n'a pas changé et que les dates de disponibilités ont été respectées, par conséquent l'ordonnancement obtenu est une solution optimale vérifiant la condition demandée. Un argument similaire (comme pour les tâches de S_1), peut être appliqué à la solution optimale juste trouvée, pour les tâches de S_2 et ceci prouve l'existence d'une solution optimale telle que chaque tâche J_i de S_1 est ordonnancée sur $P1$, à l'instant t_i et que chaque tâche de J_j de S_2 est ordonnancée sur $P2$, à l'instant t_j obtenu par l'algorithme 1. On en déduit que l'ordonnancement obtenu par l'algorithme 1 est optimal. La complexité de cet algorithme est égale à $O(nlogn)$ puisqu'elle consiste en l'arrangement des dates de disponibilités selon l'ordre croissant, des translations de tâches et permutations qui peuvent être accomplies en $O(n)$. ■

7 Conclusion

Dans ce papier, nous avons étudié le problème d'ordonnancement de tâches sur des machines parallèles et identiques sous contraintes de concordance de tâches, représentées par un graphe dit graphe de concordance dont le but est de minimiser le makespan. Nous avons proposé une formulation mathématique sous forme d'un programme linéaire en variables bivalentes et réelles et nous avons donné une nouvelle application. Aussi, nous avons présenté un état de l'art, en particulier nous avons montré le lien qui existe entre le problème d'ordonnancement sur une machine à traitement par batchs et notre problème. Ensuite nous avons établi un résultat d'inapproximabilité pour le problème général et un nouveau résultat polynomial dans le cas de deux machines.

Références

1. Bendraouche M, Boudhar M. Scheduling jobs on identical machines with agreement graph. Computers and Operations Research 2012 ; 39 382–390.
2. Graham RE, Lawler EL, Lenstra JK, Rinnooy Kan AHG. Optimisation and approximation in deterministic sequencing and scheduling : a survey, Ann. Discrete Math. 1979 ; 4, 287-326.
3. Baker BS, Coffman EG. Mutual Exclusion Scheduling. Theoretical Computer Science 1996 ; 162 : 225-243.
4. Halldorsson MM, Kortsarz G, Proskurowski A, Salman R, Shachnai H, Telle JA. Multicoloring trees.

5. Bodlaender HL, Jansen K. Restrictions of graph partition problems part I. Theoretical Computer Science 1995 ; 148 : 93-109.
6. Gardi F. Planification d'horaires de travail et théorie des graphes. In Actes du 5ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, pp. 99-100. Laboratoire d'Informatique d'Avignon (LIA, Université d'Avignon et des Pays de Vaucluse), Avignon, France.
7. Even G, Halldorson MM, Kaplan L, Ron D. Scheduling with conflicts : online and offline algorithms. Journal of scheduling 2009 ; 12 : 199-224.
8. Krarup J, De Werra D. Chromatic optimization : limitations, objectives, uses, references. European Journal of Operational Research 1982 ; 11, pp. 1–19.
9. Blazewicz J, Ecker K.H, Pesch E, Schmidt G et Weglarz J. Scheduling Computer and Manufacturing Processes. Springer-Verlag, Berlin, Allemagne. (deuxième édition) 2001.
10. Jansen K. The mutual exclusion scheduling problem for permutation and comparability graphs. Inform and Comput 2003 ; 180 2 : 71-81.
11. Gardi F. Mutual exclusion scheduling with interval graphs or related classes Part I. Discrete Appl Math 2009 ; 157 : 19-35.
12. Boudhar M, Finke G. Scheduling on a batch machine with job compatibilities. Belgium Journal of Operations Research, Statistics and Computer Science (JORBEL) 2000 ; 40 :69-80.

Un modèle de graphe pour l'ordonnancement optimisé des règles d'un système à base de règles chaînage avant

Mohammed Mahieddine[1], Farouk Hannane[2], Mohamed Benatallah[2],

[1] Laboratoire LRDSI- Equipe GLODOO Université de Blida (Algérie).
mo_mahieddine@mail.univ-blida.dz
[2] Université de Blida, F.S., Dépt. De mathématiques BP 270, Route de Soumaa, Blida(Algérie).
fhannanefr@yahoo.fr, m_benatallah@yahoo.fr

Résumé — Nous allons, dans cet article, présenter un nouveau modèle pour l'ordonnancement des règles de la base de règles, par des graphes orientés, permettant d'optimiser le temps d'inférence du moteur d'inférence chaînage avant d'un système à base de règles.

Mots-Clés- moteur d'inférence chaînage avant; système à base de règles; règle de production; graphe de relations; circuits; feedback vertex set.

1 Introduction

Feigenbaum [3] a définit un système à base de règles comme étant "un logiciel intelligent qui utilise des connaissances et un procédé d'inférence pour résoudre des problèmes. Certains problèmes sont assez difficiles à résoudre et requièrent une expertise humaine significative pour arriver à une solution. Les connaissances pour les systèmes à base de règles comprennent des faits et des heuristiques ".

Les représentations de la connaissance sont très importantes dans le domaine du raisonnement, et restent d'actualité dans le domaine de la recherche [6], parce-que pour des bases de connaissance avec des centaines ou des milliers de règles, le nombre de chemins possibles d'inférence est très grand.

Plusieurs techniques de transformation des bases de règles ont été proposées dans la littérature. Les réseaux de Pétri ont été décrits dans [10] et les graphes orientés dans [8]. Tous les graphes cités précédemment, n'ont pas eu d'application, à notre connaissance, dans la modélisation de règles de productions ayant plusieurs conditions et conclusions

Dans le travail qui suit, nous allons développer et appliquer les mécanismes qui vont permettre de faire le meilleur choix de la prochaine règle applicable à utiliser

dans l'inférence. En vue de cela nous proposons un modèle qui utilise les graphes orientés comme structure de représentation de la connaissance [7].

Dans la suite de ce papier, nous allons commencer, en section 2, par présenter le domaine des systèmes à base de règles. Nous exposons ensuite, dans la section 3, tous les détails de la modélisation proposée de la base de règles d'un système à base de règles au moyen d'un graphe orienté simple. Dans la quatrième section, nous allons présenter le logiciel que nous avons implémenté en langage java, dans le but d'automatiser le processus d'ordonnancement proposé et de tester la validité de notre modèle par l'application de ce logiciel pour opérer des tests comparatifs avec ceux produits sur une base non-ordonnée. Nous terminons le papier par une conclusion générale et quelques perspectives d'avenir.

2 Système à base de règles

Un système à base de règles [1] est un système de déduction logique basé sur un moteur d'inférence et une base de connaissances. Il est la transcription logicielle de la réflexion d'un expert dans un domaine donnée. Il est capable de déduction logique et de produire une solution qui semble la plus juste. Toutefois, il reste un outil d'aide à la décision et il reste toutefois loin de pouvoir remplacer le raisonnement d'un expert, d'ailleurs il n'est concevable que pour les domaines dans lesquels il existe des experts humains.

Un système à base de règles est principalement composé d'une *base de connaissances***,** et *d'un moteur d'inférence.*

La base de connaissances est composée d'un ensemble de faits et des règles de production.

Le moteur d'inférence utilise un algorithme qui permet, à partir d'une base de règles et d'une base de faits, d'éventuellement déduire de nouveaux faits.

A. Les principes d'un système à base de règles

Les règles de production sont un modèle de représentation des connaissances très répandu [2]. Une règle d'inférence est de la forme :

Si Condition(s) ***alors*** Conclusion(s).

La méthode de raisonnement employée dans notre cas par un moteur d'inférence est le raisonnement avec un moteur d'inférence chaînage avant.

B. Le chaînage avant

Dans le mode de *chaînage avant*, le moteur d'inférence part des faits pour arriver au but, c'est-à-dire qu'il ne sélectionne que les règles dont les conditions de la partie gauche sont vérifiées (i.e. appartiennent à la base de faits), puis applique la première de ces règles, cela va avoir comme conséquence d'ajouter la conclusion à la base de faits. Ce processus est réitéré jusqu'à ce qu'il n'y ait plus de règle applicable ou que le but soit atteint.

Voici l'algorithme du chaînage avant :

ALGORITHME DU CHAINAGE AVANT

```
ENTREE : BF (base de faits), BR (base de règles), F
(proposition à vérifier)
  DEBUT
TANT QUE F n'est pas dans BF ET QU'il existe dans BR
une règle applicable
FAIRE
      Prendre la première règle applicable R
      BR = BR - R (désactivation de R)
BF = BF union conclusion(R) (déclenchement de la règle
R, sa conclusion est rajoutée à la base de faits)
      FIN TANT QUE
      SI F appartient à BF  ALORS
     F est établi (succès)
      SINON
      F n'est pas établi (échec)
  FIN.
```

On remarque que cet algorithme n'indique pas comment choisir la prochaine règle applicable.

3 Le modèle à base de graphe proposé

3.1 Définition

La base de règles entière peut être modélisée par un graphe orienté G (V, E). Les sommets V représentent les règles Ri, et les arcs E représentent les relations entre les règles. Comme par exemple la conclusion C de la règle R_1 appartient aux conditions de la règle R_2. Nous exprimons cela au moyen d'un arc orienté allant du sommet R_1 vers le sommet R_2.

Nous allons considérer une base composée des cinq règles $\{R_1R_2R_3R_4R_5\}$ suivantes (voir Figure 3.1) qui va nous servir comme exemple pour l'application de notre technique de modélisation. Pour la simplification nous employons le calcul propositionnel de la logique.

R_1 : **SI** A **ET** B **ALORS** C
R_2 : **SI** C **ET** D **ALORS** F
R_3 : **SI** F **ET** B **ALORS** E
R_4 : **SI** F **ET** A **ALORS** G
R_5 : **SI** G **ET** F **ALORS** B

Figure.3.1 Règles de production.

Ceci nous conduit à représenter notre base de règles comme un graphe orienté. Le modèle de représentation de cette base de règles est donné par la figure 3.2.

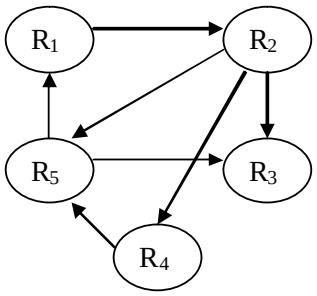


Figure.3.2 Graphe de relation des règles.

Dans une matrice d'adjacences, les lignes et les colonnes représentent les sommets du graphe.

- Un 1 à la position (i, j) signifie que le sommet i est adjacent au sommet j.
- Sinon, on place un 0.

Le graphe de relations précédent est représenté par sa matrice d'adjacence donnée en figure 3.3.

	R_1	R_2	R_3	R_4	R_5
R_1	0	1	0	0	0
R_2	0	0	1	1	1
R_3	0	0	0	0	0
R_4	0	0	0	0	1
R_5	1	0	1	0	0

Figure.3.3 La matrice d'adjacence des règles.

La matrice d'adjacence précédente va être utilisée pour calculer le demi-degré extérieur $d^+(x_i)$ et le demi-degré intérieur $d^-(x_i)$ de chaque sommet. La somme des lignes dans la matrice représente le demi-degré positif et la somme des colonnes de la matrice représente le demi-degré négatif. En théories des graphes ils sont appelés respectivement les demi-degrés des arcs sortants et ceux des arcs entrants. On calcule les demi-degrés pour les utiliser dans le problème du choix des sommets.

Afin de savoir quelle règle est obtenue à partir de quelle autre, nous allons effectuer, ce qui est connu dans la théorie de graphes par, la *mise en ordre* d'un

graphe orienté. Il est exigé que le graphe obtenu soit un graphe orienté acyclique ou *DAG* (Directed Acyclic Graph), c'est-à-dire un graphe orienté qui ne possède pas de circuit. Un circuit est un chemin du graphe dont les extrémités sont confondues. La contrainte majeure dans notre procédure réside dans l'existence de ces circuits, que nous allons par conséquent chercher à éliminer : ce problème est appelé *feedback vertex set* (*FVS*) c.-à-d. l'ensemble de sommets supprimé pour obtenir un graphe sans circuit) [9].

A partir de cette modélisation, notre problème consisterait donc à trouver le nombre minimal de sommets dans un graphe qui, une fois choisis, vont nous permettre d'en supprimer tous les circuits (i.e. *MFVS* l'ensemble des nombres minimale de sommets de *FVS)*. Dans la partie suivante nous allons aborder les techniques utilisées pour la simplification de ce genre de graphe. Bien sûr qu'aucune de ces techniques ne va modifier la solution de *MFVS* représentée par le graphe initial.

3.2 Règles de réduction de graphe

Dans cette section, nous donnons deux procédures de réduction de graphe, la première consiste de supprimer certains sommets et arcs [7], et qui sont très utiles dans la simplification des méthodes de résolution dans les graphes, et l'autre permet de décomposer le graphe en plusieurs sous graphes.

Nous éliminons tous les sommets qui contiennent que des arcs entrants ou que des arcs sortants car, si un sommet du graphe n'a pas d'arc entrant ou d'arc sortant alors celui-ci ne peut pas faire partie d'un circuit dans le graphe. Nous allons donc supprimer ce sommet du graphe ainsi que tous ses arcs adjacents, du fait du principe de l'algorithme de Morimant [5].

Nous commençons par éliminer, à partir de la matrice d'adjacence booléenne, tous les sommets qui contiennent des lignes et des colonnes complètement à zéros, Il ne va rester que les sommets dont les lignes et les colonnes sont non nulles et qui constituent des circuits. Si nous appliquons cette procédure à la matrice précédente, nous allons remarquer que le sommet R_3 comporte une ligne nulle, nous allons donc le supprimer de la matrice d'adjacence, et nous allons ainsi aboutir au digraphe réduit dont la matrice d'adjacence est donnée en figure 3.4.

	R_1	R_2	R_4	R_5
R_1	0	1	0	0
R_2	0	0	1	1
R_4	0	0	0	1
R_5	1	0	0	0

Figure.3.4 La matrice d'adjacence réduite.

La deuxième phase du prétraitement va consister à énumérer les circuits. La recherche de circuits d'un graphe G revient, en fait, à chercher les circuits du graphe réduit G' de G.

Trouver l'énumération des circuits de plusieurs graphes de taille réduite est une tache plus aisée que celle utilisant un seul graphe de taille plus importante. Cette observation nous amène à décrire la technique qui consiste à partitionner un graphe en composantes fortement connexes (CFCs).

Définition : Une composante fortement connexe CFC est un sous-ensemble de sommets tel qu'il existe un chemin entre deux sommets quelconques.

D'après la définition précédente, il est clair que tous les sommets appartenant à un même circuit du graphe initial G appartiennent à une et une seule CFC. Par conséquent, après identification des CFCs, les arcs liant les CFCs entre elles peuvent être supprimés et chaque CFC peut être traitée indépendamment des autres. Ainsi l'énumération des circuits d'un graphe est égale à l'union des circuits de toutes ses composantes fortement connexes.

K.A. Hawick et H.A. James (2008) [4], ont proposé un algorithme d'énumération des circuits pour les graphes orientés dans lequel ils ont utilisé l'idée de Johnson pour construire un arbre de recherche récursif pour l'énumération des circuits. L'algorithme, implémenté dans le langage de la programmation de D, a produit une exécution efficace et un rendement de mémoire élevé, néanmoins pour les graphes à fortes connexité, cet algorithme n'est pas praticable que pour ayant un petit nombre de sommets (moins de la quarantaine).

De notre côté nous avons choisi d'appliquer cette méthode sur le digraphe réduit précédemment, il va en résulter des circuits.

L'ensemble des circuits est $\{C_1 = \{\{R_1R_2R_5\}, \quad C_2 = \{R_1R_2R_4R_5\}\}$.

4 Construction de la matrice cyclomatique

La matrice cyclomatique C comporte des circuits et des sommets : les colonnes de la matrice représentent les circuits et les lignes représentent les sommets R_i.

La matrice cyclomatique est une matrice booléenne, dans laquelle l'existence d'un sommet dans le circuit se traduit par l'affectation d'un 1 à l'intersection de la ligne Ri et de la colonne Cj ; ce qui signifie que le sommet Ri est dans le circuit Cj ; et l'absence d'un sommet dans le circuit par l'affectation d'un 0, signifiant que le sommet Ri n'appartient pas au circuit Cj.

	C1	C2
R_1	1	1
R_2	1	1
R_3	0	0
R_4	0	1
R_5	1	1

Figure.3. 5 La matrice cyclomatique.

Nous allons utiliser la matrice cyclomatique précédente, pour calculer la somme des lignes dont nous prendrons le maximum, an sachant que la somme des lignes dans la matrice cyclomatique représente le nombre de circuits qui contiennent ce sommet.

	C_1	C_2	la somme R_{ij}	MAX(Ri)
R_1	1	1	**2**	**2**
R_2	1	1	**2**	**2**
R_3	0	0	**0**	/
R_4	1	0	**1**	/
R_5	1	1	**2**	**2**

Figure.3.6 La matrice cyclomatique et le maximum.

Nous allons prendre ensuite le sommet correspond au maximum pour l'utiliser dans le problème du choix des sommets d'un *feedback*.

Si nous choisissons le sommet qui reçoit le maximum pour l'ajouter à l'ensemble de feedback nous supprimons les circuits qui passent par ce sommet, cela va revenir, en fait, à la même procédure que celle de la matrice réduite. Le problème qui en découle est celui du choix du sommet dans le cas où il existe plusieurs sommets ayant un même maximum. Nous allons donc passer à la deuxième étape (que nous appelons procédure de choix) qui est celle du choix des sommets que nous allons ordonner par les demi-degrés des sommets.

La procédure de choix s'effectue comme suit :

- Si $d^+(R_1) =< d^+(R_2) =< \ldots =< d^+(R_n)$

  Alors nous prenons le sommet qui contient le demi-degré extérieur minimum

- Sinon $d^-(R_n) =< \ldots =< d^-(R_2) =< d^-(R_1)$

  Nous prenons le sommet qui contient le demi-degré intérieur maximum

- Si les deux cas n'existent pas, on prend alors un sommet au hasard.

4.1 Le choix des sommets

A partir de la matrice cyclomatique nous prenons le maximum de la somme des lignes. Ce maximum est égal à 2 dans la matrice précédente, et les sommets qui correspondent à ce maximum sont donc les sommets R_1, R_2 et R_5.

Nous avons $d^+(R_1)=1$, $d^+(R_2)=3$ et $d^+(R_5)=2$, nous allons donc choisir le sommet R_1 car le degré des arcs sortants est plus petit que celui de R_2 et de R_5.

4.2 Comment désactiver un circuit

Dans notre problème nous ne pouvons ni supprimer, ni dupliquer les sommets car ces sommets représentent des règles de production. La méthode pour laquelle nous avons opté pour désactiver les circuits est la suivante: Nous affectons des zéros à la ligne du sommet de choix R_1, et c'est cette matrice d'adjacence modifiée sans circuits que nous allons utiliser par la suite.

4.3 La matrice d'adjacence modifiée sans circuits

A partir de la matrice d'adjacence nous allons appliquer toutes les procédures précédentes. Nous utilisons ensuite la nouvelle matrice d'adjacence modifiée.

	R_1	R_2	R_3	R_4	R_5
R_1	**0**	**0**	**0**	**0**	**0**
R_2	0	0	1	1	1
R_3	0	0	0	0	0
R_4	0	0	0	0	1
R_5	1	0	1	0	0

Figure.3.7 La matrice d'adjacence modifiée sans circuits.

A partir de la matrice d'adjacence modifiée nous pouvons appliquer la mise en ordre d'un graphe orienté sans circuit, et nous allons supprimer, à chaque itération, tous les sommets qui correspondent à une colonne nulle, qui n'ont donc pas de sommets précédents qui leurs sont reliés.

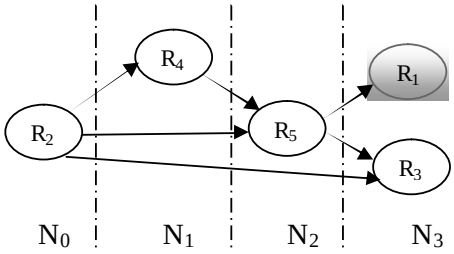


Figure.3.8 La mise en ordre d'un graphe orienté sans circuit.

1ière exécutée ; R_2 :	**SI**	C	**ET**	D	**ALORS**	F.
2 ième exécutée ; R_4 :	**SI**	F	**ET**	A	**ALORS**	G.
3 ième exécutée ; R_5 :	**SI**	G	**ET**	F	**ALORS**	B.
4ième exécutée ; R_1 :	**SI**	A	**ET**	B	**ALORS**	C.
5 ième exécutée ; R_3 :	**SI**	F	**ET**	B	**ALORS**	E.

Figure 3.9 L'ordonnancement de la base de règles.

Cela va nous faire aboutir à une base de règles ordonnée sur laquelle nous allons faire appliquer le moteur d'inférence chaînage avant d'un système à base de règles. Nous pouvons aussi faire le cycle d'exécution en deux fois en parcourant les règles dans cet ordre. Dans le cas où il y'ait des règles non exécutées nous allons faire un deuxième parcours. Donc nous avons appliqué le moteur d'inférence chaînage avant sur la base de règles une seule fois seulement dans le cas où le graphe de relation est sans circuits, et deux fois s'il contient des circuits (problème de feedback). En plus, s'il y a des règles n'exécute pas alors on à jamais exécuté.

5 Synoptique de l'algorithme global

La méthode que nous avons mise en œuvre est constituée de deux parties distinctes : La première concerne la phase de réduction, qui va nous permettre de supprimer certains sommets du graphe. Cette première partie est une méthode exacte. La deuxième partie est par contre une heuristique : on propose une heuristique basée sur une énumération des circuits du graphe borné. Cette dernière comporte plusieurs paramètres car le choix des sommets est exigé en raison de l'utilisation du *MFVS*.

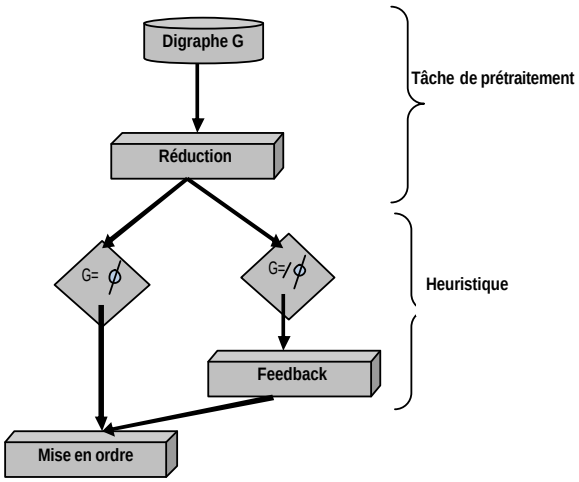


Figure.3.10 Synoptique de l'algorithme global.

6 Implémentation et teste

Suite à l'étude des modélisations existantes, nous avons choisi l'utilisation d'une modélisation à base de graphes orientés que nous avons appelé *graphes de relations*. Ce choix nous a été dicté par la considération de vouloir bénéficier des algorithmes existants dans le domaine des graphes orientés.

Pour automatiser le processus de construction de notre modèle ainsi que pour son utilisation dans les tests, nous avons réalisé une implémentation orientée objets en utilisant le langage java.

L'architecture de notre logiciel est axée sur trois principales composantes :

1. La composante *système à base de règles* comportant les classes en vue de la réalisation de l'inférence en chaînage avant à partir d'une base de faits et d'une base de règles.

2. Le composant *graphe de relation* comportant les classes produisant le comportement pour la réalisation de la mise en ordre des règles de la base de règles, à partir d'un modèle de graphe orienté.

3. d'une *interface graphique*, offrant des possibilités de manipulation et d'affichage expressives et conviviales.

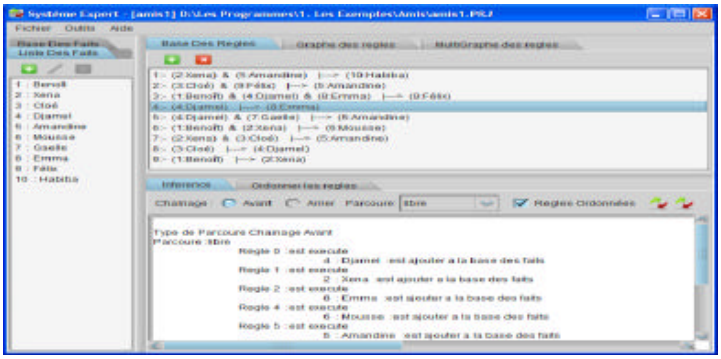


Figure 4.1 Logiciel du modèle proposé.

Après la phase de réalisation du modèle, nous validons pratiquement ce modèle, en effectuant à chaque fois une batterie de tests, afin de comparer entre le modèle proposé et la méthode utilisant des règles non-ordonnées.

Nous avons effectué plusieurs tests d'optimisation en jouant sur le changement du nombre de règles à optimiser en vue de comparer le temps d'exécution de l'inférence. Nous avons utilisé, à cet effet, une base de règles ordonnées au moyen de notre modèle et une autre non ordonnée.

Le tableau de la figure 4.1 représente une copie d'écran du fonctionnement du logiciel réalisé, affichant les résultats produits par des moyennes calculées sur 4 tests. Nous avons testé notre logiciel sur plusieurs bases de règles avec un nombre variable de règles de production.

Les Bases de Règles	B_1	B_2	B_3	B_4
Nombres des règles dans une base de règles	**50**	**100**	**150**	**200**
Temps d'exécution des Règles Ordonnée (msec)	31	62	94	125
Temps d'exécution des Règles Non Ordonnée (msec)	141	156	172	203

Figure 4.2 Tableau de test des temps d'exécution des bases de règles.

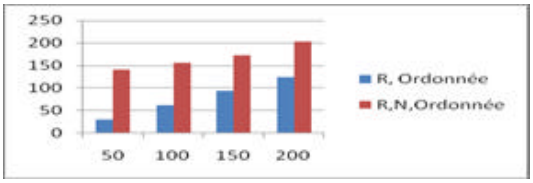


Figure 4.3 Comparaison entre les règles ordonnées et les non ordonnés.

D'après le tableau donné en figure 4.2, Nous voyons que la solution optimisée est obtenue par le modèle des règles ordonné, permettant d'atteindre ainsi un temps d'exécution minimum. L'histogramme de la figure 4.3 montre clairement le degré d'optimisation atteint, et la différence du temps d'exécution entre les deux méthodes, et cela même dans le cas où le nombre de règles est très différent entre les bases de règles.

7 Conclusion et perspectives d'avenir

Dans ce papier nous avons présenté un nouveau modèle optimisé d'ordonnancement des règles de la base de règles d'un système à base de règles chaînage avant. Le modèle proposé repose sur une structure de graphe pour modéliser la base de règles, et une technique utilisant pour sa mise en ordre. Un logiciel a été confectionné, en langage java, dans le but d'implémenter le modèle, puis nous avons utilisé ce dernier pour tester les principales phases de notre technique. Nous avons aussi présenté un histogramme comparatif des résultats des tests effectués sur une base de règles non-ordonnée et une deuxième ordonnée en utilisant le modèle d'ordonnancement proposé. Dans notre travail nous nous somme restreints à des bases de règles de grandes tailles. Ces simulations montrent assez nettement l'intérêt de l'approche proposée dans l'optimisation des bases de règles.

A l'avenir, il va nous rester à tester notre modèle sur de très grandes bases de règles, et à revoir la conception du logiciel proposé en vue d'introduire des patrons de conception.

References

1. Belaïd, S., "Intégration des problèmes de satisfaction de contraintes distribués et sécurisés dans les systèmes d'aide à la décision à base de connaissances", Thèse doctorat, université de Paul Verlaine-Metz, école doctorale IAEM Lorraine, 10 décembre 2010.
2. Brachman, R.J. and Levesque, H.J. Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, CA, 2004.
3. Fortin, N., " Conception d'outils logiciels d'aide à la décision appliqués au de vie des anodes d'une aluminerie", université de Québec à Chicoutimi, Juillet 2008.
4. Hawick, K.A., et James, H.A., "Enumerating Circuits and Loops in Graphs with Self-Arcs and Multiple-Arcs", Computer Science, Institute for Information and Mathematical Sciences, Massey University, North Shore 102-904, Auckland, New Zealand. Technical Report CSTN-013. 2008.
5. Hélary, J.M., "Algorithmique des Graphes", IFSIC, Cours C66, Juin 2004.
6. Hemmer, M.C., "Expert systems in chemistry research", ISBN 978-1-4200-5323-4, CRC Press, Taylor & Francis Group, Boca Raton London New York, 2008.
7. Lacomme P.., C. Prins C., and Sevaux M.., "Algorithmes de graphes". Eyrolles 2e edition 2003. ISBN 2-212-11385-4.
8. Siminski, R., "Graph-Based Knowledge Representations for Decision Support Systems". M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 436–444, 2007.
9. Vogel I., "Utilisation pratique du reset partiel : initialisation pour le test intègre de circuits fortement séquentiels", Thèse doctorat, Université de Montpellier. 20 December 2002.
10. Xudong, H., William C. Chu., and Hongji, Y., "A new approaches to verify rule-based systems using Petri nets". Information and Software Technology 45 (2003) 663–669.

Minimizing the makespan on two-machines flowshop scheduling problem with coupled-tasks

Nadjat Meziani[1], Ammar Oulamara [2], and Mourad Boudhar[3]

[1]University of Abderrahmane Mira Bejaia, Algeria
[2]Nancy University, France
[3]USTHB University Algiers, Algeria
ro_nadjet07@yahoo.fr, Ammar.Oulamara@loria.fr and mboudhar@yahoo.fr

Abstract. We consider the problem of the coupled tasks on two machines with the objective to minimizing the makespan such as each job consists of two operations on the first machine separated by the time interval and only one operation on the second machine. We study the complexity of one of the special cases problem of the general problem and we show that it's NP-hard. For its resolution, we suggest heuristics with numerical experiments and we present polynomial sub-problems.

Key words: Coupled task, flow shop, makespan, time lag

1 Introduction

The problem of coupled tasks scheduling was introduced for the first time by Shapiro in [10]. A noted coupled tasks (CT) consists in two different operations which are carried out in the order, separated by a time interval known as a latency or time lag. Each coupled tasks noted by the triplet (a_i, L_i, bi), represent the processing time of the first operation (a_i), latency time 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 latency time, the machine is inactive and another job can be treated.

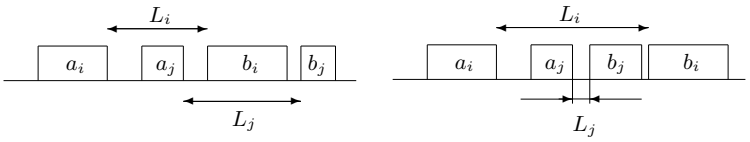


Fig. 1. Examples of interleaving jobs

According to the notation introduced by Graham et al [6], the problem of coupled tasks with one machine with the objective of minimizing C_{max} noted by $1/Coup - Task, a_i, l_i, b_i/C_{max}$ and for two machines noted by $F2/Coup - Task, a_i, l_i, b_i/C_{max}$.

The motivation of a (CT) 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 job and an exact delay is imposed between the execution of each two consecutive operations due to the chemical reactions.
Orman et al studied in [9] the (CT) 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 [3], 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 [5] to resolve a (CT) problem motivated by the time management problems of cyclic production with robots. Other researchers headed to the approximability of these problems. Thus in [1], Ageev and Baburin proposes an 7/4 and 3/2-approximation to solve the problems $1/Coup - Task, a_i = b_i = 1, L_i/C_{max}$ and $F2/Coup - Task, a_i = b_i = 1, L_i/C_{max}$ respectively. For the problems $1/Coup - Task, a_i, L_i, b_i/C_{max}$ and $F2/Coup - Task, a_i, L_i, b_i/C_{max}$, Ageev and Kononov [2] gives several results of approximability and the limits of non-approximability according to the values of a_i and b_i. Few works were realized by adding constraints to the coupled tasks. Blazewicz et al [4] 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 [12], Yu et al proved that the problem in two machines $F2/Coup - Task, a_i = b_i = 1, L_i/C_{max}$ is NP-hard. Simonin et al studied in [11] the (CT) problem in the presence of tasks of treatment.

Our problem consists of scheduling of n coupled tasks in a flow shop with two machines such as each task composed of two operations separated by a time lag on the first machine and of only one operation on the second machine in order to minimize the C_{max} which is noted by $F2/Coup-Task(1), a_i, b_i, L_i, c_i/C_{max}$. Let us note:

- a_i, b_i: processing time of the first and the second operations respectively of the job J_i on the first machine.
- L_i: time separating between the completion time of the first operation and the starting time of the second operation of the same job on the first machine
- c_i: processing time of the job J_i on the second machine.

2 The problem complexity

In this section, we show that the problem $F2/Coup - Task(1), a_i, b_i = L_i = p, c_i/C_{max}$ is NP-hard using the partition decision problem.

Theorem. 1 *The problem $F2/Coup-Task(1), a_i, b_i = L_i = p, c_i/C_{max}$ is NP-hard.*

For proof, we consider the following decision problem known as the partition with the same cardinality (PCI): let us a set $E = \{e_1, e_2, ..., e_{2n}\}$ of integers and an integer B such that $\sum_{i=1}^{2n} e_i = 2B$. Is there partition of E in $E_1 \cup E_2$ such as $\sum_{e_i \in E_1} e_i = \sum_{e_i \in E_2} e_i = B$ and $|E_1| = |E_2| = n$?. From an instance of PCI, we construct an instance for the following problem scheduling $F2C$ composed of $4n + 2$ jobs partitioned into four subsets:

- U-jobs, noted by U_i for $i = \overline{1, 2n}$;
- V-jobs, noted by V_i for $i = \overline{1, n}$;
- W-jobs, noted by W_i for $i = \overline{1, n+1}$;
- T-jobs, containing a single job T_1;

For all the jobs, we take $L_i = b_i = p$ with $i = \overline{1, 4n+2}$ and $p > B$. The values of processing time of the jobs are given in the table 1. Does there exist a schedule

Table 1. Jobs Processing times

Jobs	a_i	c_i
$U-Jobs\ (U_i)_{i=\overline{1,2n}}$	$p - e_i$	e_i
$V-Jobs\ (V_i)_{i=\overline{1,n}}$	$p + 1$	$4p$
$W-Jobs\ (W_i)_{i=\overline{1,n+1}}$	p	0
$T-Jobs\ (T_1)$	$B + p + 1 - n$	$(4n + 1)p - 2B$

with $C_{max} \leq Y$, where $Y = 4(2n + 1)p + 1$?
It is clear that this reduction is polynomial and that the problem $F2C$ belongs to NP. We show that the problem $F2C$ has a solution only if the problem PCI has a solution. For the proof, we will need the following lemmas:

Lemma. 1 *All the jobs are interleaved.*

Lemma. 2 *No job of type U is interleaved with another job of type U.*

Proof. Suppose that the PCI problem has solution, so the instance of the $F2C$ problem has solution S with $C_{max} \geq Y$. In this solution, the jobs are interleaved as shown in table 2.

Table 2. Processing time of interleaved jobs

interleaved jobs	A_i	B_i	l_i
$(V_i, U_i)_{i=\overline{1,n}}$	$4p + 1$	$4p + e_i$	$-p$
$(U_i, W_i)_{i=\overline{1,n}}$	$4p - e_i$	e_i	$-e_i$
(T_1, W_{n+1})	$B + 4p + 1 - n$	$(4n + 1)p - 2B$	$-p$

The optimal schedule of this sequence is given by the Mitten algorithm [8], by starting with the pair (V_i, U_i), $i = \overline{1, n}$, followed by the pair (T_1, W_{n+1}) and terminated by the pairs (U_i, W_i) for $i = \overline{1, n}$. Pairs (V_i, U_i) are sequenced on

the first machine between the dates 0 and $(4p+1)n$ then the pair (T_1, W_{n+1}), between the dates $(4p+1)n$ and $4np+n+3p+B+1-n = 4(n+1)p+B+1$. At the end, pairs (U_i, W_i) are sequenced between $4(n+1)p+B+1$ and $4(2n+1)p+1$. The second machine process the pairs of jobs in the same order obtained on the first machine, furthermore, it is easy to verify that the processing of each pair of jobs on the first machine is completed before the starting on the second machine, especially the pair (T_1, W_{n+1}) that ends at date $4(n+1)p+B+1$ and starts on the second machine at the date $4np+B+1$. This pair is followed by the pairs (U_i, W_i) without idle time on the machines and the scheduling length is $C_{max} = y$. Conversely, suppose that there is solution for $F2C$ problem. Using the results of Lemma 1 and Lemma 2, we show that the PCI problem has solution. For this, we show that there is a schedule of length y by interleaving all the jobs between them, which is shown in Lemma 1. We also prove that no job of U is interleaved with another job of type U, which is proved in Lemma 2.

3 Polynomial cases

In this section, we present the two following polynomial problems $F2/Coup-Task(1), a_i = a, b_i = L_i = p, c_i/C_{max}$ and $F2/Coup-Task(1), a_i = b_i = L_i = p, c_i/C_{max}$ of the studied problem.

3.1 The problem $F2/Coup-Task(1), a_i = a, b_i = L_i = p, c_i/C_{max}$

Algorithm 1: algorithm1

begin
  if $(a > p)$ **then**
    - let p_{1i} and p_{2i} the processing times of jobs on the first and the second machine respectively such as: $p_{1i} = 2p + a$ and $p_{2i} = c_i$.
    - construct a job list L in the decreasing order processing time (LPT) on the second machine and order the jobs on the two machines accord -ing to the list L.
  else
    - construct a job list according to the decreasing order (LPT) of the processing time on the second machine and order the jobs on the two machines in this order by interleaving them.
  end
end

Theorem. 2 *The algorithm 1 provides an optimal solution for the problem* $F2/Coup-Task(1), a_i = a, b_i = L_i = p, c_i/C_{max}$ *in* $O(nlogn)$.

Proof. To prove this theorem, there are two distinguished cases:
Case 1: if $(a > p)$

assume that in a feasible schedule (S) of length C_{max}, there is a job J_j, which is processed before the job J_i such as $p_{2j} < p_{2i}$. Construct another scheduling (S') in which we permute the order of processing jobs J_j and J_i on both machines. So, on the first machine the completion date of processing jobs does not change its value since the processing time of all jobs are the same and on the second machine, the length of scheduling (S') is $C'_{max} \leq C_{max}$ (see [7]).

Case 2: if $(a < p)$

Let S be a feasible schedule where all jobs are interleaved on the first machine. Let be jobs J_l and J_j interleaved with the jobs J_i and J_k respectively, then the order of processing of jobs is as follows: J_l, J_i, J_j and J_k such as $p_{2i} < p_{2j} < p_{2k}$. To permute the order of processing jobs, there are two cases:

1. Permutation of the execution order of two interleaved jobs J_j and J_k :
   let $p_{1j} = p_{1k} = a + 2p$ and $p_{2j} = c_j, p_{2k} = c_k$.
   In the scheduling S, the job J_J is treated before J_K and the completion time of the execution of the job J_K on the second machine is given by:
   $C_{2k} = max\{max\{C_{2i}, C_{1i} + p_{1j}\} + p_{2j}, C_{1i} + p_{1j} + p\} + p_{2k}$;
   $C_{2k} = max\{C_{2i} + p_{2j} + p_{2k}, C_{1i} + p_{1j} + p_{2j} + p_{2k}, C_{1i} + p_{1j} + p + p_{2k}\}$.
   If we interchange the order of processing of these jobs, we get another scheduling S' where J_k is processed before J_j. Then we must show that $C'_{2j} \leq C_{2k}$ such as C'_{2j} is the completion date of the processing of J_j on the second machine in the scheduling S'.
   If we interchange the order of the two jobs J_j and J_k on the first machine, the completion date end date of processing jobs does not change its value on the first machine (because $p_{1j} = p_{1k} = a + 2p$).
   The completion time of the job J_j on the second machine after the permutation is:
   $C'_{2j} = max\{C_{2i} + p_{2k} + p_{2j}, C_{1i} + p_{1k} + p_{2k} + p_{2j}, C_{1i} + p_{1k} + p + p_{2j}\}$
   According to the two last expressions, the values of the two first terms are equal. Similarly, for the two second terms. Concerning the two last terms, $C_{1i} + p_{1k} + p + p_{2j} < C_{1i} + p_{1j} + p + p_{2k}$ under the condition $p_{2j} < p_{2k}$. Then, $C'_{2j} \leq C_{2k}$.
2. Permutation of the order of treatment of two non interleaved jobs J_i and J_j:
   In the schedule S, the job J_i is not interleaved with the job J_j and J_i is processed before J_j. Then the completion time of treatment of J_j on the second machine is given by:
   $C_{2j} = max\{max\{C_{1l} + p, C_{1l} + p_{2l}\} + p_{2i}, C_{1l} + p + p_{1j}\} + p_{2j}$
   $C_{2j} = max\{C_{1l} + p + p_{2i} + p_{2j}, C_{1l} + p_{2l} + p_{2i} + p_{2j}, C_{1l} + p + p_{1j} + p_{2j}\}$
   If we interchange the order of processing of these jobs, we get another schedule S' where J_j is processed before J_i. We must show that $C'_{2i} \leq C_{2j}$ such as C'_{2i} is the completion time of the job J_i on the second machine in the scheduling S'.
   If we interchange the order of the two jobs J_i and J_j on the first machine, the completion time of J_k does not change its value on the first machine (since $p_{1j} = p_{1i} = a + 2p$).
   The completion time of processing of J_i on the second machine after the

permutation is:
$C'_{2i} = max\{C_{1l} + p + p_{2j} + p_{2i}, C_{1l} + p_{2l} + p_{2j} + p_{2i}, C_{1l} + p + p_{1i} + p_{2i}\}$
According to the two last expressions, the values of the two first terms are equal. Similarly, for the two second terms. Concerning the two last terms, $C_{1l} + p + p_{1i} + p_{2i} < C_{1l} + p + p_{1j} + p_{2j}$ under the condition $p_{2i} < p_{2j}$. Then, $C'_{2i} \leq C_{2j}$.

3.2 The problem $F2/Coup - Task(1), a_i = L_i = b_i = p, c_i/C_{max}$

This problem is a sub-problem of the problem $F2/Coup - Task(1), a_i = a, b_i = L_i = p, c_i/C_{max}$ when $a = p$. The following algorithm allows us to solve this problem in $O(nlogn)$.

Algorithm 2: Algorithm 2

begin

- construct a list of jobs according the decreasing order (LPT) of the processing time on the second machine.
- order the jobs on the two machines in this order by interleaving them.

end

Theorem. 3 *The algorithm* 2 *provides an optimal solution for the problem* $F2/Coup - Task(1), a_i = L_i = b_i = p, c_i/C_{max}$ *in* $O(nlogn)$.

Proof. It's the same proof as the case 2 ($a < p$) of the problem $F2/Coup - Task(1), a_i = a, b_i = L_i = p, c_i/C_{max}$ (see proof of the Theorem.2).

4 Lower bounds

In what follows, two lower bounds are proposed LB_1 and LB_2 for the problem $F2/Coup - Task(1), a_i, b_i = L_i = p, c_i/C_{max}$ such as n_1 and n_2 are the number of the interleaved jobs and non-interleaved jobs respectively.

Proposition. 1 $LB_1 = \sum_{i=1}^{\frac{n_1}{2}}(a_i+3p)+\sum_{i=1}^{n_2}(a_i+2p)+\min_{1\leq i\leq n}\{c_i\}$ *is the lower bound of the makespan.*

Proof. $\sum_{i=1}^{\frac{n_1}{2}}(a_i + 3p) + \sum_{i=1}^{n_2}(a_i + 2p)$ is the lower bound of the total completion time of the jobs on the first machine. It is deduced from the problem $1//C_{max}$. Similarly, the processing of jobs on the second machine can start only if at least the processing of one job is completed on the first machine.

Proposition. 2 $LB_2 = \min_{1\leq i\leq n}\{a_i+2p\}+\sum_{i=1}^{n} c_i$ *is the lower bound of the makespan.*

Proof. $\sum_{i=1}^{n} c_i$ is the lower bound of the total completion time of jobs on the second machine. It is deduced from the problem $1//C_{max}$. The processing of the jobs on the second machine can not begin until at least the processing of one job is completed on the first machine.

Consequently, $LB = \max\{LB_1, LB_2\}$ is also a lower bound.

5 Heuristics and numerical experimentations

5.1 Heuristics

In this section, we propose heuristics based on the interleaving of jobs for solving the problem $F2/Coup - Task(1), a_i, L_i = b_i = p, c_i/C_{max}$.

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[7] on the jobs of S.
3. Order the jobs of K according to the LPT rule of a_i.
4. Interleave a job of K with one of S.
5. Interleave the remainder jobs of K or order the remainder jobs 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 jobs of S.
3. Order the jobs of K according to the LPT rule of a_i.
4. Interleave jobs of the set K between them; if there is one job of K not interleaved, interleave it with one job of S.
5. Process jobs of S according to Johnson rule[7].

Heuristic H_3

1. Construct the sets $K = \{J_i/a_i \leq p\}$ and $S = \{J_i/a_i > p\}$.
2. Order the jobs of K and S according the LPT rule of c_i.
3. Interleave a job of K with one of S.
4. If it remains jobs of K, interleave between them and if it remains jobs of S, process them according to their order at the end of the scheduling.

5.2 Perfomance ratio

Theorem. 4 *Let C_{max_H} the length of the schedule obtained with a heuristic $H(i)_{i=\overline{1,3}}$ and $C_{max_{opt}}$ the optimal solution. The performance report is:* $\frac{C_{max_H}}{C_{max_{opt}}} \leq 2$.

Proof. Let $C_{max_1} = \sum_{i=1}^{\frac{n_1}{2}}(a_i + 3p) + \sum_{i=1}^{n_2}(a_i + 2p)$ and $C_{max_2} = \sum_{i=1}^{n} c_i$ the total processing times of jobs on the first and the second machine respectively.

Then we have: $C_{max_H} \leq C_{max_1} + C_{max_2}$

$$\leq \sum_{i=1}^{\frac{n_1}{2}}(a_i + 3p) + \sum_{i=1}^{n_2}(a_i + 2p) + \sum_{i=1}^{n} c_i$$

$$\leq \sum_{i=1}^{\frac{n_1}{2}}(a_i+3p)+\sum_{i=1}^{n_2}(a_i+2p)+\sum_{i=1}^{n} c_i+\min_{1\leq i\leq n}\{c_i\}-\min_{1\leq i\leq n}\{c_i\}$$

$$C_{max_H} \leq LB_1 + \sum_{i=1}^{n} c_i - \min_{1\leq i\leq n}\{c_i\}$$

Let: $x_1 = LB_1$, $x_2 = \sum_{i=1}^{n} c_i$ and $x_3 = \min_{1\leq i\leq n}\{c_i\}$, by replacing in the formula above, we obtain: $C_{max_H} \leq x_1 + x_2 - x_3 \ldots (1)$

Others part, we have : $C_{max_{opt}} \geq x_1 \ldots (2)$ and $C_{max_{opt}} \geq x_2 \ldots (3)$

Two cases are distinguished:

Case 1 : If $x_1 \geq x_2$

from (1) and (2), we obtain: $\frac{C_{max_H}}{C_{max_{opt}}} \leq \frac{x_1 + x_2 - x_3}{x_1}$

$$\Leftrightarrow \frac{C_{max_H}}{C_{max_{opt}}} \leq 1 + \frac{x_2 - x_3}{x_1} \leq 2$$

$$\Leftrightarrow \frac{C_{max_H}}{C_{max_{opt}}} \leq 2$$

Case 2 : If $x_2 > x_1$

from (1) et (3), we obtain: $\frac{C_{max_H}}{C_{max_{opt}}} \leq \frac{x_1 + x_2 - x_3}{x_2}$

$$\Leftrightarrow \frac{C_{max_H}}{C_{max_{opt}}} \leq 1 + \frac{x_1 - x_3}{x_2} \leq 2$$

$$\Leftrightarrow \frac{C_{max_H}}{C_{max_{opt}}} \leq 2$$

5.3 Numerical experimentations

To evaluate the performance of the proposed heuristics, we generated 100 instances uniformly distributed. For each instance, the number of jobs and processing time take their values in $\{10, 20, 50, 100, 250, 500, 1000\}$ 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 each value n of the job, 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.

Table 3. 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	H_1	H_2	H_3	H_1	H_2	H_3
$n=10$	C_{max}	13	25	62	31	1	68	74	0	26
	opt	2	2	42	8	1	21	12	0	1
	$\overline{time}$	0.06	0.03	0.36	0.04	0.03	0.36	0.07	0.03	0.06
	$\overline{dev}$	0.0173	0.0264	0.0145	0.0351	0.1086	0.0272	0.0256	0.1125	0.0656
	$mdev$	0.1256	0.1212	0.1234	0.1427	0.1867	0.1274	0.0594	0.1448	0.1496
$n=20$	C_{max}	23	7	70	38	0	62	82	0	18
	opt	0	0	58	4	0	16	7	0	2
	$\overline{time}$	0.06	0.02	0.5	0.09	0.04	0.6	0.08	0.05	0.13
	$\overline{dev}$	0.0142	0.0325	0.0092	0.0197	0.117	0.023	0.0145	0.0591	0.0679
	$mdev$	0.0789	0.1142	0.0678	0.0524	0.176	0.113	0.0293	0.0766	0.1335
$n=50$	C_{max}	25	0	75	61	0	39	90	0	10
	opt	0	0	62	0	0	9	3	0	0
	$\overline{time}$	0.1	0.06	0.86	0.1	0.04	0.87	0.1	0.06	0.13
	$\overline{dev}$	0.0062	0.0336	0.0028	0.0087	0.1261	0.0222	0.0057	0.0234	0.0729
	$mdev$	0.0275	0.0841	0.0242	0.0223	0.1548	0.0587	0.0116	0.0309	0.1056
$n=100$	C_{max}	25	0	75	70	0	30	97	0	3
	opt	0	0	69	2	0	0	3	0	0
	$\overline{time}$	0.11	0.15	0.15	0.18	0.12	1.37	0.12	0.16	0.24
	$\overline{dev}$	0.0132	0.0328	0.0001	0.0041	0.1285	0.0226	0.0029	0.0118	0.0758
	$mdev$	0.0245	0.0513	0.0003	0.0123	0.1838	0.0442	0.0058	0.0154	0.1035
$n=250$	C_{max}	14	0	86	85	0	15	100	0	0
	opt	0	0	70	1	0	0	1	0	0
	$\overline{time}$	0.24	0.29	0.38	0.48	0.47	0.5	0.53	0.44	0.52
	$\overline{dev}$	0.3241	0.3221	0.0001	0.0127	0.0041	0.0743	0.0013	0.0049	0.0753
	$mdev$	0.5216	0.4153	0.0004	0.0235	0.0063	0.0897	0.0023	0.0061	0.0912
$n=500$	C_{max}	0	0	100	100	0	0	100	0	0
	opt	0	0	74	4	0	0	5	0	0
	$\overline{time}$	0.74	0.75	1.18	0.88	0.93	1.97	1.48	1.61	1.6
	$\overline{dev}$	0.1365	0.0172	0.0002	0.0091	0.1294	0.0245	0.0006	0.0024	0.0752
	$mdev$	0.1591	0.1282	0.0006	0.0028	0.1483	0.0372	0.0012	0.0031	0.0855
$n=1000$	C_{max}	0	0	100	100	0	0	100	0	0
	opt	0	0	75	3	0	0	5	0	0
	$\overline{time}$	2.3	2.36	4.07	2.66	2.98	4.28	5.35	6.07	6.67
	$\overline{dev}$	0.2342	0.0341	0.0003	0.0043	0.1292	0.0231	0.0004	0.0012	0.0752
	$mdev$	0.4567	0.0425	0.0006	0.0125	0.1366	0.0348	0.0005	0.0014	0.0821

From the results reported in the three tables, for the processing time $a_i, c_i \in [1,50], p=10$ (table 3) and $a_i, c_i \in [1,100], p=30$ (table 4), we note that the heuristic $H3$ gives better values of C_{max} which increases by the increasing of the number of jobs. For the same values of a_i, c_i and p, $H3$ provides in many cases the values of C_{max}, which coincides with the values of the lower bound.
By increasing the value of p ($p=30$) and for $a_i, c_i \in [1,50]$ (table 3), we note that $H3$ provides better values of C_{max} for a small number of jobs $(10,20)$ compared to $H1$, but once $n=100$, it's the heuristic $H1$ that gives a good results. For

Table 4. 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	H_1	H_2	H_3	H_1	H_2	H_3
$n = 10$	C_{max}	12	6	82	18	1	81	34	12	54
	opt	0	1	59	3	0	40	5	1	12
	$\overline{time}$	0.11	0.04	0.08	0.02	0	0.09	0.05	0.03	0.07
	$\overline{dev}$	0.0148	0.0228	0.0064	0.0278	0.1046	0.0159	0.0343	0.0981	0.0361
	$mdev$	0.1588	0.0624	0.0071	0.1479	0.2014	0.0785	0.0881	0.0992	0.1082
$n = 20$	C_{max}	10	0	90	17	0	80	53	1	46
	opt	0	0	69	1	0	36	3	0	2
	$\overline{time}$	0.07	0.07	0.05	0.05	0.05	0.09	0.07	0.04	0.06
	$\overline{dev}$	0.0019	0.0591	0.0012	0.0149	0.1084	0.0102	0.0187	0.1297	0.0347
	$mdev$	0.0039	0.1254	0.0258	0.0695	0.1484	0.0781	0.0234	0.1679	0.0915
$n = 50$	C_{max}	3	0	97	29	0	71	67	0	23
	opt	0	0	57	0	0	27	3	0	0
	$\overline{time}$	0.08	0.03	0.14	0.09	0.1	0.07	0.1	0.08	0.09
	$\overline{dev}$	0.0056	0.0592	0.0019	0.0069	0.1265	0.0067	0.0073	0.1036	0.0384
	$mdev$	0.0412	0.1165	0.0487	0.0267	0.1679	0.0467	0.0181	0.1476	0.0754
$n = 100$	C_{max}	0	0	100	35	0	65	91	0	9
	opt	0	0	75	0	0	33	2	0	0
	$\overline{time}$	0.14	0.08	0.15	0.12	0.08	0.05	0.14	0.1	0.2
	$\overline{dev}$	0.0045	0.0596	0.0007	0.0031	0.1284	0.0046	0.0037	0.1053	0.0389
	$mdev$	0.0071	0.0988	0.0009	0.0126	0.1524	0.0276	0.0085	0.1315	0.0601
$n = 250$	C_{max}	0	0	100	40	0	60	98	0	2
	opt	0	0	88	1	0	30	1	0	0
	$\overline{time}$	0.32	0.24	0.33	0.29	0.32	0.33	0.39	0.23	0.4
	$\overline{dev}$	0.0086	0.0596	0.0012	0.0013	0.1341	0.0036	0.0014	0.1081	0.0377
	$mdev$	0.0147	0.0827	0.0035	0.0048	0.1478	0.0176	0.0034	0.1252	0.0542
$n = 500$	C_{max}	0	0	100	44	0	56	100	0	0
	opt	0	0	77	0	0	44	1	0	0
	$\overline{time}$	0.72	0.69	1.02	0.78	0.81	0.9	0.92	0.98	1.04
	$\overline{dev}$	0.0079	0.0595	0.0005	0.0006	0.1356	0.0023	0.0008	0.1082	0.0387
	$mdev$	0.0086	0.0712	0.0007	0.0023	0.1437	0.0123	0.0019	0.1213	0.0509
$n = 1000$	C_{max}	0	0	100	62	0	38	100	0	0
	opt	0	0	80	0	0	34	1	0	0
	$\overline{time}$	2.2	2.32	3.47	2.51	2.55	3.7	3.25	3.53	3.55
	$\overline{dev}$	0.0232	0.0475	0.0001	0.0003	0.1375	0.0021	0.0004	0.1086	0.0387
	$mdev$	0.1563	0.0698	0.0003	0.0012	0.1443	0.0089	0.0009	0.1224	0.0443

$a_i, c_i \in [1, 100]$ and $p = 50$ (table 4), we obtained better results of the values of C_{max} by using the heuristic $H3$. By increasing the number of jobs, the value of C_{max} increases by using $H1$ and becomes better and the optimum is found in several cases by the heuristics $H3$.

For $a_i, c_i \in [1, 100], p = 70$ (table 4), $a_i, c_i \in [1, 50], p = 50$ (table 3) and $a_i, c_i \in [50, 100], p = 100$ (table 5), the heuristic $H1$ gives better results for the C_{max} by increasing the number of jobs and the optimum is attained in some cases by the same heuristic.

For $a_i, c_i \in [50.100], p = 50$ (table 5) and a small number of jobs $(10, 20)$, the number of times where the C_{max} is better, is given by the heuristic $H2$ and by increasing the number of jobs, the heuristic $H3$ gives the best value of C_{max}. The optimum is reached in several cases by using $H3$ and these values increase by increasing the number of jobs. By increasing the value of $p = 70$ and for the same processing time (table 5), we note that the heuristic $H3$ provides better values of C_{max} compared to those obtained with $H1$ and $H2$ and the optimum

Table 5. 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	H_1	H_2	H_3	H_1	H_2	H_3
n=10	C_{max}	0	86	14	5	0	95	82	0	18
	opt	0	0	10	0	0	39	8	0	1
	$\overline{time}$	0.06	0.03	0.07	0.08	0.08	0.11	0.06	0.03	0.05
	$\overline{dev}$	0.0005	0.0228	0.0723	0.0057	0.1318	0.0022	0.0112	0.1144	0.0293
	mdev	0.1348	0.0786	0.0268	0.0456	0.1841	0.0323	0.0257	0.1192	0.07613
n=20	C_{max}	0	75	25	3	0	97	94	0	6
	opt	0	0	13	0	0	30	5	0	0
	$\overline{time}$	0.09	0.04	0.07	0.05	0.03	0.04	0.03	0.06	0.05
	$\overline{dev}$	0.0028	0.0089	0.0007	0.0023	0.1461	0.0007	0.0067	0.0538	0.0323
	mdev	0.0145	0.0302	0.0008	0.0202	0.2173	0.0119	0.0128	0.0619	0.0647
n=50	C_{max}	0	48	52	2	0	98	97	0	3
	opt	0	0	41	0	0	30	2	0	0
	$\overline{time}$	0.12	0.06	0.1	0.07	0.09	0.08	0.09	0.08	0.08
	$\overline{dev}$	0.0023	0.0053	0.0003	0.0009	0.1591	0.0004	0.0026	0.0218	0.0331
	mdev	0.0045	0.0344	0.0005	0.0096	0.2065	0.0063	0.0054	0.0251	0.0485
n=100	C_{max}	0	29	71	1	0	99	100	0	0
	opt	0	0	45	0	0	45	0	0	0
	$\overline{time}$	0.22	0.1	0.12	0.13	0.09	0.08	0.1	0.08	0.14
	$\overline{dev}$	0.0018	0.0038	0.0005	0.0002	0.1613	0.0015	0.0132	0.0118	0.3343
	mdev	0.0025	0.0189	0.0006	0.0033	0.2123	0.0032	0.0027	0.0124	0.0467
n=250	C_{max}	0	20	80	0	0	99	100	0	0
	opt	0	0	62	0	0	44	0	0	0
	$\overline{time}$	0.33	0.31	0.4	0.34	0.25	0.29	0.24	0.35	0.55
	$\overline{dev}$	0.0124	0.0049	0.0002	0.0078	0.1632	0.0026	0.0006	0.0043	0.0341
	mdev	0.0378	0.0142	0.0056	0.0095	0.2123	0.0039	0.0012	0.0049	0.0438
n=500	C_{max}	0	10	90	0	0	100	100	0	0
	opt	0	0	63	0	0	78	0	0	0
	$\overline{time}$	0.92	0.79	1.53	0.7	0.65	0.98	1.41	1.43	1.56
	$\overline{dev}$	0.0075	0.0054	0.0016	0.0005	0.1633	0.0002	0.0003	0.0022	0.0346
	mdev	0.0096	0.0105	0.0034	0.0012	0.1968	0.0012	0.0005	0.0024	0.0394
n=1000	C_{max}	0	2	98	0	0	100	100	0	0
	opt	0	0	19	0	0	81	0	0	0
	$\overline{time}$	2.69	2.8	5.35	2.17	2.3	3.04	4.83	5.27	5.41
	$\overline{dev}$	0.0362	0.0057	0.0006	0.0007	0.1643	0.0013	0.0001	0.0012	0.0343
	mdev	0.0541	0.0088	0.0008	0.0011	0.1873	0.0054	0.0002	0.0014	0.0377

coincides with the lower bound in many cases.
For the average execution times, the heuristic $H3$ requires a higher execution time compared to the other heuristics ones which increases with the increasing of the number of jobs. According to the preceding remarks, we notes that:
the heuristic $H3$ is better in average when the number of jobs having an a_I greater than p is important.
the heuristic $H1$ is better in average when the number of jobs having an a_I less than p is important.

6 Conclusion

In this article, we studied the complexity of flow shop problem with coupled tasks. Our problem consists in a flow shop with two machines with coupled tasks on the first machine such as each job consists in two operations on the first machine separated by a time lag and one operation on the second machine in

order to minimize the makespan. The problem is NP-hard in its general form. We have shown that one of its sub-problems ($F2/Coup-Task(1), a_i, b_i = L_i = p, c_i/C_{max}$) is NP-hard and for its resolution, we proposed heuristics with numerical experiments. We also presented some special polynomial cases for which we have developed algorithms for their resolution. Other sub-problems of the general problem will be the subject of our study later.

References

1. Ageev, A.A., Baburin, A.E.:Approximation algorithms for UET scheduling problems with exact delays. Oper. Res. Let., 35, 533-540 (2007)
2. Ageev, A.A., Kononov, A.V.: Approximation Algorithms for Scheduling Problems with Exact Delays. In WAOA, pp.1-14 (2006)
3. 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)
4. 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)
5. 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)
6. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Disc. Mathe. 5, 287-326 (1979)
7. Johnson, S.M.: Optimal two and three stage production schedules with setup time included. Nav. Res. Logi. Quar. 1, 61-67 (1954)
8. Mitten, L.G.: Sequencing n Jobs on Two Jobs with Arbitrary Time Lags. Manag. Scie. 5(3), 293-298 (1958)
9. Orman, A.J., Potts, C.N.: On the complexity of coupled-Task Scheduling. Disc. App. Math. 72, 141-154 (1997)
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. 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)

Optimisation Multi-critères Pour une Coordination Optimale des Systèmes Intelligents et Distribués

Malika BENDECHACHE[1], A.Kamel TARI[1] et Tahar KECHADI[2]

[1] Université A.Mira de Béjaia
[2] University College Dublin, Ireland

Resumé : Les systèmes distribués existant sont généralement dotés de ressources de traitement très performantes, de très grandes capacités mémoires, et des outils logiciels complexes. Avec les progrès technologiques de ces dernières années, ces systèmes sont devenus de plus en plus performants, abordables, répandus, et leur utilisation ne cesse de s'étendre à d'autres domaines d'application. Par exemple, beaucoup de systèmes intelligents y trouvent refuge, à cause de leur capacité de traitement, de communication, et d'échange. Dans la même optique, nous proposons d'étudier les systèmes intelligents qui sont dynamiques et dotés de petites ressources de calcul, de stockage, et de communications. Ceci est dû à leur faible autonomie de fonctionnement.

Dans ce papier, nous proposons d'étudier ces systèmes dans le domaine du data mining (DM), afin d'utiliser leurs points forts comme technique de résolution des problèmes de DM. Afin d'étudier le comportement de ces systèmes, nous avons modélisé ces systèmes sous forme d'un problème d'optimisation multi-objectifs, pour garantir la stabilité de leur fonctionnement.

Mots-clés : Data mining, Data mining distribué, Système Distribué, Système intelligent.

1 Introduction

Les composants numériques et l'information qu'ils véhiculent sont au cœur de notre civilisation et ne cessent d'accroître dans les décenies à venir. Notre objectif, donc, est d'essayer d'intégrer plus d'intelligence dans ces petits appareils numériques pour faire face à des tâches complexes, telles que la surveillance vidéo, régulation de trafic urbain, etc. La complexité de l'application réside principalement dans les ensembles de données, qui sont de plusieurs types, de très grandes tailles, et distribués dans l'espace et le temps. Leur manipulation nécessite des techniques d'exploration de données complexes pour extraire des connaissances nécessaires à la prise de décision rapide. Les technologies de Data Mining (DM) ont été utilisées dans de nombreux domaines (météorologie [1], finance [2], médecine[3], étude de catastrophes naturelles [4,5], etc.). Toutefois, elles sont complètement nouvelles dans cet environnement de traitement. D'ailleurs, cet environnement présente un nouveau défi qui est ; *comment concevoir des techniques de DM qui peuvent être exécutées par des systèmes à faible*

puissance de traitement, taille de mémoire réduite et aussi de faible bande passante, mais très répandus ?.

Ces contraintes doivent être prises en compte dans la conception et la mise en œuvre des algorithmes de Data Mining distribués qui devraient être rapides et efficaces. En outre, l'accès aux données et l'échange sur un réseau potentiellement très larges doivent être abordés avec des topologies réseaux robustes mais adaptatives.

Dans ce travail, nous proposons d'utiliser les points forts de systèmes distribués pour le traitement de grandes masses de données. Les algorithmes de data mining qui en découlent doivent prendre en compte non seulement des caractéristiques des données (la façon dont les données ont été distribuées, taux d'hétérogénéité, différents types de données, etc.), mais aussi la disponibilité des ressources et l'état du système globale. Dans la section qui suit nous donnerons un aperçu général sur le data mining distribué et discuterons les limites des techniques traditionnelles. Dans la section 3 nous commençons par introduire les systèmes distribués ainsi que leurs caractéristiques et challenges. Nous concluons cette section par une discussion assez brève sur les environnements de calcul distribués et leurs apports dans le domaine de data mining. Nous définissons notre système ainsi que ce modèle de fonctionnement dans la Section 4. Ce système est différent des systèmes existants, nous proposons un formalisme mathématique sous forme d'un problème multi objectifs sous un ensemble de contraintes. Notre but est d'étudier ce système afin d'extraire ces caractéristiques tout en tenant en compte des algorithmes de data mining distribués. Nous concluons et donnons les grandes lignes de notre travail dans le future proche dans la section 5.

2 Data Mining distribué

Bien que des quantités massives de données sont collectées et stockées non seulement dans des domaines scientifiques mais aussi dans les secteurs industriels et économiques, la gestion efficace de l'information utile de ces données devient un défi scientifique et un besoin urgent. Cela est encore plus critique lorsque les données recueillies sont situées sur des sites différents et appartenant à des organisations différentes [6].

Ceci a mené au développement des techniques de traitement distribués des données pour faire face aux énormes quantités de données hétérogènes et sont réparties sur un grand nombre de nœuds.

Les techniques de data mining distribués existantes éffectuent une analyse partielle des données sur l'ensemble des sites, puis génèrent des modèles globaux en effectuant une agrégation des résultats locaux. Ces deux étapes ne sont pas indépendantes puisque les approches naïves de l'analyse locale peuvent produire les modèles incorrects et ambigus. Afin d'exploiter les connaissances extraites sur les différents sites, ces techniques doivent avoir une vue de la connaissance qui non seulement facilite leur intégration, mais réduit également l'effet négatif des résultats locaux sur les modèles globaux. En bref, une gestion effi-

cace des connaissances distribuées est l'un des principaux facteurs influant sur les résultats de ces techniques [7,8,9,10].

En outre, les données qui seront recueillies à différents endroits en utilisant différentes techniques peuvent avoir différents formats et fonctions. Les techniques de data mining centralisées traditionnelles ne tiennent pas compte de toutes les applications liées aux données telles que l'évolutivité à la fois dans le temps de réponse et la précision des solutions, la distribution et l'hétérogénéité [8,11].

Quelques approches distribuées de data mining sont basées sur l'apprentissage d'ensemble, qui utilise différentes techniques pour combiner les résultats de chaque site [12]. parmi les plus citées dans la littérature : le vote majoritaire, le vote pondéré, et stacking [13,14].

Récemment, des approches distribuées très intéressantes ont été présentées. La première phase consiste à générer les connaissances locales basées sur des données locales. La seconde phase intègre les connaissances locales pour générer des résultats globaux ou des décisions. Certaines approches sont bien adaptées pour être exécutées sur les plates-formes distribuées, par exemple les algorithmes incrémentaux pour découvrir des modèles spatio-temporelle en décomposant l'espace de recherche en une structure hiérarchique, s'adressant à son application à la multi-granulaire des données spatiales sont très facile à optimiser sur la topologie du système hiérarchique distribué.

A partir de la littérature sur le data mining distribué, deux catégories de techniques sont utilisées : techniques parallèlles qui font souvent appel à des machines dédiées et des outils de communication entre les processus parallèles, et les techniques basées sur l'agrégation, qui procèdent avec une vision purement distribuée, soit sur les données, sur les modèles de base ou bien sur les plateformes d'execution. [7,8,9].

3 Environnements distribués pour les Données

Pendant de nombreuses années, les systèmes centralisés ont été l'approche standard pour le partage du contenu numérique, les processus de décision, et même de calcul. Toutefois, le matériel, logiciel, et les applications sont devenus plus complexes, par conséquent le traitement distribué est devenu nécessaire.

La plupart de ces systèmes sont composés de nœuds autonomes ayant une certaine capacité pour réaliser des tâches locales, ainsi que des décisions globales en s'appuyant sur l'information locale et les informations échangées avec son voisinage. La notion de nœuds autonomes ou d'indépendance est motivée par les points suivants [15] :

1. Fiabilité : le système fonctionne toujours même si un des noeuds du systèmes tombe en panne.
2. Chaque nœud peut fonctionner tout seul, sans coopérer avec les autres nœuds du système.

3. Chaque nœud peut gérer ses propres ressources, indépendamment de l'ensemble du système. C'est évidemment possible, uniquement si les nœuds sont équipés de renseignements sur la façon de s'occuper de leurs propres ressources et demander de l'information sur le reste du système lorsque cela est nécessaire.

Ces systèmes sont généralement composés d'un ensemble de nœuds puissants, tels que les grilles de calcul, et bien d'autres systèmes distribués existants. Ces nœuds ont une très grande puissance de traitement, l'espace mémoire très important, et les outils logiciels complexes. En d'autres termes, chaque nœud peut faire tout ce que l'ensemble du système peut faire à une échelle réduite.

Le système que nous proposons d'étudier est complètement différent du système traditionnel décrit ci-dessus. Contrairement aux systèmes traditionnels, ce système a quelques particularités qui sont :

1. Ressources à capacité réduite : faible puissance de traitement et de communication et faible capacité mémoire.
2. Un logiciel dédié : Ceci est basé sur la tâche principale que le nœud est attribué à faire.
3. Dynamique : chaque nœud peut quitter le système et le rejoindre à tout moment. La topologie du système est dynamique et chaque nœud doit mettre à jour régulièrement ses voisins, en raison du changement qui peut se produire dans le système ; (nouveaux arrivants, les mouvements de nœuds, nœuds sortants).

Avec plus d'intelligence, chaque nœud devient autonome avec la capacité de coordonner et de coopérer avec d'autres nœuds dans le but d'atteindre certains objectifs définis par l'application donnée. Ces systèmes sont de plus en plus populaires en raison des progrès dans les technologies de l'information et de communication. Pour cela, au cours des deux dernières décennies, ces systèmes ont cessé d'être le sujet marginal de l'intelligence artificielle [15]. Ils sont plutôt un sujet de recherche majeur dans les systèmes distribués et d'intelligence artificielle. Ces systèmes sont caractérisés par :

1. Système de contrôle des décisions (par exemple pour la planification, l'ordonnancement, le routage, l'exécution des tâches, etc.) sont déterminés par plus d'un nœud.
2. Les nœuds interagissent en collaboration et avec souplesse pour prendre une décision globale et définitive.
3. Aucun nœud n'a accès à toutes les informations nécessaires pour prendre une décision.
4. Les nœuds sont reliés entre aux par un réseau de communication.

4 Méthodologie

Dans ce papier nous nous intéressons à la modélisation d'un système distribué intelligent et étudier son comportement. Ce système possède à la fois les caractéristiques des systèmes Pair-à-Pair (P2P), comme la volatilité de ses noeuds

et recherche efficace des ressources, les caractéristiques des systèmes distribués classiques, comme l'hétérogénéité, l'autonomie, calcul distribué et parallèle, et enfin les caractéristiques des systèmes embarqués, comme faible puissance de calcul et de stockage.

On considère que le système est composé de N nœuds, $\{P_1, ..., P_n\}$. Chaque nœud à pour tâche de contribuer au système global pour résoudre un problème dont il est impliqué directement ou indirectement ou tout simplement répondre à une requête d'un de ses voisins. Pour étudier le comportement de ce système sur des applications de data mining, nous introduisons les quantités suivantes :

- Une fonction utilitaire, $U_i(t)$, pour évaluer la qualité des ressources d'un nœud P_i.
- Une fonction contribution, $D_i(t)$, d'un nœud P_i pour évaluer le partage de ses ressources.
- Une fonction consommation, $C_i(t)$, d'un nœud P_i pour mesurer ses avantages à faire parti du système.

Chaque nœud du système est donc caractérisé par sa contribution, sa consommation, et son voisinage immédiat. Le coût total d'un nœud peut être défini par sa participation dans le système ; $r_i D_i(\mathrm{t})$, où r_i est le nombre d'unités de resources partagées. Une ressource peut être un espace disque ou mémoire du nœud, temps d'utilisation de ses unités de calcul, poids de ses voisins immédiats, ses connaissances locales, ou toutes ces caractéristiques. Soit $b_i(\mathrm{t})$ le bénéfice total que le nœud P_i peut profiter du système. b_i est un facteur très important et il peut être, par exemple, utilisé pour décider s'il est intéressant de faire parti du système.

Le système global peut être modélisé comme un problème d'optimisation multi-critères [16]. L'optimum du système peut être vu comme une solution qui peut garantir la stabilité de fonctionnement du système. En d'autres termes, on cherche à optimiser l'utilité, la contribution, et le bénéfice d'un noeud du système, à la fois.

Les fonctions à optimiser sont :

$$D(t) = \sum_{i=1}^{N} r_i d_i(t) \quad C(t) = \sum_{i=1}^{N} c_i b_i(t) \quad U(t) = \sum_{i=1}^{N} u_i(t) \tag{1}$$

Il s'agit d'optimiser simultanément trois objectifs $D(t), B(t)$ et $U(t)$, sous un ensemble de contraintes imposées sur les resources ainsi que le fonctionnement du système. $d_i(t)$ est une fonction qui varie par rapport au temps et qui représente la contribution du noeud P_i au système à l'instant t. Par exemple, à chaque fois qu'un nœud P_i partage une unité d'une ressource r_k, on lui associe un coût c_{ik}. Donc, la contribution totale du nœud P_i est la somme de toutes les unités de ses ressources utilisées par les autres membres du système. Si on considère qu'une unité d'une ressource est toujours la même quelque soit la ressource alors le coût associé est :

$$C_i(t) = \sum_{t_0 \to t} \sum_{k=1}^{k=M} c_{ik}(t) \tag{2}$$

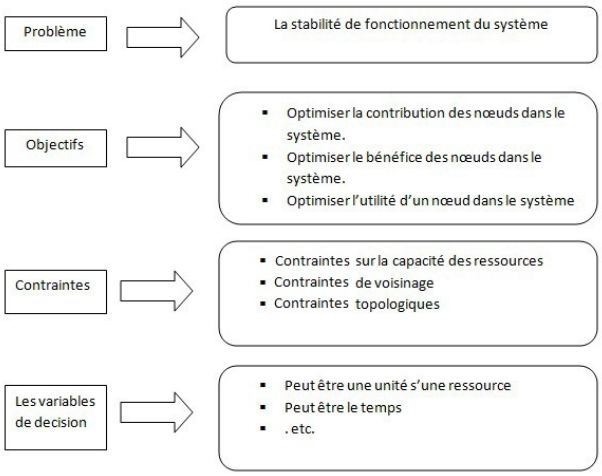


Figure 1. modélisation du système sous forme d'un POM.

Où M est le nombre d'unités de ressources que le nœud P_i partage avec les autres membres du système. Le coût total du nœud P_i pour sa participation dans le système est $C_i D_i$. Si on suppose que la contribution minimale d'un nœud est D_0, on peut définir la contribution du nœud P_i comme suit :

$$d_i(t) = \frac{D_i(t)}{D_0(t)} \tag{3}$$

A chaque fois qu'un nœud contribue par ses ressources au système c'est tous les autres nœuds qui en bénéficient avec un degré plus ou moins diffèrent. Par conséquent on peut représenter ce bénéfice en utilisant une matrice B de taille $(N \times N)$, où B_{ij} représente la contribution de P_j pour P_i, mesurée en unité de resources. Par exemple, si P_i n'est pas interessé par la contribution de P_j, alors $B_{ij} = 0$. On suppose aussi que la contribution du nœud à lui même est aussi nulle ; $B_{ii} = 0$. Dans ce qui suit, on va aussi définir quelques paramètres qui correspondent à B_{ij}.

$$b_{ij}(t) = B_{ij}/c_i(t), \quad b_i(t) = \sum_j b_{ij}(t), \quad \bar{b}(t) = \frac{1}{N}\sum_i b_i(t) \tag{4}$$

où $b_i(t)$, est le bénéfice total que P_i peut profiter du système à l'instant t. On peut définir la fonction utilité, en ce basant sur les définitions et les notations données plus haut. L'utilité qu'un noeud P_i peut tirer profit en rejoignant le système est défini comme étant le bénéfice qu'un noeud P_i peut profiter du système, en soustrayant son coût de participation.

$$U_i = pr_{ij}(D_i) \sum_j B_{ij} D_j - c_i D_i, \qquad B_{ii} = 0 \tag{5}$$

Où $pr_{ij}(D_i)$ est la probabilité que le nœud P_j accepte une demande de ressource du nœud P_i. Si on divise l'équation 5 par le terme $c_i D_0$, on déduit une nouvelle équation pour la fonction utilité comme suit :

$$u_i(t) = \frac{U_i}{c_i(t) D_0} \tag{6}$$

On peut réécrire l'equation 5 de la façon suivante :

$$u_i(t) = -d_i(t) + pr_{ij}(d_i) \sum_j b_{ij} d_j(t), \quad b_{ii} = 0 \tag{7}$$

4.1 Contraintes

1. Contrainte sur la capacité des ressources :
   Les ressources sont caracterisées par une capacité limitée qu'on peut définir comme étant une fonction $K(t)$ qui peut être différente d'un noeud à un autre, sachant que : $K_{r_i}(t) \leq \beta_i$ tel que β_i est un paramètre défini par le système et qui représente la capacité maximale qu'un noeud peut avoir dans le système.
2. Contraintes de voisinage :
   Le voisinage d'un nœud peut être défini comme étant une fonction $\varphi(vi)$, qui varie selon le critère de voisinage choisi ou selon le problème traité, comme par exemple :
   (a) *v_i dépend de la distance entre les nœuds :*
   Les voisins d'un nœud dans le système sont les nœuds les plus proches de lui. Soit P_i un nœud du système, le voisinage de P_i est défini comme suit :

   $$v(P_i) = \{P_j \in V / e = (P_i, P_j) \in E \quad et \quad \alpha(P_i, P_j) = m_d\} \tag{8}$$

   Où E est l'ensemble des arcs du système, $\alpha(P_i, P_j)$ la distance entre les nœuds P_i et P_j, et m_d est la distance maximale entre deux voisins.

   (b) *v_i dépend des ressources d'un noeud :*
   Du point de vue de data mining, le voisinage d'un nœud peut être défini par rapport à la quantité d'information (ressources qu'un nœud possède). En d'autres termes, les voisins d'un nœud sont les nœuds qui ont plus de ressources dont il est intéressé. Nous nous intéressons dans ce qui suit à ce type de voisinage.

   $$Y_{ik}(t) = \begin{cases} 1 & \text{si } P_i \text{ participe avec une ressource } k \\ 0 & \text{sinon} \end{cases} \tag{9}$$

   Nous définissons deux types de voisinage :

Voisinage Gauche : P_j est un voisin gauche de P_i si P_j envoie de l'information à P_i. Autrement dit :

$$\{\sum r_j Y_{jk}(t) \geq r_{hi} \quad ; \quad r_{hi} = r_{app} - r_i\} \tag{10}$$

Voisinage Droit : P_d est un voisin droit de P_i si P_d reçoit de l'information de P_i. Autrement dit :

$$\{\sum r_i Z_i Y_{ik}(t) \geq r_{hd} \quad ; \quad r_{hd} = r_{app} - r_d\} \tag{11}$$

Où r_i est le nombre de ressources du noeud P_i, r_d est le nombre de ressources du noeuds P_d, r_{app} est le nombre de ressources requises par l'application. r_{hi} est le nombre de ressources que le noeud P_i a besoin pour completer sa tache, r_{hd} est le nombre de ressources que le noeud P_d a besoin pour completer sa tache.

(c) *t_i dépend des ressources du système :*
Le voisinage d'un nœud peut être défini par rapport à la disponibilité des ressources du système. Le système utilise un ensemble de ressource $R = \{r_1, \cdots, r_M\}$, chaque ressource est caractérisée par un taux de disponibilité t_m. Le voisinage d'un nœud est défini par les nœuds qui ont le taux de disponibilité t_m le plus élevé.

3. Contraintes Topologiques :
La topologie du système est dynamique, c'est à dire les noeuds peuvent joindre et quitter le système à tout moment. La topologie est choisie de manière que tous les noeuds actifs (présent dans le système) doivent être joignable de n'importe quel autre noeud. Soit $Z_i(t)$ un autre symbole de Kroneker qui prend la valeur 1 si le nœud P_i est actif à l'instant t, 0 sinon.

$$Z_i(t) = \begin{cases} 1 & \text{si le nœud } P_i \text{ fait parti du système} \\ 0 & \text{sinon} \end{cases} \tag{12}$$

4.2 Le modèle Proposé

Soit $G = (V, E, W)$ un graphe pondéré non orienté, où V est l'ensemble de nœuds, E l'ensembles des arcs du graphe et W l'ensembles des poids attribués aux arcs du graphe. Soient $n = |V|$ le nombre total de noeuds et $m = |E|$ est le nombre total des arcs. Nous rappelons que $c_{i_k} \epsilon C$ est le coût que P_i partage une ressource k, μ_i est la capacité totale des ressources du noeud P_i. μ_{ij} est la capacité des ressources du noeud P_i qu'il partage avec le noeud P_j et θ_i le degré du noeud P_i. Avant de formuler le modèle proposé nous avons besoin de définir quelques variables de décision. Ces variables de decision sont booléennes.

Le problème que nous voulons résoudre est de pouvoir exécuter des algorithmes de data mining distribués sur une plateforme de calcul distribuée tout en maximisant les calculs locaux dans chaque nœud et minimisant les échanges entre les nœuds actifs. Les algorithmes de data mining distribués sont généralement constitués de plusieurs phases : phase d'accès aux données, phases

de prétraitement, phase de sélection de dimensions, phase de calcul de modèles locaux, phase d'aggregation des modèles locaux, phase d'évaluation [17]. Pour que leur exécution soit efficace les méthodes traditionnelles ne peuvent plus répondre aux exigences, leurs tailles excessivement grandes, l'hétérogéneité, synchronisation des phases de traitement et de communications, choix des données à échanger, etc. Ces algorithmes éxigent une exécution sur des plateformes où les calculs et les échanges de données sont effectués de manière intelligente [18].

Le système que nous proposons doit prendre en compte non seulement les ressources du système et leurs états dans le temps mais aussi les contraintes algorithmique et d'exécution des applications de data mining. Plus précisément, l'exécution des ces applications dépend de la disponibilité des ressources dans le temps. Par conséquent la quantité de données à échanger et à traiter dépendent de la dynamicité de cette plateforme intelligente. Dans ce papier nous proposons un formalisme mathématique de cette plateforme sous forme d'un système d'optimisation multi-critère. Sa résoludion constitue une exécution d'une tâche de data mining. Nous formulons le problème comme suit :

$$\max_{S \subset P} \sum_{P_i \in S} \sum_{k=1}^{M_i} Y_{ik}(t) \sum_{j=1}^{N} X_{ij} Z_i(t) d_i(t) \tag{13}$$

$$\min_{S \subset P} \sum_{P_i \in S} \sum_{k=1}^{M_i} c_{ik} \sum_{j=1}^{N} X_{ij} Z_i(t) b_i(t) \tag{14}$$

$$\max_{S \subset P} \sum_{P_i \in S} \sum_{j=1}^{N} X_{ij} Z_i(t) u_i(t) \tag{15}$$

Contraintes :

$$\sum_{(i,j) \in E} X_{ij} = n. \quad X_{ij} \in \{0,1\}, \quad Z_i(t) \in \{0,1\}, \quad Y_{ik}(t) \in \{0,1\}. \tag{16}$$

– `Contraintes de capacité:` Chaque noeud ne peut pas partager plus de ressources qu'il en possède. La somme des capacités des ressources partagées d'un noeud P_i doit être inférieure à la capacité maximale du même noeud P_i, autrement dit

$$\sum_{j=1}^{N} Y_{ik} Z_i(t) \mu_{ij} \leq \mu_i \tag{17}$$

– `Contraintes de voisinage:` Le nombre de voisins dépend de la fonction de distance entre les nœuds et de certains paramètres fixés par l'utilisateur ou même le système. Ces paramètres sont aussi appelés *seuils*, ils sont initialisés de façon à limiter le nombre de voisins qu'un nœud peut en avoir : $\sum_{(i,j) \in E} X_{ij} \leq \rho_i$. ρ_i est le nombre maximum de voisins.

Sous les contraintes imposées sur les resources ainsi que le fonctionnement du système. Pour résoudre ce problème on fera appel aux techniques d'heuristiques modernes telles que la théorie des jeux, les algorithmes génétiques, et les réseaux de neurones, etc.

5 Conclusion

Notre travail consiste à définir et mettre en œuvre un nouveau système distribué intelligent qui pourra prendre en compte l'aspect algorithmique ainsi que l'exécution des algorithmes de data mining distribués. Dans ce papier nous avons mis l'accent sur une étape très importante dans notre projet qui est la modélisation du système.

Nous avons défini notre système sous forme d'un problème d'optimisation multicritère afin d'optimiser plusieurs objectifs sous un ensemble de contraintes liées à la topologie ainsi qu'aux ressources, dans le but d'atteindre la stabilité du fonctionnement du système.

Notre prochaine étape consiste à inclure les algorithmes de data mining distribués et la manière dont ils seront exécutés. Le système globale sera simulé sur une architecture distribuée comme le Grid, cluster ou le cloud. L'évaluation se basera sur deux critères principaux qui sont 1) la stabilité et l'efficacité du système d'un coté et l'analyse de l'algorithme de data mining distribué qui en découle et la qualité de ses résultats sur des entrepôts de données pris comme benchmarks. En outre, comme le modèle d'optimisation est un problème complexe et NP-complet, nous allons étudier quelques méthodes de résolution à base d'heuristiques comme les réseaux de neurones, la théorie des jeux, etc.

Références

1. I.G. Olaizola, N. Aginako, and M. Labayen. Image analysis platform for data management in the meteorological domain. In *4th International Workshop on Semantic Media Adaptation and Personalization (SMAP 09)*, pages 89 – 94 89–94, San Sebastian, Spain, December 14-15 2009.
2. B. Kovalerchuk and E. Vityaev. *Data Mining and Knowledge Discovery Handbook*, chapter Data Mining for Financial Applications. Springer US, (O. Maimon and L. Rokach, Eds), 2005.
3. F.S. Khan, R.M. Anwer, O. Torgersson, and G. Falkman. Data mining in oral medicine using decision trees. *World Academy of Science, Engineering and Technology*, 37, 2008.
4. I. Aydin, M. Karakose, and E. Akin. The prediction algorithm based on fuzzy logic using time series data mining method. *World Academy of Science, Engineering and Technology*, 51, 2009.
5. P. Compieta, S. Di Martino, M. Bertolotto, and M-T. Kechadi. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages and Computing*, 18(3) :255–279, 2007.

6. J. Han and M. Kamber. *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publisher, 2nd edition, 2006.
7. L. Aouad, N.A. Le-Khac, and M-T. Kechadi. Lightweight clustering technique for distributed data mining applications. *LNCS on advances in data mining – theoretical aspects and applications*, 4597 :120–134, 2007.
8. L. Aouad, N-A. Le-Khac, and M-T. Kechadi. Grid-based approaches for distributed data mining applications. *Journal of Algorithms and Computational Technology, Multi-Science Publishing*, 2009.
9. L. Aouad, N-A. Le-Khac, and M-T. Kechadi. Performance study of a distributed apriori-like frequent itemsets mining technique. *Journal of Knowledge and Information Systems*, 2009.
10. M. Whelan, N-A. Le Khac, and M-T. Kechadi. Performance evaluation of a density-based clustering method for reducing very large spatio-temporal dataset. In *in Proc. of International Conference on Information and Knowledge Engineering*, Las Vegas, Nevada, USA., July 18-21 2011.
11. M. Bertolotto, S. Di Martino, F. Ferrucci, and M-T. Kechadi. Towards a framework for mining and analysing spatio-temporal datasets. *International Journal of Geographical Information Science*, 21(8) :895–906, 2007.
12. N-A. Le-Khac, L. Aouad, and M-T. Kechadi. Knowledge map layer for distributed data mining. *Journal of ISAST Transactions on Intelligent Systems*, 1(1), 2008.
13. P.K. Chan and S.J. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *12th International Conference on Machine Learning*, pages 90–98, 1995.
14. C.R. Reeves. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc. New York, NY, USA, 1993.
15. D. Soshnikov. An approach for creating distributed intelligent systems. In *Workshop on Computer Science and Information Technologies*, Moscow, Russia, 1999.
16. M. Gen, R. Cheng, and L. Lin. *Network Models and Optimization : Multiobjective Genetic Algorithm Approach*. Springer, 2008.
17. Y. Kodratoff. Technical and scientific issues of kdd (or : is kdd a science ?). In *LNAI on Algorithmic Learning Theory, K. Jantke, and T. Shinohara, T. and Zeugmann (Eds.)*, volume 997, pages 261–265. Springer Verlag, 1997.
18. A. Abdellah Bedrouni, M. Mittu, A. Boukhtouta, and J. Berger. *Distributed Intelligent Systems : A Coordination Perspective*. Springer, 2009.

Bases de données et Systèmes d'Information

Managing Dynamic Protocol Substitution in Web Services Environment

Ali Khebizi[1], Hassina Seridi-Bouchelaghem[2], Imed Chemakhi[3], and Hychem Bekakria[3]

[1] LabStic Laboratory, 08 May 45 University, Guelma -Algeria-
ali.khebizi@gmail.com
[2] LABGED Laboratory, University Badji Mokhtar Annaba, Po-Box 12, 23000, Algeria seridi@labged.net
[3] Computer science Institute, 08 May 45 University, Guelma -Algeria-
Chemakhi.imed@gmail.com,hychem.bekakria@gmail.com

Abstract. As Web services become the dominant technology for integrating distributed information systems, enterprises are more interested by these environments for developing and deploying their applications. However, enterprises socio-economic environments are more and more subject to changes which impact directly business processes published as Web services. In parallel, if at change time some instances are running, the business process evolution will impact the equivalence and substitution classes of the actual service. In this paper, we present an equivalence and substitution analysis in dynamic protocol evolution context. We suggest an approach to identify residual services that can substitute a modified one, where ongoing instances are pending. Our analysis is based on protocol schema matching and on real execution traces. The proposed approach has been implemented in a software tool which is a very useful change management tool for decision support that provide some useful functionalities for protocol managers.

Keywords: Service protocol, Protocol equivalence, Protocol substitution, Dynamic evolution, Execution path, Execution trace.

1 Introduction

Web services are the new generation of distributed software components. They generate a lot of enthusiasm among different socio-economic operators's which favourite these environments to deploy applications at a large scale. In Web services environments, standardization is a key concept and actors uses standards based XML (WSDL [1], UDDI[2] and SOAP [3]) to publish, discover, invoke and compose distributed software. Consequently, intra and inter enterprises applications integration is more flexible, easy and transparent for software developers. Moreover, integration process is accelerated among internet stakeholders, resulting in greater flexibility of exchanges and increased fluidity of transactions through the Web.

In Web services technology, two elements are fundamental for providing a high interactivity level between service providers and service requesters. The first one is service interface, described via the standard WSDL that allow specifying the services input/output and available operations. The second element is service protocol (**Business Protocol**), which describes the provider's business process logic. A Business process consists of a group of business activities undertaken by one or more organizations in pursuit of some particular goal [5]. For example, booking flight tickets, managing a bank account, payroll and B2B transactions. In addition, Business process specifies the service external behaviour by providing constraints on operations order, temporal constraints [6] and transactional constraints [7]. Those descriptors are fundamental in order to promote correct conversations between providers and customers; as service operations can't be invoked in any way. However, if a service protocol is published in the Web (its interface and its protocol), at a moment during its life cycle, it can be invoked by some clients that are currently conducting operations to meet their specific needs. Furthermore, in large public applications (e-commerce, e-government, electronic library, ...), thousand or hundreds of clients are invoking the same service at the same time and every one has reached a particular execution level. In parallel, as enterprises are open systems, changes are permanent and inevitable. Consequently, business processes may evolve to adapt to environment changes that affect real word. In this context, related service protocols must be updated, otherwise services execution may produce incoherences and inconsistency when they are invoked. This context is called **dynamic protocol evolution.**
In dynamic protocol evolution, service evolution is expressed through the creation and decommission of its different versions during its lifetime [11]. It's possible that the evolved service may not be able to satisfy initial customer requirements, as its description has changed but the customer needs are stable. Furthermore, some services may fails and clients must find new services that can replace actual one. In this situation, it is crucial to find others services that satisfy both initial and new customer needs in order to replace the actual one.
Services substitution analysis deal with checking if two services satisfy the same functionalities; if they support the same conversation messages [5]. This concept is very useful in case of service failure, in order to search an other one to replace it. In some cases, this analysis can serve to search and locate a new service with the same functionalities but with a higher quality of service (Qos). It can also be used to test whether a new proposed version, that expresses evolution or maintenance requirements, is yet, equivalent to an obsolete one and for finding new services that can support conversations required by standards like ebXML [9], xCBL [10] and Rosetanet [8]. Service protocol update induces challenges for filtering which services that were already compared to the old version and identified as an alternative class, remain equivalents or can replace the evolved service. The major constraint is related to active instances that have already executed some operations based on the old version. In this case, we must deal with historical executions and we must take into account past executions in substitution analysis process.

In this paper we are interested to dynamic protocol evolution and we focus on change impact analysis on service protocol substitution and equivalence. A set of formal methods are exposed to check if new service version, can be yet substituted by the hole class, or partially subclass set of services that were discovered corresponding to the obsolete version.
The remainder of the paper is structured as follow. We start by describing the problem and exposing our motivations, in section 2. In section 3, we propose our formal approach and algorithms for managing substitution aspects in dynamic protocol evolution context. Section 4, describes system architecture and software tool implementation. We expose related works in section 5 and conclude with a summary and directions for future works in section 6.

2 Problem and Motivations

Every organisation (enterprises, administrations, banks, ...etc.) is an open system which is, eventually, impacted by environment changes induced by law updates, enterprises merging, procedure improvements and so on. In order to survive, organizations must adapt there business processes to these changes. Today's organizations information systems reflect real business processes and are exposed on the Web as services (interfaces and protocols). Consequently, every business process changes induce, immediately, these two descriptions update. The challenge in dynamic protocol evolution context is to identify, among the set of already identified class substitution services, the subset of those that can, yet, replace an actual service after its specification changes, with respect to past interactions.
Addressing service protocols substitution analysis, after protocol evolution, responds to the following motivations:

1. Ensuring service execution continuation for active instances by providing service reserves that can be used alternatively.
2. Ensuring correct interactions between customers and providers by specifying the new service substitution class in order to avoid conversation inconsistency between provider and customer protocols.
3. In dynamic environments, like Web services, transactions are long duration and resources consumer. After protocol changes, it's inconceivable to restart execution from scratch because loss of work is catastrophic for customers who had spend much time running applications.
4. In real time systems and critical applications (aeronautics, e-commerce, medical systems, control systems, manufactures,...), brutal service stop is catastrophic for organizations and can cause damage to property and human lives. It is imperative, in these systems to treat with precision and accuracy services that can substitute an evolved or failed one.
5. In large public applications (e-government, e-learning, e-commerce...), a large number of active instances are pending at a given time. Consequently, manual management of these instances is cumbersome and an automatic support is required to ensure that only pertinent services are proposed for

substitution process. By automatic monitoring of change management, we provide specification of service protocols for eventual substitution in a transparent manner.

The main issue is to manage protocol substitution with respect to historical traces. Starting a new search query for locating new services, based on the new version, is expensive and in addition returned services can be inconsistent with the old version and does not comply with past executions .
To illustrate our motivations, we present in **Fig.1** a real world scenario.

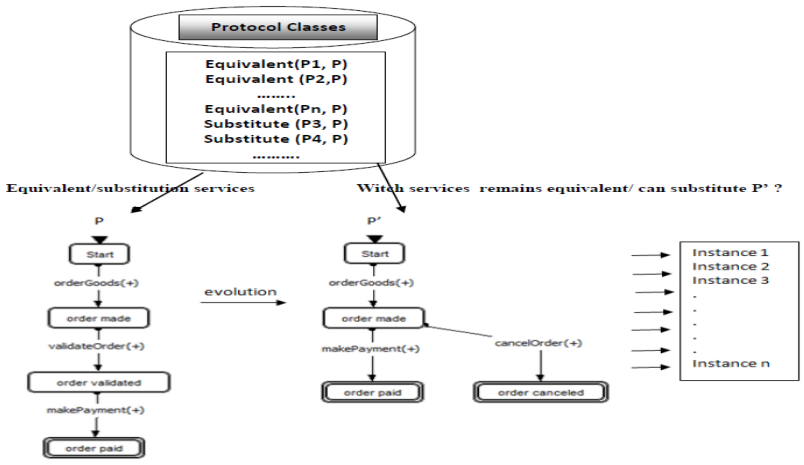


Fig. 1. Which services can substitute P' after evolution ?

In this scenario, service protocol P have some equivalent services $(P_1, P_2, \ldots, P_n)$ and other services $(P_3, P_4, \ldots P_m)$ can substitute it in the cases of needs. However, service P has evolved for different reasons to a new version P'. Consider for example that evolution operations added a new message *cancelOrder* and removed the message *Order validated*. At evolution time, active instances (instance 1, instance 2,...), are running and have reached a particular execution level (some have just started, others instances are performing operations *Ordermade* or *OrdreValidated*. In order to be able to substitute service P, in case of problems, protocol manager want to know: **Which protocols remain in conformance with the new protocol specification and can replace it** ?
To answer this question, we propose a formal approach for managing protocols substitution in dynamic evolution context. Our analysis is based on two complementary aspects: Protocol schema matching and historical execution traces.

3 Analysing Substitution in Dynamic Protocol evolution

One of the most challenging issues in dynamic protocol evolution context is to find potential protocols for substitution, where instances are running according to old protocol. To address this analysis, we introduce three fundamental concepts: **service protocol model, execution path and execution trace**.

- **Service protocol**: We use finite state machine to represent service protocols. In this model states represent different phases that a service may go through during its interaction with a requester. Transitions are triggered by messages sent by the requester to the provider or vice versa [4],[5]. A message corresponds to operation invocation or to its reply, as shown in **Fig 1**. A finite state machine is described by the tuple: $P = (S, s_0, F, M, R)$, consisting of:
  - S : A finite set of states.
  - $s_0 \in S$: is the protocol initial state.
  - F: Set of final states machine, with $F \subset S$.
  - M: Finite set of messages.
  - $R \subset (S \times S \times M)$: Transitions set, each one involves a source state to a target state following the message receipt. We note (s, s', m) the transition from s to s' after invoking message m.
- **Execution trace**: Service behaviour traces is a finite sequence of operations $(a, b, c, d, e, ...)$. It represents events that service has invoked, from its beginning to the actual state. We note : $trace(P, i)$ to express the execution trace performed by an active instance i in a protocol P.
  For example at time t, trace of instance 10 which has performed the two first operations of protocol P is :
  $trace(P, 10) = orderGoods.validateOrder$
  After executing operation: $makePayement$ at time $t + \triangle t$, the new trace is:
  $trace(P, 10) = orderGoods.validateOrder.makePayement$
- **Complete Execution path**: Represents the sequence of states and messages from an initial state to a final one. We note : $expath(P)$. For instance, protocol P' of **Fig.1** has two complete execution paths:
  $execpath(P) = \{Start.orderGoods.order\ made.makePayement.order\ Paid$,
  $Start.orderGoods.order\ made.cancel\ Order.order\ Canceled\}$.

3.1 Structural approach based protocol schema

Let P and P', respectively old and new service versions after operating changes.
$E_P = \{P_i\ (i = 1 \dots n)\}$: Is the services set **equivalent** to P.
We note $Equi(P_i, P)$, the equivalence relationship between services P_i and P.
Two service are equivalent if they can be used interchangeably and they provide the same functionalities in every context [5]. Every service $Pi \in E_P$ can replace P.
$Equi(P_i, P) \Leftrightarrow \forall(i = 1 \dots n)(expath(P_i) \subset expath(P)) \wedge (expath(P) \subset expath(P_i))$.
($\wedge$ is the logic **and** operator). **(1)**

Let $S_P = \{P_j \ (j = 1 \dots m)\}$: The services set that can substitute P. We note $Subst(P_j, P)$, the substitution relationship.
A service can substitute an other one if it provides, at least, all the conversations that P supports [5] (complete execution paths).
$Subst(P_i, P) \Leftrightarrow \forall (i = 1 \dots m)(expath(P) \subset expath(P_i)$ **(2)**.
Based on this formalization we notice that if protocol P has evolved to a new version P', equations **(1)** does not remain valid. So we want to identify the protocols subset that satisfy equation **(2)**, in order to provide services that can replace the evolved protocol.
From equation **(2)**:$Subst(P_i, P') \Leftrightarrow \forall (i = 1 \dots m)\ (expath(P') \subset expath(P_i)$. **(3)** . We conclude:
Lemma 1:: $\forall i$ satisfying $Equi(P_i, P)$, protocol (P_i) can substitute the new reduced version P', if changes related to protocol evolution are reductionist.
Reducing Protocol specification is application of change operations including:

- Loops removal: Elimination of repeated operations.
- Final sub-paths removal: Reduction of sub-procedures.
- Operations and messages removal: Change in activities and procedure reduction.
- Complete paths removal: Entire cancellation of a procedure.
- sub-protocols removal: Useful in cases of companies mergers and takeovers.

This change operation are very frequent in practice and respond to operational needs like procedures cancellation, reducing tasks, business processes alignment, and so one. However, when changes are additive, substitution analysis must consider the protocol difference. Protocol difference between two protocols P' and P describe the set of all complete execution paths of P' that are not common with P [5]. We note P'/P this difference.
Substitution analysis consists to examine each protocol in the class S_P, with the aim to identify possible protocols that can substitute P'.
Because equation **(1)** no longer holds, we must comply with equation **(2)**.
In order to replace P', each protocol $P_i \in S_P$ must be able to replace the new requirements (the difference P'/P).
$Subst(P_i, P') \Rightarrow Subst(P_i, P'/P) \Leftrightarrow \forall (i = 1 \dots m)\ (expath(P'/P) \subset expath(P_i)$ **(4)**. We conclude:
Lemma 2: If changes are additive, protocols subset $\subset S_P$ which are containing the difference P'/P can substitute the new extended version P'.
Additive changes are operations performing:

- Adding loops: Provide more flexibility, opportunities...
- Adding sub-paths: Procedure improvement, new laws, adaptation.
- Adding messages and operations: For procedure refinement.
- Adding new complete paths: Expansion of activity areas of an organisation.
- Adding sub-protocols: Importing activities.

3.2 Execution traces based analysis

Protocol schema based analysis is rigid and does not take into account the actual execution for active instances. Really, it's possible that a protocol $P_i \in S_P$ can't

substitute an evolved one in general, but by taking into account execution traces, it can do that for specific instances. As an example, let a protocol P and its active instances $i_1, i_2, \ldots, i_n$, as mentioned in **Fig.1**. In parallel, protocol changes have added new states and messages to a particular path: *part-path*. After analysing active instances execution traces, we observe that all instances have't executed this unexpected path *part-path*. In this case, even if we can't replace P' with a protocol $P_i \in S_P$, basing on protocol schema analysis, we can substitute it basing on real execution traces, because changes do not impact real instances. We notice that execution traces may inform protocol managers on how to proceed with substitution analysis. We propose two substitution analysis based execution traces: **Historical execution paths and State execution paths**.

Historical execution paths substitution analysis Let *histpath*, a protocol P historical execution path executed by an active instance i. During its execution, instance i has invoked an operations sequence : $a, b, c, d, e, \ldots$, and let *futurpath*: future paths not yet executed by this instance.
If P' is the new version of P, after changes and S_P is the protocol set that can substitute P, we are interested by filtering instances that are not concerned with changes. We consider protocol changes as the difference between P' and $P : P'/P$.
In this situation, if protocol $P_i \in S_P$ can't substitute P', contrarily, it can substitute it for the instances subset that have not executed this difference.
We note: $Subst(P_i, P')/Occur_j$: The substitution of P' by P_i for occurrence j. $Subst(P_i, P')/Occur_j (i = 1 \ldots n, j = 1 \ldots m)$ is possible if :
$(histpath(occur_j) \notin allpaths(P'/P)$. **(5)**.
$Allpaths(P'/P)$ is the hole possible paths set generated by protocol difference P'/P. This means that, substitution is possible if actual instance i has executed an old path not affected by changes.

State Based Substitution Analysis Historical execution path analysis is more general and based on the hole historical execution paths. Although, protocols $P' \in S_P$ can't replace P in the general case, substitution is possible for some states. So, we need to compute which states are not affected by changes. Substitution analysis must deal with this kind of traces by selecting protocol services that substitute active service by considering actual state and future execution path.
As an example, consider the execution path from **Fig. 1**: If a subset of actual instances are in the state: *Order made*, so their execution trace are : *begin.order made*. A service $P_i \in S_P$ can substitute P' if it can replace it from the actual state and for future execution paths. We don't consider past execution paths because changes occurs after the state *(Order made)*. We formalize this analysis as follows:
Let *futurpaths* the future execution path set of an active instance (all possible future paths), and s the actual state of instance i.
Because we don't know which paths will be token by active instance i during

its life cycle , so we must take into account in our analysis, all possible future paths that instance i can execute. Furthermore, instance i must comply with historical path (form initial state to the actual one). We formalize bellow this analysis:
$Subst(P_i, P')/state(s) : (i = 1 \dots n)$ if :
$(futurpaths(state(s)) \subset allpaths(P'/P)) \wedge (trace(P, i) \in histpath(P_i))$.**(6)**

3.3 Algorithms

We present here **Substitution-based-schema** algorithm to calculate schema protocol based substitution analysis presented in section 3.1.
Algorithm 1: Substitution-based-schema
Input: $P = (S, s_0, F, M, R), P' = (S', s'_0, F', M', R'), P_i = (S_i, s_{0i}, F_i, M_i, R_i)$.
Output: Decision on substitution.
Begin
1. substitution:= True
2. Completpath= $\phi, Completpath' = \phi$
3. CompletPath:= RecursivePaths (S, s_0, F, M, R)
4. CompletPath':= RecursivePaths (S', s'_0, F', M', R')
5. $For\ i = 1\ to\ n$ * n is protocol number *
6. $While\ (path \in completpath)\ Do$
7. $If\ (path \notin CompletPath')\ Do$
8. $substitution := False$
9. $break$
10. EndIf
11. EndWhile
12. EndFor
13. Return (substitution)
14. End **Substitution-based-schema**.
the **Substitution-schema-based** algorithm take as input: old and new version of protocol P and the protocol set that can substitute the old version P. As output it provide decision on substitution by computing potential services remaining that can yet replace P.
The algorithm calculate *completepath* of P and *completepath'* of the new version P' (line 3, 4). By traversing initial protocols class which satisfy inclusion path condition (line 5,6) it will test if every path is included in *completepath'*. If the test is negative, the substitution is impossible for the protocol being tested (line 7,8).
Algorithm 2: RecursivePaths (P, s, F)
Input:$P = (S, s_0, F, M, R)$
Output: All Execution Paths
Begin
1. Path := "" * a variable for any path; initially empty *
2. Paths:= φ * possible paths set *
3. Return-Paths:= φ * temporary variable for cumulate path *
4. For $(s, s', m) \in M$ Do

5. $s' := target - state(m)$
6. if $(s' \notin F)$ Then
7. $Return_Paths$:= RecursivePath (P, s', F) * Recursive Procedure Call *
8. For $(R_path \in Return_Paths)$ Do
9. $Path := s + "." + m + "." + R_Chemin$
10. $Paths := Paths \cup Path$
11. EndFor
12. Else
13. $Paths := Paths \cup (s + "." + m + "." + s')$ * Final state reached *
14. EndFor
15. Return Paths
16. End. Recursivepaths

Recursive-Paths Algorithm, computes recursively all possible paths in a protocol definition, from an initial state to a final one. This algorithm is used in **Substitution-based-schema** algorithm.

4 System Architecture and Software tool presentation

We have implemented the software performing substitution analysis in Java-Eclipse environment. Service protocols are implemented as automata and saved in XML files.

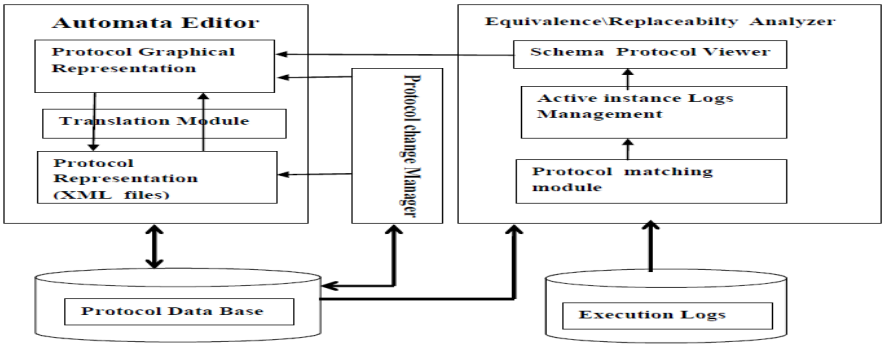


Fig. 2. System Architecture for Managing Dynamic Substitution

The software performs some operational functions useful for protocol manager, like protocol description, operating changes in protocol definition and protocol specification checking. **Fig. 3**.

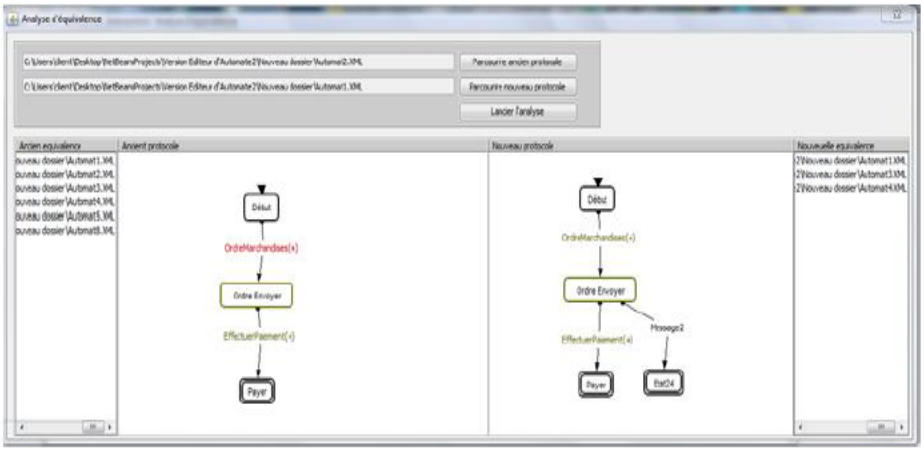


Fig. 3. Protocol specification, evolution, and static Equivalence-Substitution

The system kernel provides checking static equivalence and substitution. Based on schema definitions or on execution traces, the system strength is dynamic analysis. This analysis allows user to select a particular protocol in protocol databases, operate changes and then proceed to change impact analysis on protocol substitution. Change impact analysis is performed basing on schema definition or on execution traces.
We have simulated a local protocol database for tests. After providing protocols specification and evolution, the first step is protocol difference computing (based on protocol definition) **Fig. 4**. Then, we have introduced an aleatory execution instance for each protocol, conducting to a execution logs databases. The system filters the protocol database, analyses logs directory and searches for the remaining service protocols which can substitute the evolved version for a specific instance**Fig. 5**. We visualize, below some screen-shots of the the software tool.

5 Related Work

Protocol management and analysis had benefited for a lot off contributions, from protocol schema matching to static evolution. But, dynamic protocol analysis did not receive all the interest it deserves. In [4],[5] authors presents a general framework for representing, analysing and managing Web service protocols, however this analysis is restricted to static context. In [6], protocol description is enriched with temporal constraints and the same static analysis was performed. In [12], authors had proposed some change operators and patterns specification for changes, but change impact analysis was not studied. In [13], authors studies the compatibility between old and new protocol version and dynamic replaceabilty

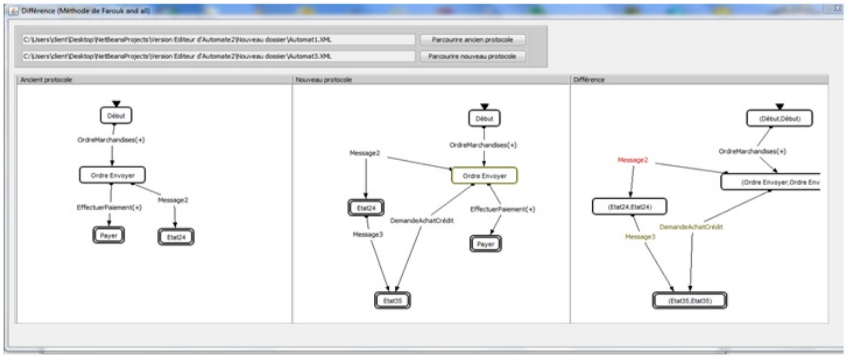


Fig. 4. Substitution analysis based on protocol schema definition

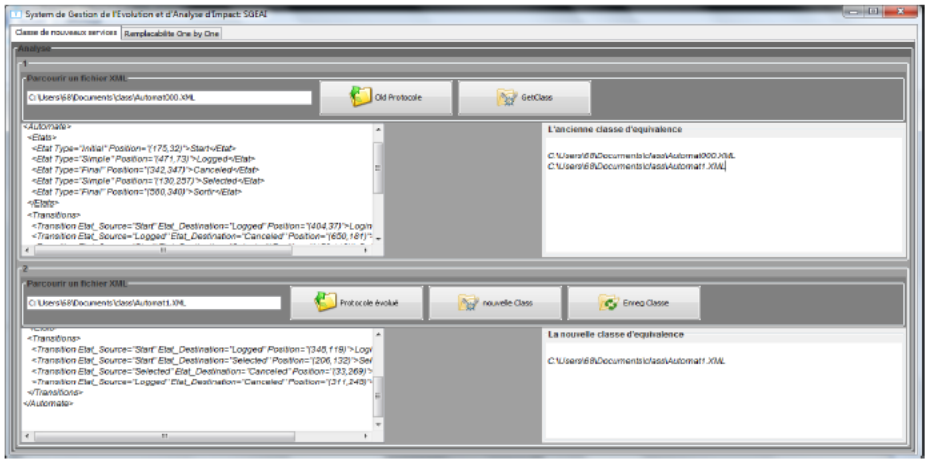


Fig. 5. Substitution analysis provide services substitution class for an evolved protocol

analysis had been presented in therms of compatibility between old and new version only. No comparison with other services was made. Our work responds in a consistent manner to the previous deficiencies.

6 Conclusion and future Work

In this article we have studied and formalized substitution problem inherent to dynamic protocol evolution. We have proposed an approach and a software tool in order to identify service protocols that can, yet replace an evolved one.
As future work, we plan to address protocol substitution analysis for richer protocols descriptions, such timed and transactional constraints automata. In addition, we aim to specify protocol changes more formally by identifying evolution patterns and by their classification with respect to induced impact on protocol substitution and compatibility.

References

[1] R. Chinnici and al. Web Services description Language (WSDL) version 2.0 June 2007. http://www.w3.org/TR/wsdl20/

[2] T. Bellwood and al. UDDI Version 3.0.2 UDDI Spec Technical Committee Draft, 2004. http: //uddi.org/pubs/uddi-v3.htm/

[3] M. Gudgin and al. SOAP version 1.2, July 2001. http://www.w3.org/TR/2001/WD-soap12-20010709/

[4] B. Benatallah and al : Web Service Conversation Modeling A cornerstone for E-Business automation, IEEE Internet computing 8 (1) (2004) 46-545 WSC

[5] B. Benatallah and al : Representing, Analysing and Managing Web Service Protocols. Data Knowledge Ingineering. 58 (3): 327-357, 2006.

[6] J. Ponge and al: Fine-Grained Compatibility and Replaceability Analysis of Timed Web Service Protocols. ER 2007: 599-614

[7] A. Khebizi: External Behavior Modeling Enrichment of Web Services by Transactional Constraints, ICSOC PhD Symposium, December 2008. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-421/paper12.pdf

[8] Rosetanet : http://www.rosettanet.org/.

[9] ebXML Technical Architecture Specification v1.0.4 February 2001, http://ebxml.org/specs/ebTA.pdf.

[10] http://www.xcbl.org/.

[11] Gustavo Alonso, Fabio Casati, Hurumi Kuno,Vijay Machiraju : Web services concepts Architectures and applications, Edition Springer Verlag Berlin 2004.

[12] Barbara Weber and al : Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems , 2008

[13] Ryu, S. H. and al, 2008. Supporting the dynamic evolution of Web service protocols in service-oriented architectures. ACM Trans. Web 2, 2, Article 13, 46 pages. DOI = 10.1145/1346237.1346241 http://doi.acm.org/10.1145/1346237.1346241.

Positionnement des progiciels d'historisation parmi les solutions de gestion de données

Brice Chardin[1], Jean-Marc Lacombe[2] et Jean-Marc Petit[1]

[1] Université de Lyon, CNRS
[1] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[2] EDF R&D, France

Résumé Pour gérer les données de ses systèmes de production d'électricité, EDF a fait le choix de systèmes dédiés à ce cas d'application : les progiciels d'historisation. Ces produits « de niche » ont évolué en parallèle des autres systèmes de gestion de données, en se spécialisant pour ce segment du marché. Dans cet article, nous cherchons à répondre à la question suivante : comment se positionnent les progiciels d'historisation de données parmi les systèmes de gestion de données ? Pour cela, nous examinons les différences avec trois autres catégories de systèmes : SGBDR, systèmes de gestion de flux de données et systèmes NoSQL ; puis définissons un benchmark dérivé du contexte industriel d'EDF.

Mots-clé: Progiciels d'historisation, benchmark, données de capteurs, flux de données

1 Introduction

EDF est un des principaux producteurs d'électricité dans le monde. En France, le groupe exploite un parc constitué de centrales nucléaires, hydrauliques, thermiques à flammes (gaz, charbon, fioul), éoliennes et solaires.

Ces systèmes de production d'électricité, répartis sur plus de cinq cent sites, sont largement instrumentés : au total, environ un million de capteurs constituent les éléments de base des systèmes d'acquisition de données, et échangent de l'ordre du milliard d'enregistrements par jour. L'exploitation de ce volume important de données nécessite dans un premier temps de disposer de moyens d'accès. Pour cela, des systèmes de gestion de bases de données (SGBD) se chargent de leur archivage et de leur mise à disposition. L'archivage de ces données industrielles est un problème complexe. En effet, il s'agit de stocker un grand nombre de données – sur la durée de vie des installations, soit plusieurs décennies – tout en supportant la charge des insertions temps réel et des requêtes d'extraction et d'analyse soumises au SGBD.

Au sein de ces centrales, le système d'information est constitué dans un premier temps d'un ensemble de dispositifs informatiques utilisés en temps réel pour l'exploitation et la surveillance des installations : des systèmes de contrôle-commande prennent en charge des actions automatiques sur le procédé, tandis

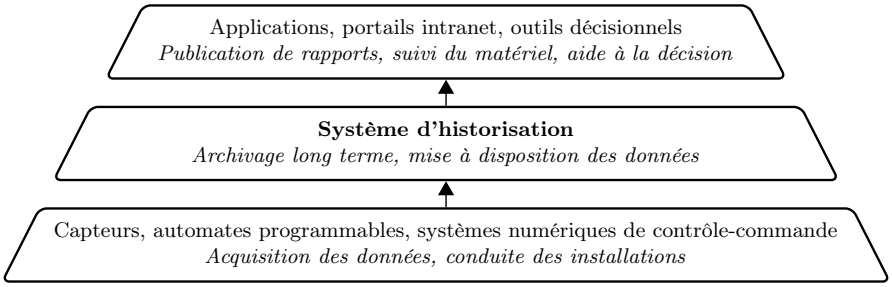


Figure 1. Position des systèmes d'historisation dans le système d'information de production

que d'autres systèmes acquièrent et traitent les données de production pour permettre leur visualisation et assister les opérateurs dans leurs activités de contrôle, de suivi des performances ou d'optimisation de la conduite. À ce niveau, les données sont principalement issues des capteurs ou des rapports d'état du contrôle-commande, mais également de saisies manuelles et d'applications métiers spécifiques pour certains traitements.

L'accès direct à ces données est délicat à cause des impératifs de sûreté : toute intervention sur les données doit se faire sans impacter le fonctionnement de l'existant. À EDF, cet accès est fourni par ce qu'il est convenu d'appeler un « système d'historisation », chargé d'archiver les données de production. Ces données sont alors mises à disposition de diverses applications sans contraintes temps réel (suivi de matériel, outils décisionnels, etc.) et d'acteurs extérieurs à la centrale (ingénierie, R&D, etc.). La figure 1 reprend l'organisation du système d'informations des centrales d'EDF, et présente le positionnement des systèmes d'historisation comme intermédiaires fondamentaux pour l'accès aux données de production.

À l'échelle d'une centrale, des milliers de séries temporelles sont ainsi archivées par le système d'historisation. Chaque série, désignée par un identifiant de « repère », est une séquence de mesures horodatées, généralement représentées par un ensemble de paires date-valeur – où la valeur peut posséder plusieurs dimensions, typiquement la mesure et une métadonnée sur sa qualité. Les périodes d'échantillonnage varient selon les repères, allant de 200 ms – pour la vitesse de rotation de la turbine par exemple – à quelques minutes. Certains repères correspondent à des variables de type événement, dont l'acquisition se produit lors d'un changement d'état – l'ouverture ou la fermeture d'une vanne par exemple.

Pour manipuler des données structurées, les systèmes de gestion de bases de données relationnels (SGBDR) sont la solution privilégiée depuis une quarantaine d'années. Cependant, ceux-ci sont peu utilisés dans ce contexte industriel très spécifique soumis à des contraintes de performances en insertion. Depuis les années 90, EDF a fait le choix d'une catégorie de SGBD dédiée à ce cas d'ap-

plication : les progiciels d'historisation. Ces systèmes ont continué à évoluer en parallèle des SGBDR, en se spécialisant pour ce segment du marché.

Ces progiciels d'historisation sont des solutions propriétaires avec des coûts de licence de l'ordre de plusieurs dizaines de milliers d'euros, et dont le fonctionnement interne n'est pas dévoilé. D'un autre coté, de nombreuses solutions open-source existent pour manipuler de telles données. Les SGBDR en font partie, mais également certains SGBD « NoSQL » dont l'interface d'accès simplifiée peut être adaptée au contexte. Cependant, leurs différences avec les progiciels d'historisation restent mal connues, tant au niveau des fonctionnalités que des performances : alors que de nombreux benchmarks permettent d'évaluer les SGBD dans divers environnements d'utilisation, il n'en existe à notre connaissance aucun à ce jour pour l'historisation de données industrielles.

Dans ce contexte, notre contribution vise à donner des éléments de réponse sur le positionnement des progiciels d'historisation parmi les SGBD. Pour cela, nous présentons dans un premier temps les progiciels d'historisation, en analysant les différences avec les SGBDR, les systèmes de gestion de flux de données (*Data Stream Management System* ou DSMS) et les SGBD NoSQL. Nous proposons ensuite un benchmark, inspiré du fonctionnement de l'historisation des données des centrales nucléaires, et générant des requêtes d'insertion, de mise à jour, de récupération et d'analyse des données. Des résultats expérimentaux sont présentés pour le progiciel d'historisation InfoPlus.21 [2], le SGBDR MySQL [5], et le SGBD NoSQL Berkeley DB [4].

2 Progiciels d'historisation de données

Les progiciels d'historisation de données – comme InfoPlus.21 [2], PI [6] ou Wonderware Historian [3] – sont des progiciels propriétaires conçus pour archiver et interroger les données issues de l'automatique industrielle. Leurs fonctionnalités de base sont toutefois proches de celles des systèmes de gestion de base de données (SGBD), mais spécialisées dans l'historisation de séries temporelles.

Ces progiciels d'historisation de données – ou *data historians* – ont pour but de collecter en temps réel les données en provenance du procédé industriel et de les restituer à la demande. Pour cela, ils sont capables de gérer les données de plusieurs milliers de capteurs : leur historique, pouvant porter sur des décennies de fonctionnement, est alimenté par un flux constant de données en insertion, tout en effectuant des calculs et en répondant aux besoins des utilisateurs.

Les progiciels d'historisation supportent des rythmes d'insertion élevés, et peuvent ainsi traiter plusieurs centaines de milliers d'évènements – soit quelques mégaoctets de données – à la seconde. Ces performances sont permises par une conception spécifique des buffers d'insertion, qui conservent les valeurs récentes en mémoire volatile pour les écrire ensuite sur disque, triées chronologiquement. La fenêtre temporelle associée à ce buffer impose des contraintes fortes sur l'horodatage des données. Ces contraintes peuvent ainsi porter sur un intervalle de validité, par exemple uniquement des dates passées ou appartenant à une fenêtre temporelle donnée, ou encore sur la séquentialité des données, ie. les insertions

doivent être effectuées en respectant leur ordre chronologique. Le non-respect de ces contraintes peut entraîner une diminution notable des performances, voire l'impossibilité d'insérer les données.

Le modèle de données utilisé par les progiciels d'historisation est un *modèle hiérarchique*, permettant de représenter les données suivant la structuration physique des installations et faciliter ainsi la consultation de données similaires groupées par sous-systèmes. Dans le domaine de la production d'électricité par exemple, les repères peuvent être regroupés dans un premier temps par site d'exploitation, puis par unité de production, et enfin par système élémentaire (e.g. le système de production d'eau déminéralisée).

Les progiciels d'historisation proposent également une vision relationnelle des données archivées, accessible à l'aide d'une interface SQL. Les progiciels d'historisation peuvent cependant ne pas se conformer à la norme SQL dans son ensemble. De plus, les progiciels d'historisation ne sont pas conçus pour gérer des bases de données relationnelles dans le cas général, certains types de données n'étant pas supportés (image, texte, BLOB, etc.).

En plus du SQL, les progiciels d'historisation fournissent également une interface dédiée pour les insertions, mises à jour et récupération des données. Les insertions sont fonctionnellement comparables aux requêtes SQL "INSERT", en évitant l'analyse syntaxique et les conversions de types. L'interface de récupération des données cependant diffère significativement du SQL. Les extractions peuvent être définies avec des conditions de filtrage (typiquement avec des seuils de valeur ou des vérifications du champ qualité), du ré-échantillonnage ou des calculs d'agrégats sur des intervalles temporels – par exemple pour calculer la moyenne horaire. Bien que les conditions de filtrage et la définition d'intervalles soient traduisibles simplement en SQL, l'interpolation de valeurs (avec divers algorithmes d'interpolation : par pallier, linéaire, etc.) peut être fastidieuse à définir, tant en SQL qu'avec des interfaces NoSQL usuelles, en particulier lorsque plusieurs séries temporelles – possédant leur propre période d'échantillonnage – sont concernées, comme pour le produit de deux séries par exemple.

Néanmoins, les entrepôts de paires clé-valeur ordonnées fournissent des méthodes d'accès NoSQL proches, comme les curseurs de Berkeley DB [4]. Ces curseurs peuvent être positionnés sur une valeur de clé, et être incrémentés ou décrémentés suivant l'ordre des clés – pour récupérer les valeurs consécutives d'une série temporelle dans ce contexte. Malgré tout, l'interface des progiciels d'historisation est spécialisée, et donc combine de nombreux algorithmes et traitements usuels adaptés aux besoins industriels, en plus de l'extraction de données brutes.

Concernant la politique tarifaire, les progiciels d'historisation sont proposés avec des licences propriétaires. Il est difficile d'estimer le coût de ces licences car leur prix est négocié pour chaque client. L'ordre de grandeur de ce coût est de plusieurs dizaine de milliers d'euros par serveur, avec des tarifs dégressifs en fonction du nombre de repères – allant de 10 à 0.5 euros par repère. Les progiciels

d'historisation proposent une offre de support et de maintenance, dont le coût annuel correspond à un pourcentage du prix de la licence.

Synthèse Les progiciels d'historisation sont des produits conçus et vendus pour un usage industriel spécifique. Les autres SGBD peuvent avoir des cadres d'application plus variés, pour un coût potentiellement moins important, mais n'incluent généralement pas la plupart des fonctionnalités métier que possèdent les progiciels d'historisation. Ces systèmes ne supportent typiquement pas les protocoles de communication industriels, ni les algorithmes de compression avec perte spécifiques aux séries temporelles, l'interpolation ou le ré-échantillonnage. Dans l'ensemble, les progiciels d'historisation peuvent être caractérisés par :

- une structure de schéma hiérarchique simple, basée sur les repères,
- une architecture centralisée,
- une conception optimisée pour l'archivage long-terme d'un grand volume de données ordonnées chronologiquement, où la date joue un rôle fondamental,
- des algorithmes de compression adaptés,
- une "interface NoSQL" avec filtrage, ré-échantillonnage et calcul d'agrégats,
- une interface SQL,
- pas de transactions,
- uniquement des données de capteur (pas d'image, de blob, etc.),
- des applications intégrées spécialisées pour les données industrielles,
- une politique tarifaire basée sur le nombre de repères.

En quelque sorte, les progiciels d'historisation sont des SGBD non relationnels – donc « NoSQL » – qui ont su s'imposer sur un marché de niche. Pour autant, ils ne correspondent à aucune des catégories de SGBD existantes.

3 Benchmark adapté à l'historisation de données

Une évaluation basée sur des critères fonctionnels est importante, mais ne permet pas d'avoir une idée quantitative sur les capacités de traitement de chaque système. Par ailleurs, les éditeurs des progiciels d'historisation ne publient pas les capacités de traitement de leurs solutions. L'utilisation d'un benchmark est donc le seul moyen d'évaluer les différences de performances entre ces systèmes. Cependant, cette comparaison ne s'avère pas si facile, les fonctionnalités, les interfaces et le modèle de données sous-jacents étant différents.

Pour cela, nous nous concentrons sur des opérations (requêtes) simples sur un schéma de base de données générique. Nous proposons donc un benchmark et l'utilisons pour évaluer un progiciel d'historisation, un entrepôt de paires clé-valeur ordonnées et un SGBDR.

Alors que de nombreux benchmarks existent pour les systèmes de gestion de base de données relationnels, comme TPC-C ou TPC-H [7,8], il n'en existe, à notre connaissance, pas pour les progiciels d'historisation. L'idée de comparer ces systèmes à l'aide d'un benchmark existant – adapté aux SGBDR – est donc naturelle. Cependant, il ne nous a pas semblé possible de mettre en oeuvre

un benchmark du Transaction Processing Performance Council (TPC) pour les raisons suivantes :

- Les progiciels d'historisation ne respectent pas les contraintes ACID et ne permettent pas les transactions.
- L'insertion au fil de l'eau est une opération primordiale pour les systèmes d'historisation, ce qui exclut les benchmarks insérant les données par groupements, comme TPC-H.
- Les progiciels d'historisation sont conçus pour traiter des séries temporelles. Il est nécessaire que le benchmark manipule ce type de données pour que les résultats soient significatifs.

Les benchmarks pour les DSMS, comme *Linear Road* [1] auraient aussi pu être envisagés ; mais les progiciels d'historisation ne supportant pas les requêtes continues, leurs mises en œuvre auraient été impossibles.

Pour comparer les performances des progiciels d'historisation avec d'autres SGBD, nous définissons un benchmark basé sur un scénario reprenant le fonctionnement de l'historisation des données des centrales nucléaires d'EDF. Dans ce contexte, les données sont issues de capteurs répartis sur le site d'exploitation et agrégées par un démon servant d'interface avec le système d'historisation. Pour les insertions, ce benchmark simule le fonctionnement de ce démon et génère pseudo-aléatoirement les données à insérer. Ces données sont alors accessibles par des applications ou utilisateurs distants, qui peuvent envoyer des requêtes pour mettre à jour, récupérer ou analyser ces données. Après la phase d'insertion, ce benchmark génère un ensemble simple mais représentatif de ce type de requêtes.

3.1 Modèle de données

Ce benchmark manipule les données selon un schéma minimal, centré sur les données de séries temporelles. Pour chaque type de variable – analogique ou booléen – une table de description est définie (resp. `ana` et `bool`). Les mesures sont stockées dans des tables différentes (resp. `anavalues` et `boolvalues`). La figure 2 présente le schéma logique utilisé pour ce benchmark.

Chaque repère est en particulier associé à un identifiant (`AnaId` ou `BoolId`). Pour les données analogiques, la table de description contient également une période d'échantillonnage théorique (`Interval`) et deux seuils délimitant les valeurs critiques basses (`Threshold1`) ou hautes (`Threshold2`).

Les séries temporelles sont stockées dans les tables `anavalues` et `boolvalues`, qui contiennent l'identifiant (`AnaId` ou `BoolId`) – en tant que clé étrangère –, la date de la mesure avec une précision à la milliseconde (`Date`), la valeur (`Value`) et un tableau de huit bits pour les méta données (`Quality`).

Pour que ce benchmark soit compatible avec les modèles de données hiérarchiques utilisés par les progiciels d'historisation, le modèle relationnel défini précédemment ne peut pas être imposé. Dans la figure 3, nous proposons un modèle hiérarchique équivalent, permettant de représenter les mêmes données et d'exécuter des requêtes fonctionnellement équivalentes.

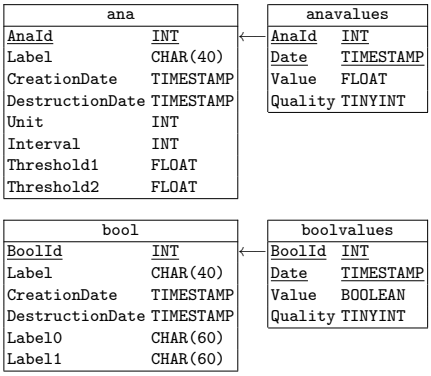


Figure 2. Schéma logique de la base de données

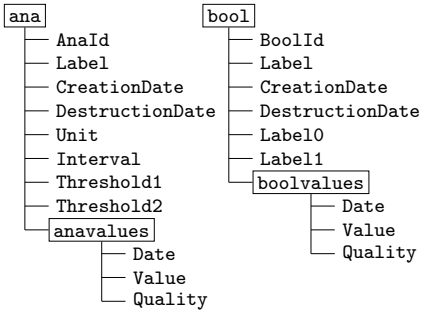


Figure 3. Schéma logique équivalent pour le modèle hiérarchique

3.2 Requêtes

Ce benchmark définit douze requêtes représentatives de l'usage d'EDF pour évaluer les performances de chaque système, et identifier les optimisations spécifiques à certains types de requêtes. Pour conserver une définition simple et faciliter l'analyse des performances, les interactions entre les requêtes ne sont pas prises en comptes : les requêtes sont exécutées une par une dans un ordre établi. En particulier, l'évaluation des performances malgré une charge continue en insertion n'est pas considérée, même si cela correspond à une situation plus réaliste. De même, les traitements spécifiques aux séries temporelles proposés par les progiciels d'historisation ne font pas partie des requêtes exécutées par le benchmark, car leur équivalent en SQL standard peut s'avérer compliqué à définir (par exemple pour calculer une moyenne pondérée par les intervalles temporels – variables – des mesures). Pour chaque requête, les équivalents NoSQL doivent permettre d'obtenir les mêmes résultats.

Insertion des données
R0.1 Insertion de valeur analogique.
Paramètres `ID`, `DATE`, `VAL` et `QUALITY`.

```
INSERT INTO anavalues VALUES
  ([ID],[DATE],[VAL],[QUALITY]);
```

R0.2 Insertion de valeur booléenne.

Modification des données La mise à jour de données est une opération rare pour les systèmes d'historisation. Le benchmark considère cependant l'impact des mises à jour sur les performances.

R1.1 Mise à jour d'une valeur analogique et de son champ qualité.
Paramètres `VAL`, `ID` et `DATE`.

```
UPDATE anavalues
SET Value = [VAL], Quality = (Quality | 128)
WHERE AnaId = [ID] AND Date = [DATE];
```

R1.2 Mise à jour d'une valeur booléenne et de son champ qualité.

Extraction de données brutes

R2.1 Valeurs analogiques brutes.
Paramètres `ID`, `START` et `END`.

```
SELECT * FROM anavalues
WHERE AnaId = [ID] AND Date BETWEEN [START] AND [END]
ORDER BY Date ASC;
```

R2.2 Valeurs booléennes brutes.

Calcul d'agrégats

R3.1 Dénombrement de données analogiques.
Paramètres `ID`, `START` et `END`.

```
SELECT count(*) FROM anavalues
WHERE AnaId = [ID] AND Date BETWEEN [START] AND [END]
```

R3.2 Dénombrement de données booléennes.

R4 Somme de valeurs analogiques.

R5 Moyenne de valeurs analogiques.

R6 Minimum et Maximum de valeurs analogiques.

Filtrage sur la valeur

R7 Dépassement de seuil critique.
Paramètres `ID`, `START` et `END`.

```
SELECT Date, Value FROM ana, anavalues
WHERE ana.AnaId = anavalues.AnaId AND ana.AnaId = [ID]
  AND Date BETWEEN [START] AND [END]
  AND Value > ana.Threshold2;
```

R8 Dépassement de valeur.

Calcul d'agrégats avec filtrage sur plusieurs séries

R9 Repère avec valeurs anormales.
Récupère le repère dont les valeurs ne sont, le plus souvent, pas comprises entre ses deux seuils critiques entre deux dates.
Paramètres `START` et `END`.

```
SELECT Label, count(*) as count FROM ana, anavalues
WHERE ana.AnaId = anavalues.AnaId
  AND Date BETWEEN [START] AND [END]
  AND (Value > Threshold2 OR Value < Threshold1)
GROUP BY ana.AnaId ORDER BY count DESC LIMIT 1;
```

Opérations sur les dates

R10 Vérification de la période d'échantillonnage.
Récupère les repères dont la période d'échantillonnage ne respecte pas la valeur `Interval` donnée dans la table `ana`.
Paramètres `START` et `END`.

```
SELECT values.AnaId, count(*) as count FROM ana,
( SELECT D1.AnaId, D1.Date,
         min(D2.Date-D1.Date) as Interval
  FROM anavalues D1, anavalues D2
  WHERE D2.Date > D1.Date AND D1.AnaId = D2.AnaId
    AND D1.Date BETWEEN [START] AND [END]
  GROUP BY D1.AnaId, D1.Date
) as values
WHERE values.AnaId = ana.AnaId
  AND values.Interval > ana.Interval
GROUP BY values.AnaId ORDER BY count DESC LIMIT 1;
```

Extraction de valeurs courantes Ces deux requêtes ne possédant pas de paramètre, elles ne sont exécutées qu'une seule fois pour éviter d'utiliser le cache sur les requêtes – stocker leur résultats pour ne pas avoir à les réévaluer. Elles permettent de récupérer les valeurs les plus récentes pour chaque repère de la base.

R11.1 Valeurs analogiques courantes.

```
SELECT AnaId, Value FROM anavalues,
WHERE (AnaId, Date) IN
( SELECT AnaId, max(Date) FROM anavalues
  GROUP BY AnaId )
ORDER BY AnaId;
```

R11.2 Valeurs booléennes courantes.

4 Expérimentation du benchmark

Ce benchmark a été exécuté avec le progiciel d'historisation InfoPlus.21, le SGBDR MySQL et le SGBD NoSQL Berkeley DB. Les progiciels d'historisation sont des solutions propriétaires avec des conceptions distinctes et donc des

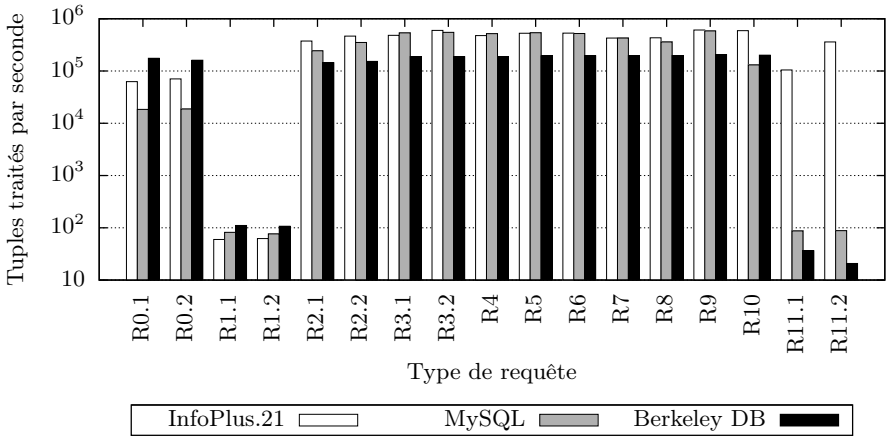


Figure 4. Capacités de traitement

performances différentes. Nous avons choisi l'un des plus répandus, InfoPlus.21, utilisé à EDF dans le domaine nucléaire. Nous avons retenu MySQL pour sa facilité d'utilisation et sa pérennité avec une communauté d'utilisateurs importante, essentiels pour un usage industriel. Enfin, nous avons choisi l'entrepôt de paires clé-valeur ordonnées Berkeley DB pour nos expérimentations. Cette catégorie de système NoSQL est particulièrement adaptée aux requêtes usuelles basées sur des intervalles de clés (*range queries*). MySQL (avec le moteur de stockage InnoDB) comme Berkeley DB ont été optimisés pour ce contexte d'utilisation, notamment en désactivant la gestion de transactions.

Pour chaque système, le serveur de test possède un processeur Xeon Quad Core [3] E5405 2.0 GHz, 3 GB de RAM et trois disques durs de 73 GB 10K.

500 000 000 tuples de données sont insérés pour chaque type – analogique et booléen – ce qui correspond à 11.5 GB sans compression. 1 000 000 mises à jour sont ensuite effectuées, suivies de 1 000 requêtes R2 à R8, 100 requêtes R9 et R10, et 1 requête R11.1 et R11.2. Les paramètres des requêtes sont générés de manière à accéder, en moyenne, à 100 000 tuples pour les requêtes R2 à R8, et 10 000 000 tuples pour les requêtes R9 et R10.

4.1 Résultats

Les capacités de traitement sont données dans la figure 4, en indiquant le nombre de tuples traités à la seconde.

Pour l'analyse des résultats expérimentaux, il est possible de regrouper les requêtes présentant des performances similaires par catégories. On distingue quatre

3. Pour ces tests, seul un cœur est activé a cause du manque d'optimisation de notre progiciel d'historisation pour des insertions multi-threadées.

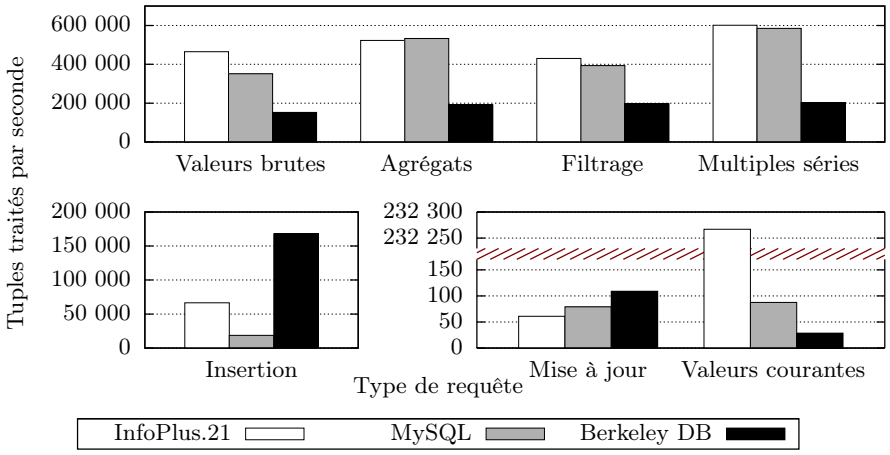


Figure 5. Capacités de traitement par catégorie

catégories de requêtes : les insertions (R0.1 et R0.2), les mises à jour (R1.1 et R1.2), les *range queries* (R2.1 à R10) et l'extraction des valeurs courantes (R11.1 et R11.2). Parmi les *range queries*, on identifie quatre types de requêtes : R2.1 et R2.2 pour les valeurs brutes, R3, R4, R5 et R6 pour les agrégats, R7 et R8 pour le filtrage, et R9 et R10 pour l'interrogation de multiples repères.

La figure 5 donne alors un aperçu des différences de performances en regroupant les requêtes similaires. Pour cette analyse, la requête R2.1 n'est pas prise en compte à cause des activités en arrière plan dues aux mises à jour, qui dégradent les performances de cette requête. La requête R10 est également ignorée parmi les résultats MySQL, à cause de mauvaises performances probablement dues à la définition des procédures stockées utilisées pour son traitement.

Comme on pouvait s'y attendre, les progiciels d'historisation gèrent les insertions efficacement par rapport aux SGBDR : InfoPlus.21 atteint 66 500 insertions par seconde (ips), soit environ 3.2 fois mieux qu'InnoDB et ses 20 500 ips. Toutefois, Berkeley DB atteint 168 000 ips, soit 2.5× mieux qu'InfoPlus.21. Ce résultat est à relativiser car Berkeley DB est utilisé en tant que librairie, sans mécanismes de communication entre processus, ce qui peut avoir un impact important sur les performances par rapport à MySQL ou InfoPlus.21.

L'extraction des valeurs courantes (R11.1 et R11.2) est le deuxième point fort prévisible des progiciels d'historisation, étant donnée leur conception particulière où les valeurs les plus récentes sont conservées en mémoire. Cette opération est plus rapide de plusieurs ordres de grandeur par rapport à MySQL (×1 850) ou Berkeley DB (×6 140).

Pour les autres requêtes d'analyse [4], MySQL et InfoPlus.21 présentent des performances très proches, avec au plus de 25% de différence entre ces deux

4. R2.1 et R10 exclues.

systèmes. Leurs capacités de traitement sont comprises entre 350 000 et 610 000 tuples par seconde selon les requêtes. Berkeley DB, de part la simplicité de son interface d'accès, présente des performances homogènes, mais sensiblement moins bonnes, autour de 190 000 tuples par seconde.

5 Conclusion

L'historisation de données industrielles est un contexte applicatif pour lequel les SGBDR sont peu utilisés, au profit de produits "de niche" spécialisés pour les traitements spécifiques des applications sous-jacentes. Les raisons de cette segmentation du marché sont historiques : elles se basent sur les contraintes de performances auxquelles les systèmes d'historisation sont soumis. Pourtant, même si ces progiciels d'historisation ont été conçus pour supporter une charge particulièrement importante en insertion de données, les résultats du benchmark que nous avons proposé mettent en évidence des performances du même ordre de grandeur pour les SGBDR et systèmes NoSQL : avec une configuration adaptée, MySQL et Berkeley DB pourraient supporter les charges identifiées à EDF et donc pourraient être compétitifs du point de vue "gestion de données".

Cependant, la comparaison fonctionnelle met en avant la force des progiciels d'historisation : ces fonctionnalités « métier » tendent à prendre le pas sur les performances. Les clients métiers et la facilité d'intégration (communication avec les autres systèmes, configuration originale adaptée au contexte, etc.) constituent une différence importante avec les autres catégories de SGBD. Néanmoins, pour s'affranchir du coût de licence ou pour intégrer le système d'historisation dans un environnement où les ressources matérielles sont limitées voire où un progiciel d'historisation n'est pas utilisable (système d'exploitation non supporté, etc.), un SGBD conventionnel pourrait être mis en œuvre dans ce contexte applicatif.

Références

1. A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. S. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts. Linear Road : A Stream Data Management Benchmark. In *VLDB'04 : 30th International Conference on Very Large Data Bases*, pages 480–491, 2004.
2. Aspen Technology. *Database Developer's Manual*, 2007. Version 2006.5.
3. Invensys Systems. *Wonderware Historian 9.0 High-Performance Historian Database and Information Server*, 2007.
4. M. A. Olson, K. Bostic, and M. I. Seltzer. Berkeley DB. In *FREENIX'99 : 1999 USENIX Annual Technical Conference, FREENIX Track*, pages 183–191, 1999.
5. Oracle. *MySQL 5.5 Reference Manual*, 2011.
6. OSIsoft. *PI Server System Management Guide*, 2009. Version 3.4.380.
7. Transaction Processing Performance Council. *TPC Benchmark C Standard Specification*, 2007.
8. Transaction Processing Performance Council. *TPC Benchmark H Standard Specification*, 2008.

Les Concepts Sont-ils de Bons Candidats à l'Indexation?

Fatiha Boubekeur-Amirouche[1], Wassila Azzoug[2], Mohand Boughanem[3],

[1]Université Mouloud Mammeri, 15000 Tizi-Ouzou, Algérie,
[2]Université M'Hamed Bouguerra, 35000 Boumerdès, Algérie,
[3]Université Paul Sabatier de Toulouse, 31000 Toulouse, France
amirouchefatiha@mail.ummto.dz, azzoug_wassila@umbb.dz, boughane@irit.fr

Résumé. L'indexation par les concepts (ou indexation conceptuelle) en recherche d'information (RI), a pour objet de représenter les documents (et requêtes) par des entités sémantiques, les concepts, plutôt que par des entités lexicales, les mots, qu'ils contiennent. Le but étant de retrouver les documents sémantiquement pertinents pour une requête utilisateur. Dans ce papier, nous proposons d'expérimenter différentes variantes d'une approche d'indexation par les concepts. Les concepts sont identifiés par un processus de désambiguïsation qui s'appuie sur l'utilisation conjointe de deux ressources linguistiques : WordNet et WordNetDomains. L'évaluation expérimentale de notre approche d'indexation a montré des résultats très satisfaisants.

Mots-clés: Recherche d'information, Indexation conceptuelle, WordNet, WordNetDomains, Désambiguïsation.

1 Introduction

Un système de recherche d'information (SRI) a pour but de sélectionner, dans une collection de documents préalablement enregistrée, l'ensemble des documents pertinents pour une requête utilisateur.

L'une des étapes clé dans ce processus de recherche, est l'indexation. L'indexation consiste à construire une représentation simplifiée, l'index, des documents (et requêtes) dans le but de faciliter la recherche. Dans les systèmes de RI classiques, l'index est constitué d'un ensemble de termes (généralement des mots clés simples) pondérés, sensés représenter au mieux le contenu sémantique du document (ou de la requête) auquel il est associé.

L'autre étape clé dans le processus de RI est l'appariement document-requête. L'objectif est de comparer l'index des documents et celui de la requête afin de retrouver les documents pertinents pour la requête. Dans les SRI classiques, l'appariement document-requête est lexical et se base sur la présence ou l'absence d'un mot de la requête dans le document. Or il est bien connu que les mots de la langue sont ambigus. Un même mot peut avoir des sens différents (cas des homonymes) dans le document et dans la requête, et différents mots utilisés dans le document et la requête peuvent avoir un même sens (cas des synonymes). L'appariement lexical est incapable de traiter cette ambiguïté de la langue naturelle.

De ce fait, des documents pertinents pour la requête ne seront pas retrouvés, et des documents non pertinents seront retournés à l'utilisateur.

L'indexation conceptuelle tente de pallier les problèmes de l'appariement lexical en utilisant pour la recherche, des index sémantiques au lieu de simples mots clés. De tels index sont construits à partir de concepts. Un concept correspond généralement au sens associé à un mot dans un dictionnaire, un thésaurus ou une ontologie….

L'indexation conceptuelle se base en général sur trois étapes : (1) l'identification des termes d'index, (2) la désambiguïsation de ces termes et (3) la pondération des concepts.

La première étape a pour objet d'identifier les termes représentatifs du contenu du document. Il s'agit d'une étape d'indexation classique basée sur des techniques linguistiques de tokénisation, lemmatisation et élimination de mots vides, et éventuellement sur des techniques plus avancées d'identification de collocations.

Les mots de langue sont par nature ambigus. Un mot peut être associé à différents sens. La seconde étape de l'indexation conceptuelle a pour objectif de désambiguïser le sens d'un mot, c'est-à-dire d'identifier le sens correct du mot, dans son contexte d'utilisation dans le document. Pour ce faire, les approches de désambiguïsation s'appuient sur des ressources linguistiques telles que les dictionnaires automatisés, ou encore les ontologies (la ressource la plus utilisée [2], [3], [11], [20], [21] étant la base lexicographique WordNet [14]), qui constituent des sources d'évidence pour les définitions et sens des mots. Le principe de la désambiguïsation consiste généralement à classer chaque sens possible d'un mot selon un score basé sur sa distance sémantique par rapport aux sens des autres mots du même contexte dans le document. Dans une approche plus récente [11], les auteurs se basant sur WordNet et WordNetDomains, proposent une désambiguïsation à deux niveaux : d'abord désambiguïser le domaine correct d'un mot dans le document, puis désambiguïser ce mot dans le domaine ainsi identifié. Le domaine correct d'un mot est celui qui maximise ses occurrences dans le contexte local du mot cible. En nous basant sur un principe similaire, nous avons proposé dans [4] de désambiguïser les domaines sur la base d'une mesure de probabilité d'appartenance du mot cible et des mots du contexte à un domaine donné. Le domaine qui maximise cette mesure est retenu comme domaine correct du terme dans le document. La désambiguïsation d'un mot dans son domaine est basée sur un score cumulé de ses proximités sémantiques [12], [13] aux autres mots de son contexte.

La troisième étape a pour objet d'associer à chaque concept un poids numérique représentant son degré d'importance dans le document. L'objectif est de pouvoir ordonner les résultats de la recherche par ordre de pertinence (généralement décroissant). (La pondération n'est pas l'objet de ce papier).

Notre objectif à travers ce papier est de répondre à la question suivante : « **les concepts sont-ils de bons candidats à l'indexation des documents en RI?**». Pour répondre à cette question, nous proposons d'implémenter différentes variantes de notre approche (théorique) d'indexation sémantique proposée dans [4], et de les évaluer par rapport à des approches d'indexation classique basée mots-clés.

La suite du papier est structurée comme suit : La section 2 introduit des notions préliminaires sur WordNet et WordNetDomains, ainsi que des définitions utilisées

dans le reste du papier. La section 3 rappelle notre approche d'indexation conceptuelle. La section 4 est dédiée à l'évaluation de notre approche, les résultats expérimentaux y sont présentés. La section 5 présente des travaux connexes. La section 6 conclut le papier.

2 Notions Préliminaires

2.1 WordNet et WordNetDomains

WordNet est une base lexicographique électronique qui couvre la majorité des noms, verbes, adjectifs et adverbes de la langue Anglaise, qu'elle structure en un réseau de nœuds et de liens. Les nœuds sont des concepts. Ils sont constitués par des ensembles de termes synonymes appelés *synsets*. Les liens représentent des relations sémantiques entre synsets, dont par exemple la relation de subsumption (ou relation *is-a*) qui permet d'associer un concept classe (l'hypernyme) à un concept sous-classe (l'hyponyme).

WordNetDomains est une extension de WordNet dans laquelle les synsets ont été étiquetés par des labels de domaines. Ces domaines sont organisés dans une hiérarchie définissant la relation de spécialisation/généralisation entre eux. La racine de cette hiérarchie est le domaine *Top-Level*. Un domaine particulier, le domaine *Factotum*, indépendant de la hiérarchie issue de *Top-Level*, regroupe tous les sens des mots qui n'appartiennent à aucun domaine particulier mais qui peuvent apparaître avec des termes associés à d'autres domaines.

2.2 Définitions et Notations

Soit m un mot d'un texte à indexer.

1. On appelle occurrence de m, toute instance m_i de m dans le texte. Une instance m_i du mot m apparaît dans une seule phrase.
2. On appelle contexte global du mot m_i, l'union de toutes les phrases contenant au moins une occurrence de m_i dans le document. Le contexte global du mot sera noté par : ζ_{G_i}.

3 Approche Proposée

L'indexation sémantique vise à représenter un document par un ensemble de concepts pondérés qui décrivent au mieux son contenu.

Le processus d'indexation du document s'effectue en trois étapes: (1) l'identification des termes d'index, (2) la désambiguïsation des termes d'index et (3) la pondération des concepts.

3.1 Identification des Termes d'Index

Le but de cette étape est d'identifier en se basant sur WordNet l'ensemble des termes (simples et collocations) représentatifs du document.

L'algorithme mis en œuvre (détaillé dans [4]) se base sur une liste prédéfinie φ_{Coloc} de toutes les collocations de WordNet pour identifier:

- l'ensemble ξ_{Expres} des collocations (mots composés ou expressions) du document, correspondant à des collocations dans φ_{Coloc}.
- l'ensemble $\xi_{Simples}$ des mots simples du document ayant une entrée dans WordNet,
- l'ensemble ξ_{Orphel} des mots orphelins (mots simples du document n'ayant pas d'entrée dans WordNet).

3.2 Désambiguïsation des Termes

Les collocations étant des expressions quasiment désambiguïsées, l'étape de désambiguïsation concerne uniquement les mots simples ayant des entrées dans WordNet, soit donc l'ensemble des termes de $\xi_{Simples}$.

Chaque terme de $\xi_{Simples}$ peut avoir plusieurs sens possibles. Le but de cette étape est de sélectionner le sens adéquat du terme dans le document. Notre approche de désambiguïsation se base sur le contexte global du mot (tel que défini en section 2.2).

L'approche de désambiguïsation proposée est une approche à trois niveaux :

(1) dans le premier niveau, il s'agit de déterminer la catégorie syntaxique (nom, verbe, …) du mot *m* dans son contexte. L'identification de la catégorie syntaxique des mots d'un texte donné est réalisée en utilisant un tagger syntaxique.

(2) le second niveau, permet d'identifier le domaine correspondant à l'usage du mot dans son contexte. L'identification des domaines s'appuie sur l'utilisation de *WordNetDomains*. Ce niveau de désambiguïsation permettra de limiter le nombre de sens du terme qui seront examinés dans le niveau suivant de désambiguïsation.

(3) le troisième niveau de désambiguïsation consiste alors à sélectionner parmi les sens possibles du terme dans le domaine sélectionné, celui qui est sensé le définir au mieux dans son contexte.

Identification de la catégorie syntaxique des mots simples. La catégorie syntaxique d'un mot dans un contexte d'utilisation donné est un premier indicateur de son sens dans ce contexte. Pour identifier les catégories syntaxiques des mots d'un document,

nous utilisons le Stanford POS Tagger[1]. Cette étape permet non seulement de limiter le nombre de sens mais aussi de préciser les sens d'un terme qui seront examinés lors des prochains niveaux de désambiguïsation.

Désambiguïsation au niveau des domaines. Un mot dans $\xi_{Simples}$ peut posséder plusieurs sens dans WordNet, relativement à la catégorie syntaxique qui lui est associée dans son contexte. Les sens de WordNet sont étiquetés dans *WordNetDomains* par des labels de domaines. Un sens peut appartenir à un ou plusieurs domaines.

Partant de l'hypothèse que le domaine probable d'un mot dans le document est celui qui maximise sa similarité avec les domaines des autres termes (mots simples ou collocations) du document, nous attribuons à chaque domaine D_j associé à un sens du mot m_i un score de désambiguïsation (formule (1)). Le domaine D_j qui maximise ce score est sélectionné comme domaine adéquat pour le mot m_i dans le contexte considéré.

$$Score(D_j) = \arg\max_j \left(\sum_{t_k \in \zeta_{G_i}} \sum_{k | t_k \in \{\xi_{simples} \cup \xi E_{xpres}\}} Sim(D_j, D_k) \right). \quad \textbf{(1)}$$

Où :

ζ_{G_i} désigne le contexte global de m_i.

$Sim(D_j, D_k)$ désigne la similarité entre les domaines D_j et D_k.

Pour mesurer la similarité entre les domaines D_j et D_k, nous utilisons et adaptons la formule de Wu-Palmer [22] à la hiérarchie *Top-Level* de WordNetDomains, ce qui donne:

$$Sim(D_j, D_k) = \frac{2 * profondeur(D^*)}{profondeur(D_j) + profondeur(D_k)}. \quad \textbf{(2)}$$

Où :

D^*: est le domaine le plus spécifique qui subsume D_j et D_k dans la hiérarchie de WordNetDomains.

profondeur(D)* : est le nombre d'arcs entre la racine de WordNetDomains (*Top-Level*) et le domaine D^*.

profondeur(D_j) : est le nombre d'arcs entre le domaine *Top-Level* et le domaine D_j en passant par le domaine D^*.

[1] http://nlp.stanford.edu/software/tagger.shtml

Remarque 1. La formule de similarité est appliquée aux seuls domaines de la hiérarchie *Top-Level*. Le domaine *factotum*, indépendant de cette hiérarchie n'est pas considéré dans cette désambiguïsation.

Désambiguïsation des sens des mots. A l'issue de l'étape précédente, tout mot m_i de $\xi_{Simples}$ est associé à un seul domaine D_j dans son contexte. Deux cas peuvent se présenter :

- soit m_i possède un seul sens dans D_j, dans ce cas il est désambiguïsé.
- soit m_i possède plusieurs sens dans D_j, dans ce cas il est ambigu. Il faut le désambiguïser. Nous proposons une désambiguïsation des seuls sens appartenant à D_j. L'objectif est alors de sélectionner parmi ces sens, le sens correct pour le mot m_i dans son contexte.

Soit

- S_i l'ensemble de tous les synsets associés au mot m_i,
- D l'ensemble, non redondant, de tous les domaines associés aux éléments de S_i ,
- $S_{i(j)}$ est l'ensemble des synsets de S_i appartenant au domaine D_j ,
- $S_{i(j)}[k]$ le kième élément de l'ensemble $S_{i(j)}$.

Pour désambiguïser le mot m_i dans son domaine, on associe à chacun de ses sens $S_{i(j)}[k]$ de l'ensemble $S_{i(j)}$, un score basé sur sa proximité sémantique avec les autres sens associés aux mots de son contexte dans leurs domaines respectifs. Le concept $S_{i(j)}[k]$ ayant le plus grand score est alors retenu comme sens adéquat pour le mot m_i dans son contexte. Formellement:

$$S_{i(j)}[k] = Arg\max\left(\sum_{\substack{l \mid m_l \in \zeta_{G_i} \\ l \neq i}} \sum_{1 \leq n \leq \left|S_{l(m)}\right|} sim\left(S_{i(j)}[k], S_{l(m)}[n]\right)\right). \quad (3)$$

Où $sim\left(S_{i(j)}[k], S_{l(m)}[n]\right)$ est la similarité sémantique entre les concepts $S_{i(j)}[k]$ et $S_{l(m)}[n]$ calculée sur la base de la mesure de Resnik [17].

L'index sémantique du document est ainsi constitué de l'ensemble des concepts identifiés.

3.3 Pondération des Concepts

Nous avons dans un premier temps choisi de pondérer les concepts avec un schéma de pondération classique (les schémas de pondération *tf*idf* et *Okapi-BM25* [18] ont été testés).

4 Evaluation Expérimentale

4.1 Collection de Test

Vu la complexité des calculs induits par les méthodes d'identification des termes d'index, de désambiguïsation de concepts inhérentes à notre approche, nous avons mené nos expérimentations sur un sous-ensemble de la collection TIME. La collection TIME est une petite collection composée de 425 documents issus d'articles de presse du magazine Time de 1963. Elle propose en outre un nombre suffisamment important de requêtes (83) et des jugements de pertinence. Le sous-ensemble de la collection TIME que nous avons utilisé se compose de 260 documents, et de 10 requêtes ainsi que des jugements de pertinence associés.

4.2 Protocole d'Evaluation

Nous avons évalué notre approche sur un système de RI basé sur le modèle vectoriel. Dans ce système, que nous avons implémenté, les index des documents et requêtes, assimilés à des vecteurs de termes pondérés, sont comparés à travers la mesure du cosinus classiquement utilisée dans le modèle vectoriel. L'évaluation est faite selon le protocole TREC. Pour chaque requête, les 1000 premiers documents restitués par le système sont examinés, et les précisions *P@x* à différents points *x* (*x* = 5, 10, 20, 50, 100, 200, 500, 1000) ainsi que la précision moyenne *Avg_P* sont calculées. La précision au point *x*, P@x, est le ratio des documents pertinents parmi les *x* premiers documents restitués. *Avg_P* est la moyenne des *P@x*, *x*=5..1000. Il s'agit ensuite de comparer les résultats obtenus à partir de notre approche à ceux restitués par un système de référence (ou baseline).

Dans nos expérimentations, nous avons considéré deux baselines : la première (notée TF_IDF) correspond à une indexation classique basée sur les mots clés pondérés par *tf*idf*, la seconde (notée BM25) correspond à une indexation classique basée sur les mots clés pondérés par Okapi-BM25.

4.3 Evaluation de l'Approche d'Indexation Sémantique

Nous avons mené plusieurs expérimentations pour évaluer l'impact de l'index sémantique sur l'efficacité de la recherche. Pour atteindre ces objectifs, nous avons

comparé aux résultats issus des baselines TF_IDF et BM25, les résultats issus des index suivants:

(1) index *Sem_TF_IDF*, index sémantique issu de notre approche (ce dernier est composé des concepts et des mots orphelins), pondéré par *tf*idf*.

(2) index *Sem_BM25*, notre index sémantique pondéré par *Okapi-BM25*.

(3) index (*Sem+Bas)_TF_IDF*, index sémantique issu de notre approche combiné avec les mots clés issus de l'index classique (ceci permettrait de compenser les erreurs de désambiguïsation), pondéré par *tf*idf*.

(4) index (*Sem+Bas)_BM25*, index sémantique issu de notre approche combiné avec les mots clés issus de l'index classique, pondéré par *Okapi-BM25*.

Evaluation de l'approche d'indexation par les concepts. Les comparaisons réalisées entre les baselines et notre index sémantique sont représentées à travers les graphiques suivants (Fig. 1):

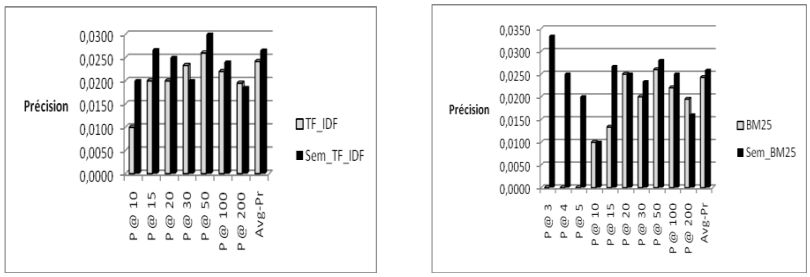


Fig. 1. (a) TF_IDF *vs* Sem-TF_IDF (b) BM25 *vs* Sem-BM25

Notre index sémantique, pondéré respectivement par *tf*idf* puis par *Okapi-BM25* présente de meilleurs résultats que les baselines respectives TF-IDF et BM25. De plus, l'index Sem-BM25 présente de meilleurs résultats que Sem-TF_IDF.

Evaluation de l'approche d'indexation par l'index combiné {concepts + mots clés}. L'idée à travers ces expérimentations est de tester l'apport de notre index sémantique par rapport à l'index classique. La combinaison des deux index apporterait-elle de meilleurs résultats? Pour répondre à cette question, les comparaisons représentées en Fig. 2 suivante ont été réalisées.

Les graphiques montrent clairement que les résultats de l'index combiné (Sem+Bas) (index sémantique+index classique) sont meilleurs que ceux d'un index classique, tant avec la pondération *tf*idf* qu'avec la pondération *Okapi-BM25*.

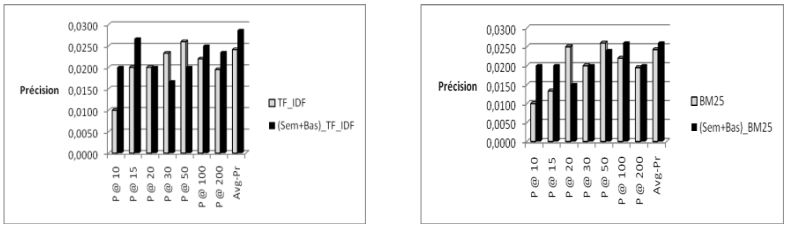


Fig. 2. (a) TF_IDF *vs* (Sem+Bas)_TF_IDF (b) BM25 *vs* (Sem+Bas)_BM25

Néanmoins, comme le montre la figure 3 suivante, l'index combiné (Sem+Bas) est globalement plus performant que l'index sémantique. Ceci peut en particulier s'interpréter par le fait que la désambiguïsation (dans l'index sémantique), pas toujours précise (des sens peuvent avoir été mal désambiguïsés), peut avoir éliminé certains termes et par conséquents les documents (peut être pertinents) correspondants. Tandis que l'index (Sem+Bas) a permis de les réintroduire par le biais des mots clés ajoutés à l'index (ce qui a pour effet d'augmenter la précision).

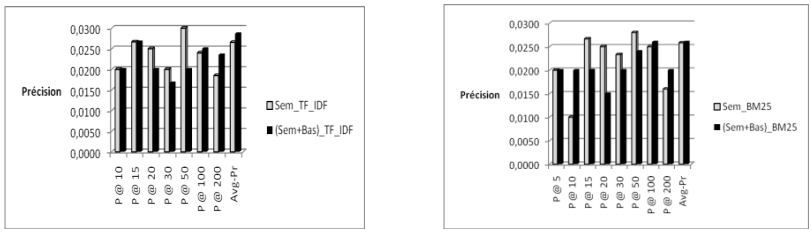


Fig. 3. (a) Sem_TF_IDF *vs* (Sem+Bas)_TF_IDF (b) Sem_BM25 *vs* (Sem+Bas)_BM25

Il en ressort de ces expérimentations que notre index sémantique, combiné ou non aux mots clés, est d'un apport certain dans l'amélioration des résultats de la recherche.

5 Travaux Connexes

L'indexation sémantique basée sur les concepts a été utilisée en RI dans le but d'enrichir les représentations des documents et requête, soit en remplacement de la représentation classique basée mots-clés [21], [15], soit en combinaison avec cette dernière [16], [1], [6]. L'idée est que la représentation des documents et requêtes par les concepts (avec ou en remplacement des mots-clés) permet d'avoir un modèle de

recherche moins dépendant des termes spécifiques utilisés [19]. En effet, dans un tel modèle, des documents pertinents peuvent être retrouvés même s'ils ne partagent aucun terme avec la requête (ce qui permet de pallier au problème de synonymie [6]), et des documents non pertinents qui partagent des termes avec la requête ne sont pas retrouvés (palliant ainsi au problème de polysémie).

Le potentiel de l'indexation sémantique basée-concepts à effectivement pallier aux limitations de l'indexation classique et à améliorer les performances de la recherche a été exploré par de nombreux chercheurs en RI. Les conclusions rapportées stipulent que l'indexation basée sur les concepts seuls n'est pas toujours concluante, tandis que l'indexation combinée {concepts+mots-clés} est performante. Ainsi, l'approche d'indexation par les sens (synsets) de WordNet introduite dans [21], expérimentée sur six collections de test a montré pour chacune une nette diminution des performances de la recherche dans le cas d'utilisation des collections désambiguïsées par rapport aux mêmes collections non désambiguïsées. La raison invoquée est une désambiguïsation imprécise des sens des mots. Cependant, même avec une désambiguïsation acceptable (taux de précision de 60%), l'approche de [20] n'a apporté aucune amélioration des performances de la recherche par rapport à l'approche classique basée mots-clés. Par ailleurs, des gains de performances de l'indexation sémantique par rapport à l'indexation classique basée mots clés sont rapportés dans [7], [10], [5], [3]. Dans [7], l'indexation par les synsets a apporté un gain de performance de 29% par rapport à l'indexation classique. Dans [10], les auteurs ont rapporté que leur modèle d'indexation par les concepts issus d'une ontologie de domaine, comparé à un modèle vectoriel classique, assurait un haut degré de précision et de rappel (de l'ordre de 90% chacun). Dans l'approche [3] basée sur l'indexation sémantique par les synsets de WordNet, les résultats de la recherche sur la collection Muchmore[2] ont apporté des gains de précision de l'ordre de 50% par rapport à l'indexation basée sur les mots clés. Par ailleurs, l'indexation combinée concepts+ mots-clés a produit de meilleurs résultats que l'approche basée concepts. Le gain de performances étant de l'ordre de 20% par rapport à l'indexation par les concepts seuls, et de plus de 70% par rapport à une indexation classique basée mots-clés. Dans [9], les auteurs comparent les performances de recherche de trois types d'approches d'indexation automatique : l'approche basée concepts (issue du système de RI conceptuelle SAPHIRE [8]), l'approche basée mots clés et l'approche combinée mots+concepts. Les tests ont été réalisés sur trois collections de test issues de bases de données médicales. Les résultats rapportés ont montré que l'indexation conceptuelle n'a apporté aucune amélioration des performances de la RI par rapport à une indexation basée mots-clés, tandis que l'indexation combinée mots-clés+concepts a produit des résultats équivalents à ceux d'une indexation classique. A contrario, dans [2], les auteurs rapportent que leur approche d'indexation par les synstes de WordNet combinés aux mots-clés apportait un gain de performance de 26% par rapport à l'indexation classique basée mots-clés.

A l'instar de ces approches, notre approche évaluée dans ce papier apporte également un gain de performances par rapport à l'indexation classique aussi bien quand les sens sont utilisés seuls que lorsqu'ils sont combinés aux mots-clés dans la représentation sémantique des documents et requêtes. Notre approche a été testée sur

[2] http://muchmore.dfki.de/about.html

un sous-ensemble de la collection TIME. Les résultats obtenus sont comparés relativement à ceux de deux baselines: la première basée mots-clés pondérés par tf-idf, et la seconde basée mots-clés pondérés par Okapi-BM25. Ces résultats montrent que l'approche basée sur les synsets seuls apporte une amélioration de 9,5% par rapport à la baseline {mots-clés + tf-idf}, et une amélioration de 6,2% par rapport à la baseline {mots-clés + Okapi-BM25}. Par ailleurs, l'approche combinée concepts+mots-clés produit de meilleurs résultats comparativement à l'approche d'indexation classique avec des gains de performance de 18,2% et de 7% respectivement par rapport à la baseline {mots-clés + tf-idf} et à la baseline {mots-clés + Okapi-BM25}.

6 Conclusion

Nous avons présenté dans ce papier, une approche d'indexation sémantique basée sur l'utilisation conjointe de WordNet et de WordNetDomains. Notre contribution a principalement porté sur la proposition d'une approche d'identification des concepts et leur utilisation comme termes d'index. Les résultats de l'évaluation de l'approche, sur une petite collection, ont montré son efficacité par rapport à des approches classiques basées mots-clés. Des travaux sont en cours en vue de sa validation expérimentale à plus grande échelle.

Bibliographie

1. Bast, H., Chitea, A., Suchanek, F. and Weber, I.: Ester: Efficient search on text, entities, and relations. In *SIGIR*, 2007. 671–678.
2. Baziz, M., Boughanem, M., Aussenac-Gilles N. A: Conceptual Indexing Approach based on Document Content Representation. Dans CoLIS5 : Fifth International Conference on Conceptions of Libraries and Information Science, Glasgow, UK, 4 juin 8 juin 2005. Lecture Notes in Computer Science LNCS Volume 3507/2005, Springer-Verlag, Berlin Heidelberg, p.171-186.
3. Boubekeur, F., Boughanem, M., Tamine, L., Daoud, M.: Using WordNet for Concept-based document indexing in information retrieval. Dans: Fourth International Conference on Semantic Processing (SEMAPRO 2010), Florence, Italy, Octobre 2010.
4. Boubekeur, F., Azzoug, W., Chiout, S., Boughanem, M. : Indexation sémantique de documents textuels. Dans: 14e Colloque International sur le Document Electronique (CIDE14), Rabat, Maroc, Décembre 2011.
5. Egozi, O., Gabrilovich, E. and Markovitch, S.: Concept-Based Feature Generation and Selection for Information Retrieval. Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008).
6. Egozi, O., Markovitch, S., and Gabrilovich, E.: Concept-Based Information Retrieval using Explicit Semantic Analysis. ACM Transactions on Information Systems, Volume 29 Issue 2, April 2011.
7. Gonzalo, J., Verdejo, F., Chugur, I. and Cigarrin, J.: Indexing with wordnet synsets can improve text retrieval. In *COLING/ACL Workshop on Usage of WordNet for NLP*. 1998.

8. Hersh, W.R, Hickam, D.H., Haynes, R.B., McKibbon, K.A.: Evaluation of SAPHIRE: A Concept-Based Approach to Information Retrieval. SCAMC 15. 1991; 808-812.
9. Hersh, W.R, Hickam, D.H., and Leone, T.J.: Words, concepts or Both: Optimal Indexing Units for Automated Information Retrieval. Proc 16th Annu Symp Comput Appl Med Care 1992:644–8.
10. Khan, L.R., Mc Leod, D., and Hovy, E.: Retrieval effectiveness of an ontology-based model for information selection. In The VLDB Journal (2004)13, pp. 71–85.
11. Kolte, S. G., Bhirud, S. G.: Word Sense Disambiguation using WordNetDomains. In First International Conference on Emerging Trends in Engineering and Technology. 2008 IEEE DOI 10.1109/ICETET.2008.231
12. Lesk, M.E.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a nice cream cone. In Proceedings of the SIGDOC Conference. Toronto, 1986.
13. Lin, D.: An information-theoretic definition of similarity. In Proceedings of 15th International Conference On Machine Learning, 1998.
14. Miller, G.: WordNet: A Lexical database for English. Actes de ACM 38, pp. 39-41.
15. Ratinov, L., Roth, D. and Srikumar, V.: Conceptual search and text categorization. Technical Report UIUCDCS-R-2008-2932, UIUC, CS Dept.
16. Ravindran, D. and Gauch, S.: Exploiting hierarchical relationships in conceptual search. In *CIKM*, 238–239.2004.
17. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, Journal of Artificial Intelligence Research (JAIR), 11, 1999, (p. 95-130).
18. Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M.: Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA, November 1994.
19. Styltsvig H. B. Ontology-based information retrieval. Ph.D. thesis, Dept. Computer Science, Roskilde University, Denmark.
20. Uzuner, O., Katz, B., Yuret, D.: Word Sense Disambiguation for Information Retrieval. AAAI/IAAI 1999: pp. 985-986.
21. Voorhees, E. M.: Using WordNet to disambiguate word senses for text retrieval. Association for Computing Machinery Special Interest Group on Information Retrieval. (ACM-SIGIR-1993): 16thAnnual International Conference on Research and Development in Information Retrieval, 171–180. (1993).
22. Wu, Z. and Palmer, M.: Verb semantics and Lexical selection. Proceedings of the 32th Annual Meetings of the Association for Computational Linguistics, pp. 133-138. 1994.

Towards a Spatio-temporal Interactive Decision Support System for Epidemiological Monitoring

Coupling SOLAP and Datawarehouse

Farah Amina Zemri, Djamila Hamdadou, Karim Bouamrane
Computer Science Département, Sciences Faculty, Université
Of Oran Es-Senia, *BP 1524, El-M'Naouer, 31000, Oran, Algéria*
{zemri_farah, dzhamdadoud, kbouamranedz }@yahoo.fr

Abstract. The primary objective of this work is to provide a spatiotemporal decision support system destined to public health specialists. The aim is to best meet the needs for senior decision makers in terms of epidemiological monitoring. The proposed approach EPISOLAP allows for better riding the infectious diseases spread problem and earlier detection of epidemic outbreaks. The proposed decision model is designed to ensure a high quality of spatial queries processing with minimum of response time; all this in one decision platform with spacial datawarhouse and navigation tool SOLAP. This latter is specifically designed for the exploration of spatial data. The hybridization of two main aspects of OLAP and GIS tools is possible, which are the implementation of the multidimensional model by using cubes and hyper cubes technology and mapping applications of GIS tools.

Keywords: Geographic Information System (GIS), Spatial On Line Analysis Processing (SOLAP), Spatial System Decision Support (SDSS); Data Warehouse (DW) Public Health (PH); Epidemiological Monitoring (EM).

1. Introduction

It is possible to record on a given territory the different outbreaks of a disease that has a risky for certain populations, and to monitor development on the territory and among these populations. We can develop and test new hypotheses, in some cases, the link between the disease, its symptoms, environment and socio-economic factors which likely spread the disease. Our contribution in this study is to achieve these goals using tool integration between GIS and OLAP technology. This will make their benefits to ensure better spatiotemporal analysis and thus help, effectively the public health actors.

2. Related Works

Many works and studies were carried out in the field of spatial decision support using SOLAP technology in various application Domains (forestry and forest management, transport, health, archeology, sports, Risk management university recruitment. etc.). These studies are mainly those that are in the geomantic research center (GRC) of Laval University in Quebec in the last ten years. This new scientific and technical field including, according to a systems approach, all the means of acquiring and managing spatial data used in the production process and management the territorial information [10]. We may mention the work of [18], where the authors used an integrated SOLAP approach to develop a system for exploring space-time

interactive data bank of corporate information in collaboration with the Ministry of Transport of Quebec. This is to see the relationship between road quality and the number of road accidents for a good maintenance management. In [16], the authors have exploited a GIS dominant approach to develop an application in the field of public health by taking advantage for SOLAP technology to explore the relationship between respiratory health and the environment. Another application SOLAP in [13] is proposed. It is developed in the field of risk management of erosion potential to inform citizens about the situation, and give them the opportunity to compare and discuss some scenarios for intervention. In [17], the author develops another SOLAP application on the students to count them according to their origins, to better plan future recruitment efforts. This work was compared with the transactional approach of GIS. The result showed that the new decision-making approach is better in terms of use simplicity, timeliness and quality of display and processing spatial queries. In the work [20], a SOLAP integrated approach was developed in the field of sport to analyze the performance of athletes in relation to weather and mechanical system. This is by using a satellite positioning (GPS), exploration tools and Analysis SOLAP providing better assessment and better tracking of athletes. Increasing needs of Spatial OLAP applications called up on the search studies to fulfill specific needs. We quote for exemple Spatial OLAP 3D application used in a three-dimensional archaeological excavations to compare different batches (stratigraphic units) [14], hypermedia SOLAP in [12] where the data are complex and diverse. This tool exploits all the resources of geospatial data that provides a good documentary analysis on the subject, however, it should manage heterogeneous and distributed data. In the work of [9], the authors used mini cubes for mobile environments (PDA, mobile phones, etc..): the creation of mobile Spatial OLAP; it was proposed in [19] a Spatial OLAP integrated with communicating agents to increase the level of automating solve problems related to the heterogeneity of models of spatial data cubes. In the work of [11], a support system based on technology JMAP Spatial OLAP KHEOPS Technologies was developed by the National Institute of Public Health of Quebec to track disease of occidental Nile virus in real time. The institute has built a system fully capable of supporting the effort of monitoring, prevention and fight against this disease.

3. Elements of Epidemiological Monitoring (EM)

Epidemiological monitoring is a process of decision making. Figure 1 illustrates the technical approach adopted in the process of epidemiological monitoring as it does on the ground for the development of decision making in disease monitoring [22].

3.1 Data registration

Registration of health data is based on notifiable diseases priority.

3.2 Data Analysis

The analysis of health data is a basic process operating base that includes, on the one hand, media handling and notification, and on the other, an analysis of data supplemented by interpreting of results.

3.3Interpretation of analytical results

In this step, the social and environmental factors of the spread of the epidemic need to be found. This aspect has been studied on the socioeconomic status of patients and their living conditions, as well as the influence of these factors on the disease of tuberculosis.

3.4 Monitoring Action and Measures Taken

After the data analysis and interpretation of results, epidemiological monitoring needs to be complemented by monitoring action, which includes: retro information, establishment of means of control, prevention and control, and blocking measures of any epidemic process following stages of contagious disease eradication [7].

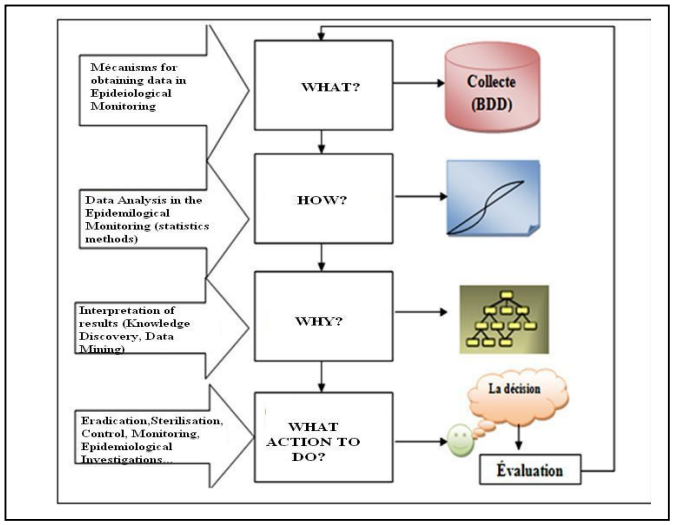


Fig 1. Process of decision making in the Epidemiological Monitoring

4. Proposed Approach

By following the general architecture of an OLAP application and choosing the integrated approach Spatial OLAP [2] appropriate to the needs for our application, we propose a decision-making approach EPISOLAP for epidemiological monitoring which to facilitate decision-making for actors in public health [21]. The system of Interactive Decision Support (DSS) proposed in the present work is based on the Spatial OLAP tool and provides as indicators of decision support tables and charts as well as the cartographic results returning the results of queries in using "hypercubes" technology. Figure 2 gives an overview of proposed EPISOLAP system.

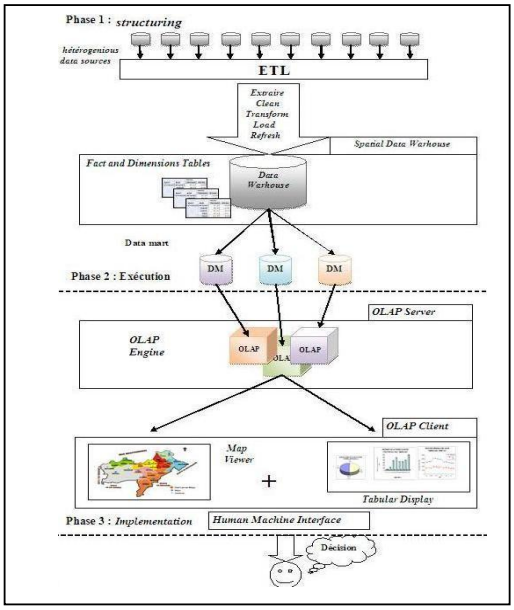


Fig 2. Decision Support System in Epidemiological Monitoring.

EPISOLAP system follows three steps (Structuration, Exploitation, Implementation) of the decision space approach proposed by Pictet [1].

4.1 Structuring phase

In this phase, the monitoring data of the various heterogeneous sources undergo a series of treatments in data warehousing process ETL (Extraction, Transformation, Loading) then is integrated into a single spatial data warehouse. The data warehouse is designed as a spatial database by UML stretched over by spatial PVL (Spatial Plug-in for Visual Languages) to support the spatial data. The different DIMENSIONS are modeled by primitives with their graphic notations (called pictogram) [1]. The relational database needs to be structured in a special format called star schema model which is derived from its configuration containing a central object, called fact table connected to a number of objects called radial dimension tables containing attributes defining each member of the dimensions [5]. The data store or data mart is a localized implementation of a data warehouse for single use, and its dimensions are chosen according to the needs for our application and desired measures figured in the client application [5].

4.2 Execution phase

The server retrieves data from SQL queries and interprets the data in a multidimensional view before presenting them to the client module. This is according to different modes of display graphics in tables, histograms or camembert or map by a

geographical information system. The OLAP engine brings together the members of DIMENSIONS according to different levels of granularity to facilitate posting on the OLAP client application. The data is precompiled and therefore prepared for display to give a power incalculable treatment of spatial queries. These different modes of representation are displayed at the final decision maker through a man-machine interface.

4.3 Implementation phase

This phase involves the interpretation of test results and the discovery of knowledge in order to facilitate the decision-making for health actors. We focused our study on the first two phases only (structuring and execution) for the collection and analysis of epidemiological monitoring data to study how these data evolve and promote the human brain through associations or correlations between phenomena studied, in order to improve the interpretation the results. Figure 3 summarizes the different steps of the two phases of structuring and operation.

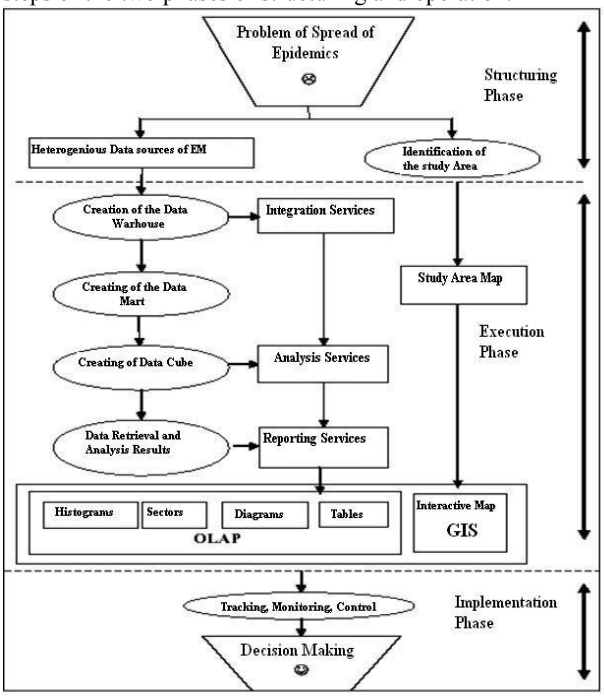


Fig 3. Phases and stages of Proposed Decision-Making Model

5. Case study

Tuberculosis disease is distinguished as a notifiable disease. It is one of the most important causes of mortality in Algeria, and economically, its treatment is so costly. Oran, which is one of the western regions of the country, still suffers from the disease tuberculosis.

5.1 Identification on the study area

In this study, the target is a group of Oran region residential areas where poor populations are concentrated. These settlements were the source of contamination for various reasons: the influx of people into the city of Oran, the rural exodus caused by economic problems and / or of security or those related to natural disasters (depopulation, earthquakes ...). The aim therefore needs to be geo-location of populated "poor" areas in order to observe the scene or the TBC epidemic could spread rapidly and widely. The identification of urban settlements and rural district allows the circumscription of the epidemic spread phenomenon.

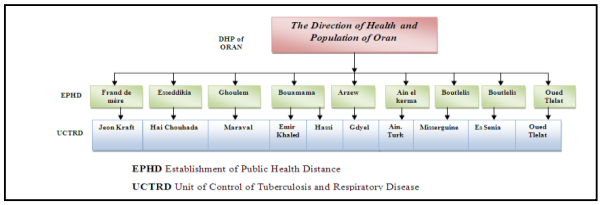


Fig 4. Health structures distributed over the region of Oran

Monitoring of this epidemic in the region of Oran is one of challenges of the Directorate of Health and Population of Oran (DHP). Implementation of Interactive Decision Support System (DSS), will allow better monitoring of the disease and its spread over the geographical map of the region of Oran while identifying areas with a foster epidemic.

5.2 Scenario of epidemiological monitoring

The patient detected in a Basic Health Unit (BHU) with the disease of tuberculosis after receiving the necessary analysis has to be transferred to the Service Control of Tuberculosis and Lung Disease (UCTLD) corresponding to his residence where he receives the necessary treatment for six months. Each UCTLD (Unit of Control of Tuberculosis and Respiratory Disease) through its own ECPH (Establishment of Close of Public Health) needs to periodically report his patients, as shown in Fig 4 to the Department of Health and Population (Oran DHP) and the Department of Epidemiology and Preventive Medicine (SEPM) eventually. He then initiates the search of infectious cases as soon as one case of tuberculosis diseas is received.

5.3 Datawarhousing process

We will try to adopt bellow, the decision-making approach for epidemiological monitoring in the proposed schema in Fig 8 on the monitoring of tuberculosis disease in the region of Oran. The informational part of our study is represented by the socio-economic and medical data of populations (patients with tuberculosis) inhabiting the study area. Each UCTRD has a data source implemented in a database of its region where all information about the patient can be found, such as treatment..... etc.

The ten data sources of ten UCTRDs needs to be integrated into a single data warehouse in Health and Population Department. This is done through a process of datawarhousing of data. As shown in the diagram of figure 5. Datawarhouse defined as a tool for collecting and storing (archiving), information processing and integration

from multiple heterogeneous sources [6] fits well with the modeling of decision information system of the region Oran.

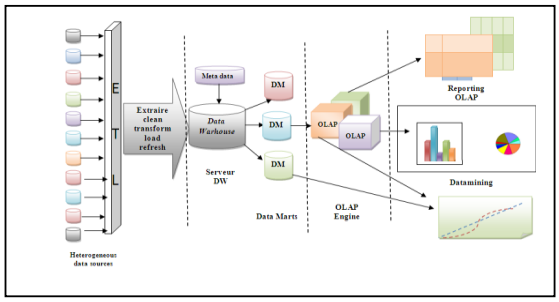


Fig 5. Datawarhousing process

After the confirmation of data by the DHP, we start a cleanup operation of this data from the ten disparate and heterogeneous sources all over across the region. In addition to this homogenization operation is performed to eliminate duplicates (this is due to find such a patient following his treatment in error in two different care structures).The next operation consists of the localization of TB cases, according to inform the address of the patient, in the municipality of his place of residence, the statistical calculation of incidence rates and detection of outbreaks epidemics, as well as the end, the exploitation of these data and their graphical and tabular through tables, graphs sectors and map display.

5.4 Design of Spatial Datawarhouse

We will identify in this part dimensions, dimension members, the facts and the measures we inted to calculate and see portrayed in the graphic display and / or mapping.

Given data confidentiality, it is not possible to have access to detailed data, i.e. those at the individual level. Our database will be designed to manage the categories of individuals and the phenomenon of epidemic spread rather than individuals themselves.

5.4.1Dimensions: Each dimension has one hierarchy.

The temporal dimension*:* with three levels.

1) Time: All-year-Quarter.

The spatial geometry dimension

2) Territory: a cartographic representation of members at each level.

- **Municipality:** is the detailed level with the basic health units (BHU).
- **Socio-Health Sector:** 10 members or ten units containing control TB and Respiratory Diseases of the region.
- **The region of Oran:** Country of Oran.

3) Thematic dimension

Sex

- **Detailed level:** man (code: M) or woman (code: W)
- **Up level**: "All".

Age

- **Detailed level:** 0, 1.2..... 89, 90 +
- **Age groups:** (0-4, 5-15, 15-49, 49-60, 60 +.)
- **Up level:** "All"

Type of the disease: based on disease location.

- **Detailed level**: pulmonary tuberculosis (Code: TP) and Extra pulmonary TB (code: TEP)
- **Up level**: "All"

5.4.2 Facts: granularity as the incidence rate of disease on each type in each BHU and for each quarter.
5.4.3 Measures: describing these facts are the **number of cases recorded** and the **average incidence rate.**

.

The diagram in Figure 6 shows the model dimensions by the Spatial PVL using the pictograms of Perceptory tool [4].

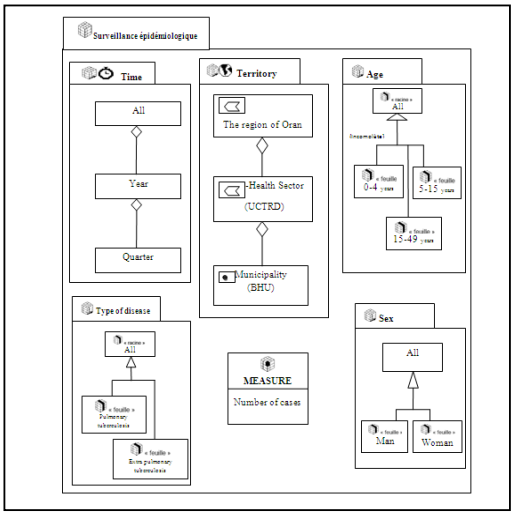


Fig 6. Using Multidimentional Pictograms with Perceptory

We have chosen the star model to design the relational database in our application, ie structuring tables in several dimension tables and one fact table as shown Figure 7.

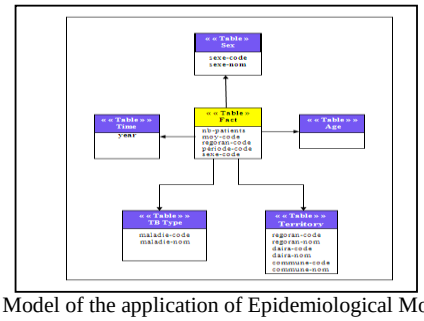


Fig 7. Star Model of the application of Epidemiological Monitoring

5.5 Spatial Data Cube

The star model will be used to structure the multidimensional model of our spatial data cube. Here is what it looks like our geospatial data cube. It was here limited to three dimensions only, but it is up to four, or possibly five dimensions.

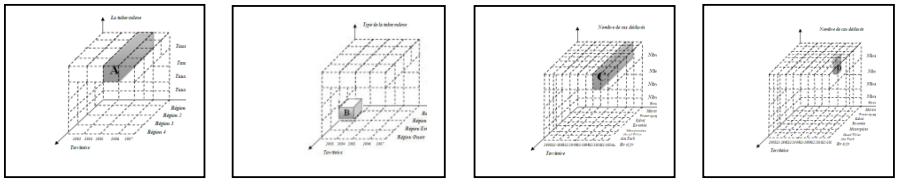


Fig 8. Illustration of Multidimentional Cube

5.6 Experimental Results

Request1

The number of recorded cases of the disease of Extra-Pulmonary Tuberculosis (EPT) in the region of Aezew in 2009 [23].

Figure 9.a Extraction Figure 9.b Transformation Figure 9.c Loading

Fig 9. The three Steps of Execution of the Query on the Region of Arzew

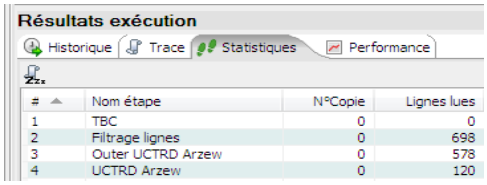


Fig 10. Result of the Query on the Region of Arzew

Request 2

We can calculate the number of cases in each UCTRD (Unit of Control of Tuberculosis and Respiratory Diseases) by following the three steps shown in Figure 9 [23].

Table 1. Results Queries Table on UCTRD

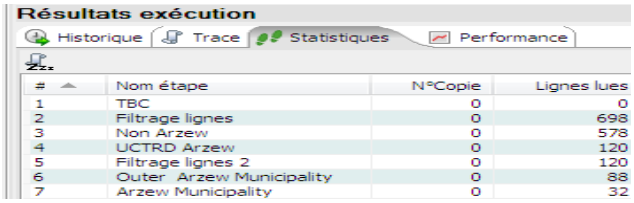
Transformations	**UCTRD**	Number of cases of TEP registred in 2009	Queries execution time (secondes)
Transformation1	AET	**28**	**1.7**
Transformation2	Arzew	**120**	**1.8**
Transformation3	Bouamama	**23**	**3.7**
Transformation4	El Amir Khaled	**89**	**2.1**
Transformation5	Es senia	**127**	**1.8**
Transformation6	Hai echouhada	**121**	**1.7**
Transformation7	Jean Kraft	**44**	**1.6**
Transformation8	Misserguine	**42**	**1.3**
Transformation9	Maraval	**73**	**1.7**
Transformation10	O.Tlelet	**15**	**1.1**

Discussion of the table

We can clearly detect the region of Es Senia, Hai Chouhada Arzew which are epidemic outbreaks with respective numbers of cases 127, 121 and 120 new cases registered in 2009. The execution time query is calculated in seconds and is almost the same for all requests. Taking the case of UCTRD of Arzew, to better monitor the EPT in this region, we continue our investigations on the disease of tuberculosis in the town of Arzew only [23].

Request 3

What is the number of cases in the town of Arzew only?



#	Nom étape	N°Copie	Lignes lues
1	TBC	0	0
2	Filtrage lignes	0	698
3	Non Arzew	0	578
4	UCTRD Arzew	0	120
5	Filtrage lignes 2	0	120
6	Outer Arzew Municipality	0	88
7	Arzew Municipality	0	32

Fig 11. Result of the query on the municipality of Arzew

Knowing that the region contains 10 municipalities of Arzew and a total with 120 cases over the entire region in Arzew, divided on 10 municipalities makes an average of 12cas/town. The number of 32 in the town of Arzew is much higher than the average then the town of Azew represents an outbreak of epidemic the whole region of Arzew [23].

6. Conclusion

SOLAP technology is an important means for access to relevant information; it offers the ease of use and speed of treatment and execution of applications by the solution adopted. It is also destined to non-computer users who do not have knowledge of the language of SQL Inquiring. This decision tool is a real help to quickly respond to various questions of the actors in public health.

7. Bibliography

1. Yvan Bédard, Méthodes et outils pour l'exploitation des données géospatiales. volet 10: Modélisation des bases de données géospatiales Centre de recherche en géomatique Université Laval, Québec.10-11 juin 2002, Tunis.
2. Yvan Bédard, Notions avancées de bases de données SIG. Volet 8: Entrepôts de données. Centre de recherche en géomatique. Université Laval, Québec. 10-11 juin 2002. Tunis.
3. Yvan Bédard, Marie-Josée Proulx, Sonia Rivest, 2005: "Enrichissement du OLAP pour l'analyse géographique : exemples de réalisations et différentes possibilités technologiques, " Soumis à la première journée francophone sur les entrepôts de données et analyse en ligne, Lyon, 10 juin 2005.
4. Yvan Bédard, Suzie Larrivée , Présentation de Perceptory: les extensions spatiales.departement de sciences géomatiques.université LAVAL de Quebec.Canada.2009
5. Yvan Bédard, Notions avancées de bases de données SIG : OLAP et SOLAP, Centre de recherche en géomatique Université Laval, Québec.Canada.2009.
6. Boussaid. O, Introduction aux systèmes d'information décisionnels. Cours de master recherche ECD.Laboratoire ERIC- université de Lyon 2, France, 2010.
7. Bouziani Mustapha. Les pathologies infectieuses : Aspects épidémiologiques et prophylactique ; 2000.
8. Direction de la Prévention Institut National de Santé Publique, Manuel de la lutte antituberculeuse à l'usage des personnels medicaux–Programme National de Lutte Contre la Tuberculose.Edition 2007.
9. Etienne Dubé, Thierry Badard, Yvan Bedard , Service Web de constitution en temps réel de mini-cubes SOLAP pour clients mobiles.une architecture orientée services pour l'utilisation mobile des données géo-decisionnelles. Centre de recherche en

Géomatique(CRG), haire de recherche industrielle en base de données géospatiales décisionnelles.SAGEO'2007.

10. P. Gagnon et D.J. Coleman: CISM Journal ACSGC Vol. 44, No 4,p. 383-389,1990.
11. Germain Lebel, M.A,M. Sc, surveillance du virus du Nil occidental.Instutut national de snté publique du Quebec. Version 2009.
12. Rosemarie McHugh, Francis Bilodeau, Sonia Rivest,Yvan Bédard, Michel Michaud Analyse du potentiel d'une application SOLAP pour une gestion efficace de l'érosion des berges en Gaspésie Îles-de-la-Madeleine. Centre de recherche en Géomatique(CRG), Département des sciences géomatiques, université laval, Quebec, chaire de recherche industrielle en base de données géospatiales décisionnelles.2006
13. Rosemarie McHugh 1,2 Stéphane Roche 1,Yvan Bédard 1,2Vers une solution SOLAP comme outil participatif. Centre de Recherche en Géomatique, Chaire de recherche industrielle CRSNG en bases de données géospatiales décisionnelles, Université Laval, Pavillon Casault, G1K 7P4 Québec (Qc) Canada.SAGEO, 2007
14. Miquel, M., Y. Bédard & A. Brisebois, 2002, Conception d'entrepôts de données géospatiales à partir de sources hétérogènes, exemple d'application en foresterie, Ingénierie des Systèmes d'information,Vol. 7, No. 3, p. 89-111.
15. J.Pictet, « Dépasser l'évaluation environnementale, procédure d'étude et insertion dans la décision globale». Collection Meta, Presses Polytechnique et Universitaires Romandes,Lausanne, Suisse,1996.
16. Proulx M-J, Y.Bédard, M. Nadeau, P. Gosselin & G. Lebel, Géomantique et santé environnementale résultant du projet ICEM/SE. *Géomatique 2002, 30-oct., Montréal,Qc, Canada.* Martin Nadeau1, Sonia Rivest1, Yvan Bédard1Pierre Gosselin2 et Germain Lebel2 Centre de recherche en géomatiqueUniversité Laval Sainte-Foy (Québec) Canada.
17. Proulx, M.J. & Y. Bédard. Le potentiel de l'approche multidimensionnelle pour l'analyse de données géospatiales en comparaison avec l'approche transactionnelle des SIG., Colloque Géomatique 2004 - Un choix stratégique! Montréal, 27-28 octobre., Montréal, Canada
18. Rivest, S., P. Gignac, J. Charron & Y. Bédard, Développement d'un système d'exploration spatio-temporelle interactive des données de la Banque d'information corporative du ministère des Transports du Quebec. Colloque Géomatique 2004 - Un choix stratégique! Montréal, 27-28 octobre., Montréal, Canada
19. Tarek Sboui*, Mehrdad Salehi**, Yvan Bédard***, Sonia Rivest Catégorisation des problèmes d'intégration des modèles des cubes de données spatiales.Fouille de données complexes dans un processus d'extraction des connaissances.8èmes Journées Francophones. Extraction et Gestion des Connaissances. Sophia Antipolis 29 janvier 2008.
20. Veilleux, J-P, M. Lambert, R. Santerre, & Y. Bédard, 2004, Utilisation du système de positionnement par satellites (GPS) et des outils d'exploration et d'analyse SOLAP pour l'évaluation et le suivi de sprotifs de haut niveau, Colloque Géomatique 2004 - Un choix stratégique! Montréal, 27-28 octobre. Montréal, Canada.
21. Zemri Farah Amina, Hamdadou Djamila, Bouamrane Karim. Vers un outil d'aide à la décision spatio-temporelle « Spatial On Line Analysis Processing » : Application dans la surveillance des épidémies. 2emes Doctoriales STIC'11 ; université de Tebessa.20 et 21 avril 2011.
22. Zemri Farah Amina, Hamdadou Djamila, Bouamrane Karim. Towards an Spatio-temporal Interactive Decision Support System for the Epidemiological Monitoring: Coupling SOLAP and Datawarehouse. 2nd Annual European Decision Science Institute EDSI'11 Conference Oestrich-Winkel, Germany. June 24-25, 2011.
23. Zemri Farah Amina, Hamdadou Djamila, Bouamrane Karim. Couplage SOLAP & Datawarhouse: Outil interactif d'aide à la décision spatio-temporelle. épidémies Application dans la surveillance des épidemies. 1eres Doctoriales JDLIO'11 ; université d'Oran.31 Mai et 1 juin 2011.

Optimisation III

Un schéma numérique d'optimisation globale unidimensionnelle des fonctions non convexes et applications

Mohamed Rahal et Abdelkader Ziadi
Laboratoire de Mathématiques Fondamentales et Numérique
Département de Mathématiques, Université Ferhat Abbas, Sétif.

Résumé Dans ce travail, on présente un algorithme pour résoudre un problème de minimisation globale unidimensionnelle sans contrainte de type $\min_{x\in[a,b]} f(x)$ où f une fonction continue non différentiable et non lipschitzienne. L'algorithme présenté est basé sur le recouvrement de l'intervalle faisable par un ensemble fini d'intervalles. Les centres des intervalles construit, forment une suite récurrentes de type $x_{k+1} = x_k + r_k$ qui converge vers le minimiseur global de f que l'on cherche. L'avantage de ce schéma c'est qu'il ne dépend d'aucun calcul auxiliaires, à part des informations de la fonction objectif sur $[a, b]$. On propose une modification de cet algorithme pour cette classe de fonctions. Enfin, on donne des exemples numériques et des applications sur la résolution numérique des systèmes d'équations algébriques non linéaires.

Mots clés: Optimisation globale sans contraintes, Fonction non convexe, Méthode de recouvrement

1 Introduction

Considérons le problème de minimisation globale unidimensionnelle:

$$(\mathbf{MGU}) \quad \begin{cases} \min f(x) \\ t.q.\ x \in [a, b] \end{cases}$$

où f une fonction réelle continue dans l'intervalle $[a, b]$ de $\mathbb{R}$.
Les problèmes d'optimisation globale sont présents dans plusieurs domaines de recherche en mathématiques et en technologie. L'existence de multiples minima locaux d'une fonction objectif non convexe rend les problèmes

d'optimisation globale très difficile à résoudre. Le problème d'optimisation globale (**MGU**) est de trouver $x^* \in [a,b]$ tel que $f(x^*) \leq f(x)$ pour tout $x \in [a,b]$. Notons que la continuité de f sur le compact $[a,b]$ garanti l'existence du minimum global de f sur $[a,b]$. Dans tout ce travail, on suppose que f est donnée sous l'hypothèse suivante:

Hypothèse. Soit la fonction $f(x)$ définie dans l'intervalle $[a,b]$ de $\mathbb{R}$, f est dite höldérienne dans $[a,b]$, s'il existe des constantes $h = h(f,[a,b]) > 0$ et $0 < \alpha < 1$ telle que

$$|f(x) - f(y)| \leq h\,|x-y|^{\alpha}\,, \text{ pour chaque } x, y \in [a,b]\,. \tag{1.1}$$

Propriétés

• Evidemment, une fonction höldérienne f est lipschitzienne sur $[a,b]$ quand $\alpha = 1$, les fonctions lipschitziennes sont une classe spéciale des fonctions höldérienness.

• Si $f(x)$ est une fonction höldérienne de constantes $h > 0$ et $0 < \alpha < 1$ sur $[a,b]$, alors elle est de même pour toute constante $h^{'} > h$ et $\alpha^{'} < \alpha$ sur $[a,b]$.

• Bien que les fonctions höldériennes sont continues, elle peuvent être non-différentiables. Intuitivement, les fonctions höldériennes telles que α soit assez petit sont beaucoup plus irrégulières que celles où α est assez grand (ce qui explique le fait que $\alpha < 1$, car c'est le cas non-différentiable qui nous intéresse).

Soit $\varepsilon > 0$ la précision exigée avec laquelle le minimum global est atteint.
De nombreuses approches utilisant une fonction auxiliaire ont été proposé pour trouver un minimiseur global de problèmes d'optimisation globale continue [3-7].
Dans le cas unidimensionnel, Gourdin et al.[4] ont proposés une extension de la méthode de Piyavskii pour la minimisation globale des fonctions höldériennes où les paramètres h et α sont connues à priori. Pour cela, ils ont utilisé la technique Branch and Bound qui est basée sur la construction des sous estimateurs paraboliques par morceaux. A chaque itération, la méthode nécéssite la détermination du minimiseur global d'une fonction unimodale convexe mais non différentiable. Cela revient à résoudre une équation algébrique non linéaire et non dérivable. L'algorithme qu'on va présenté dans ce travail, évite la résolution d'une équation algébrique non linéaire. C'est un algorithme itératif de recouvrement non-uniforme inspirés par Yu. G. Evtushenko [1], [2] pour les fonctions lipschitziennes. L'algorithme est alors simple et à l'avantage d'éviter l'utilisation des calculs auxiliaires ce qui permet de réduire considérablement le temps de calcul. La convergence de

l'algorithme est également étudiée et des expériences numériques réalisées sur des fonctions de test montrent l'efficacité de l'algorithme.

2 L'idée générale des méthodes de recouvrement

Soit $[a,b]_*$ l'intervalle des solutions du problème (**MGU**); f_* le minimum global de la fonction objectif $f(x)$. Introduisons l'intervalle des solutions ε-approximées du problème (**MGU**):

$$[a,b]_*^{\varepsilon} = \{x \in [a,b] : \ f(x) \leq f_* + \varepsilon\}. \tag{1.2}$$

Evidemment, $[a,b]_* \subset [a,b]_*^{\varepsilon} \subset [a,b]$. Dans la plupart des problèmes pratiques, il suffit de trouver au moins un point $x_{opt} \in [a,b]_*^{\varepsilon}$ et de prendre la valeur $f_{opt} = f(x_{opt})$ comme une valeur optimale de f_*. D'autre part, il est nécessaire de déterminer la valeur du minimum global avec la précision requise ε et de trouver au moins un point x_{opt} où cette valeur approchée est obtenue.
Considérons l'ensemble unidimensionnel des points $x_1, x_2, ..., x_k$, où chaque point $x_i \in [a,b]$. A chaque point x_i, on cherche la valeur de f et on cherche le minimum record R_i comme:

$$R_i = \min_{1 \leq j \leq i} f(x_j) = f(x_l), \quad 1 \leq l \leq i. \tag{1.3}$$

Aussi, on cherche un des minimiseurs record x_l. Il résulte de (1.3) que la suite R_i est décroissante. Avec chaque point x_i, nous associons son voisinage $[a,b]_i$ tel que $x_i \in [a,b]_i$.
Considérons les ensembles suivant X_i et leur unions:

$$X_i = \{x : x_i \in [a,b]_i, \ \ f(x) \geq R_i - \varepsilon\}, 1 \leq i \leq k, \qquad V_k = \bigcup_{i=1}^{k} X_i \tag{1.4}$$

La collection des ensembles $X_1, X_2, ..., X_k$ couvre l'intervalle faisable $[a,b]$ si:

$$[a,b] \subset V_k. \tag{1.5}$$

Théorème 1.1. *Supposons que l'ensemble des points admissibles $x_1, x_2, ..., x_k$ et la collection corespondante d'ensembles $X_1, X_2, ..., X_k$ satisfont la condition* (1.5). Alors, *le minimiseur record x_r trouvé de la condition $R_k = f(x_r)$, appartient à l'ensemble $[a,b]_*^{\varepsilon}$.*

Ce théorème exprime l'idée principale de la méthode de recouvrement non-uniforme: au lieu de trouver le minimum global dans $[a,b]$, on trouve le

minimum global sur les sous intervalles dont l'union contient $[a,b]$. La condition (1.4) peuve facilement être vérifiée par la construction des bornes inférieures de $f(x)$ sur l'intervalle $[a,b]_i$ ou X_i. Des minorants de la fonction f satisfaisant la condition (1.1) sont présentés dans la section suivante.

3 Minimisation des fonctions höldériennes

La partie la plus importante dans l'algorithme de recouvrement non-uniforme est la détermination d'au moins une certaine partie de l'ensemble X_i.
Pour construire un recouvrement de l'intervalle $[a,b]$, il faut savoir construire des minorants de f.
Supposons que la fonction f satisfait la condition (1.1). Par conséquent, nous obtenons le minorant de f sur l'intervalle $[a,b]$:

$$\forall\ x,y \in [a,b]\ , \qquad f(y) - h\,|x-y|^{\alpha} \leq\ f(x). \tag{1.6}$$

Utilisant ce minorant, nous prouvons le théorème suivant:

Théorème 1.2 *Soit f une fonction höldérienne de paramètres $h > 0$ et $0 < \alpha < 1$, définie sur un intervalle $[a,b]$ de $\mathbb{R}$. Supposons que f soit évaluée aux points $x_1, x_2, ..., x_k$. Posons $m_k^* = \min\limits_{1\leq i\leq k}\{f(x_i)\}$ et désignons par $(I(x_i,r_{ik}))_{1\leq i\leq k}$ une famille d'intervalles de centres $\{x_i\}_{1\leq i\leq k}$ et de rayons r_{ik}. Si on a $[a,b] \subset \bigcup\limits_{i=1}^{k} I(x_i,r_{ik})$ avec $r_{ik} = \left(\frac{f(x_i)-m_k^*+\varepsilon}{h}\right)^{\frac{1}{\alpha}}$ alors m_k^* est le minimum global, à ϵ près, de f sur $[a,b]$.*

Preuve. D'après l'inégalité (1.6) et pour $y \in [a,b]$ fixé, si un certain x vérifie

$$m_k^* - \varepsilon \leq\ f(y) - h\,|x-y|^{\alpha}\,, \tag{1.7}$$

alors $m_k^* - \varepsilon \leq f(x)$. On considère les intervalles

$$I(x_i,r_{ik}) = \{x \in \mathbb{R},\ \ |x-x_i| \leq r_{ik}\}\,;$$

de l'inégalité (1.7) on a $|x-y| \leq \left(\frac{f(y)-m_k^*+\varepsilon}{h}\right)^{\frac{1}{\alpha}}$,

et pour $y = x_i$, on doit donc avoir les rayons r_{ik} donnés par $r_{ik} = \left(\frac{f(x_i)-m_k^*+\varepsilon}{h}\right)^{\frac{1}{\alpha}}$.
On peut montrer facilement que pour tout $i = 1, ..., k$ et pour tout $x \in I(x_i,r_{ik})$ on a $m_k^* - \varepsilon \leq f(x)$. Soit maintenant $M = \min\limits_{x\in[a,b]} f(x) = f(x_0)$,

$x_o \in [a,b]$, on a $x_0 \in \bigcup_{i=1}^{k} I(x_i, r_{ik})$, donc il existe $i_0 \in \{1,2,...,k\}$ tel que $x_0 \in I(x_{i_0}, \left(\frac{f(x_{i_0})-m_k^*+\varepsilon}{h}\right)^{\frac{1}{\alpha}})$ par conséquent

$$|x_{i_0} - x_0| \leq \left(\frac{f(x_{i_0}) - m_k^* + \varepsilon}{h}\right)^{\frac{1}{\alpha}},$$

donc

$$|x_{i_0} - x_0|^{\alpha} \leq \frac{f(x_{i_0}) - m_k^* + \varepsilon}{h}. \tag{1.8}$$

Puisque f est höldérienne on a $|f(x_{i_0}) - f(x_0)| \leq h\,|x_{i_0} - x_0|^{\alpha}$,
d'après (1.8) on a $|f(x_{i_0}) - f(x_0)| \leq f(x_{i_0}) - m_k^* + \varepsilon$.
D'où $m_k^* - M \leq \varepsilon$. On déduit que m_k^* est un minimum global de f.
Donc si la réunion des intervalles $I(x_i, r_{ik})$ ne couvre pas $[a,b]$, le minimum global peut être atteint dans $[a,b] \backslash \bigcup_{i=1}^{k} I(x_i, r_{ik})$. Par conséquent, les intervalles $I(x_i, r_{ik})$ peuvent être omis de l'ensemble faisable $[a,b]$; on cherche la solution dans la partie restante. Le problème (**MGU**) aura une solution lorsque la réunion des intervalles couvre complètement $[a,b]$.
La représentation donnée suggére une méthode constructive pour résoudre le problème (**MGU**). En résumé, la méthode consiste à procéder ainsi: supposons que pour une certaine suite de points $\{x_k\}$ le record m_k^* est déterminé par $m_k^* = \min\limits_{1 \leq i \leq k} \{f(x_i)\}$. La suite des points x_i et les rayons r_{ik} de l'intervalle sont stockés dans une mémoire. Si au nouveau point x_{k+1} on a $f(x_{k+1}) < m_k^*$, on pose $m_{k+1}^* = f(x_{k+1})$ et on remplace le terme r_{ik} par r_{ik+1}. Si les intervalles I_{ik+1} recouvrent l'intervalle $[a,b]$, alors le calcul sera stoppé; sinon, on prend un nouveau point x_{k+2} et on continue. L'ensemble $[a,b]$ est recouvert par des intervalles de différents rayons (non-uniformisé). Considérons un point x_i auquel $m_i^* = f(x_i)$, $1 \leq i \leq k$, on a évidemment $f(x_i) = m_i^* \geq m_k^*$ d'où,

$$r_{ik} = \left(\frac{f(x_i) - f_{\varepsilon} + \varepsilon}{h}\right)^{\frac{1}{\alpha}} \geq \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}.$$

Donc le plus petit rayon est au point x_i auquel $m_i^* = f(x_i)$ i.e., $\left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}$. On prend donc $x_1 = a + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}$, parce qu'à l'initialisation $f(x_1) = m_1^*$. Avec cette valeur de x_1, on gagne du temps et on est sûr de ne pas avoir ignoré le minimum global au voisinage de ce point. En effet, si

$$x^* = \arg\min_{x \in [a,b]} f(x) \in [a, a + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}]$$

alors:

$$|f(x_1) - f(x^*)| \leq h\,|x_1 - x^*|^{\alpha} \leq \varepsilon.$$

En général, pour $i \geq 1$, la suite itérative $(x_i\ )$ est définie par le schéma suivant:

$$x_{i+1} = x_i + \left(\frac{f(x_i) - f_\varepsilon + \varepsilon}{h}\right)^{\frac{1}{\alpha}} + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}.$$

Ce choix nous permet de ne pas rater le minimum global de f, car les intervalles I_i se rencontrent. On arrête quand k vérifie $[a,b] \subset \overset{k}{\underset{i=1}{\cup}} I(x_i, r_{ik})$.
Si $x_k < b$ et $x_k + r_{kk} \geq b$, alors le dernier point de la suite est x_k.
Si $x_k + r_{kk} < b$ et $x_k + r_{kk} + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}} \geq b$ le dernier point de la suite sera $x_{k+1} = b$.

Algorithme
1. Initialisation.
Poser $k = 1,\ x_1 = a + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}},\ x_\varepsilon = x_1,\ f_\varepsilon = f(x_\varepsilon)$
2. Etapes $k = 2, 3, ...$
Poser $x_{k+1} = x_k + \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}} + \left(\frac{f(x_k) - f_\varepsilon + \varepsilon}{h}\right)^{\frac{1}{\alpha}}$
Si $x_{k+1} > b - \left(\frac{\varepsilon}{h}\right)^{\frac{1}{\alpha}}$, alors stop.
Sinon, déterminer $f(x_{k+1})$.
Si $f(x_{k+1}) < f_\varepsilon$, alors poser $x_\varepsilon = x_{k+1}, f_\varepsilon = f(x_{k+1})$
Poser $k = k + 1$ et aller à 2.

4 Modification de l'algorithme non-uniforme

Dans [1], [2], plusieurs modifications ont été proposées. On propose dans ce travail une modification pour le cas des fonctions höldériennes. Pour cela on donne le résultat suivant:

Théorème 1.3 *Soit f une fonction höldérienne de paramètres $h > 0$ et $0 < \alpha < 1$, définie sur un intervalle $[a,b]$ de $\mathbb{R}$. Supposons que f soit évaluée aux points $x_1, x_2, ..., x_k$. Posons $m_k^* = \min\limits_{1 \leq i \leq k} \{f(x_i)\}$ et désignons par $(I(x_i, r_{ik}))_{1 \leq i \leq k}$ une famille d'intervalles de centres $\{x_i\}_{1 \leq i \leq k}$ et de rayons $r_{ik} = \frac{f(x_i) - m_k^* + \frac{\varepsilon}{2}}{C_{\varepsilon/2}}$ où*

$$C_\varepsilon = \begin{cases} \dfrac{\alpha h^{\frac{1}{\alpha}}}{\varepsilon^{\frac{1-\alpha}{\alpha}}} & \text{pour} \quad \alpha \in \left\{\frac{1}{m}, m \in \mathbb{N}^* \setminus \{1\}\right\} \\ \dfrac{\alpha h^{\frac{\alpha+1}{\alpha}}}{\varepsilon^{\frac{1}{\alpha}}} & \text{pour} \quad \alpha \in]0,1[\setminus \left\{\frac{1}{m}, m \in \mathbb{N}^* \setminus \{1\}\right\}. \end{cases}$$

Si on a $[a,b] \subset \overset{k}{\underset{i=1}{\cup}} I(x_i, r_{ik})$, *alors* m_k^* *est le minimum global de* f *sur* $[a,b]$.
Donnons d'abord le résultat suivant:

Théorème 1.4 *Soit* f *une fonction höldérienne de paramètres* $h > 0$ *et* $0 < \alpha < 1$, *définie sur un intervalle* $[a,b]$ *de* $\mathbb{R}$. *Alors il existe une constante* $C_\varepsilon > 0$ telle que

$$\forall \varepsilon > 0 \text{ et } \forall\, x, y \in [a,b], \qquad |f(x) - f(y)| \leq C_\varepsilon |x - y| + \varepsilon \quad , \tag{1.9}$$

On donne le lemme suivant:

Lemme 1. Soit $\delta > 0$ et $\alpha \in]0,1[$, alors il existe une constante $k > 0$ telle que:

$$z^\alpha - kz - \delta \leq 0 \qquad \forall\, z > 0 \tag{1.10}$$

Preuve. En effet:
1) Si $\alpha = \frac{1}{m}$, $m \in \mathbb{N}^* \setminus \{1\}$, on a

$$z \leq (kz + \delta)^m$$

Le développement de cette inégalité en utilisant la formule du Binôme de Newton, nous obtenons, pour tout $k > 0$:

$$\begin{aligned} (kz+\delta)^m &= \sum_{i=0}^{m} \binom{m}{i} k^i z^i \delta^{m-i} = \delta^m + mkz\delta^{m-1} + ... + k^m z^p = \\ &= mkz\delta^{m-1} + R(z,k,\delta), \end{aligned}$$

où $\binom{m}{i} := \frac{m!}{i!(m-i)!}$ et $R(z,k,\delta) > 0$, le reste du developpement, d'où

$$(kz + \delta)^m \geq mkz\delta^{m-1},$$

et par conséquent

$$kz + \delta \geq (mk\delta^{m-1})^{\frac{1}{m}} z^{\frac{1}{m}},$$

donc

$$z^{\frac{1}{m}} \leq \frac{k}{(mk\delta^{m-1})^{\frac{1}{m}}} z + \frac{\delta}{(mk\delta^{m-1})^{\frac{1}{m}}}.$$

En prenant $k = \frac{1}{m\ \delta^{m-1}}$, on obtient l'inégalité (1.10).

2) Si $\alpha \neq \frac{1}{m}$, $m \in \mathbb{N}^* \setminus \{1\}$, $\exists\, p \in \mathbb{N}^*$ tel que

$$\frac{1}{p+1} < \alpha < \frac{1}{p} \quad ,$$

par conséquent,

$$z^{\alpha} \leq \frac{\delta^{-p}}{p+1} z + \delta \leq \alpha \delta^{-\frac{1}{\alpha}} z + \delta$$

d'où (1.10).

Preuve du théorème 1.4 De la condition (1.1) et du lemme 1, si on pose $z = |x-y|$ dans (1.10), alors il existe une constante $k > 0$ telle que

$$|x-y|^{\alpha} \leq k\,|x-y| + \delta \tag{1.11}$$

avec

$$k \quad = \begin{cases} \alpha \delta^{\frac{\alpha-1}{\alpha}} & \text{pour } \alpha \in \left\{ \frac{1}{m}, m \in \mathbb{N}^* \setminus \{1\} \right\} \\ \\ \alpha \delta^{-\frac{1}{\alpha}} \text{ pour } \alpha \in]0,1[\setminus \left\{ \frac{1}{m}, m \in \mathbb{N}^* \setminus \{1\} \right\}. \end{cases}$$

Posons $h\delta = \varepsilon$, et puisque f est höldérienne, et à partir de (1.1) et (1.11) on déduit le résultat du théorème 1.3 ■

Remarque. On peut obtenir une autre constante $C_{\varepsilon}^{'}$ plus petite que C_{ε}. En effet, prouvons le lemme suivant:

Lemme 2. Soit $h, \varepsilon > 0$ et $0 < \alpha < 1$. Considérons la fonction $g_{\alpha} :]0, \infty[\to \mathbb{R}$ défine par

$$g_{\alpha}(z) = \frac{hz^{\alpha} - \varepsilon}{z}.$$

Alors

$$\sup_{z>0} g_{\alpha}(z) = \frac{\alpha\varepsilon}{1-\alpha} \left(\frac{h}{\varepsilon}(1-\alpha) \right)^{\frac{1}{\alpha}}.$$

Preuve. On observons que l'unique solution de l'équation algébrique

$$g_{\alpha}^{'}(z) = 0,$$

est $z_0 = \left(\frac{\varepsilon}{h(1-\alpha)} \right)^{1/\alpha} \in]0, \infty[$. La fonction $g_{\alpha}(z)$ est croissante sur $]0, z_0[$ et décroissante sur $]z_0, \infty[$. D'où, le maximum global de $g_{\alpha}(z)$ est atteint en z_0. Ce qui achève la preuve du lemme 2.
Maintenant à partir de l'inégalité (1.9) on déduit:

$$\forall \varepsilon > 0 \text{ et } \forall\ x, y \in [a,b], \qquad C_{\varepsilon}^{'} \geq \frac{|f(x) - f(y)| - \varepsilon}{|x-y|}.$$

Et comme f est höldérienne, alors on a

$$\frac{|f(x)-f(y)|-\varepsilon}{|x-y|} \leq \frac{h\,|x-y|^{\alpha}-\varepsilon}{|x-y|}$$

Cependant, on peut prendre

$$C_{\varepsilon}^{'} = \underset{x,y\in[a,b]}{\text{Sup}} \frac{h\,|x-y|^{\alpha}-\varepsilon}{|x-y|} \geq \underset{x,y\in[a,b]}{\text{Sup}} \frac{|f(x)-f(y)|-\varepsilon}{|x-y|}\ .$$

Posons, $|x-y|=z$, en utilisant le lemme 2. Nous obtenons la constante $C_{\varepsilon}^{'}$ pour laquelle l'inégalité (1.9). est vérifiée. On peut montrer facilement que $C_{\varepsilon}^{'} < C_{\varepsilon}$.

Preuve du théorème 1.3
De l'inégalité (1.9) on a $\forall \varepsilon > 0$ et $\forall\ x, y \in [a,b]\,,\ \exists C_{\varepsilon} > 0$, telle que

$$f(y) - C_{\varepsilon/2}\,|x-y| - \frac{\varepsilon}{2} \leq f(x).$$

Avec la modification qu'on a vu, pour $y \in [a,b]$ fixé, si un certain x vérifie

$$m_k^* - \varepsilon \leq\ f(y) - C_{\varepsilon/2}\,|x-y| - \frac{\varepsilon}{2}, \tag{1.12}$$

alors $m_k^* - \varepsilon \leq f(x)$. Si on pose $y = x_i$ dans l'inégalité (1.12) on obtient $|x-x_i| \leq \frac{f(x_i)-m_k^*+\frac{\varepsilon}{2}}{C_{\varepsilon/2}}$ on peut alors recouvrir l'intervalle $[a,b]$ par la famille d'intervalle

$$I(x_i, r_{ik}) = \{x \in \mathbb{R},\ \ |x-x_i| \leq r_{ik}\}\,;$$

et la constante C_{ε} est donnée dans le théorème 1.3.

L'algorithme modifié
1. Initialisation.
Poser $k=1,\ x_1 = a + \frac{\varepsilon}{2C_{\varepsilon/2}},\ x_{\varepsilon} = x_1,\ f_{\varepsilon} = f(x_{\varepsilon})$
2. Etapes $k = 2, 3, ...$
Poser $x_{k+1} = x_k + \frac{f(x_k)-f_{\varepsilon}+\frac{\varepsilon}{2}}{C_{\varepsilon/2}}$
Si $x_{k+1} > b$ alors stop.
Sinon, déterminer $f(x_{k+1})$.
Si $f(x_{k+1}) < f_{\varepsilon}$, alors poser $x_{\varepsilon} = x_{k+1}, f_{\varepsilon} = f(x_{k+1})$
Poser $k = k+1$ er aller à 2.

5 Résolution d'un système d'équations non-linéaires

La résolution d'un système d'équations algébriques est une activité de base en analyse numérique. Dans le cas où le système est linéaire, il existe plusieurs approches pour déterminer les solutions(méthode d'élimination de Gauss, matrice inverse, etc.). Dans le cas où les équations dans le système ne sont pas linéaires ou polynomiales, il est difficile de déterminer les racines du système. On pourrait bien résoudre ce système par la méthode de Newton mais cela nécessite l'utilisation des dérivées. Si les fonctions f_i sont seulement continues, l'utilisation d'une méthode d'optimisation peut s'imposer. Nous allons voir l'application des algorithmes d'optimisation globale qu'on a vu précédemment, sur le système suivant:
Considérons le système d'équations algébriques non linéaires:

$$f_i(x) = 0,\ 1 \leq i \leq n, \qquad (\mathbf{S})$$

avec f_i, des fonctions höldériennes de paramètres $h_i > 0$ et $\alpha_i < 1$, (pour tout $1 \leq i \leq n$), et définies sur un même intervalle $[a, b]$ et à valeurs dans $\mathbb{R}$.Dans le cas où les fonctions f_i sont lipschitziennes, le système $(\mathbf{S})$ a été étudié par plusieurs auteurs [3]. Dans notre travail nous allons étendre cette étude dans le cas où les fonctions f_i sont höldériennes et peuvent être non-différentiables.
La solution du système $(\mathbf{S})$ est un vecteur $x^* \in \mathbb{R}$ tel que $f_i(x^*) = 0$, pour chaque $1 \leq i \leq n$.
D'abord, introduisons sur l'intervalle $[a, b]$, la fonction $H : [a, b] \to \mathbb{R}^n$ par

$$H(x) = (f_1(x), ..., f_n(x)) .$$

Les solutions approximatives de l'équation:

$$H(x) = 0, \quad x \in [a, b],$$

seront données par les points de l'ensemble suivant

$$[a, b]_\epsilon = \{x \in [a, b],\ \|H(x)\| \leq \epsilon\} ,$$

où ϵ est la précision et $\|.\|$ la norme euclidienne. On a le résultat suivant:

Lemme 3. *Si f_i sont des fonctions höldériennes sur $[a, b]$ de paramètres $h_i > 0$ et $\alpha_i < 1$, pour $1 \leq i \leq n$, alors la fonction $f(x) = \|H(x)\|$ est höldérienne sur $[a, b]$ de paramètres $h = (\sum_{i=1}^{n} h_i^2)^{1/2}$ et $\alpha = \min_i \alpha_i$.*

Preuve. Soit $x, y \in [a,b]$, le fait que f_i sont höldériennes, nous avons:

$$\begin{aligned} |f(x)-f(y)| &= |\|H(x)\| - \|H(y)\|| \leq \|H(x)-H(y)\| \\ &\leq \left(\sum_{i=1}^{n} h_i^2 \, |x-y|^{2\alpha_i}\right)^{\frac{1}{2}} \leq (\sum_{i=1}^{n} h_i^2)^{1/2}. \, |x-y|^{\min \alpha_i}. \end{aligned}$$

Le système (**S**) se ramène à un problème d'optimisation globale. Les solutions de (**S**) sont précisément les minimiseurs globaux du problème suivant

$$\min_{x \in [a,b]} f\ (x). \qquad (\mathbf{P})$$

Proposition *$x^* \in [a,b]$ est une solution du système* (**S**) *si et seulement si:*

$$0 = f_* = \|H(x^*)\| = \min \{\|H(x)\|, \, x \in [a,b]\}.$$

Preuve. Si le système (**S**) admet une solution, alors $f_* = 0$; sinon, $f_* > 0$. La preuve est une conséquence immédiate de la propriété de la norme, i.e., $\|z\| \geq 0$, $\forall z$ et $(\|z\| = 0 \Leftrightarrow z = 0)$.
D'après la proposition précédente, le problème d'optimisation (**P**) contient toutes les informations du système (**S**). On voit bien que la condition $\min_{[a,b]} f(x) > 0$ est satisfaite si et seulement si le système (**S**) n'admet aucune solution; et dans le cas où $\min f([a,b]) = 0$, l'ensemble des solutions du problème (**P**) coincide avec l'ensemble des solutions du système (**S**).
Exemple. Considérons le système d'équations algébriques

$$\begin{cases} g_1(x) = \sqrt{\frac{9}{4} - x^2} - \frac{\sqrt{5}}{2} = 0 \\ g_1(x) = \left|\sin(\frac{\pi}{2}x)\right| \left|\frac{\sqrt{2}-x}{\sqrt{2}-1}\right|^{\frac{1}{3}} - x = 0 \\ g_1(x) = -\cos(x+\frac{\pi}{2}-1) e^{1-\frac{\sqrt{\left|\sin \pi(x+\frac{\pi}{2}-1)-\frac{1}{2}\right|}}{\pi}} = 0. \end{cases} \qquad (S')$$

Les fonctions $g_1(x)$, $g_2(x)$ et $g_3(x)$ du système (S') sont höldériennes sur l'intervalle $[-1.5, 1.5]$ respectivement de constantes $(h_1 = \sqrt{3},\ \alpha_1 = 1/2)$, $(h_2 = 5.4,\ \alpha_2 = 1/3)$ et $(h_3 = 7.3,\ \alpha_3 = 1/2)$.
Le système (S') est équivalent au problème d'optimisation globale suivant:

$$\min_{x \in [-1.5, 1.5]} g(x),$$

où $g(x) = \|H(x)\| = \sqrt{(g_1(x))^2 + (g_2(x))^2 + (g_3(x))^2}$, est höldérienne de constante $h = \sqrt{h_1^2 + h_2^2 + h_3^2} = 9.2439$ et $\alpha = \min(\alpha_1, \alpha_2, \alpha_3) = 1/3$.
En appliquant l'algorithme de recouvrement non-uniforme modifié, on obtient pour $\epsilon = 0.1$, la solution du (S') $x = 1.0000062$ (solution exacte est $x = 1$).

6 Conclusion

Dans ce travail on a proposé une modification d'une méthode itérative de recouvrement non-uniforme pour le cas des fonctions höldériennes définie sur un intervalle $[a, b]$ de $\mathbb{R}$. L'algorithme ainsi donné est simple et n'exige pas d'autres calculs auxiliaires tels la construction des fonctions minorantes, ou la fonction f d'être différentiable. Le schéma de l'algorithme ne dépend que des informations de la fonction objectif. La convergence a été étudiée et on a appliqué l'algorithme pour la résolution des systèmes d'équations algébriques non linéaires.

References

[1] Yu. G. Evtushenko, *Algorithm for finding the global extremum of a function(case of a non-uniforme mesh)*, USSR Comput. Mathem. and Phys., 11, No. 6, 1390-1403. (1971).
[2] Yu. G. Evtushenko, V. U. Malkova, and A. A. Stanevichys, *Parallelization of the Global Extremum Searching Process*, Automation and Remote Control, Vol. 68, No. 5, pp. 787-798. (2007).
[3] R. Horst and P. M. Pardalos, *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht. (1995).
[4] E. Gourdin, B. Jaumard, and R. Ellaia , *Global Optimization of Hölder function*, J. of Global Optimization, Vol. 8, pp. 323-348. (1996).
[5] D. Lera and Ya. D. Sergeyev, *Global Minimization Algorithms for Hölder functions*, BIT, Vol 42, No. 1, pp. 119-133. (2002).
[6] M. Rahal and A. Ziadi, *A new extension of Piyavskii's method to Hölder functions of several variables*, Applied mathematics and Computation. Vol. 197, pp. 478-488. (2008).
[7] M. Rahal, *Extension de certaines méthodes de recouvrement en optimisation globale*. Thèse de Doctorat en sciences, Université Ferhat Abbas, Sétif, (2009).

Approche de Généralisation Multicritères Pondérés Appliquée au Thème Bâti

Khalissa Derbal, Kamel Boukhalfa, Zaia Alimazighi

LSI Laboratory, Computer Science Department, Faculty of Electronic an Computer Science, USTHB, BP 32 EL ALIA 16111 Bab Ezzouar Algiers.
kderbal@usthb.dz, kboukhalfa@usthb.dz, Alimazighi@wissal.dz

Résumé. Le processus de généralisation cartographique a connu une importante évolution ces deux dernières décennies. De nombreux travaux entrepris par des chercheurs pluridisciplinaires ont marqué cet axe. Le linéaire routier était le thème le plus traitée par la communauté scientifique car il constitue le thème dominant sur une carte. Le thème bâti, s'est imposé par sa dépendance à ce dernier en particulier dans l'accomplissement des projets d'aménagement urbain. Les approches de généralisation du bâti, sont peu nombreuses et majoritairement inspirées par les travaux sur les thèmes linéaires. Dans cet article, nous proposons une approche de généralisation conçue pour le traitement du bâti, que nous caractérisons de multicritères pondérés. Elle est basée sur : (1) la capitalisation et la formalisation du savoir-faire d'un expert cartographe et (2) une méthode d'orchestration d'opérateurs de généralisation connus dans la littérature. Notre approche a été implémentée dans l'environnement SIG « *ArcGis 9.2* » en utilisant un jeu de données vecteurs provenant de l'Institut National de Cartographie et Télédétection (INCT). Nous présentons le résultat des différentes étapes de traitement, ainsi que le résultat final qui représente la région étudiée après généralisation.

Mots clés : Base de Données Géographique(BDG), Généralisation Cartographique, Thème Bâti, Opérateurs de Généralisation, Contraintes **Cartographique.**

1 Introduction

L'information géographique a connu une large utilisation au cours de ces dernières années grâce au développement de la cartographie numérique et des technologies de l'information. La donnée géographique est devenue désormais accessible. Chaque utilisateur désire avoir une carte personnalisée. Cependant, ceci est irréalisable avec les cartes de base que disposent les institutions de production cartographique car elles sont conçues à des échelles spécifiques et avec un coût très élevé. Ces institutions se sont donc orientées vers une approche alternative capable de produire autant de représentations cartographiques que de besoins exprimés par les utilisateurs avec un moindre coût. Il s'agit du *processus de généralisation*. La généralisation consiste donc, à dériver à partir d'un jeu de données très détaillé décrivant une localisation sur

la surface de la terre, de multiple représentations pouvant être sauvegardées ou non. La complexité du processus a fait que son automatisation a rapidement imposé aux chercheurs la décomposition du traitement en tâches élémentaires. Différents travaux se sont donc, focalisés sur le développement d'algorithmes interprétant les différents opérateurs de généralisation [6][8]. D'autres travaux se sont intéressés à des thèmes particuliers au détriment d'autres, c'est le cas du linéaire (réseau routier, réseau hydrographique…) et ce, afin d'arriver à des résultats validant le processus lui-même [2][9][4]. Nous soulignons aussi, que le *linéaire routier* était jusque-là le plus abordé car il constitue le thème dominant sur une carte. Ce thème est intimement lié au *thème bâti* (tout ce qui est construit sur la surface de la terre) notamment lorsqu'il s'agit d'aménagement urbain. De plus, le processus de généralisation n'est pas complètement automatisé malgré les efforts des chercheurs, il demeure un axe de recherche active [14]. Tout ceci a constitué notre motivation pour aborder cet axe dans le cadre d'une collaboration avec l'*INCT*. Après le thème routier [13], nous nous intéressons dans cet article au thème bâti. Nous proposons donc une approche de généralisation du thème bâti, dans laquelle nous formalisons les connaissances et le savoir-faire d'un expert cartographe tout en exploitant différents opérateurs de généralisation. Le choix de l'application de l'un ou l'autre des opérateurs est soumis à une démarche décisionnelle basée sur la détermination du degré d'importance d'un objet. Cette mesure est calculée à partir des poids associés à l'ensemble des critères décrivant pertinemment un objet géographique.

Cet article est organisé comme suit : dans la section 2, nous introduisons quelques concepts de base liés à la généralisation. Dans la section 3 nous présentons un état de l'art à l'issue duquel nous avons proposé notre approche. Dans la section 4, nous décrivons de façon détaillée l'approche proposée. La section 5 présente les différentes étapes du traitement ainsi que la phase d'implémentation. Nous terminons ce papier par une conclusion et quelques perspectives.

2 La généralisation

La généralisation est définie par l'ACI (Association de Cartographie Internationale) comme étant "la représentation simplifiée de détails en fonction de l'échelle et des objectifs de la carte". Dans cette définition nous retenons deux aspects: (1) Les besoins qui sont l'échelle de représentation et l'objectif de la carte, et (2) Les moyens utilisés qui sont la sélection et la représentation simplifiée de détails [8]. Ces deux aspects sont interprétés par les concepts d'échelle, de contraintes cartographiques et de conflits que nous présentons en fin de la présente section.

Lors de la réduction de l'échelle d'une carte, cette dernière devient *illisible.* Certains objets deviennent imperceptibles, d'autres se chevauchent ou se superposent générant ainsi le phénomène de *conflit.* Pour la rendre cette carte lisible, *o*n procède par filtrage d'information où on ne garde sur cet espace réduit que les objets utiles pour l'objectif visé. Les objets jugés moins utiles sont alors éliminés tandis que d'autres peuvent subir des transformations (amplification, déplacement, etc.) qui sont réalisées grâce aux *opérateurs de généralisation (algorithmes)* largement étudiés dans la littérature [8]. La figure 1 illustre l'utilité de la généralisation.

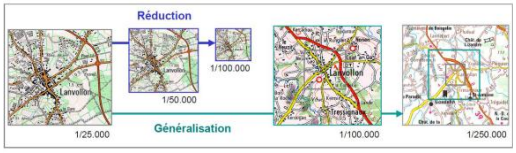


Figure 1. L'apport de la généralisation

2.1 Echelle/Niveau de détail

Ce concept, connu par *LoD* (*Level of Detail*), a été largement abordé par les chercheurs [7] afin de mettre l'accent sur deux facteurs qui gèrent la même information dans deux contextes différents : celui de la *carte*, en tant qu'espace réduit et celui de la *base de données* en tant qu'espace infini. Dans le premier contexte le rôle de l'échelle est majeur car elle détermine la taille de la carte, la sélection des objets et leur représentation par leur emprise ou un symbole en fonction de leurs tailles. Donc, l'échelle contrôle le volume d'information sur une carte. Par contre dans le deuxième contexte, *LoD* englobe quatre notions principales pour être défini exactement : la précision, l'exactitude, la résolution géométrique et la résolution sémantique. Ainsi, pour produire une carte lisible il faut d'abord définir une échelle qui permet de visualiser le niveau de détail souhaité.

2.2 Contraintes cartographiques

La généralisation cartographique a pour but de satisfaire un ensemble de contraintes qui peut être hiérarchisé en quatre groupes ; *contraintes de lisibilité*, *contraintes de respect de forme*, *contraintes d'organisation spatiale* et *contraintes d'harmonie globale* [5]. Les contraintes de lisibilité, par exemple, sont celles qui sont à l'origine de la nécessité de généraliser un ensemble de données cartographiques. Elles sont indépendantes des objectifs de la carte. Elles sont là pour assurer que tout objet graphique isolé est perceptible, que sa forme peut être discernée, que deux objets voisins sont séparés et que les différents paliers sont respectés. Ces contraintes sont pratiquement réalisées via ce qu'on appelle un seuil (seuil de perception, de séparation et de densité maximale) [2][3].

2.3 Phénomène de conflit

Une carte est produite à partir d'une *Base de Données Cartographique(BDC)* qui n'est autre qu'une *base de données géographique* enrichie de symboles qui représentent pertinemment les objets géographiques affichés ou imprimés. La symbolisation des objets provoque le phénomène de conflit lors de la réduction d'échelle. Trois types de conflit sont alors définis : *l'auto- conflit* où deux parties d'un même objet entrent en conflit (se superposent par exemple), *l'intra-conflit* qui concerne l'interaction de l'objet avec ses voisins de la même couche et enfin *l'inter-conflit* qui apparait lors de la superposition de deux couches différentes. Par exemple,

dans le cas du thème bâti, l'intersection point/polygone est considérée comme un inter-conflit. La figure 2 illustre un exemple d'intra-conflit pouvant apparaitre lors du traitement du thème bâti.

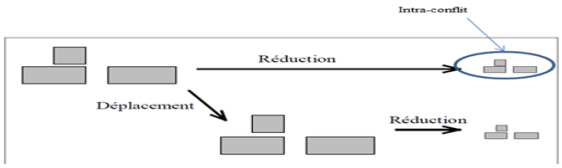


Figure 2. Exemple d'intra-conflit dans le thème Bâti

3 Etat de l'art

L'automatisation du processus de généralisation demeure une tâche difficile à réaliser malgré les efforts fournis depuis plus d'une trentaine d'années. Différentes approches de généralisation cartographique inspirés par la pratique et le raisonnement des experts cartographes, ont été développées. Une des premières approches est *l'approche séquentielle* qui se base sur l'application séquentielle d'algorithmes de généralisation. Son inconvénient majeur est de ne pas tenir compte de l'environnement des objets à généraliser. Une telle approche ne reflète donc pas le raisonnement du cartographe qui observe les objets tout en visant un objectif global (objectif de la carte). Par conséquent ces objets sont traités dans un contexte et non pas de façons indépendantes. Pour pallier ce problème, de nouvelles approches dites *à base de connaissance* ont vu le jour. Dans ces approches, le raisonnement intellectuel inspiré de la pratique manuelle du cartographe a été intégré. Ainsi les insuffisances qui ont marqué la classe des approches séquentielles ont été prises en considération (le choix des algorithmes, leur séquence et les paramètres à appliquer). La classe d'approches à base de connaissance englobe entre autres *les systèmes experts à base de règles* [1][2], *les systèmes à base de contraintes*[5][3] et *systèmes multi-agents* [10][12]. *Les approches à base de règles* souffrent de la rigidité des règles. Ces derniers ont été remplacées par les contraintes afin de donner plus de flexibilité aux systèmes développés, donnant naissance ainsi aux *approches à base de contraintes*. Différents travaux se sont penchés sur la détermination de ces contraintes [5] (voir section 3). Cependant, un problème essentiel a surgit et qui se résume dans le fait que la résolution de la violation d'une contrainte peut générer de nouveaux conflits plus important. Il faut donc choisir *l'ordre de satisfaction des contraintes* qui est un processus très complexe [11].

Dans la figure 3, nous présentons une classification de ces différentes approches, inspirée d'une description détaillée présentée dans [11]. Cette diversité d'approches informe sur l'intérêt grandissant que les chercheurs ont porté à cet axe. Les approches à base de connaissance, à base de contraintes et multi-agents ont été appliquées au *thème bâti* [5][8]. Cependant le nombre de travaux qui ont abordé ce thème reste insignifiant car il est souvent lié à d'autres thèmes en particulier le *thème routier*. De plus, le traitement du bâti est toujours conditionné par les particularités de la zone

traitée *(zone urbaine dense, zone rurale, etc.)*[9]. Ceci rend les résultats très spécifiques à la région étudiée et ne peuvent être appliqués systématiquement à d'autres régions qui ne présentent pas les mêmes caractéristiques. Or, une généralisation de qualité est celle qui permet une représentation adaptée aux besoins des utilisateurs. L'approche que nous proposons utilise ces concepts tout en proposant *des critères* permettant de s'adapter à différentes situations. Par exemple, le critère *zone* que nous proposons et qui prend deux valeurs *nord* et *sud* pondérées différemment nous informe implicitement sur la densité du bâti. Ceci va guider le choix de l'opérateur à appliquer. La mesure du degré d'importance d'un objet sera moyennement élevée si sa zone d'appartenance est le *sud* (moins de conflits), ce qui favorisera le choix de l'opérateur *d'amplification* pour le mettre en évidence.

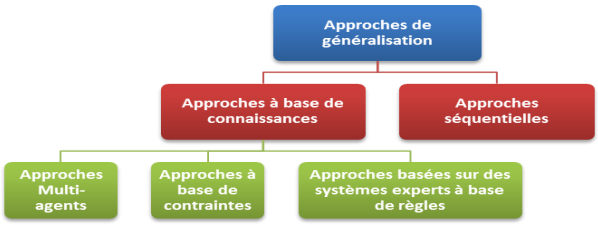


Figure 3. Classification des approches de généralisation cartographique.

4 Approche proposée

L'approche de généralisation que nous proposons dans cet article est basée sur la capitalisation et la formalisation des connaissances d'un expert cartographe. Ce dernier a pendant longtemps pratiqué la généralisation, de façon manuelle puis de façon interactive. Nous nous sommes donc, inspirés de son raisonnement intellectuel vis-à-vis de ce processus. Le thème qui constitue l'objet de cette étude est le *thème bâti* à deux dimensions. Il est représenté sous deux formes géométriques vectorielles qui sont : la forme ponctuelle et la forme polygonale. La représentation d'un objet dans l'une ou l'autre de ces deux formes dépend de l'importance de l'objet dans un niveau de détail donné (qui correspond à l'échelle au niveau cartographique). A titre d'exemple une agglomération est représentée par un polygone à une grande échelle (*1/50000*) et par un point à une petite échelle (*1/200000*). Afin de mener notre étude, nous avons considéré trois principales hypothèses : (1) les données vecteurs utilisées sont initialement fournies à une *échelle origine détaillée* et seront généralisées à une *échelle cible moins détaillée* (par exemple échelle origine =1/50000 et échelle cible=1/200000 ou 1/500000) (2) le thème bâti regroupe deux couches correspondantes aux deux primitives point et polygones (la couche ponctuelle et la couche polygonale). Comme ces deux couches présentent des caractéristiques différentes nous les traitons séparément et (3) nous adoptons la classification des opérateurs définie dans [9]. Nous soulignons que chaque opération est décrite par un triplet : <*Contrainte à satisfaire, Règle qui représente la condition*

à vérifier, Opérateur à appliquer>. Nous nous alignons ainsi aux approches à base de connaissance.

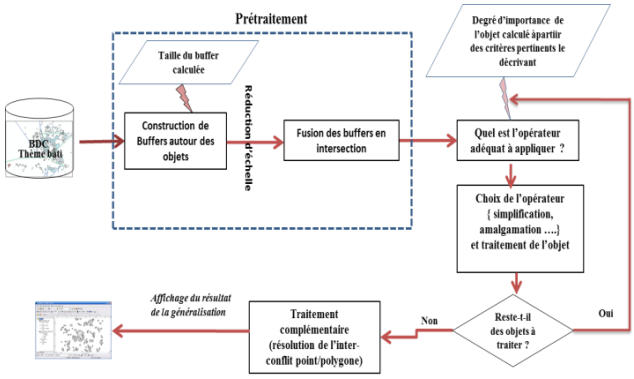


Figure 4. Etapes de traitement selon l'approche proposée

4.1 Description de l'approche

L'approche de généralisation que nous proposons s'articule autour de trois phases de traitement appliquées à chacune des couches : *bufferisation, fusionnement et orchestration*. Un traitement complémentaire permet de prendre en charge le phénomène d'inter-conflit (point/polygone) lors de la superposition des couches polygonale et ponctuelle.

La phase de bufferisation, consiste à construire des buffers autour des objets. Un buffer est une zone de taille calculée en fonction du seuil de perception et de l'échelle cible de la carte (voir la section 4.2). Cette zone permet de prévenir le phénomène de conflit à l'échelle d'arrivée.

La *phase de fusionnement consiste à fusionner* tous les objets dont leurs buffers se chevauchent lors de la réduction d'échelle. Ces deux premières phases (bufferisation et fusionnement) constituent un prétraitement efficace car elles permettent de réduire le nombre d'intra-conflits pouvant être générés.

La phase d'orchestration fait appel aux différents opérateurs de généralisation adaptés aux traitements des objets polygonaux et des objets ponctuels (*simplification*/*élimination amalgamation*, *simplificatio*n, *déplacement*, etc). Nous proposons une méthode d'orchestration en fonction du degré d'importance de chaque objet. Cette mesure est calculée à partir des poids attribués à chacun des critères décrivant un objet. A titre d'exemple, le critère *nom* informe sur le nom attribué à l'objet dans la BDC selon une nomenclature donnée (nous adoptons celle de l'INCT). A chaque nom est attribué un poids en fonction de son importance général et son importance pour l'objectif visé de carte. L'enchaînement de ces différentes étapes de traitement est illustré par la figure 4.

4.2 Traitement de la couche polygonale

Le traitement de la couche polygonale est détaillé via les étapes suivantes :

Bufferisation : La dimension du buffer est égale au seuil_de_percetion* échelle_cible. Le seuil correspond à la taille minimale d'un figuré ou d'un élément graphique perçu sur une carte. Ce seuil peut être défini comme suit : Diamètre minimal d'un point : 0.2 mm, Epaisseur minimale d'une ligne : 0.1 mm (0.08 mm dans certains cas), largeur minimale du côté d'un carré plein : 0.4 mm, etc. Ainsi, par exemple pour l'échelle d'arrivée 1/200000 avec un seuil_de_perception = 0.8mm, la dimension du buffer = 160m terrain.

Fusion des buffers **:** Cette opération est décrite par trois éléments de base :
- *La contrainte à satisfaire* : élimination préliminaire du conflit
- *Règle* : « respecter l'orientation ».
- *Opérateurs* : [Simplification : agrégation (fusion)].

L'algorithme de fusion ne fait que négliger les frontières entre les objets fusionnés.

Orchestration du processus : Dans cette phase, chaque objet est classé selon la valeur de son degré d'importance, le choix de l'opérateur à appliquer en dépend. Chaque objet polygone a donc, son *degré d'importance* qui correspond à la somme des coefficients (poids) attribués aux critères pertinents décrivant l'objet. Nous avons défini 3 critères : *la superficie de l'objet, la zone* où se situe l'objet (*nord* ou *sud* du pays) et le *nom* de l'objet par rapport à la nomenclature proposée par l'INCT. Ces critères, sont choisis suite à une étude des spécificités d'un extrait de la base de données géographique de l'INCT. A chacun de ces critères, nous attribuons un poids calculé de façon expérimentale et que nous avons formulé comme suit :
-La superficie est pondérée par *poids_superficie.* Nous avons établi expérimentalement trois intervalles : *Petite Superficie (PS), Moyenne Superficie (MS)* et *Grande Superficie(GS)* auxquels nous attribuons respectivement les coefficients suivants : *Co_PS, Co_MS* et *Co_GS*.
- La zone est pondérée par un coefficient nommé *Co_zone.*
- Le nom du bâti est pondéré par *Co_nom.*

La somme des coefficients associés aux différents critères définis ci-dessus détermine le degré d'importance de l'objet.

Som_Co_Poly = Co_superficie (un parmi les trois cités précédemment) + Co_zone + Co_nom.

A l'issue de cette classification des objets, un choix de l'opération à effectuer pourrait être désormais effectué. Faut-il supprimer objet, l'amplifier, le transformer ou le préserver ? Chacune de ces opérations est décrite dans le tableau 1.

4.3 Traitement de la couche ponctuelle

La même démarche est suivie dans le traitement de la couche ponctuelle, avec la redéfinition des critères décrivant l'objet : la *zone* où se situe l'objet (*nord* ou *sud* du pays), le *nom de l'objet* et *nature de l'objet*. Ce dernier est un qualificatif descriptif d'une importance particulière pour notre patrimoine. A titre d'exemple la nature

d'objet '*koubba'* qui signifie un mausolée est très significatif car il arrive souvent que toute une région porte le nom de ce mausolée ; la région de *Ami-moussa* dans la wilaya de *Relizane* en est un exemple. Parmi ces noms, nous citons quelques-uns adoptés par l'INCT : *koubba, maison carrée, maison rectangulaire, cimetière musulman, cimetière chrétien*.

Tableau 1. Opérateurs de généralisation pouvant être appliqués à la couche polygonale.

Opération	Description
Supprimer objet	*Contrainte* : harmonie globale. *Règle* : « degré d'importance faibe ». *Opération* : [Simplification : sélection/élimination (suppression)].
Amplifier objet	*Contrainte* : la lisibilité (seuil de séparation) *Règle* : « il ne percute pas un autre objet ». *Opération* : [Caricature : amplification].
Transformer l'objet	*Règle* : «préserver sa position en repère absolu ». *Opération* : [Caricature : amélioration de la géométrie (fractalisation)].
Conserver l'objet	Certains objets seront conservés et ne nécessitent aucun traitement

A l'issue de cette classification une décision concernant l'opération à effectuer sera efficacement prise : supprimer l'objet ponctuel, le fusionner avec d'autres ou bien le conserver. La description de ces opérations est donnée dans le tableau 2.

Tableau 2. Opérateurs de généralisation pouvant être appliqués à la couche ponctuelle.

Opération	Description
Supprimer point	*Contrainte* : harmonie globale. *Règle* : « l'objet ne doit pas être important ». *Opération* : [Simplification : sélection/élimination (suppression)].
Fusionner point	*Contrainte* : éliminer l'empattement des objets. *Règle* : « respecter l'orientation ». *Opération* : [Simplification : agrégation (fusion
Conserver point	Certains objets seront conservés et ne nécessitent aucun traitement

4.4 Traitement complémentaire (résolution de l'inter-conflit) :

En plus des traitements cités ci-dessus, d'autres traitements, peuvent être envisagés. Un premier consiste à satisfaire la contrainte de lisibilité (seuil de séparation). Pour cela, nous procédons par calcul de distance entre l'objet polygonale et les autres objets avoisinants (toutes couches confondues). Si la distance calculée est inférieure au seuil fixé, l'objet polygonal de plus grande superficie subira une érosion de façon à ce qu'il conserve le seuil de séparation. Cette opération est décrite dans le tableau 3.

Un second traitement est basé sur l'aspect sémantique. En effet s'il y a un conflit entre le résultat du traitement de la couche ponctuelle et les objets qui ont subi une transformation de géométries, nous supprimons le point de la couche ponctuelle, car ce dernier est moins important par rapport à un objet qui peut représenter un bâtiment administratif ou industriel. De plus si le point supprimé représente une référence telle qu'un *mausolée*, ce nom est reporté sur l'objet conservé.

Tableau 3. Description de l'opération d'érosion.

Opération	Description
Erosion	Contrainte : contrainte de lisibilité (seuil de séparation). Règle : « préserver au mieux sa forme globale. Caricature : déplacement (érosion).

5 Expérimentation

Notre approche a été implémentée dans un environnement dédié au traitement de l'information géographique, *SIG ArcGis 9.3*. Nous avons utilisé un jeu de données vecteurs à une échelle donnée (échelle origine) plus détaillée que l'échelle cible.

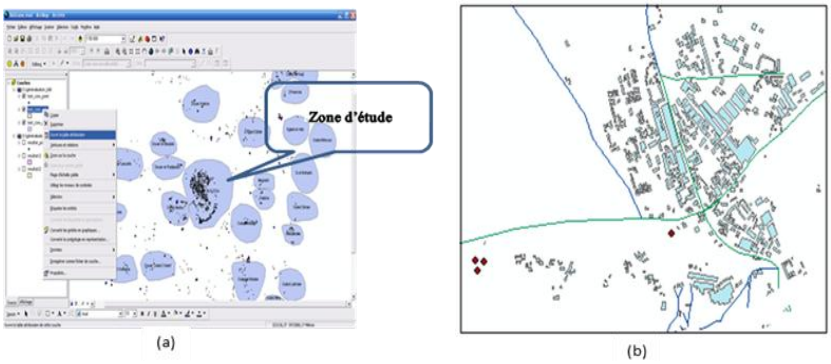


Figure 5. Région d'ami moussa dans la wilaya de Relizane à une échelle donnée (a),la même région agrandie (b).

La figure 5 représente la Zone d'étude : « *Ami moussa* » située dans la wilaya de *Relizane* au nord-ouest de l'Algérie. Les surfaces bâties sont symbolisées par des polygones et des points (figure 9.a). Un extrait de la *geodatabase* est présenté dans la figure 6. Dans la figure 7, nous présentons la zone d'étude après réduction d'échelle, elle est particulièrement marquée par l'apparition du phénomène de conflit qui rend les objets imperceptibles. Le processus de généralisation proposé passe par les trois phases de notre approche ; bufferisation, fusion et orchestration, en plus du traitement finalisant la carte généralisée. Chaque phase génère un résultat intermédiaire que nous présentons via les figures 9.a et 10. La figure 8 représente l'interface principale

de l'application développée. Le résultat de la première phase du traitement est présenté dans la figure 9.b et ramené à l'échelle cible (1/200000) dans la figure 9.c.

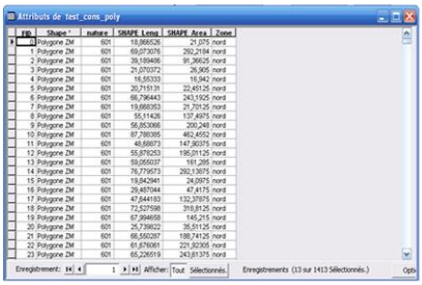


Figure 6. Extrait de la géodatabase de la zone d'étude

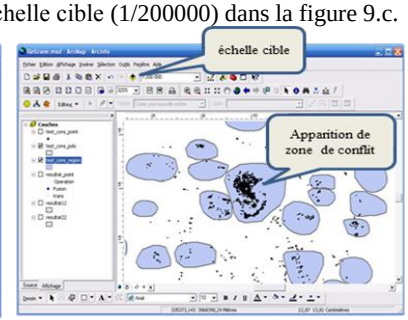


Figure 7. Apparition des zones de conflits lors de la réduction d'échelle.

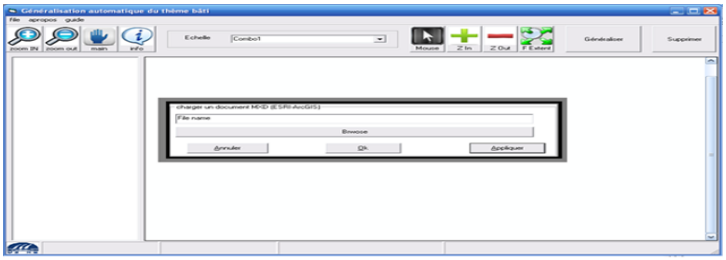


Figure 8. Interface principale de l'application

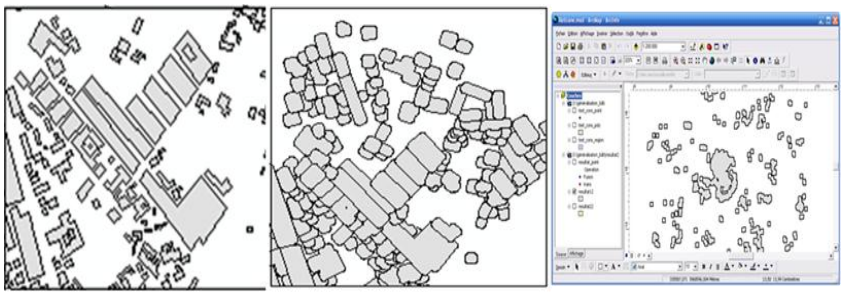


Figure 9. La zone d'étude à l'échelle d'origine (a), Résultat de la bufferisation en agrandi (b), résultat ramené à l'échelle cible (c).

Le traitement de la troisième phase consiste à exploiter la mesure qui représente le *degré d'importance* de chaque objet (la somme des poids des différents critères) dans

le but de décider de l'opérateur adéquat à appliquer. Le résultat obtenu se trouve dans la figure 10. Nous soulignons que les objets de la couche polygonale sont transformés.

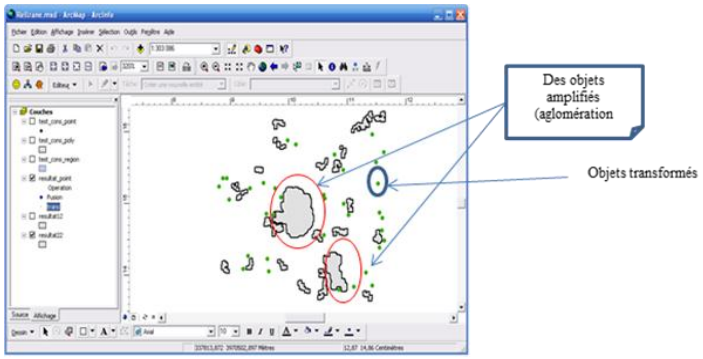


Figure 10. Résultat du traitement de la couche polygonale

Le traitement de la couche ponctuelle est illustré par la figure 11.a et 11.b (résolution d'intra-conflit). Le traitement de l'inter-conflit point/polygone sera résolu en appliquant l'opérateur d'élimination (figure 11.c) à condition de conserver la valeur sémantique que porte l'objet ponctuel, s'il représente un point de repère (cas d'un mausolée par exemple). Pour cela ce nom sera affecté à l'objet polygonal qui représente une agglomération.

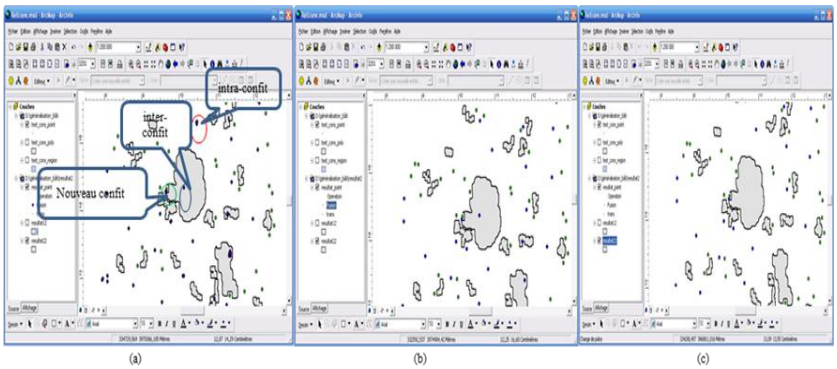


Figure 11. Apparition de différents types de conflits couche ponctuelle/couche polygonale(a), résolution de l'intra-conflit (b) et résolution de l'inter-conflit(c).

Pour bien illustrer l'utilité de la généralisation et la difficulté de la tâche accomplie, nous concluons cette section par la présentation de la zone d'étude à l'échelle cible avant et après généralisation (voir figure 12).

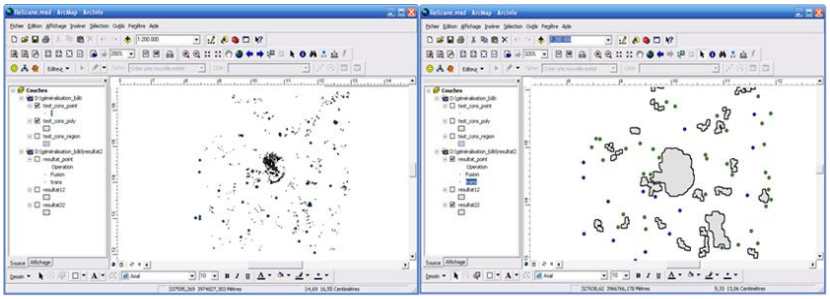


Figure 12. La zone d'étude avant généralisation(a), après généralisation(b)

6 Conclusion :

La complexité de l'automatisation d'un processus de généralisation cartographique réside dans la complexité du traitement qui devra permettre la génération d'une carte lisible et exploitable sans abimer son contenu ni créer de nouveaux conflits. Ceci peut être réalisé par une orchestration d'opérateurs de généralisation (quel est l'opérateur le plus adéquat à appliquer ? et quand l'appliquer ?). Le présent papier contribue à répondre à ces questions à travers une approche de généralisation multicritères pondérés. Cette approche aborde le concept d'orchestration ; les opérateurs ne sont pas appliqués systématiquement mais de façon décisionnelle. Notre expérimentation a porté sur le thème bâti peu abordé dans la littérature. Nous avons présenté les résultats intermédiaires afin de mettre en évidence l'efficacité de la pondération que nous avons proposée. Dans des travaux connexes nous nous sommes intéressés à la résolution du phénomène de *l'auto-conflit* dans un processus de généralisation du linéaire routier [13].
Comme perspectives à ce travail, nous envisageons d'évaluer quantitativement notre approche en la comparant à d'autres approches appliquées au bâti. Néanmoins, ceci reste difficile car comme nous l'avons évoqué dans la section 3(état de l'art), les approches de généralisation sont d'une manière générale conditionnées par les caractéristiques de la zone étudiée. Nous proposons également d'aborder le phénomène inter-conflit routier/bâti car ces deux thèmes sont intimement liés et constituent ensemble, les principaux acteurs de la planification urbaine.

Remerciements
Nous tenons à remercier, Mr A. Koucha (expert cartographe à l'INCT), Mme R. Guemdani, Mr A. Kaddour Djebbar, cadres de l'INCT ainsi que Mr Benabid Abdelkarim pour leur participation active à l'élaboration du prototype.

Références

1. S.Shea et R .McMaster 1989. "*Cartographic generalization in a digital environment: when nd how to generalize".* Actes d'AutoCarto 9, ASPRS/ACSM.
2. MacMaster 91, « Conceptual frameworks for geographical knowledge » dans Buttenfield & R McMaster (ed) Map Generalization, Harlow, Longman, p.21, 39.
3. Müller J. C., Weibel R., Lagrange J.P. and Salge F., (1995). Generalization: State ofthe Art and Issues. In Müller J.C., Lagrange J.P. and Weibel R. (eds.): Gis and Generalization: Methodology and Practice, Bristol, Taylor & Francis, pp. 3-18.
4. Plazanet 1996. « Analyse de la geometrie des objets lineaires pour l'enrichissement des bases de donnees geographiques ». Rapport de thèse en Sciences de 'Information Géographique, Université de Marne-la-Vallée.
5. N.REGNAULD 98, « Généralisation du bâti : structure spatiale de type graphe et représentation cartographique ». Thèse de doctorat : informatique. Paris, 11 mai 1998.
6. A. RUAS. Modèle de généralisation de données géographiques à base de contraintes et autonomie. Thèse de doctorat : science de l'information géographique. Paris, 09 avril 1999.267p.
7. Christelle Vangenot, "Multi-représentation dans les bases de données géographiques", PhD thesis, école polytechnique fédérale de laussane, 2001.
8. S.Mustière. Apprentissage supervisé par la généralisation cartographique. Thèse de doctorat en informatique.Paris, 08 juin 2001.246p.
9. C.DUCHENE. « Généralisation cartographique par agents communicants : le modèle CartACom », application aux données cartographiques en zone rurale.Paris6, 11 juin 2004.
10. Jabeur N. (2006). A multi-agents system for on-the-fly web map generation and spatial conflict, PHD thesis. Université LAVAL, Québec, Janvier 2006.
11. M.N.SABO 07. « Intégration des algorithmes de généralisation et des patrons géométriques pour la création des objets auto-généralisants (SGO) afin d'améliorer la généralisation cartographique à la volée ». Thèse de doctorat : sciences géomatiques. Université laval. Québec, 2007.
12. Gaffuri J. (2008) Généralisation automatique pour la prise en compte de thèmes champ : le modèle GAEL, Ph.D thesis, Université paris-est, École doctorale ICMS, France.
13. K.Derbal Amieur, Z.Alimazighi « Approche de résolution de l'auto-conflit dans un processus de généralisation automatique du linéaire routier ». Veille Stratégique Scientifique et Technologique VSST'2010, 25 au 30 octobre 2010, Toulouse, France.
14. Cécile DUCHÊNE, Recherches en généralisation, bilan et perspectives Laboratoire COGIT, Journées de la Recherche IGN 2012.

Résolution par la méthode de relaxation d'un problème de contrôle optimal avec une entrée libre

Louadj Kahina [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.
louadj_kahina@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 avec une entrée libre et des contraintes sur les états initiaux et finaux. Dans ce problème, le critère est la somme pondérée de deux termes correspondant, d'une part à un critère de minimisation de la commande par rapport à une valeur de celle-ci conduisant asymptotiquement à un état stationnaire du système à réguler. En appliquant le principe du minimum de Pontriaguine, nous pouvons déterminer analytiquement et numériquement la commande optimale et vérifier ainsi la concordance des résulats obtenus.
Mots clés: méthode de relaxation, méthode de tir, principe du minimum de Pontriaguine, contrôle optimal.

1 Introduction

Dans les travaux [1], [2] et [3] des algorithmes numériques de relaxation permettant de déterminer la loi de commande optimale d'un système sans contrainte sur l'état initial et sur l'état final ont été proposés et analysés. De même en couplant la méthode de tir et la méthode de relaxation, on a proposé dans [7], un algorithme numérique permettant de déterminer la loi de commande optimale d'un système avec contrainte sur l'état final. Dans notre étude, nous présentons une méthode pour déterminer analytiquement et numériquement la solution d'un problème de contrôle optimal avec un temps terminal fixé ainsi qu'une contrainte sur l'état final et initial. Nous considérons ici un problème de contrôle optimal avec une entrée libre; de plus, le critère à minimiser est une somme pondérée entre d'une part, la distance entre l'état du système et un état désiré et d'autre part, la distance entre la commande nominale et une commande conduisant asymptotiquement à un état stationnaire. Nous présentons préalablement les équations décrivant le modèle mathématique et, en utilisant le principe du minimum de Pontriaguine, nous donnons une caractérisation de la solution du problème à résoudre, permettant ainsi de déterminer la commande optimale en fonction de la variable d'état adjoint. Nous comparons ensuite, sur un exemple simple, la solution analytique à la solution numérique calculée en utilisant la méthode de

relaxation couplée à la méthode de tir [5]; nous terminons en donnant quelques remarques sur les performances de l'algorithme numérique (vitesse de convergence et temps de calculs).

2 Position du problème

Soit le système dynamique suivant:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ x(0) = z \in X_0 = \{z \in \mathbf{R^n}, \mathbf{Gz} = \gamma, \mathbf{d}_* \leq \mathbf{z} \leq \mathbf{d}^*\}, \mathbf{x(t^*)} = \mathbf{x}^*, \mathbf{t} \in [\mathbf{0}, \mathbf{t}^*], \\ u(t) \in U_{ad}. \end{cases} \quad (1)$$

où $x(t)$ est un n-vecteur représentant l'état du système à l'instant t, $x(0) = z$ est la condition initiale et $x(t^*) = x^*$ 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_{ad} est l'ensemble de commandes admissibles. A, B sont des $n \times n$- et $n \times r$-matrices données, G est une matrice de l lignes et de n colonnes et de $rang\ G = l \leq n$, γ est un $l-$ vecteur. De plus, on suppose que A est l'opposé d'une $M-$matrice.

On cherche une commande admissible $\hat{u}$ qui transfère le système d'un état initial $x(0)$ vers un état final x^* fixé et minimisant la fonction coût J définie par:

$$J(u) = \frac{1}{2} \int_0^{t^*} \left(x^t Q x + u^t N u \right) dt,$$

les matrices Q et N sont symétriques, Q est définie non-négative et N est définie positive.
L'Hamiltonien est donné par:

$$H\left(x(t), p(t), u(t), t\right) = \frac{1}{2} \left[x^t Q x + u^t N u \right] + p^t \cdot \left[Ax + Bu \right].$$

Cherchons maintenant la commande qui minimise l'Hamiltonien; cela revient à chercher $\hat{u}(t)$, tel que:

$$H\left(\hat{x}(t), \hat{p}(t), \hat{u}(t)\right) \leq H\left(x(t), p(t), u(t)\right); \forall u(t) \in U_{ad}, \forall t \in [0, t^*].$$

Les équations d'optimalité s'écrivent:

$$\begin{cases} \frac{dx}{dt} = \frac{\partial H}{\partial p} = Ax + Bu; x(0) = z, x(t^*) = x^*, \\ -\frac{dp}{dt} = \frac{\partial H}{\partial x} = A^t p(t) + Qx(t), \\ \frac{\partial H}{\partial u} = 0 = Nu(t) + B^t p(t). \end{cases} \quad (2)$$

Ces équations sont connues sous le nom d'équations d'Hamilton-Pontriaguine.

Condition de transversalité sur $p(t)$: De manière générale, lorsque l'on prend en compte un coût terminal, le critère à minimiser s'écrit

$$J = g(t^*, x(t^*)) + \int_0^{t^*} f(x(t), u(t), t)dt,$$

où g est le coût terminal, l'état final étant fixé, on a donc classiquement:

$$\begin{cases} \text{il existe } (k_0, k_1) \neq 0, \text{ tel que} \\ \\ \varphi(0, z, t^*, x(t^*), k_0, k_1) = g(0, z, t^*, x(t^*)) + (\psi_0(0, z)|k_0) + (\psi_1(t^*, x(t^*))|k_1), \end{cases} \tag{3}$$

où $\psi_0(0, z) = 0$ représente des contraintes sur les conditions initiales,
et $\psi_1(t^*, x(t^*)) = 0$ représente des contraintes sur les conditions finales avec ψ_0 et ψ_1 de classe C^1 par rapport à x.

$$\begin{cases} p(0) = \frac{\partial \varphi}{\partial z}(0, z, t^*, x(t^*), k_0, k_1), \\ \\ p(t^*) = -\frac{\partial \varphi}{\partial x(t^*)}(0, z, t^*, x(t^*), k_0, k_1), \end{cases} \tag{4}$$

k_i, $i = 1, 2$ étant les multiplicateurs de Lagrange.
Dans notre problème, $g(t^*, x(t^*)) = 0$, alors les conditions de transversalité (3) et (4) sur le vecteur adjoint s'écrivent:

$$\begin{cases} \text{il existe } (k_0, k_1) \neq 0, \text{ tel que} \\ \\ \varphi(0, z, t^*, x(t^*), k_0, k_1) = (\psi_0(0, z)|k_0) + (\psi_1(t^*, x(t^*)|k_1), \end{cases} \tag{5}$$

$$\begin{cases} p(0) = \frac{\partial \varphi}{\partial z}(0, z, t^*, x(t^*), k_0, k_1), \\ \\ p(t^*) = -\frac{\partial \varphi}{\partial x(t^*)}(0, z, t^*, x(t^*), k_0, k_1). \end{cases} \tag{6}$$

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 $x(0) = z$ et d'une condition finale $x(t^*) = x^*$. Par contre, la seconde équation, correspondant à l'équation de l'état adjoint, n'est munie d'aucune condition initiale ou d'aucune condition terminale exploitable pratiquement. On va donc utiliser la méthode de tir pour en déduire la valeur de $p(0)$.

2.1 La méthode de tir simple

La méthode de tir permet d'obtenir la valeur de $p(0)$ nécessaire à la résolution du problème aux deux bouts obtenus par application du principe de Pontriaguine. Si on est capable, à partir de la condition de minimisation de l'Hamiltonien d'exprimer le contrôle en fonction de $(x(t), p(t))$ alors on obtient un système différentiel de la forme $\dot{\upsilon}(t) = F(t, \upsilon(t))$ où $\upsilon(t) = (x(t), p(t))$ et, où les conditions

initiales et finales s'écrivent sous le forme $R(\upsilon(0), \upsilon(t^*)) = 0$. Finalement, on obtient le problème aux valeurs limites suivant:

$$\dot{\upsilon}(t) = F(t, \upsilon(t)), R(\upsilon(0), \upsilon(t^*)) = 0. \tag{7}$$

La solution du problème de Cauchy est:

$$\dot{\upsilon}(t) = F(t, \upsilon(t)), \upsilon(0) = \upsilon_0.$$

La fonction de tir est définie par:

$$\varphi(t^*, \upsilon_0) = R(\upsilon_0, \upsilon(t^*, \upsilon_0)).$$

Soit $\upsilon(t, \upsilon_0)$, le problème (7) est équivalent à:

$$\varphi(t^*, \upsilon_0) = 0. \tag{8}$$

Il s'agit donc de déterminer un zéro de la fonction de tir φ. Comme l'équation (8) représente un système algébrique non linéaire, si l'on connaît une approximation de υ, ce système peut se résoudre par la méthode de Newton. Dans notre problème, φ s'écrit $\varphi = x(t^*) - x^*$. Rappelons la formulation de la méthode de Newton; il s'agit à présent de résoudre numériquement $\varphi(\upsilon) = 0$, où φ est une fonction de classe C^1; le principe de la méthode de Newton est le suivant:
A une étape k donnée, soit υ^k une approximation d'un zéro υ de φ; donc υ s'écrit $\upsilon = \upsilon^k + \Delta\upsilon^k$, et on a alors:

$$0 = \varphi(\upsilon) = \varphi(\upsilon^k + \Delta\upsilon^k) = \varphi(\upsilon^k) + \frac{\partial\varphi}{\partial\upsilon}(\upsilon^k) \cdot (\upsilon - \upsilon^k) + o(\upsilon - \upsilon^k),$$

ce qui entraîne,

$$\frac{\partial\varphi}{\partial\upsilon}(\upsilon^k) \cdot (\upsilon - \upsilon^k) = -\varphi(\upsilon^k),$$

où $\frac{\partial\varphi}{\partial\upsilon}(\upsilon^k)$ est la matrice Jacobienne de l'application $\upsilon \mapsto \varphi(\upsilon)$ calculée quand $\upsilon = \upsilon^k$; or on ne connaît la fonction $\upsilon \mapsto \varphi(z)$ que numériquement. On va donc utiliser un procédé de dérivation numérique basé sur la méthode des différences finies. Pour calculer $\frac{\partial\varphi}{\partial\upsilon}(\upsilon^k)$, nous retiendrons une approximation de $\frac{\partial\varphi}{\partial\upsilon}(\upsilon^k)$ très général (voir [4]), incluant pour des choix particuliers des paramètres la méthode des différence finies symmétriques et définie ci-dessous,

$$\frac{\partial\varphi_i}{\partial\upsilon_j}(\upsilon^k) \simeq \frac{1}{h_{ij}}[\varphi_i(\upsilon + \sum_{k=1}^{j} h_{ik}e^k) - \varphi_i(\upsilon + \sum_{k=1}^{j-1} h_{ik}e^k)],$$

où h_{ij} sont des paramètres de discrétisation correspondant à la $i^{éme}$ équation et à la $j^{éme}$ variable, et e^k est le $k^{éme}$ vecteur de la base canonique. Soit $\Delta_{ij}(\upsilon, h)$ une approximation de $\frac{\partial\varphi_i}{\partial\upsilon_j}(\upsilon)$; si l'approximation par différences finies est consistante alors,

$$\lim_{h\to 0} \Delta_{ij}(\upsilon, h) = \frac{\partial\varphi_i}{\partial\upsilon_j}(\upsilon), i, j = 1, ..., n.$$

On pose,

$$J(x,h) = (\Delta_{ij}(v,h)).$$

De manière générale, on a à considérer l'itération suivante:

$$v^{k+1} = v^k - J(v^k, h^k)^{-1} \cdot \varphi(v^k).$$

Le problème de la convergence de ce processus itératif est résolu grâce à un résultat du livre d'Ortega et Rheinboldt [4]; en effet si les pas de discrétisation h_{ij} sont petits et tendent vers zéro, on est assuré de la convergence de ce processus.

3 Résolution numérique

Pour résoudre le problème, nous considérons le couplage de la méthode de relaxation avec la méthode de tir, cette dernière étant destinée à calculer $p(0)$. Les étapes de la méthode sont:

1. Approximation initiale de la commande $u^0(t), t \in [0, t^*]$, et de l'état adjoint $p^0(0)$.
2. $r \leftarrow 0$
3. **Tant que** convergence$> \varepsilon$ **faire**
   - Détermination de l'état $x^r(t)$ et de l'état adjoint $p^r(t)$ composante par composante séquentiellement pour $t \in [0, t^*]$ par intégration numérique, pour le temps croissant.

$$\begin{cases} \frac{dx^r}{dt} = Ax^r + Bu^r(t), \\ x(0) = z. \end{cases} \tag{9}$$

$$\begin{cases} -\frac{dp^r}{dt} = A^t p^r + Qx^r, \\ p^r(0) \end{cases} \tag{10}$$

   où $p^r(0)$ est calculé par la méthode de tir.
   - Détermination de la commande $u^{r+1}(t)$

$$u^{r+1}(t) \leftarrow -N^{-1}B^t p^r(t). \tag{11}$$

   - Convergence $\leftarrow |u^{r+1}(t) - u^r(t)|$.
   - Détermination de la fonction de tir:

$$\varphi(p) = x^r(t^*) - x^*.$$

   - 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 2 Les étapes (9) à (11) 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.

4 Exemple numérique

On considère le problème suivant:

$$\begin{cases} \textit{Déterminer} \quad u \in U_{ad} \quad \textit{tel que}, \\ \\ J(u) \leq J(v), \ \forall \ v \in U_{ad}, \end{cases}$$

où

$$J(u) = \frac{1}{2} \int_0^{t^*} ((x - x_d)^2 + ku^2) dt. \tag{12}$$

sous les contraintes suivantes:

$$\begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ \\ x(0) = z \in X_0 = \{z \in \mathbf{R^2} : \mathbf{Gz} = \gamma, \ \mathbf{0 \leq z_1 \leq 20}, \ \mathbf{-10 \leq z_2 \leq 10}\}, \\ \\ \kappa x(T) = g, \end{cases} \tag{13}$$

où $G = (1, 2), \ \gamma = 3, \ \kappa = (1, 0), \ g = 2, \ t^* = 2, \ a > 0, \ b > 0.$
Dans le modèle mathématique précédent, x_d correspond à un état désiré, u est la commande, k permet de doser deux critères distincts à minimiser, un critère de précision et un autre de minimisation d'énergie et U_{ad} est l'ensemble des commandes admissibles. On prend $N = k.1$, avec évidemment $k > 0$ pour vérifier l'hypothèse de définie positivité de N.
On peut reformulé le problème (13) d'une autre manière:

$$\begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ \\ x_1(0) = z_1, \ x_2(0) = z_2 \ \texttt{tel que} \ z_1 + 2z_2 = 3, 0 \leq z_1 \leq 20, \ -10 \leq z_2 \leq 10, \\ \\ x_1(t^*) = 2. \end{cases} \tag{14}$$

La forme matricielle du système d'état s'écrit:

$$\begin{pmatrix} \dot{x}_1 \\ \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -b & a \\ \\ a & -b \end{pmatrix} \begin{pmatrix} x_1 \\ \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix} u,$$

et le critère de coût est reformulé comme suit:

$$J(u) = \frac{1}{2} \int_0^{t^*} [(x_1 - x_{1d}, x_2 - x_{2d})^t \begin{pmatrix} 1 & 0 \\ \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 - x_{1d} \\ \\ x_2 - x_{2d} \end{pmatrix} + ku^2] dt.$$

L'Hamiltonien du système est donné par:

$$H(x(t),p(t),u(t),t) = \frac{1}{2}((x_1 - x_{1d})^2 + (x_2 - x_{2d})^2 + ku^2) \\ +p_1(t)(-bx_1 + ax_2) + p_2(t)((ax_1 - bx_2) + p_2(t)u.$$

Les équations d'optimalité sont données par:

$$\begin{cases} \begin{cases} \frac{dx_1}{dt} = \frac{\partial H}{dp_1} = -bx_1 + ax_2, \\ \frac{dx_2}{dt} = \frac{\partial H}{dp_2} = ax_1 - bx_2, \\ z_1 + 2z_2 = 3, \end{cases} \\ \begin{cases} -\frac{dp_1}{dt} = \frac{\partial H}{dx_1} = -b\psi_1 + a\psi_2 + x_1 - x_{1d}, p_1(0) = \alpha_1,\ p_1(t^*) = \alpha_2, \\ -\frac{dp_2}{dt} = \frac{\partial H}{dx_2} = a\psi_1 - b\psi_2 + x_2 - x_{2d}, \quad p_2(0) = \alpha_3,\ p_2(t^*) = \alpha_4, \end{cases} \\ \frac{\partial H}{\partial u} = 0 = ku + p_2(t), \end{cases}$$

où les coefficients $\alpha_i, i = 1,4$ seront calculés à partir des conditions de transversalité:

$$\begin{cases} \text{il existe } (k_0,k_1) \neq 0, \text{tel que} \\ \varphi(0,z,t^*,x(t^*),k_0,k_1) = \ (\psi_0(0,z|k_0) + (\psi_1(t^*,x(t^*)|k_1), \end{cases}$$

où

$$\psi_0(0,z) = \ Gz - \gamma,$$

$$\psi_1(t^*,x(t^*)) = \ \kappa x(t^*) - g,$$

$$\varphi(0,z,t^*,x(t^*),k_0,k_1) = \ (Gz - \gamma|k_0) + (\kappa x(t^*) - g|k_1).$$

$$\begin{cases} p(0) = \frac{\partial \varphi}{\partial z}(0,z,t^*,x(t^*),k_0,k_1) = \ G^t k_0, \\ p(t^*) = -\frac{\partial \varphi}{\partial x(t^*)}(0,z,t^*,x(t^*),k_0,k_1) = \ -\kappa^t k_1, \end{cases}$$

$$\begin{cases} (p_1(0),p_2(0)) = (1,2)k_0 = (k_0,2k_0), \\ (p_1(t^*),p_2(t^*)) = -(1,0)k_1 = (-k_1,0). \end{cases}$$

Les coefficients $\alpha_i,\ i = \{1,2,3,4\}$, dépendent des paramètres $k_0,\ k_1$, qui seront calculés en utilisant la méthode de tir. Cette méthode consiste à résoudre le

système suivant en utilisant les conditions aux limites, $x_1(t^*) = 2$, $x_2(t^*) = 0$, $x_1(0) + 2x_2(0) = 3$:

$$\begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ z_1 + 2z_2 = 3, \\ k_1 = -p_1(t^*), \\ k_0 = p_1(0), \\ u = -\frac{p_2(t)}{k}. \end{cases}$$

La résolution de ce problème sera faite de deux manières différentes, analytiquement et numériquement.

Solution numérique En optimisation classique, minimiser une fonction H par rapport à un paramètre u implique dans un premier temps de chercher les points stationnaires du système, c'est-à-dire les valeurs u pour lesquelles $\frac{\partial H}{\partial u}(x, u, p) = 0$, puis d'étudier en ces points la positivité de la matrice Hessienne $\frac{\partial^2 H}{\partial^2 u}(x, u, p)$. On aura alors:

$$\frac{\partial^2 H}{\partial^2 u}(x, u, p) = k;$$

pour assurer la positivité de $\frac{\partial^2 H}{\partial^2 u}(x, u, p)$ on a nécessairement $k > 0$, c'est une condition naturelle pour le problème considéré.
On a à résoudre le système suivant:

$$\begin{cases} \dot{v_1} = -bv_1 + av_2 + u, \\ \dot{v_2} = av_1 - bv_2, \\ \dot{v_3} = bv_3 - av_4 - v_1 + x_{1d}, \\ \dot{v_4} = -av_2 + bv_4 - v_2 + x_{2d}, \\ u = -\frac{v_4}{k} \\ v_1(0) \in \mathbf{R}, v_{\mathbf{2}}(\mathbf{0}) \in \mathbf{R}, \\ v_3(0) \in \mathbf{R}, v_{\mathbf{4}}(\mathbf{0}) \in \mathbf{R}. \end{cases}$$

Soit $v(t, 0, 0, p_1, p_2)$ une solution du système au temps t avec les conditions initiales $(v_1(0), v_2(0), v_3(0), v_4(0))$.

On construit une fonction de tir qui est une équation algébrique non linéaire de la variable p à l'instant $t = 0$; cette fonction de tir est calculée par une procédure d'intégration numérique d'équations différentielles ordinaires (Euler, Runge-Kutta, etc); la fonction de tir s'écrit:

$$\varphi(\psi) = \begin{pmatrix} \upsilon_1(2,0,0,p_1,p_2) - 2 \\ \upsilon_2(2,0,0,p_1,p_2) \end{pmatrix}.$$

Le problème à résoudre s'écrit alors: Déterminer $p(0)$ tel que $\varphi(p(0))$ soit nul ce qui revient à déterminer $x(t^*)$ 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 φ.
2. l'algorithme de résolution de $\varphi(p) = 0$.

Appliquons l'algorithme vu au paragraphe (3-1) à notre exemple:

- Approximation initiale de l'état adjoint $p^{(0)}(0)$ et de la commande $u^{(0)}(t)$, $t \in [0, t^*]$.

- $r \leftarrow 0$.

- Tant que pas de convergence faire
- Détermination de l'état $x^{(r)}(t)$ et de l'état adjoint $p^{(r)}(t)$, successivement composante par composante, $t \in [0, t^*]$ par intégration numérique dans le sens direct de l'équation d'état et de l'équation d'état adjoint.

$$\begin{cases} \frac{dx_1^{(r)}}{dt} = -bx_1^{(r)} + ax_2^{(r)}, \\ \frac{dx_2^{(r)}}{dt} = ax_1^{(r)} - bx_2^{(r)} + u^{(r)}(t), \\ z_1 + 2z_2 = 3, \end{cases}$$

$$\begin{cases} -\frac{dp_1^{(r)}}{dt} = -bp_1^{(r)} + ap_2^{(r)} + x_1^{(r)} - x_{1d}, \, p_1^{(r)}(0) \\ -\frac{dp_2^{(r)}}{dt} = ap_1^{(r)} - bp_2^{(r)} + x_2^{(r)} - x_{2d}, \quad p_2^{(r)}(0) \end{cases}$$

  où $p_1^{(r)}(0)$, $p_2^{(r)}(0)$ sont calculés par la méthode de tir.
- Détermination de la commande $u^{(r+1)}(t)$:

$$u^{(r+1)}(t) \leftarrow -\frac{p_2^{(r)}(t)}{k}, \quad t \in [0, t^*].$$

- Convergence $\leftarrow$ norme $(u^{(r+1)} - u^{(r)})$.
- Détermination de la fonction de tir.

- Solution de l'équation de tir par la méthode de Newton.

– Actualiser la valeur de $p^{(r+1)}(0)$

$$p^{(r+1)}(0) \leftarrow p^{(r)}(0) + correction.$$

– $r \leftarrow r+1$.
fin de tant que.

La convergence de cette méthode itérative est un problème ouvert. Nous conjecturons qu'elle peut se montrer par des techniques de contraction analogue à celle utilisées dans [3]; elle découle certainement du fait que A est l'opposé d'une $M-$ matrice.

Solution exacte Pour calculer de manière analytique la solution optimale $x(t)$, et la commande optimale correspondante $u(t)$ du problème (12)-(14), nous avons utilisé les équations d'optimalité ainsi que la condition de transversalité sur $p(t)$. La solution des équations d'état $x_1(t)$ et $x_2(t)$ sont données par:

$$\begin{aligned} x_1(t) = \lambda[\frac{b^2-a^2+C_1^2}{2b} - C_1]e^{C_1 t} + \beta[\frac{b^2-a^2+C_1^2}{2b} + C_1]e^{-C_1 t} \\ + \mu[\frac{b^2-a^2+C_2^2}{2b} - C_2]e^{C_2 t} + \alpha[\frac{b^2-a^2+C_2^2}{2b} + C_2]e^{-C_2 t} \\ + \frac{b^2-a^2}{2b}\nu + \frac{1}{2}x_{1d} + \frac{a}{2b}x_{2d}. \end{aligned} \tag{15}$$

et

$$\begin{aligned} x_2(t) = \lambda[\frac{b^2-a^2-C_1^2}{2a} - C_1(\frac{b^2+a^2-C_1^2}{2ab})]e^{C_1 t} + \beta[\frac{b^2-a^2-C_1^2}{2a} \\ + C_1(\frac{b^2+a^2-C_1^2}{2ab})]e^{-C_1 t} + \mu[\frac{b^2-a^2-C_2^2}{2a} - C_2(\frac{b^2+a^2-C_2^2}{2ab})]e^{C_2 t} \\ + \alpha[\frac{b^2-a^2-C_2^2}{2a} + C_2(\frac{b^2+a^2-C_2^2}{2ab})]e^{-C_2 t} + \frac{a^2-b^2}{2a}\nu \\ + \frac{b}{2a}x_{1d} + \frac{1}{2}x_{2d}. \end{aligned} \tag{16}$$

La solution des équations d'état adjoint $p_1(t)$ et $p_2(t)$ sont données par:

$$p_1(t) = \lambda e^{C_1 t} + \beta e^{-C_1 t} + \mu e^{C_2 t} + \alpha e^{-C_2 t} + \nu. \tag{17}$$

$$\begin{aligned} p_2(t) = \lambda[\frac{a^2+b^2-C_1^2}{2ab}]e^{C_1 t} + \beta[\frac{a^2+b^2-C_1^2}{2ab}]e^{-C_1 t} + \mu[\frac{a^2+b^2-C_2^2}{2ab}]e^{C_2 t} \\ + \alpha[\frac{a^2+b^2-C_2^2}{2ab}]e^{-C_2 t} + \frac{a^2+b^2}{2ab}\nu + \frac{1}{2a}x_{1d} - \frac{1}{2b}x_{2d}. \end{aligned} \tag{18}$$

Les constantes étant déterminées par les conditions aux limites suivantes: $z_1 + 2z_2 = 3,\ x_1(t^*) = 2,\ p_1(0) = k_0,\ p_1(0) = 2k_0,\ p_1(t^*) = -k_1,\ p_2(t^*) = 0.$ Pour plus de détails nous renvoyons à [6].

Comparaison des deux approches L'algorithme numérique a été implémenté en utilisant le progiciel Matlab. En particulier, on a utilisé les fonctions *ode*45 et $fsolve$. Dans cet exemple, la méthode converge indépendemment du point de départ $p(0) = (-1.8706, 342.0562)$.

L'expérimentation numérique a été effectuée pour les valeurs de $a = 1$, $b = 3$ et $k = 2$. On déduit que la solution exacte et la solution numérique correspondent parfaitement (voir figure 1 et figure 2).

La performance de la procédure numérique est résumé dans le tableau suivant,

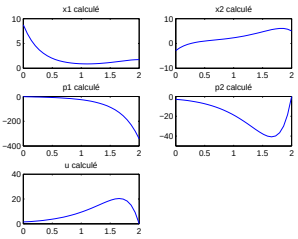


Fig. 1. solution approchée

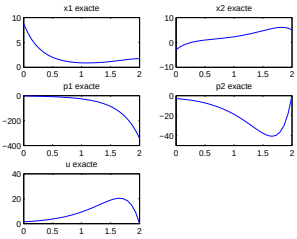


Fig. 2. solution exacte

pour différentes valeurs de k.

k	temps d'exécution C.P.U	Nombre d'itération
0.5	0.2028	3
1	0.2340	4
1.5	0.2496	4
2	0.2808	5
2.5	0.3276	5

Notons que l'algorithme converge rapidement, pour un nombre d'itérations très petit nécessaire pour atteindre la convergence. De plus, le temps de calcul utilisé est très faible.

References

1. D.Gien, B.Lang, J.C.Miellou, L.Raffort, P.Spiteri, Commande optimale de systèmes complexes, RAIRO Automatique, Systems Analysis and Control vol. 18, N=2, 1984, pp.209-224.
2. B.Lang, J.C.Miellou, P.Spiteri, Asynchronous relaxation algorithms for optimal control problems. Mathematical and Computers in Simulations 28(1986) 227-242.
3. J.C.Miellou, P.Spiteri, A parallel asynchronous relaxation algorithm for optimal control problem. Proceeding of the International Conference on Mathematical Analysis and its Apllications, Kuwait 1985.

4. J.M.Ortega and W.C.Rheinboldt, Iterative solution of nonlinear equations in several variables. Academic Press, New York, 1970.
5. E.Trelat, Contrôle optimal: théorie et applications, Vuibert, collection Mathématiques Concrêtes, 2005.
6. K.Louadj, P.Spiteri et M.Aidene, Résolution d'un problème de contrôle optimal avec une contrainte sur l'état final et initial par la méthode de relaxation, 2010.
7. K.Louadj, P.Spiteri, M.Aidene et F.Messine, La solution analytique et numérique d'un problème de contrôle optimale à bicritères, ISOR'11, USTHB, Algérie.

Primal-Dual Method for Solving a Linear-Quadratic Multivariate Optimal Control Problem

BIBI Mohand Ouamer and KHIMOUM Noureddine

Laboratory of Modelisation and Optimization of Systems (LAMOS)
University of Béjaia 06000, Algeria.
mohandbibi@yahoo.fr khimoumnordine@hotmail.com

Abstract. In this paper, we intend to minimize a quadratic terminal functional on the trajectories of a linear dynamic system. The admissible controls are bounded multivariate functions, which are piecewise constant and satisfy terminal phase constraints.
The algorithm of optimization is constructed on the basis of new methods of linear and quadratic programming and of their applications in the constructive theory of optimal control.
An iteration of the primal-dual algorithm is based on the principle of diminution of suboptimality estimate that permits to estimate the difference between the current value and the optimum value of the cost function. In the present paper, we will develop a constructive method for the case, where the control is a multivariate function.

Keywords: Optimal Control, Support Maximum Principle, Support Auxiliary Problem, Dual Method, Finishing Procedure.

1 Problem Statement and Definitions

Consider the following optimal control problem:

$$\min \quad J(u) = c^T x(t_*) + \frac{1}{2} x^T(t_*) D x(t_*), \tag{1}$$

$$\dot{x}(t) = Ax(t) + Bu(t), \quad t \in T, \quad x(0) = x_0, \tag{2}$$

$$Gx(t_*) = g, \quad d^- \leq u(t) \leq d^+, \quad t \in T = [0, t_*], \tag{3}$$

where A and D are square matrices of order n, with $D^T = D \geq 0$, B is an $n \times r$-matrix, G an $m \times n$-matrix, with $\text{rang}(G) = m < n$. The functions $u(t) = (u_1(t), \ldots, u_r(t))$ and $x(t) = (x_1(t), \ldots, x_n(t))$ represent respectively the value of the multivariate control and the vector of state at time t; c, g, d^- and d^+ are vectors of corresponding dimension. Furthermore, we suppose that the dynamic system (2) is controllable in the sense of Kalman.

The piecewise continuous function $u(t), t \in T$, and the corresponding trajectory $x(t), t \in T$, are called the admissible control and the admissible trajectory respectively if they satisfy the constraints (2) and (3). We choose on the set T a subset of isolated moments $T_s = \{t_j, j \in J_s\}$, $J_s = \{1, \ldots, j_s\}$, $j_s \leq m$ and for every moment t_j in T_s we associate a set of indices $I_j \subset I$ such that $\sum_{j \in J_s} |I_j| = m$, where $I = \{1, 2, \ldots, r\}$. Set $I_s = \{I_j, j \in J_s\}$, $Q_s = \{I_s, T_s\}$ and form the matrix $P_s = P(Q_s) = (p_i(t_j), i \in I_j, j \in J_s)$, where $p_i(t)$ is the i^{th} column of the matrix $P(t) = Gq(t), t \in T$. The matrix function $q(t), t \in T$, verifies the differential equation $\dot{q}(t) = -Aq(t), t \in T, q(t_*) = B$.

Definition 1. *The set $Q_s = \{I_s, T_s\}$ is called a support control of the problem (1)-(3) if the matrix P_s is regular. The pair $\{u, Q_s\}$ formed by the admissible control u and the support Q_s is called an admissible support control.*

Definition 2. *The support control $\{u, Q_s\}$ is said to be non degenerate, if for any moment t_j of T_s and for any index $i \in I_j, j \in J_s$, one of the two following conditions is satisfied:*
i) In the neighborhood of t_j, the component $u_i(t), t \in T$, is non critical.
ii) The point t_j is a point of discontinuity of the function $u_i(t), t \in T$.

2 Support Maximum Principle

To the support control $\{u, Q_s\}$, we assign the vector of multipliers

$$y^T = \left((Dx(t_*) + c)^T q_i(t_j), i \in I_j, j \in J_s\right)^T P_s^{-1}, \tag{4}$$

and the solution of the conjugate system

$$\dot{\psi}(t) = -A^T \psi(t), t \in T, \psi(t_*) = G^T y - Dx(t_*) - c,$$

where $q_i(t)$ is the i^{th} column of the $n \times r$-matrix $q(t)$. We form the hamiltonian:

$$H(x, \psi, u) = \psi^T (Ax + Bu).$$

Theorem 1 (Support Maximum Principle). *[7] Let $\{u, Q_s\}$ be an admissible support control. For the optimality of u, it's sufficient and, in the case of the non degeneracy of $\{u, Q_s\}$, also necessary that the following maximum condition holds:*

$$H(x(t), \psi(t), u(t)) = \max_{d^- \leq v \leq d^+} H(x(t), \psi(t), v), t \in T. \tag{5}$$

3 Suboptimality Criterion

Let $\{u, Q_s\}$ a support control where $u(t), t \in T$, is an admissible control and $x(t), t \in T$, its corresponding trajectory. Consider another admissible control $\bar{u}(t) = u(t) + \triangle u(t), t \in T$, and its corresponding trajectory $\bar{x}(t), t \in T$. Then the increment of the functional defined in (1) is:

$$\begin{aligned}\triangle J(u) &= J(\bar{u}) - J(u) \\ &= \int_0^{t_*} (Dx(t_*) + c)^T q(t) \triangle u(t) dt + \Gamma \\ &= \sum_{i=1}^{r} \int_0^{t_*} (Dx(t_*) + c)^T q_i(t) \triangle u_i(t) dt + \Gamma, \end{aligned} \tag{6}$$

where $\Gamma = \frac{1}{2} \triangle x^T(t_*) D \triangle x(t_*) \geq 0$, $q_i(t)$ is the i^{th} column of the matrix $q(t) = F(t_*)F^{-1}(t)B$, $F(t) = AF(t)$, $F(0) = I_n$, I_n, is an identity matrix of order n.

The continuous function $\Delta^T(t) = (\Delta_i(t), i \in I)^T = \psi^T(t)B, t \in T$, is called a cocontrol, and we define

$$T_i^+ = \{t \in T : \Delta_i(t) > 0\} \text{ and } T_i^- = \{t \in T : \Delta_i(t) < 0\}, i = \overline{1, r}.$$

So the increment of the functional takes the following final form:

$$\begin{aligned}\triangle J(u) &= -\int_0^{t_*} \triangle^T(t) \triangle u(t) dt + \Gamma \\ &= -\sum_{i=1}^{r} \int_0^{t_*} \triangle_i(t) \triangle u_i(t) dt + \Gamma \end{aligned} \tag{7}$$

The following number:

$$\beta(u, Q_s) = \sum_{i=1}^{r} \int_{T_i^+} \Delta_i(t)[d_i^+ - u_i(t)]dt + \sum_{i=1}^{r} \int_{T_i^-} \Delta_i(t)[d_i^- - u_i(t)]dt \tag{8}$$

is called the suboptimality estimate since it always verifies the following inequality $J(u) - J(u^0) \leq \beta(u, Q_s)$. Thus, for $\beta(u, Q_s) \leq \varepsilon$, the admissible control u is an ε-optimal one.

4 Algorithm

The proposed algorithm is constructed without quantization of the continuous dynamic system (2) and consists of three procedures : control transformation, support transformation and finishing. The procedure of control transformation aims at solving a finite dimensional quadratic programming problem called the

support auxiliary problem [5], whose the solution allows to construct an admissible support control $(\bar{u}, \tilde{Q}_s)$ such that $J(\bar{u}) < J(u)$. The second transformation is used to obtain via a dual control method a new support $\bar{Q}_s$, satisfying $\beta(\bar{u}, \bar{Q}_s) \leq \beta(\bar{u}, \tilde{Q}_s)$. These transformations should be repeated until the conditions of the passage to the finishing procedure are obtained. The latter consists in solving a system of equations by means the Newton method in order to get the optimal control.

4.1 Control Transformation

Suppose that $\varepsilon \geq 0$ is given and let $\{u, Q_s\}$ be an initial support control such that $\beta(u, Q_s) > \varepsilon$. Let's do one iteration $\{u, Q_s\} \to \{\bar{u}, \tilde{Q}_s\}$ such that $J(\bar{u}) < J(u)$. Choose two numbers $\alpha > 0$ and $h > 0$ (parameters of the algorithm) and construct the sets

$$T_\alpha = \{t \in T : \eta(t) \leq \alpha\},\ T_* = T \setminus T_\alpha,$$

where $\eta(t) = \min\limits_{i \in I} |\triangle_i(t)|, t \in T$. Divide the set T_α into N intervals $[\tau_j, \tau^j], j = \overline{1, N}$, in such a way that $\tau_j < \tau^j \leq \tau_{j+1}$, $\tau^j - \tau_j \leq h$, $T_s \subset \{\tau_j, j = \overline{1, N}\}$, $u_i(t) = u_{ij} = \text{const}, t \in [\tau_j, \tau^j[, j = \overline{1, N}, i = \overline{1, r}$.

Calculate

$$\beta_{ij} = -\int_{\tau_j}^{\tau^j} \triangle_i(t)dt,\ z_{ij} = \int_{\tau_j}^{\tau^j} q_i(t)dt,\ v_{ij} = \int_{\tau_j}^{\tau^j} p_i(t)dt,\ j = \overline{1, N}, i = \overline{1, r},$$

$$\beta_{N+1} = -\sum_{i=1}^{r} \int_{T_*} \triangle_i(t)\alpha_i(t)dt,\ z_{N+1} = \sum_{i=1}^{r} \int_{T_*} q_i(t)\alpha_i(t)dt,\ v_{N+1} = \sum_{i=1}^{r} \int_{T_*} p_i(t)\alpha_i(t)dt,$$

where

$$\alpha_i(t) = \begin{cases} d_i^+ - u_i(t), \text{ if } \triangle_i(t) > \alpha, \\ d_i^- - u_i(t), \text{ if } \triangle_i(t) < -\alpha. \end{cases}$$

Assume $S = \{1, 2, \ldots, N, N{+}1\},\ l = \{l_{11}, l_{12}, \ldots, l_{1N}, \ldots, l_{r1}, \ldots, l_{rN}, l_{N+1}\}$, $\beta = \{\beta_{11}, \beta_{12}, \ldots, \beta_{1N}, \ldots, \beta_{r1}, \ldots, \beta_{rN}, \beta_{N+1}\},\ Z = \{z_{11}, z_{12}, \ldots, z_{1N}, \ldots, z_{r1}, \ldots, z_{rN}, z_{N+1}\}$.
The vectors l, β are of dimension $(Nr + 1)$ and the matrix Z is of order $N \times (Nr + 1)$.

We consider the following support auxiliary problem:

$$\begin{aligned} &\min \quad \beta^T l + \frac{1}{2} l^T Z^T D Z l, \\ &\sum_{i=1}^{r} \sum_{j=1}^{N} v_{ij} l_{ij} + v_{N+1} l_{N+1} = 0, \\ &d_i^- - u_{ij} \leq l_{ij} \leq d_i^+ - u_{ij}, j = \overline{1, N}, i = \overline{1, r}, 0 \leq l_{_{N+1}} \leq 1. \end{aligned} \tag{9}$$

We can solve the problem (9) by the method [8] and let l^{ε_1} be the ε_1-optimal feasible solution. Then the admissible control $\bar{u}(t), t \in T$, defined by the relations:

$$\bar{u}_i(t) = u_{ij} + l_{ij}^{\varepsilon_1}, t \in [\tau_j, \tau^j[, j = \overline{1,N},$$
$$\bar{u}_i(t) = u_i(t) + l_{N+1}^{\varepsilon_1}\alpha_i(t), t \in T_*, i = \overline{1,r},$$

verifies the inequality

$$J(\bar{u}) \leq J(u).$$

4.2 Support Transformation

Let $\{\bar{u}, \tilde{Q}_s\}$ be the support control found after the resolution of the problem (9). Calculate by the formulas (4) the cocontrol $\tilde{\triangle}^T(t) = \tilde{\psi}^T(t)B, t \in T$, corresponding to the support control $\{\bar{u}, \tilde{Q}_s\}$ and construct the quasi-control $\tilde{\omega}(t) = (\tilde{\omega}_1(t), \ldots, \tilde{\omega}_r(t), t \in T)$, where

$$\begin{cases} \tilde{\omega}_i(t) = d_i^+, & \text{if } \tilde{\triangle}_i^T(t) > 0; \\ \tilde{\omega}_i(t) = d_i^-, & \text{if } \tilde{\triangle}_i^T(t) < 0; \quad i = \overline{1,r}. \\ \tilde{\omega}_i(t) \in [d_i^-, d_i^+], & \text{if } \tilde{\triangle}_i^T(t) = 0. \end{cases}$$

The quasi-trajectory $\tilde{\kappa}(t), t \in T$, verifies the equation $\dot{\tilde{\kappa}}(t) = A\tilde{\kappa}(t) + B\tilde{\omega}(t)$, $t \in T$, $\tilde{\kappa}(0) = x_0$.

Introduce two parameters $\mu_1 > 0$ and $\mu_2 > 0$ and denote $T_i^* = \{t \in T : \text{ sign } \tilde{\triangle}_i(t) \neq \text{ sign } \triangle_i(\tilde{\omega}, t)\}$, $i = \overline{1,r}$, where $\triangle_i(\tilde{\omega}, t), t \in T$, is the cocontrol constructed with the pair $\{\tilde{\omega}, \tilde{Q}_s\}$. If

$$\|G\tilde{\kappa}(t_*) - g\| \leq \mu_1, \qquad |T_i^*| \leq \mu_2, \qquad i = \overline{1,r}, \tag{10}$$

then pass to the finishing procedure described bellow.

Suppose that the conditions (10) are not fulfilled. If $G\tilde{\kappa}(t_*) = g$, then we construct a new admissible control in the form

$$\bar{\bar{u}}(t) = \bar{u}(t) + \theta(\tilde{\omega}(t) - \bar{u}(t)), \; t \in T, \; 0 \leq \theta \leq 1,$$

where $\theta = \min\{1, \theta_{\gamma_0}\}$, with

$$\gamma_0 = (\tilde{\kappa}(t_*) - \bar{x}(t_*))^T D(\tilde{\kappa}(t_*) - \bar{x}(t_*)),$$

$$\theta_{\gamma_0} = \begin{cases} \dfrac{\beta(\bar{u}, \tilde{Q}_s)}{\gamma_0}, & \text{if } \gamma_0 > 0, \\ \infty, & \text{if } \gamma_0 = 0, \end{cases}$$

where $\bar{x}(t), t \in T$, is the corresponding trajectory to the control $\bar{u}$.
We have the relation

$$J(\bar{\bar{u}}) - J(\bar{u}) = -\theta\beta(\bar{u}, \tilde{Q}_s) + \theta^2 \frac{\gamma_0}{2}.$$

Remark 1. If $\gamma_0 = 0$, then $\theta = 1$ and $\bar{\bar{u}}$ is optimal. If $\theta < 1$, then we will start a new iteration with the support control $\{\bar{\bar{u}}, \tilde{Q}_s\}$.

If $G\tilde{\kappa}(t_*) \neq g$, we will consider the dual problem of the problem (1)-(3) defined by:

$$\begin{aligned} \max_{\lambda} \quad & L(\lambda) = -\frac{1}{2}\kappa^T D\kappa + y^T g - \psi^T(0)x_0 + \int_0^{t_*} v^T(t)d^-dt - \int_0^{t_*} w^T(t)d^+dt \\ & \psi^T(t)B + v^T - w^T(t) = 0,\ v(t) \geq 0,\ w(t) \geq 0,\ t \in T, \\ & \dot{\psi} = -A^T\psi, \psi(t_*) = G^T y - D\kappa - c, \end{aligned} \tag{11}$$

where $\lambda = (\kappa, y, v(t), w(t), t \in T)$, $\kappa \in \mathbb{R}^n$, $y \in \mathbb{R}^m$.

By fixing the vector $\kappa = \bar{x}(t_*)$, the problem (11) becomes linear

$$\begin{aligned} \max \quad & L_{\bar{u}}(\lambda) = -\frac{1}{2}\bar{x}^T(t_*)D\bar{x}(t_*) + y^T g - \psi^T(0)x_0 + \int_0^{t_*} v^T(t)d^-dt - \int_0^{t_*} w^T(t)d^+dt \\ & \psi^T(t)B + v^T - w^T(t) = 0,\ v(t) \geq 0,\ w(t) \geq 0,\ t \in T, \\ & \dot{\psi} = -A^T\psi, \psi(t_*) = G^T y - D\bar{x}(t_*) - c, \end{aligned} \tag{12}$$

Calculate the $m-$vector

$$\gamma(\tilde{I}_s, \tilde{J}_s) = (\gamma_{ij}, i \in \tilde{I}_j, j \in \tilde{J}_s) = P^{-1}(\tilde{Q}_s).[g - G\tilde{\kappa}(t_*)],$$

and set

$$|\gamma_{i_0 j_0}| = \max|\gamma_{ij}|, i \in \tilde{I}_j, j \in \tilde{J}_s.$$

Construct another according dual solution $\bar{\lambda} = (\bar{y}, \bar{v}(t), \bar{w}(t), t \in T)$ of the problem (12), where $\bar{\lambda} = \tilde{\lambda} + \sigma\triangle\lambda$; $\bar{y} = \tilde{y} + \sigma\triangle y$; $\bar{v}(t) = \tilde{v}(t) + \sigma\triangle v(t)$; $\bar{w}(t) = \tilde{w}(t) + \sigma\triangle w(t)$; $\delta(t) = (\delta_1(t), \delta_2(t), \ldots, \delta_r(t))$, $t \in T$, $\sigma \geq 0$. Set

$$\triangle y^T = e^T P^{-1}(\tilde{Q}_s)\ \text{sign}\ \gamma_{i_0 j_0}, \delta(t) = \triangle y^T P(t),$$

where $e = (e_{ij}, i \in \tilde{I}_j,\ j \in \tilde{J}_s)$, $e_{ij} = 0$, $(i,j) \neq (i_0, j_0)$, $e_{i_0 j_0} = 1$.
Determinate the fonctions $\sigma_i(t)$, $t \in T$, $i = \overline{1,r}$:

$$\sigma_i(t) = \begin{cases} -\dfrac{\tilde{\triangle}_i(t)}{\delta_i(t)}, \text{ si } \tilde{\triangle}_i(t)\delta_i(t) < 0; \\ 0, \qquad \text{si } (\tilde{\triangle}_i(t) = 0,\ \delta_i(t) > 0,\ \tilde{\omega}_i(t) \neq d_i^+) \text{ ou } (\tilde{\triangle}_i(t) = 0,\ \delta_i(t) < 0,\ \tilde{\omega}_i(t) \neq d_i^-)\ ; \\ \infty, \qquad \text{sinon.} \end{cases}$$

Construct the sets:

$$\begin{cases} T(i,\sigma) = \{t \in T : \sigma_i(t) < \sigma\}, \\ T^+(i,\sigma) = \{t \in T(i,\sigma) : \hat{\triangle}_i(t) > 0, \qquad i = \overline{1,r} \\ T^-(i,\sigma) = \{t \in T(i,\sigma) : \hat{\triangle}_i(t) < 0. \end{cases} \tag{13}$$

Remark that:

$$\text{sign } \bar{\triangle}_i(t) = \begin{cases} -\text{sign } \tilde{\triangle}_i(t), & \text{si } t \in T(i,\sigma); \\ \text{sign } \tilde{\triangle}_i(t), & \text{si } t \in T \setminus T(i,\sigma). \end{cases}$$

Calculate the quality criterion of the problem (12):

$$\begin{aligned} L_{\bar{u}}(\bar{\lambda}) - L_{\bar{u}}(\tilde{\lambda}) &= \sigma|\gamma_{i_0 j_0}| + \sigma \sum_{i=1}^{r} \int_0^{t_*} \delta_i(t)\tilde{\omega}_i(t)dt + \sigma \sum_{i=1}^{r} \int_0^{t_*} (\triangle v_i(t) d_i^- - \triangle w_i(t) d_i^+)dt \\ &= \sigma|\gamma_{i_0 j_0}| + \sum_{i=1}^{r} (d_i^+ - d_i^-) \left(\int_{T^+(i,\sigma)} [\tilde{\triangle}_i(t) + \sigma\delta_i(t)]dt - \int_{T^-(i,\sigma)} [\tilde{\triangle}_i(t) + \sigma\delta_i(t)]dt \right). \end{aligned}$$

Therefore, the rate of change of the quality criterion in the direction $\triangle\lambda$ is:

$$\alpha(\sigma) = |\gamma_{i_0 j_0}| - \sum_{i=1}^{r} (d_i^+ - d_i^-) \int_{T(i,\sigma)} |\delta_i(t)|dt \tag{14}$$

We have by construction :

$$\alpha(0) = |\gamma_{i_0 j_0}| > 0, \alpha(\bar{\sigma}) < \alpha(\sigma), \forall \bar{\sigma} > \sigma$$

Found $\sigma_0 \geq 0$ such that $\alpha(\sigma_0 - \xi) > 0$, $\alpha(\sigma_0 + 0) \leq 0$, for all $0 < \xi < \sigma_0$.
Let $(\tau_{j_1}, i_1) \in \{(t,i) : t \in T, i = \overline{1,r}\} \setminus \{(\tau_j, i) : i \in \tilde{I}_j, j \in \tilde{J}_s\}$ a pair such that

$$\tilde{\triangle}_{i_1}(\tau_{j_1}) + \sigma_0 \delta_{i_1}(\tau_{j_1}) = 0, \delta_{i_1}(\tau_{j_1}) \neq 0.$$

Change the support $\tilde{Q}_s$ by $\hat{Q} = \{\hat{I}_s, \hat{T}_s\}$, $\hat{I}_s = \{\hat{I}_j, j \in \hat{J}_s\}$, $\hat{T}_s = \{\tau_j, j \in \hat{J}_s\}$, as follows:

If $j_1 \notin \tilde{J}_s$ and $\tilde{I}_{j_0} = \{i_0\}$, then, we set $\hat{J}_s = (\tilde{J}_s \setminus j_0) \cup j_1$, $\hat{I}_j = \tilde{I}_j$, $j \in \hat{J}_s \setminus j_1$, $\hat{I}_{j_1} = \{i_1\}$.
If $j_1 \notin \tilde{J}_s$ and $|\tilde{I}_{j_0}| > 1$, then, we set $\hat{J}_s = \tilde{J}_s \cup j_1$, $\hat{I}_j = \tilde{I}_j$, $j \in \tilde{J}_s \setminus j_0$, $\hat{I}_{j_0} = \tilde{I}_{j_0} \setminus i_0$, $\hat{I}_{j_1} = \{i_1\}$.
If $j_1 \in \tilde{J}_s$ and $|\tilde{I}_{j_0}| > 1$, then, we set $\hat{J}_s = \tilde{S}_s$, $\hat{I}_j = \tilde{I}_j$, $j \neq j_0$, $j \neq j_1$, $\hat{I}_{j_0} = \tilde{I}_{j_0} \setminus i_0$, $\hat{I}_{j_1} = \tilde{I}_{j_1} \cup i_1$.
If $\tilde{I}_{j_0} = \{i_0\}$, $j_1 \in \tilde{S}_s$, then, we set $\hat{J}_s = \tilde{J}_s \setminus j_0$, $\hat{I}_j = \tilde{I}_j$, $j \neq j_1$, $\hat{I}_{j_1} = \tilde{I}_{j_1} \cup i_1$.

By changing the support $\tilde{Q}_s$ by a new support $\bar{Q}_s$, the value of the objective function in the problem (12) will increase in quantity $\int_0^{\sigma_0} \alpha(\sigma)d\sigma$. So, we have

$$\beta(\bar{u}, \bar{Q}_s) = \beta(\bar{u}, \tilde{Q}_s) - \int_0^{\sigma_0} \alpha(\sigma)d\sigma. \tag{15}$$

If $\beta(\bar{u}, \bar{Q}_s) \leq \varepsilon$, then we stop the resolution process of the problem (1)-(3). In the contrary case, we will start a new iteration with $\{\bar{u}, \bar{Q}_s\}$ or we will pass to the finishing procedure if conditions (10) are fulfilled for $\{\bar{u}, \bar{Q}_s\}$.

4.3 Finishing Procedure

Suppose that the conditions (10) are verified for the quasi-control $\tilde{\omega}(t), t \in T$, and the quasi-trajectory $\tilde{\kappa}(t), t \in T$, corresponding to the support control $\{\bar{u}, \tilde{Q}_s\}$. Let $\{t_1, \ldots, t_s\}$, $m \leq s \leq n$, be the set of all points of T such that $0 \leq t_1 < \ldots < t_s \leq t_*$, $\tilde{\triangle}_i(t_j) = 0, i \in I_j \neq \emptyset$, $\tilde{\triangle}_i(t_j) \neq 0$, $i \in I \setminus I_j$, $j \in J = \{1, \ldots, s\}$. Consider the case where $|I_j| = 1$ and $\dot{\tilde{\triangle}}_i(t_j) \neq 0, i \in I_j, j \in J$. The finishing procedure consists in looking for the solution $y, \tau = (\tau_j, j \in J)$ of $(m + s)$ equations

$$\begin{aligned} &\sum_{i \in I_j, j \in J} d_{ij} \int_{t_j}^{\tau_j} p_i(t)dt + g - G\tilde{\kappa}(t_*) = 0, \\ &\triangle_i(y, \tau, \tau_j) = 0, i \in I_j, j \in J, \end{aligned} \tag{16}$$

where
$d_{ij} = (d_i^+ - d_i^-)$ sign $\dot{\tilde{\triangle}}_i(t_j)$, $\triangle_i(y, \tau, t) = y^T p_i(t) - (D\kappa(t_*, \tau) + c)^T q_i(t)$;
$\kappa(t, \tau), t \in T$, is the trajectory corresponding to the control $\omega(t, \tau) = (\omega_i(t, \tau), i \in I)$, $t \in T$, defined as follows : let $t_{j_1}, \ldots, t_{j_p}$, $0 \leq p \leq s$, be the roots of the component $\tilde{\triangle}_i(t)$ on the set T. If $p = 0$, then we put $\omega_i(t, \tau) = \tilde{\omega}_i(t), t \in T$. If $p \geq 1$, we define $\omega_i(t, \tau)$ as follows:

$$\omega_i(t, \tau) = \begin{cases} d(i, j_1), & t \in [0, \tau_{j_1}[; \\ d(i, j_q), & t \in [\tau_{j_{q-1}}, \tau_{j_q}[, q = \overline{2, p}; \\ d_i^+ + d_i^- - d(i, j_p), & t \in [\tau_{j_p}, t_*], \end{cases}$$

where $d(i, j) = \frac{1}{2}(d_i^+ + d_i^- - d_{ij})$.

We can solve the system (16) by the Newton method, starting with the initial approximation $y^{(0)} = y_0$, $\tau^{(0)} = (t_j, j \in J)$, where y_0 is the vector (4), constructed with the pair $\{\tilde{\omega}, \tilde{Q}_s\}$. So, for enough small parameters μ_1 and μ_2, the optimal control has the form:

$$u^0(t) = \omega(t, \tau), t \in T.$$

5 Numerical Example

For illustration of the described algorithm, consider the problem of the minimal distance between two material points, whose the equations of motion are as follows:

$$\begin{cases} \ddot{y}_1 = u_1, & y_1(0) = 0, & \dot{y}_1(0) = 0, \\ \ddot{y}_2 = u_2, & y_2(0) = 2, & \dot{y}_2(0) = 0. \end{cases} \tag{17}$$

The controls $u_1(.)$ and $u_2(.)$ are subject to $|u_1(t)| \leq 1$, $|u_2(t)| \leq 1$, $t \in T = [0, t_*]$, $t_* = 1$. It's required to find admissible controls $u_1^0(.)$, $u_2^0(.)$ such that at the terminal instant the material points will have a same speed and the distance between them will be minimal.

If we set $x_1 = y_1$, $x_2 = \dot{y}_1$, $x_3 = y_2$, $x_4 = \dot{y}_2$, $x = (x_1, x_2, x_3, x_4)$, $u = (u_1, u_2)$, then this example presents a special case of problem (1)-(3), with $c = 0$, $g = 0$, $x_0 = (0, 0, 2, 0)$, $d^- = (-1, -1)$, $d^+ = (1, 1)$, $G = (0, 1, 0, -1)$,

$$D = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Calculate the matrix $q(t) = (q_1(t), q_2(t))$ and the vector $p(t) = (p_1(t), p_2(t))$, $t \in T$:

$$q(t) = \begin{pmatrix} 1 - t & 0 \\ 1 & 0 \\ 0 & 1 - t \\ 0 & 1 \end{pmatrix}, \quad p(t) = (1, -1).$$

The control $u(t) = (u_1(t), u_2(t)) = 0$, $t \in [0, 1]$, is admissible. Then we have $x(1) = (0, 0, 2, 0)$, $J(u) = 2$.
To the initial admissible control, we assign $Q_s = \{I_s, T_s\}$, $I_s = \{I_1\}$, $I_1 = \{1\}$, $T_s = \left\{\frac{1}{4}\right\}$. With the pair $\{u, Q_s\}$, we obtain $P_s = 1$, $y = -\frac{3}{2}$, $\triangle(t) = (\triangle_1(t), \triangle_2(t))$, $\triangle_1(t) = -\triangle_2(t) = -2t + \frac{1}{2}$, $t \in [0, 1]$. We choose $\varepsilon = 0$ and we calculate the suboptimality estimate $\beta(u, Q_s) = 1,25 > \varepsilon$. Then we construct the quasicontrol $\omega(t) = (\omega_1(t), \omega_2(t))$, $t \in [0, 1]$:

$$\omega_1(t) = 1,\ \omega_2(t) = -1,\ t \in \left[0, \frac{1}{4}\right[,$$
$$\omega_1(t) = -1,\ \omega_2(t) = +1,\ t \in \left[\frac{1}{4}, 1\right],$$

and the quasitrajectory $\kappa(t), t \in [0, 1]$. Thus

$$\kappa(1) = \left(\frac{-1}{16}, \frac{-1}{2}, \frac{33}{16}, \frac{1}{2}\right), \quad y(\omega) = \frac{-51}{32},$$

$$\triangle(\omega, t) = (\triangle_1(\omega, t), \triangle_2(\omega, t)), \quad \triangle_1(\omega, t) = -\triangle_2(\omega, t) = -\frac{17}{8}t + \frac{17}{32}, \quad t \in [0, 1].$$

We remark that the functions $\triangle_i(t)$, $\triangle_i(\omega, t)$, $i = 1, 2$, have the same roots and signs. Since $|G\kappa(1) - 0| = 1$, we can pass to the finishing procedure, with $\mu_1 = 1$ and $\mu_2 = 0$. We solve the system (16) by the Newton method, starting with $y^{(0)} = y(\omega) = -\frac{51}{52}$, $\tau^{(0)} = \frac{1}{4}$. Hence, $y^0 = -\frac{3}{4}$, $\tau^0 = \frac{1}{2}$ and the optimal support

control $\{u^0, Q_s^0\}$, with $T_s^0 = \left\{\frac{1}{2}\right\}$, has the form:

$$u_1^0(t) = 1,\ u_2^0(t) = -1,\ t \in \left[0, \frac{1}{2}\right[,$$
$$u_1^0(t) = -1,\ u_2^0(t) = 1,\ t \in \left[\frac{1}{2}, 1\right].$$

Thus, at the terminal instant the value $J(u^0)$ and the minimal distance d between the two material points are equal respectively to $J(u^0) = \frac{9}{8}$ and $d = \frac{3}{2}$, with $\dot{y}_1(1) = \dot{y}_2(1) = 0$.

References

1. R. Gabasov and F. M. Kirillova. New Linear Programming Methods and there Applications to Optimal Control Problems. Proc. I Workshop on Control Applications of Nonlinear Programming, Denver, USA, 1979. Pergamon Press, 1980.
2. R. Gabasov, F. M. Kirillova and O. I. Kostyukova. Algorithms for Solving Linear Optimal Control Problems, Doklady AN SSSR, 1984, T. 274, No. 5, pp. 1048-1052.
3. S. V. Gnevko. Optimization Methods of Dynamic System with several Entries. Izvestia ANBSSR. Seria Fiz-Mat. naouk, 1985, No. 5, pp. 26-32.
4. M. O. Bibi and O. I. Kostyukova. Optimization of a Linear Control System with respect to Quadratic Terminal cost Functional. Doklady AN BSSR, 1986, T. 30, No. 1, pp. 16-19.
5. N. V. Balashevich, R. Gabasov, and F. M. Kirillova, Numerical Methods for Open Loop and Closed Loop Optimization of Linear Control Systems, Computational Mathematics and Mathematical Physics, 2000, Vol. 40, pp. 799-819.
6. R. Gabasov, F. M. Kirillova and N. S. Pavlenok. Constructing Open-Loop and Closed-Loop Solutions of Linear-Quadratic Optimal Control Problems, Computational Mathematics and Mathematical Physics, 2008, Vol. 48, No. 10, pp. 1715-1745.
7. M. O. Bibi. Optimal Control of a Quadratic Problem with a Piecewise Linear Entry. Proc. of the 5th International Conference of Operations research (CIRO'10), from 24 to 27 may 2010, University of Marakkech, Morroco, pp. 1-4.
8. B. Brahmi and M. O. Bibi. Dual Support Method for Solving Convex Quadratic Programs. - Optimization, 2010, Vol. 59, No. 6, pp. 851-872.
9. R. Gabasov, F. M. Kirillova and O. I. Kostyukova. Direct Accurate Method to Optimize a Linear Dynamic Multi-input System. -Avtomatika i Telemechanika, 1986, 6, 6-13.
10. O. I. Kostyukova. Optimization of a Linear Dynamic Multi-input System. -IZV. AN BSSR, Seria Fiz-mat. Nauk, 1990 No. 5, pp. 16-21.
11. A. S. Chernushevich. Method of Support Problems for Solving a Linear-Quadratic Problem of Terminal Control. -Int. J. Control, 1990, 52(6), 1475-1488.

Traitement d'images

Cartographie des feux de forêts par segmentation des images satellitaires

Habib Mahi, Nabila Benkablia, Sarah Rabia Cheriguène

Centre des Techniques Spatiales, Division Observation de la Terre, 1, Avenue de la Palestine, B.P. 13, 31200 Arzew Algérie
mahihabib@yahoo.fr, benkablia@gmail.com, sarah.cheriguene@yahoo.fr

Résumé. Le but de cette étude est la mise en place d'une chaine de traitement semi-automatisée pour la cartographie des zones incendiées en se basant uniquement sur des images satellitaires acquises après un feu de forêt. Le processus est composé de quatre phases : 1) Application de l'algorithme de segmentation JSEG, 2) Fusion de régions par la règle de Minimum d'hétérogénéité, 3) Extraction des régions dont les valeurs de l'indice de végétation « NDVI » sont faibles et enfin 4) Purification des régions d'intérêts. Les tests ont été conduits sur des images issues du satellite Landsat-7/ ETM+ de la région de Tlemcen en Algérie. Les résultats obtenus on été comparés d'une part à une classification supervisée basée SVM et avec des données de terrain, d'autre part. Les différentes comparaisons laissent apparaitre l'efficacité de l'approche préconisée.

Mots Clés: Images satellitaires, segmentation, algorithme JSEG, indice de végétation NDVI, classification supervisée, SVM.

1 Introduction

Au cours des sept dernières années, le nord de l'Algérie, et en particulier, lors des saisons estivales, est ravagé par environ 20 à 30 incendies majeurs, induisant une diminution du patrimoine forestier d'environ 19 570 à 66 580 hectares par an. Face à ce risque, la cartographie des dommages post-feux s'avère nécessaire. En effet, une cartographie précise des zones brulées va permettre, d'une part, la mise en place d'un plan de restauration des zones touchées ainsi que l'allocation des ressources nécessaires pour cette fin, et d'autre part, l'analyse spatiale des facteurs déclenchant pour une meilleure compréhension du comportement des feux de forêts.

De part à leur large couverture spatiale, leur répétitivité temporelle et leur coût d'acquisition faible, les données de télédétection peuvent contribuer significativement à la cartographie des zones brûlées et par conséquent, à l'estimation des dégâts. Plusieurs techniques relatives à l'estimation de ces dégâts par le biais des données satellitaires ont été utilisées avec succès. Citons à titre non exhaustif, celles basées sur la comparaison de deux images dont une acquise avant et l'autre après feux [1]-[2], celles basées sur des méthodes de classifications supervisées [3]-[4], ou encore celles faisant appel à la photo-interprétation [5].

Dans cette étude, nous essayons de cartographier les zones brûlées à partir d'une seule image acquise après feu en appliquant l'algorithme de segmentation JSEG (*J-SEGmentation*) modifié. Le choix de l'algorithme segmental JSEG résulte en fait de sa prise en compte de l'information couleur et texturale, mais aussi par son mode opératoire qui consiste à prétraiter l'image (filtrage et quantification couleur) avant de procéder à une segmentation spatiale. En revanche, l'algorithme JSEG souffre de deux limitations qui affectent les résultats de la segmentation. L'une est causée par le paramètre de quantification couleur qui détermine la distance minimale entre deux couleurs quantifiées, une valeur élevée de ce paramètre induit un fort lissage de l'image et par conséquent une perte d'informations. A l'inverse, une petite valeur génère une sur-segmentation, d'où la difficulté de choisir une valeur optimale pour ce paramètre. Aussi la sélection d'une valeur appropriée pour ce paramètre de quantification dépend fortement de la thématique à rechercher. L'autre limitation est liée au paramètre de fusion qui prend en compte exclusivement le critère couleur d'hétérogénéité. Enfin, il est difficile de définir une combinaison appropriée de paramètres pour la quantification et la fusion de régions pour trouver les régions d'intérêt.

Pour pallier ces inconvénients, nous proposons une méthodologie qui repose sur quatre étapes. Premièrement, l'algorithme JSEG est appliqué en fixant le paramètre de quantification à 50 de telle sorte que nous obtenons une image sur-segmentée. Ensuite, les régions contigües ayant des attributs de couleur et de géométrie similaires seront fusionnées, la règle de Minimum d'Hétérogénéité est utilisée dans cette deuxième étape. La troisième étape consiste à extraire les zones brûlées dont la valeur moyenne de l'indice NDVI (*Normalized Difference Vegetation Index*) est la plus faible. Enfin, la dernière étape consiste à supprimer les pixels mal segmentés.

Le reste de l'article est structuré de la manière suivante : après une brève présentation de l'algorithme JSEG dans la section 2.1, la règle de Minimum d'Hétérogénéité est détaillée dans la section 2.2. La classification supervisée basée SVM (*Support Vector Machines*) ainsi que l'indice de végétation normalisé sont décrits dans les sections 2.3 et 2.4, respectivement. Dans la section 3, nous présentons les résultats expérimentaux ainsi que des comparaisons quantitatives avec des résultats issus de la classification supervisée par SVM d'une part, et des vérités terrain, d'autre part. Enfin, nous conclurons en soulignant les perspectives qu'offre cette étude.

2 Aspects méthodologique

Cette section présente brièvement l'algorithme JSEG, la fusion par la règle de Minimum d'Hétérogénéité, la classification par SVM et enfin l'indice de végétation normalisé NDVI.

2.1 Algorithme JSEG

Développé par Deng et Manjunath en 2001 [6], l'algorithme JSEG est un algorithme de segmentation d'images couleurs texturées, opérant en deux étapes. La première

étape ayant pour but de pratiquer une quantification couleur sur l'image à traiter (cette étape est pilotée par le paramètre de quantification, noté q), la seconde faisant une segmentation spatiale (cette étape est pilotée par le paramètre de fusion, noté m).
L'un des problèmes majeur en segmentation d'images couleurs peut être la grande quantité de couleurs formant ces images, c'est pourquoi l'algorithme JSEG exerce une quantification afin de réduire le nombre de couleurs. Cette quantification regroupe les pixels en quelques teintes dominantes tout en conservant une bonne représentation de l'image de départ. Cette quantification effectue dans un premier temps un filtrage non linéaire par l'algorithme PGF (*Peer Group Filtering*) [7] qui lisse l'image et réduit le bruit. L'algorithme remplace chaque pixel dans l'image par la somme pondérée de ses voisins en privilégiant ceux qui sont de couleur proche. Les paramètres statistiques locaux sont employés comme des poids affectant une valeur faible aux zones texturées et une valeur plus importante pour les zones lisses. Suite à cela, et afin de faire la quantification couleur la méthode utilise un algorithme de Lloyd Généralisé (GLA pour *Generalized Lloyd Algorithm*) qui exécute la quantification dans l'espace de représentation couleur Lab. Les couleurs similaires, c'est à dire, d'écart colorimétrique inférieur à un seuil, sont regroupées et représentées par une seule et même couleur, ce qui réduit le nombre total d'étiquettes. Finalement, l'image quantifiée est formée en assignant à chaque pixel l'étiquette qui correspond à sa couleur initiale. La seconde étape de l'algorithme JSEG est d'opérer une segmentation spatiale sur l'image filtrée et quantifiée pour générer l'image J. Le critère d'homogénéité utilisé et celui s'inspirant du critère discriminant de Fisher [8].
Sur la figure 1, nous illustrons les différentes étapes de l'algorithme JSEG.

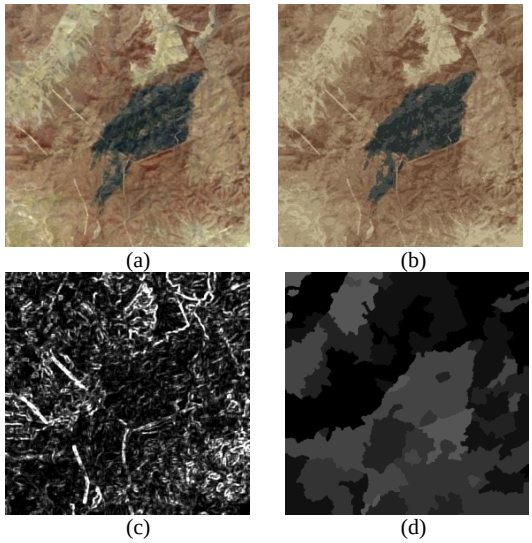


Fig. 1. Segmentation par l'algorithme JSEG: (a) image originale, (b) carte des classes avec q = 8, (c) carte-J et (d) image des segments avec m = 0.2.

2.2 Règle de Minimum d'Hétérogénéité

Une fois la pré-segmentation par l'algorithme JSEG réalisée, l'étape suivante consiste à fusionner les régions contigües ayant des caractéristiques spectrales et géométriques similaires en appliquant la règle de Minimum d'Hétérogénéité [9]-[10]-[11]. La fonction de fusion notée f est définie comme étant la somme de deux termes pondérés chacun par un poids. Le premier terme désigne la différence d'hétérogénéité spectrale et le second la différence d'hétérogénéité géométrique. Dans les deux termes la différence est opérée entre la nouvelle région obtenue après fusion et les deux régions ayant servies à sa constitution.

$$f = w_{spectrale}\Delta h_{spectrale} + w_{géométrique}\Delta h_{géométrique} \,. \quad (\mathbf{1})$$

avec $w_{spectrale} \in [0,1]$, $w_{géométrque} \in [0,1]$ et $w_{spectrale} + w_{géométrique} = 1$. La différence d'hétérogénéité spectrale $\Delta h_{spectrale}$ est définie de la manière suivante:

$$\Delta h_{spectrale} = \sum_{b=1}^{B} w_b \left[n_{R1\cup R2}\delta_{b,R1\cup R2} - \left[n_{R1}\delta_{b,R1} + n_{R2}\delta_{b,R2} \right] \right]. \quad (\mathbf{2})$$

avec B le nombre de bandes spectrales utilisées, $w_b \in [0,1]$ le poids associé à chaque bande spectrale b, $n_{R1\cup R2}$, n_{R1}, n_{R2} le nombre de pixels dans les régions $R1 \cup R2$, $R1$ et $R2$ respectivement. $\delta_{b,k}$ est donnée par:

$$\delta_{b,k} = \sigma_{b,k}\mu_{b,k} \,. \quad (\mathbf{3})$$

Où $\sigma_{b,k}$ et $\mu_{b,k}$ sont l'écart-type et la moyenne de la région $k \in \{R1 \cup R2, R1, R2\}$ dans la bande spectrale b.
La différence d'hétérogénéité géométrique $\Delta h_{géométrique}$ est un paramètre décrivant l'amélioration de la forme au niveau de la compacité et de la rugosité du bord de la région susceptible d'être formée:

$$\Delta h_{géométrique} = w_{compacité}\Delta h_{compacité} + w_{lissage}\Delta h_{lissage} \,. \quad (\mathbf{4})$$

avec

$$\Delta h_{compacité} = n_{R1\cup R2}\frac{l_{R1\cup R2}}{\sqrt{n_{R1\cup R2}}} - \left[n_{R1}\frac{l_{R1}}{\sqrt{n_{R1}}} + n_{R2}\frac{l_{R2}}{\sqrt{n_{R2}}} \right]. \quad (\mathbf{5})$$

$$\Delta h_{lissage} = n_{R1\cup R2}\frac{l_{R1\cup R2}}{b_{R1\cup R2}} - \left[n_{R1}\frac{l_{R1}}{b_{R1}} + n_{R2}\frac{l_{R2}}{b_{R2}} \right]. \quad (\mathbf{6})$$

où $w_{compacité} \in [0,1]$, $w_{lissage} \in [0,1]$ et $w_{compacité} + w_{lissage} = 1$ sont des paramètres de pondération permettant l'adaptation de l'hétérogénéité géométrique aux objets d'intérêts dans une image, l_k et b_k représentent le périmètre de la région k et le périmètre de sa boite englobante avec $\in \{R1 \cup R2, R1, R2\}$.

Notons enfin que la fusion entre deux régions voisines est opérée quant la valeur de f est inférieure à un certain seuil ε défini par l'utilisateur.

2.3 Classification par SVM

Introduit par Vapnik [12] dans les années 80, les SVM sont définies comme étant un système d'apprentissage basé sur des théories linéaires statistiques. Contrairement aux algorithmes de classification classique, comme l'algorithme de maximum de vraisemblance, les SVM présentent une classe de classifieurs non paramétriques, ce qui implique que les données en entrée peuvent ne pas obéir à une loi de distribution normale. L'autre avantage des SVM réside dans leur capacité à résoudre des problèmes linéaires avec un petit nombre d'échantillons d'apprentissage [13]. De part ces avantages les SVM ont été largement utilisés dans la classification des images médicales [14]-[15] et satellitaires [16]-[17]-[18].
L'idée de base des SVM est de rechercher l'hyperplan optimal dans l'espace de données. Cet hyperplan de séparation est défini par un ensemble de points appartenant à l'ensemble des échantillons d'apprentissage, appelé aussi les vecteurs supports. Dans le cas bi-classes, considérons un ensemble d'échantillons formé de N points $\{x_i\}_{i=1..N}$, où chaque point est décrit par un vecteur de d dimension, et soit $y_i \in \{-1, +1\}$ la classe d'étiquette correspondante au point x_i. La fonction de classification F qui détermine l'étiquette y d'un point représenté par un vecteur $\vec{V}$ est définie par:

$$\mathrm{F}(\vec{V}) = y = sign\big(\langle \vec{\omega}, \vec{V} \rangle + b\big). \tag{7}$$

Où $sign$ représente la fonction signe, $\vec{\omega}$ le vecteur normal, b le biais et $\langle . \rangle$ le produit scalaire. On définit aussi la marge qui est donnée par $\|\vec{\omega}\|^{-1}$ avec $\|\vec{\omega}\|$ la norme du vecteur $\vec{\omega}$.
Trouver un hyperplan de séparation optimal entre les deux classes revient à maximiser la marge en minimisant $\|\vec{\omega}\|$. Cette optimisation se traduit par le formalisme Lagrangien de la façon suivante:

$$\mathcal{L}_p = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_{i=1}^{N} \alpha_i y_i \left(\langle \vec{\omega}, \vec{V} \rangle + b\right) + \sum_{i=1}^{N} \alpha_i. \tag{8}$$

Où $\mathcal{L}_p$ désigne le Lagrangien primal, $\alpha_i \geq 0$ représente le multiplicateur de Lagrange du $i^{ème}$ point d'apprentissage. Le problème d'optimisation revient donc à minimiser $\mathcal{L}_p$ par rapport à $\vec{\omega}$ et b tout en le maximisant par rapport à α_i. Cela implique que le gradient de $\mathcal{L}_p$ s'annule par rapport à $\vec{\omega}$ et b:

$$\begin{cases} \vec{\omega} = \sum_{i=1}^{N} \alpha_i y_i \vec{V_i} \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases}. \tag{9}$$

Après intégration de ces conditions dans l'équation du Lagrangien primal, nous obtenons le Lagrangien dual donné par:

$$\mathcal{L}_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \vec{V_i}, \vec{V_j} \rangle. \tag{10}$$

Cette équation permet de calculer les α_i. Notons que $\alpha_i > 0$ pour les vecteurs supports et $\alpha_i = 0$ pour les vecteurs situés à l'extérieur des hyperplans définissant la marge optimale. On peut déduire $\vec{\omega}$ de l'équation 1. Finalement, la fonction de classification F s'écrira sous la forme:

$$\mathrm{F}(\vec{V}) = sign\left(\sum_{i=1}^{Ns} \alpha_i y_i \langle \overrightarrow{S_i}, \vec{V} \rangle + b\right). \quad (11)$$

Dans le cas ou les données en entrée ne sont pas linéairement séparable, il est nécessaire de les projeter dans un espace de dimension supérieure. Cette projection fait appel à une fonction appelée fonction noyau qu'on notera K. La fonction de classification F s'écrira dans le cas non linéaire sous la forme:

$$\mathrm{F}(\vec{V}) = sign\left(\sum_{i=1}^{Ns} \sum_{i=1}^{N} \alpha_i y_i K(\overrightarrow{S_i}, \vec{V}) + b\right). \quad (12)$$

Le noyau peut être sigmoïd, polynomial ou RBF dans le cas de cette étude.
Conçus initialement pour le cas bi-classes, les SVM ont été adaptées au cas multiclasses en faisant appel à deux approches à savoir, une contre une ou une contre tous. Dans la première approche, on construit $M(M-1)/2$ fonctions de décision, ce qui revient à séparer les classes deux à deux (M étant le nombre de classes). Dans la seconde approche, pour chaque classe, on détermine un hyperplan séparant celle-ci des autres classes. On obtient ainsi M fonctions de décision.

2.4 Indice de Végétation Normalisé

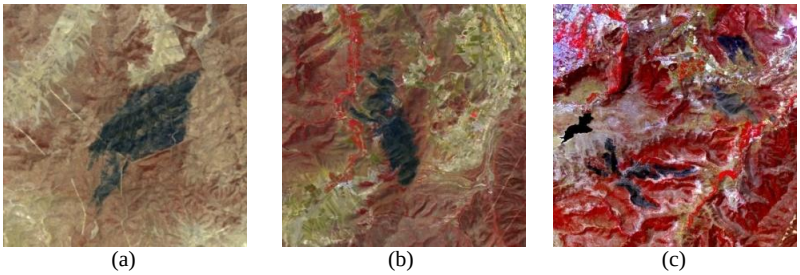
Les indices de végétation sont des formules empiriques conçues pour fournir des mesures quantitatives qui sont souvent en rapport avec la biomasse et l'état de la végétation. L'indice le plus communément utilisé est le NDVI [19]. Il est calculé à partir des bandes Rouges (R) et Proche Infra-Rouge (PIR) comme suit :

$$NDVI = \frac{PIR-R}{PIR+R}. \quad (13)$$

Le NDVI fournit une valeur entre -1 et +1. Une valeur de NDVI proche de 1 indique une grande densité de végétation saine. A l'inverse une valeur de NDVI négative indique une végétation malsaine ou une absence de végétation.

3 Expérimentations

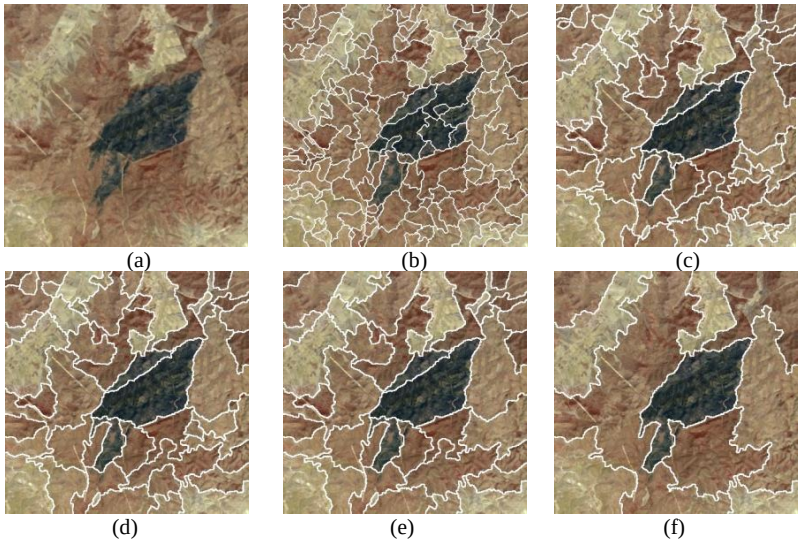
La méthodologie adoptée a été implémentée sous l'environnement MATLAB et sur un processeur Intel Dual-Core cadencé à 1.7 GHz et 2 GB de mémoire vive. Les images utilisées dans les expérimentations (Fig. 2) sont extraites d'une scène Landsat-7 couvrant la région de Tlemcen et acquise en date du 10/08/2010. Les images ont une taille de 400 x 400 pixels, une résolution spatiale de 30 mètres et sont composées des bandes Vert, Rouge et Proche Infra-Rouge.
La méthodologie adoptée sera détaillée sur l'image de la figure 2.a, pour ensuite donner les résultats finaux relatifs aux figures 2.b et 2.c.



(a) (b) (c)

Fig. 2. Images utilisées lors des expérimentations.

La première étape de notre processus consiste à appliquer l'algorithme de segmentation JSEG en fixant le paramètre de quantification q à 50. Le résultat est illustré dans la figure 3.b pour l'image de la figure 2.a. Cette valeur de quantification procède à une réduction du nombre de couleurs distinctes de l'image, sans pour autant réduire la qualité de l'information, et permet d'extraire les couleurs représentatives différenciant les régions voisines dans l'image.



(a) (b) (c)

(d) (e) (f)

Fig. 3. (a) Image originale, (b) Segmentation par l'algorithme JSEG avec q=50 et m=0, (c), (d), (e), (f) résultats obtenus après fusion des régions avec un seuil de 5000, 10000, 20000 et 24000 respectivement.

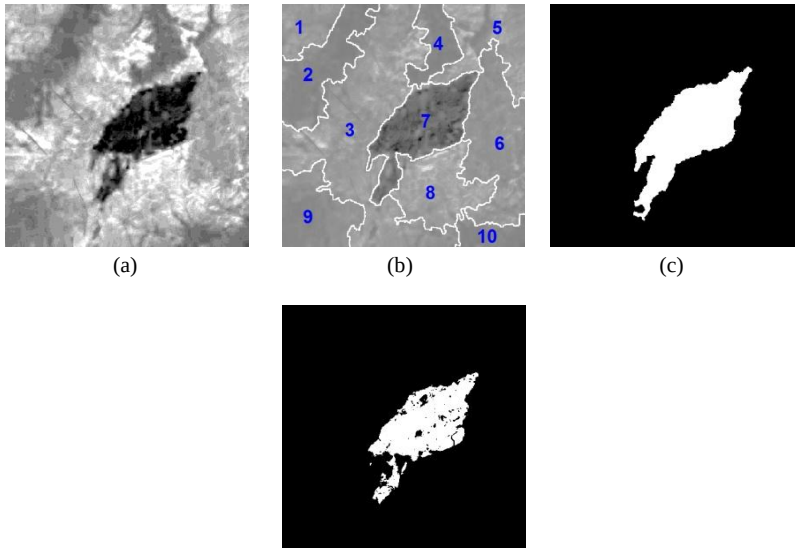
Dans la deuxième étape, la règle de Minimum d'Hétérogénéité a été utilisée pour regrouper les régions ayant à la fois une similitude spectrale et géométrique. Les figures 3.b, 3.c, 3.e et 3.f représentent les résultats avec différents valeurs du seuil ε. Les valeurs de pondérations sont : $w_{spectrale}$=0.7, $w_{géométrique}$=0.3, w_b=0.33 (pour les trois canaux) et enfin $w_{compacité} = w_{lissage}$ =0.5. On remarque que l'écart entre

les valeurs du seuil ε est fonction du nombre de pixels des régions à fusionner. En d'autre terme, si les deux régions candidates pour la fusion sont formées d'un nombre restreint de pixels, la valeur obtenue de la fonction d'hétérogénéité f est faible, ce qui implique une petite valeur du seuil, et inversement.

La troisième étape du processus consiste à repérer les régions dont la moyenne des valeurs du NDVI est la plus faible. Pour ce faire et dans un premier temps, l'image NDVI (Fig. 4.a) est calculée en utilisant l'équation 13. Ensuite, et pour chaque région obtenue après fusion, nous calculons la moyenne des valeurs du NDVI (Fig. 4.b). Sur la figure 4.c, nous présentons la région dont la moyenne des NDVI est la plus faible, et elle correspond au feu de forêt. Enfin, la figure 4.d illustre la région d'intérêt finale après élimination des pixels intrus. Cette dernière étape est effectuée en appliquant la formule empirique suivante :

$$Dist\left(ndvi\left(Pixel \in R_{feu\ de\ fôret}\right), médiane\_ndvi(R_{feu\ de\ fôret})\right) > Dist\left(ndvi\left(Pixel \in R_{feu\ de\ fôret}\right), max\ \_ndvi(R_{feu\ de\ fôret})\right). \quad \textbf{(14)}$$

Où $Dist$ désigne la distance Euclidienne.



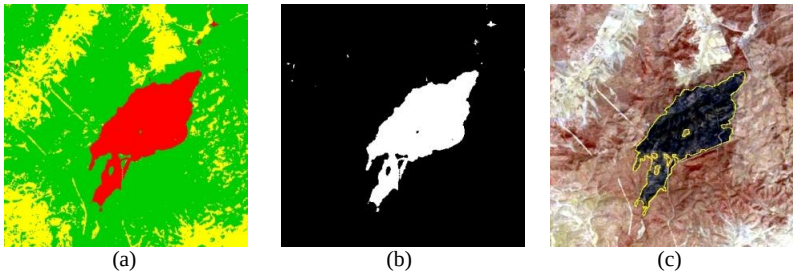
(a) (b) (c)

(d)

Fig. 4. (a) Image NDVI, (b) Superposition des limites des régions sur l'image NDVI, (c) Masque obtenu sur la région d'intérêt et (d) Région d'intérêt finale.

Afin de valider les performances de notre méthode, et dans un premier temps, une classification supervisée par SVM a été réalisée sur la même image avec trois (03) classes thématiques : Feu de forêt (en rouge), Forêt (en vert) et Sol dégradé (en jaune). Sur les figures 5.a et 5.b nous présentons l'image des classes résultante ainsi que le masque sur la région d'intérêt extrait de cette dernière.

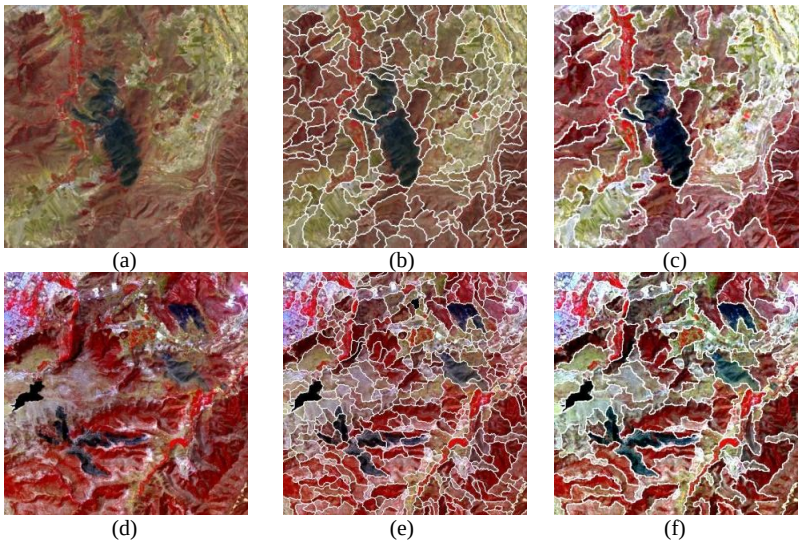
Dans un second temps, le résultat obtenu par notre approche et celui obtenu par classification ont été comparés à la vérité terrain fournie par un photo-interprète et obtenue par la digitalisation manuelle de la région d'intérêt sur l'image satellitaire (Fig. 5.c). A cet effet, nous utilisons la quantification géométrique comme paramètre de comparaison, pour la caractérisation des feux de forêts. Ainsi les descripteurs de formes retenus sont le périmètre et la superficie.



(a) (b) (c)

Fig. 5. (a) Images des classes obtenu par SVM, (b) Masque obtenu sur la région d'intérêt et (c) Vérité terrain.

Comme cité dans le début de cette section, nous avons appliqué les mêmes traitements sur les deux autres images utilisées dans les expérimentations. Les résultats sont illustrés dans la figure 6. Les valeurs de pondérations retenues pour la phase de fusion des régions voisines sont identiques à celles utilisées pour la première image.

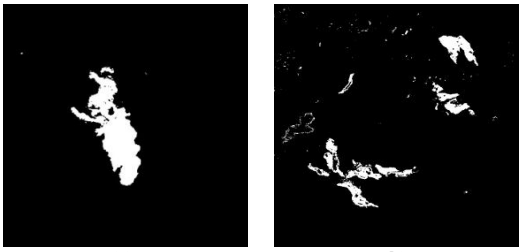


(a) (b) (c)

(d) (e) (f)

Fig. 6. (a), (d) images originales, (b), (e) Segmentation par l'algorithme JSEG avec q=50 et m=0, (c), (f) résultats obtenus après fusion des régions avec un seuil de 4000 et 45000 respectivement.

La figure 7 représente les résultats obtenus à partir des images originales de la figure 6.



(a) (b)

Fig. 7 : Résultats finaux de la démarche :(a) issu de l'image 6.a, (b) issu de l'image 6.b.

Les résultats quantitatifs des différentes expérimentations sont listés dans le Tableau 1. Dans ce tableau, on y présente la superficie et le périmètre des régions brulées calculés par l'approche adoptée, par la classification supervisée basée SVM et enfin par photo-interprétation (Vérité Terrain). L'analyse comparative des moyennes des écarts entre les superficies estimées par notre approche à celles obtenues par vérité terrain est de l'ordre de 0.71%. Cette moyenne est largement inférieure à celle calculée entre les superficies estimées par la classification et la vérité terrain, qui est de l'ordre de 1.81%.

Cet écart est dû principalement au fait que la classification basée SVM utilise un critère de similarité exclusivement spectral. En d'autre terme, les pixels ayant des réponses spectrales similaires sont regroupés dans la même classe, même s'ils n'appartiennent pas réellement au même objet thématique, cas des pixels appartenant à la classe eau et ombre qui seront classés dans la classe feux de forêt. Afin de lever cette confusion, il est fortement suggéré d'intégrer d'autres sources conjointement avec les données spectrales.

Tableau 1: Superficies en hectare et périmètres en kilomètre des zones brulées des différentes approches.

	Processus adopté		Classification supervisée par SVM		Vérité Terrain	
Images	Superficie	Périmètre	Superficie	Périmètre	Superficie	Périmètre
1	1416.69	45.45	1598.13	45.72	1474.65	46.33
2	776.43	31.95	817.38	32.76	752.91	30.82
3	1371.69	79.44	1593.81	98.78	1238.40	75.95

Finalement, nous abordons l'aspect de temps de calcul nécessaire lors de la phase de fusion des différentes régions issues de la segmentation par JSEG (q=50, m=0). Les différents calculs ont été effectués sur un PC équipé d'un processeur P4 cadencé à 3.0 GHz avec 2.0 Go de mémoire vive et en utilisant l'environnement MATLAB.

Le Tableau 2 présente le temps de calcul obtenu en fonction du nombre de régions fusionnées pour chaque image test. Nous remarquons que le temps augmente si l'écart entre le nombre de région initial et celui final augmente.

Tableau 2: Temps de calcul en fonction du nombre de régions fusionnées.

Images	Nombre de Régions Initial	Nombre de Régions Final	Nombre de Régions fusionnées	ε	Temps en secondes
1	92	13	79	24000	27.9337
2	210	16	194	4000	69.0297
3	433	31	402	45000	193.8562

4 Conclusions

Nous avons présenté dans cet article une nouvelle approche pour la cartographie des feux de forêts en appliquant l'algorithme de segmentation JSEG modifié. Cette approche est basée uniquement sur une image satellitaire acquise après feu et est composée de quatre étapes : une segmentation par l'algorithme JSEG; fusion de région basée sur les descripteurs spectraux et géométriques; utilisation des valeurs de NDVI pour l'extraction des régions d'intérêts et enfin élimination des pixels intrus.
Les expérimentations menées sur les images satellitaires, l'évaluation quantitative des résultats obtenus et les comparaisons effectuées à la fois avec la classification supervisée par SVM et la vérité terrain témoignent l'applicabilité et l'efficacité de la présente étude.

La principale limitation de notre approche est le nombre de paramètres à ajuster lors de la phase de fusion de régions. Un paramétrage adaptatif n'est pas évident à mettre en place, mais pourra être inscrit comme perspective à ce travail. De plus, le temps de calcul reste à optimiser lors de la phase de fusion des différentes régions issues de la segmentation par l'algorithme JSEG.

Enfin, il serait souhaitable de tester notre approche sur des données satellitaires à haute et à très haute résolution spatiale, et d'utiliser l'indice de brillance qui permet de discriminer entre les sols nus et les sols brulés conjointement avec le NDVI.

Références

1. Bastarrika, A., Chuvieco, E., Martin, M.P.: Mapping burned areas from Landsat TM/ETM+ data with a two-phase algorithm: Balancing omission and commission errors. Remote Sensing of Environment 115, pp. 1003-1012 (2011).
2. Phua, M.H., Tsuyuki, S., Furuya, N., Lee, J.S. Detection deforestation with a spectral change detection approach using multitemporal Landsat data: A case study of Kinabalu park, Sabah, Malaysia. Journal of Environmental Management 88, pp. 784-795 (2008).
3. Zammit, O., Descombes, X. and Zerubia J.: Assessment of different classification algorithms for burnt land discrimination. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3000-3003 (2007).
4. Cao, X., Chen, J., Matsushita, B., Imura, H., and Wang, L.: An automatic method for burn scar mapping using support vector machines. IEEE International Journal of Remote Sensing, vol. 30, n. 3-4, pp. 577--594 (2009).
5. Leblon, B., Merzouki, A., MacLean, D.A., LaRocque, A.: Phot-interpretation and remote sensing at the Faculty of forestry and Environmental Management, UNB. The Forestry Chronicle, vol. 84, n. 4, pp. 534-538 (2008) .

6. Deng, Y., Manjunath, B.S.: Unsupervised Segmentation of Colour-Texture Regions in Images and Video. IEEE Transactions on Pattern Analysis and Machine Intelligent, vol. 23, n. 8, pp. 800--810 (2001).
7. Kenny, C., Deng, Y., Manjunath, B.S., Hewer, G.: Peer Group Image Enhancement. IEEE Transactions on Image Processing, vol. 10, n. 2, pp. 326--334 (2001)
8. Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. John Wiley & Sons, New York (1970).
9. Li, H. T., Gu, H.Y., Han, Y.S., Yang, J.H. and Han, S.S.: An Efficient Multi-Scale Segmentation for High-Resolution Remote Sensing Imagery Based On Statistical Region Merging and Minimum Heterogeneity Rule. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4, pp. 1257--1262 (2008).
10. Huang, X., ZHANG, L.: A comparative study of spatial approaches for urban mapping using hyperspectral ROSIS images over Pavia City, northern Italy. IEEE International Journal of Remote Sensing, vol. 30, n. 11-12, pp. 3205--3221 (2009).
11. Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I., Heynen, M. : Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS Journal of Photogrammetry & Remote Sensing 58, pp. 239-258 (2004).
12. Vapnik V. N.*:* Statistical learning theory. New York, Wiley (1998).
13. Cristianini, N. and Shaew-Taylor, J.: An introduction to support vector machines and other kernel based learning methods. Cambridge, U.K.: Cambridge Univ. Press (2000).
14. Guyon, I., Weston, J., Barnhill, S., Vapnik V. N.: Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning, 46. pp. 389-422 (2002).
15. Vanitha, L., Venmathi, A.R. Classification of Medical Images Using Support Vector Machine. Internal Conference on Information and Network Technology, vol.4, pp. 63-67, Singapore (2011).
16. Guo, B., Gunn, S. R., Damper, R. I. and Nelson, J. D. B. Customizing kernel functions for SVM-based hyperspectral image classification. IEEE Transactions on Image Processing, 17 (4). pp. 622-629 (2008).
17. Demir, B., Ertürk, S.: Empirical mode decomposition of hyperspectral images for support vector machine classification. IEEE Transactions on Geoscience and Remote Sensing, vol.48, no.11, pp.4071-4084 (2010).
18. Tarabalka, Y., Fauvel, M., Chanussot, J., Benediktsson, J.A. : SVM and MRF-Based Method for Accurate Classification of Hyperspectral Images. IEEE Transactions on Geoscience and Remote Sensing Letters, vol.7, no.4, pp. 736-740 (2010).
19. Rouse, J.W., Hass, R.H., Schell, J.A. and Deering, D.W.: Monitoring Vegetation Systems in the Great Plains with ERTS. In third ERTS symposium (1973).

Segmentation d'images médicales 3D par un contour actif rapide

Ouardia Chilali[1], Ahcen Ait-Menguellet[1], Arezki Slimani[1], Hamid Meziani[1],

[1] Dépatement d'Automatique, Faculté de Génie électrique et d'informatique, Université Mouloud MAMMERI, Tizi-ouzou, Algérie.
chilalikarima@yahoo.fr

Résumé. Dans cet article, une combinaison de modèles de contours actifs est réalisée afin d'accélérer la segmentation des images médicales 3D. En plus de cette combinaison, une nouvelle fonction *level set* accélérée est introduite. En premier lieu, le modèle trouvé était en 2D. Son extension en 3D était très essentiel. Seulement, le passage était très simple d'autant qu'il s'agissait de la représentation implicite des contours actifs. Des résultats très intéressants sont obtenue, en appliquant l'approche sur des images 2D, mais surtout sur différents types d'images médicales.

Mots clés: Contour actif, *level set* accélérée, image 3D, Chan et Vese, DICOM.

1 Introduction

Le plus souvent, l'information que l'on cherche à extraire peut être obtenue sur une image en deux dimensions (2D) mais un nombre grandissant de problèmes nécessitent d'appréhender la scène observée en trois dimensions. Pour atteindre un tel objectif, un grand nombre de techniques de traitement d'images 3D ont été mises au point. Toutefois, il peut sembler simple de traiter des images tridimensionnelles lorsque l'on est habitué à manipuler des images en deux dimensions. Il suffit de généraliser, à la troisième dimension, des opérations connues de traitement d'images 2D. C'est le cas de la méthode des contours actifs.

Un contour actif est une structure géométrique évoluant, itérativement, de manière à s'ajuster aux frontières des objets recherchés. Il est généralement représenté par une courbe dans une image 2D ou une surface dans une image 3D. La forme et la position initiale de la courbe ou de la surface sont fournies de manière manuelle ou automatique. Selon la représentation de la courbe ou la surface, deux types de contours actifs peuvent se distinguer : contours actifs explicites [1] et contours actifs implicites [2]. Ces derniers sont plus avantageux mais souffre du facteur temps.

Pour cela, l'objectif de notre article est de développer un contour actif implicite 2D rapide et de l'étendre vers le 3D, pour l'utiliser en segmentation d'images médicales. Pour se faire, nous allons, dans la section suivante, annoncer des généralités et des notions de base sur l'imagerie tridimensionnelle, en général, et l'imagerie médicale 3D, en particulier. La section 3 se consacrera au modèle des

surfaces actives implicites qui sont une extension des contours actifs implicites. Nous intéresserons, beaucoup plus, au modèle de *Chan* et *Vese* [3] et nous adopterons une méthode d'accélération du processus de segmentation. Par la suite, et dans la section 4, nous présentons le modèle de l'approche adoptée, ainsi que l'algorithme implémenté pour l'amélioration du temps de traitement. La dernière section sera consacré aux tests et résultats d'application de notre approche adopté. Ce travail s'achèvera par une conclusion générale.

2 Imagerie tridimensionnelle

Dans beaucoup d'applications tridimensionnelles [4], la scène est souvent représentée par un tableau à 3 dimensions de volumes élémentaires qu'on appelle plus simplement un *voxel.* C'est un élément tridimensionnel d'image généralement cubique, qui correspond à une intensité. Un voxel est composé : de surfaces, d'arêtes et de sommets. Cela permet de définir un voisinage n-connexe (n = 6, 18 ou 26) du *voxel P*(Fig. 1).

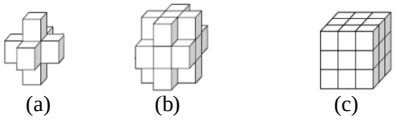


Fig. 1. Voisignage d'un voxel: (a) 6-*voisins de P, (b)18-voisins de P et (c) 26-voisins de P.*

La figure 2 représente la représentation numérique d'une image médicale.

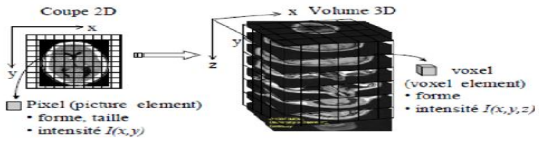


Fig. 2. La représentation numérique d'une image 2D et 3D.

Les processus actuels d'imagerie médicale, particulièrement l'IRM ou le scanner X hélicoïdal, réalisent, de façon directe ou dans le cadre de reconstructions, des acquisitions volumiques [4]. La visualisation du volume la plus simple est la reconstruction multi-planaire (Multi-Planaire Reconstruction (MPR), en anglais), correspondant au système de référence en anatomie (Fig. 3).

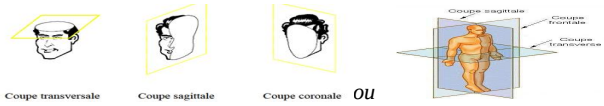


Fig. 3. Les plans de base en imagerie radiologique.

2.1 Volume et surface

La figure 4 illustre la dualité des deux approches.

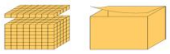


Fig. 4. Dualité volume surface en imagerie médicale volumique.

Les forces et faiblesses de ces deux approches peuvent se résumer de la façon suivante :

- → *Approches volumiques :* bénéficient naturellement des potentialités visuelles considérables, issues de la nature de données échantillonnées. Les outils d'analyse que l'on peut y appliquer sont directement issus des outils "classiques" du traitement d'image correspondants à la 3D (filtrage, segmentation, morphologie mathématique, etc.).
- → Approches surfaciques : agissent à un niveau en fait plus élevé : celui de la structure des données tridimensionnelle que l'on souhaite visualiser ou représenter. Dans tous les cas, une approche **reconstructrice** est nécessaire (reconstruction ou modélisation).

3 Contour actif

Le contour actif a été introduit par *Kass*, *Witkin* et *Terzopoulos* en 1988 sous le nom de *snake* (serpents) ou courbe minimisante [5] et [6]. Il s'agit d'une méthode semi-interactive dans laquelle l'opérateur place dans l'image, au voisinage de la forme à détecter, une forme initiale de contour qui sera amenée à se déformer sous l'action de plusieurs forces : une énergie propre, une énergie potentielle et une énergie externe. Plusieurs améliorations de ce modèle initial ont permet de diversifier le modèle en plusieurs autres modèles tels le contour actif ballon [7], le contour actif GVF [8], les *contours actifs géométriques*[9], Le modèle du *contour actif géodésique* [10], etc. Il est possible de classer ces forces en deux groupes, suivant qu'elles agissent uniquement sur l'information contour [11], ou qu'elles prennent en compte l'information région [12].

3.1 Contour actif 3D

Le modèle du contour actif a été étendu en 3D, donnant ainsi naissance à la surface active ou le maillage actif (selon la représentation implicite ou explicite, respectivement) [13], afin de segmenter des objets dans des images 3D. Les maillages actifs sont des représentations explicites discrètes. L'information, stockée et manipulée, est un ensemble de sommets interconnectés (les points noirs dans l'exemple de la figure 5) par un ensemble d'arêtes. La surface est déformée par modifications des coordonnées des sommets.

Fig. 5. Illustration d'un maillage.

Deux principaux types de maillage sont souvent rencontrés : les maillages triangulaires et les maillages simplexes [13] (Fig. 6).

Maillage triangulaire Maillage simplexe

Fig. 5. Les deux types de maillages.

Remarque 1. Dans le langage courant, le terme surface active représente les contours actifs 3D, en générale. De ce fait, on parle de maillage explicite pour la représentation explicite et de surface implicite pour la représentation implicite.

3.2 Surface active

La représentation des *levels set* par le modèle des surfaces actives est l'extension du modèle des contours actifs 2D ; telle que la fonctionnelle d'énergie est donnée par [14] :

$$E(S, c_I, c_0) = \iint_s \alpha g dS + \lambda_I \iiint_{inside(S)} (I - c_I)^2 \, dxdydz + \lambda_0 \iiint_{outside(S)} (I - c_0)^2 \, dxdydz \quad \textbf{(1)}$$

Avec : S est le niveau de surface définissant la partition de l'image, *dS* est l'élément de surface, c1 et c0, sont initialement, des variables inconnus (sont, respectivement, l'intensité moyenne des voxels à l'intérieur et à l'extérieur de la surface), *I=I*(x, y, z) l'intensité de l'image au point (x, y, z) Ω ; $\Omega \subset \Re^3$ le domaine de l'image et g une fonction définie par :

$$g = 1/(1 + \lambda e \, f) \quad \textbf{(2.a)}$$

Où f signifie le bord de l'image :

$$f = (\nabla G_\sigma * I) \quad \textbf{(2.b)}$$

Sans oublier que α, λe ,λI, λo sont des paramètres positifs fixés.
Comme pour le 2D, la fonctionnelle d'énergie peut être exprimé par une fonction de ϕ, dite fonction *level-set,* de la manière suivante :

$$E(\phi, c_I, c_0) = \iiint_\Omega [\alpha g \delta(\phi) |\nabla \phi| + \lambda_I H(\phi)(I - c_I)^2 + \lambda_0 (1 - H(\phi))(I - c_0)^2] \, dxdydz \quad \textbf{(3)}$$

Où :H est une fonction Heaviside et δ est une fonction Dirac.
A partir d'une initialisation $\phi_{t=0}(x, y, z)$, la minimisation de (3) est accomplie en laissant évoluer la fonction *level set* en fonction du temps.

L'extension du modèle ne soulève pas des difficultés théoriques significatives, grâce à la nature multidimensionnelle du formalisme d'ensemble de niveaux. Le problème majeur de cette méthode est sa lenteur. Ainsi, plusieurs méthodes d'accélération ont été soit rajouter ou développer pour remédier à ce problème (*Narrow band*, *Fast matching*, etc.) surtout pour ce qui concerne le 3D, vu le nombre important d'informations mises en jeu. Notre objectif était d'adopter une méthode accélérée afin de diminuer le temps de traitement.

4 L'approche adoptée

Notre but était de trouver une méthode plus rapide, réduisant le temps de calcul d'évolution, et comme nous l'avons déjà évoqué, ce modèle adopté combine pour sa construction, les avantages des deux modèles, à savoir : celui dans [15] et [17], ainsi que celui dans [16]. Le premier fait introduire une force signé de pression (signed pression force (SPF)), tandis que le deuxième introduit une fonction *level set* accélérée.

4.1 La force signée de pression

C'est une force générée par la différence de la moyenne des intensités à l'intérieur et à l'extérieur de l'objet (c_1 et c_2 respectivement) et leurs moyennes. Cette force applique une pression incitant le contour à ce rétrécir ou s'accroitre selon son signe. La formule de la SPF est donnée par l'équation qui suit [17] :

$$SPF = \frac{I - \frac{C_1 + C_2}{2}}{max\left(\left|I - \frac{C_1 + C_2}{2}\right|\right)} \quad (4)$$

Cette force correspond à la force externe (F_{ext}).

4.2 La fonction level set accélérée

Nous prendrons une nouvelle présentation de la fonction *level-set*, à la différence de l'ancienne présentation sous forme d'une distance signée : c'est celle développée par les auteurs de [16]. Soit ϕ la fonction *level set*, à valeurs réelles, ayant pour support le domaine image Ω. La surface C , qu'on nomme aussi front, est définie comme le niveau zéro de ϕ , à l'instant *t* donnée :

$$C = \{x / \ \phi(x, t) = 0\} \quad (5)$$

Habituellement, la fonction ϕ est initialisée comme la distance euclidienne signée à la surface C. Le signe dépend de l'appartenance du point x à l'intérieur ou à l'extérieur de C . Ici, nous choisissons que ϕ soit négative à l'intérieur de C :

$$\mathcal{R}_{in} = \{x/\ \phi(x,t) < 0\} \tag{6}$$

La base de l'optimisation de [16] est d'utiliser une alternative à la distance euclidienne. La valeur de ϕ en un point indique le statut de ce point par rapport à la surface : À l'intérieur, à l'extérieur, sur le bord intérieur ou sur le bord extérieur. Nous introduisons deux ensembles L_{in} et L_{out}, contenant les points, respectivement, sur les bords intérieur et extérieur. Un point appartient à L_{in} s'il est à l'intérieur et qu'au moins un de ses voisins est à l'extérieur et inversement pour L_{out} :

$$\begin{aligned} L_{in} &= \{x/\phi(x) < 0\ \exists y \in N(x)\ tel\ que\ \phi(y)\ > 0\} \\ L_{out} &= \{x/\phi(x) < 0\ \exists y \in N(x)\ tel\ que\ \phi(y)\ < 0\} \end{aligned} \tag{7}$$

Où $N(x)$ est le voisinage discret de *x* (définie par la norme de connexité 4 en 2D et 6 en 3D) tel que *:*

$$N(x) = \{y \in D / \sum_{k=1}^{k} |y_k - x_k| = 1\} \tag{8}$$

Les ensembles L_{in} et L_{out} sont implémentés en listes chainées vérifiant : $L_{in} \subset \mathcal{R}_{in}$ (région interne) et $L_{out} \subset \mathcal{R}_{out}$ (région externe) comme représentée en figure 6. La fonction *level-set* est définie [16] comme une fonction ayant des signes opposés à l'intérieur et à l'extérieur du contour (surface) C telle que :

$$\phi = \begin{cases} 3 & pour\ les\ valeurs\ à\ l'extérieur\ de\ C \\ 1 & pour\ les\ valeurs\ appartenant\ à\ L_{out} \\ -1 & pour\ les\ valeurs\ appartenant\ à\ L_{in} \\ -3 & pour\ les\ valeurs\ à\ l'extérieurde\ C \end{cases} \tag{9}$$

Cette nouvelle présentation de la fonction *level-set* permet d'accélérer la convergence de l'évolution des contours (surfaces) actifs.

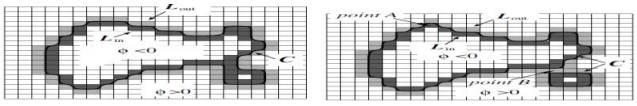


Fig. 6. Illustration des deux listes L_{in} et l_{out}.

Par rappel, l'équation d'évolution de ϕ est donnée par l'expression suivante [12] :

$$\frac{\partial \phi}{\partial t} = \delta(\phi)\left[\mu div\left(\frac{\nabla \varphi}{|\nabla \varphi|}\right) - \nu - \lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2\right] \tag{10}$$

Où $div\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)$ est la courbure qui permet de garder une courbe lisse. Elle représente la F_{int}. Le reste de l'équation est la F_{ext}. Lorsque la vitesse *F* (qui représente la somme des F_{ext} et F_{int} (équation (10))) est négative, la valeur ϕ diminue localement, ce qui a pour effet de dilater la surface au point considéré. A l'inverse, la surface se rétracte si $F > 0$. Dans cette méthode seul le signe de *F* est pris en compte (ϕ réduite à un nombre restreint de valeurs, seuls les points voisins du front sont concernés par l'évolution). A chaque itération *F* n'est calculé que pour les points L_{in} et L_{out} .Ainsi, les points de L_{out} passent dans L_{in} si F< 0 , tandis que les points de L_{in} passent dans L_{out} si F 0.Leur valeur est affecté à -1 ou 1 selon le cas. La région est ensuite mise à jour, de façon à ce qu'un point de L_{in} soit enlevé s'il n'a aucun voisin dans L_{out}, et inversement. De cette façon, les voxels changent rapidement de statut, et le front évolue de manière accélérée. Avant de présenter les détails de l'algorithme, on définie deux procédures : *switch_in* et *switch_out* [16].

Le modèle du contour actif a été étendu en 3D, donnant ainsi naissance à la surface active ou le maillage actif (selon la représentation implicite ou explicite, respectivement) [18], afin de segmenter des objets dans des images 3D.

4.3 Algorithme d'implémentation

L'algorithme de [16] est décrit comme suit :

- Step 1 : initialisation de F, et les deux listes L_{out} , L_{in}
- Step 2 : calcul de la force F pour tous les points dans L_{out} et L_{in}
  TantQue critère d'arrêt non vérifié faire
- Step 3 : //Evolution extérieure.

```
Pour tout  x ∈ Lout  faire
      Si F(x) > 0  alors
      switch_in (x)
    Finsi
  FinPour
//Elimination des points redondants dans Lin.
Pour tout point x ∈ Lin faire
      Si   ∀y ∈ N(x)  ϕ(y) < 0  alors,
         Lin ← Lin\{x}
      ϕ(x) ← -3
    FinSi
  FinPour
      // Evolution intérieure.
```

```
                Pour tout point x ∈ Lin  faire
                    si  F(x) < 0
                     switch_out(x)
                   FinSi
                FinPour
    //Elimination des points redondants dans Lout.
          Pour tout point x ∈ Lout faire
             Si ∀y ∈ N(x)  ϕ(y) > 0 alors
                 Lout ← Lout\{x}
                ϕ(x) ← 3
             FinSi
           FinPour
        FinTanque
```

- Step 4 : si la condition est satisfaite, terminer l'algorithme si non retourner à step 2.

Lorsque les listes L_{out} et L_{in} ne peuvent plus évoluer, nous considérons que la fonction *level set* a atteint son état d'équilibre. Le critère d'arrêt est tel que :

$$F(x) \leq 0 \quad \forall x \in L_{out} \qquad \text{et} \qquad F(x) \geq 0 \quad \forall x \in L_{in} \tag{11}$$

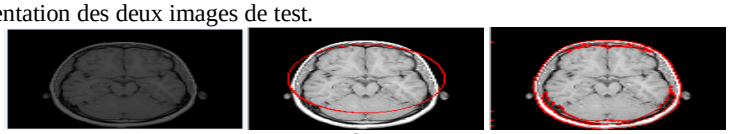
Remarque 2. L'équation d'évolution est composée de deux composantes extérieur et intérieur F_{ext} et F_{int} . Le terme de régularisation (F_{int}) peut être remplacé par $\Delta\phi$, qui est le Laplacien de ϕ. On peut dire que l'évolution de la fonction *level-set* (ϕ) avec le Laplacien, est équivalent au filtrage avec le noyau Gaussien indépendamment, du critère d'évolution afin de régulariser le contour (surface), et cela, en effectuant un produit de convolution après chaque itération du processus de convergence. C'est ce que nous allons appliquer dans notre cas.

Remarque 3. L'algorithme implémenté pour l'imagerie 3D n'est qu'une extension du celui de 2D, sauf qu'en 3D l'élimination des points redondants, se fait au voisinage des voxels ce qui nous mène à les éliminer selon les surfaces, les arêtes et les sommets, où L_{out} et L_{in} sont des ensembles de voxels appartenant au front. ϕ Sera considérée comme une surface active. La vitesse *F* est calculée en fonction des voxels contenus dans L_{in} et L_{out} .

5 Tests et résultats

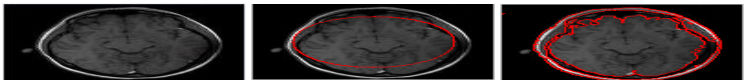
5.1 Application sur des images 2D

Dans cette partie, nous avons choisie pour la segmentation, deux types d'images médicales « Brain1 » (Fig. 7.a) et « Brain2 » (Fig. 8.a) de dimensions 128X128 et 131X131 respectivement, afin de faire une évaluation comparative entre l'approche adoptée et la méthode de *Chan et Vese*. Les figures 8 et 9, montrent les résultats de segmentation des deux images de test.



(a) (b) (c)

Fig. 8. Test sur l'image brain1 : (a) image originale, (b) l'image Initialisée par cercle automatique et (c) résultat après 10 itérations



(a) (b) (c)

Fig. 9. Test sur l'image brain1 : (a) image originale, (b) l'image Initialisée par cercle automatique et (c) résultat après 10 itérations

Le tableau ci-dessous illustre les différentes éstimations en temps d'évolution du contour pour les deux modèles appliqués sur les deux images brain1 et brain2 aprés 10 itérations.

Table 1. Les différentes éstimations en temps (en seconde) d'évolution du contour.

Images	Approche adoptée	Modèle C-V
Brain1 (128X128)	15,52	16,28
Brain2 (131X131)	31,22	35,25

A partir de cette comparaison nous pouvons dire que notre approche est plus rapide que celle de *Chan* et *Vese*. Ainsi, nous allons l'appliquer en imagerie médicale 3D.

5.2 Application sur des images 3D

Nous avons, premièrement, pris une image Matlab dite « *mri* », de dimension 128X128X27 (Fig. 10).

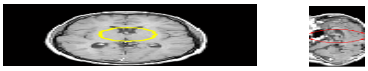


Fig.10. Initialisation manuelle par petite sphére.

Les figures ci-dessous (Fig. 11 et Fig. 12), nous montre le résultat de segmentation sur des coupes centrales selon les trois plans.

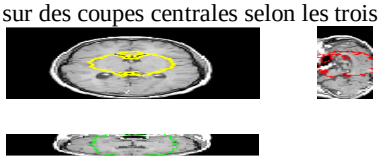


Fig.IV.11 Résultat de segmentation après 20 itérations.

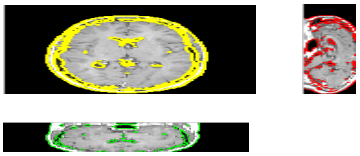
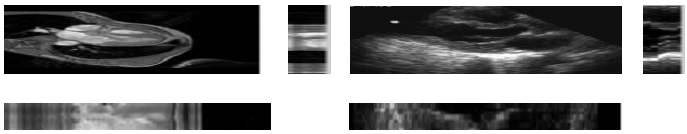


Fig. 12. Résultat de segmentation après 100 itérations.

5.2 Application sur des images 3D de format DICOM

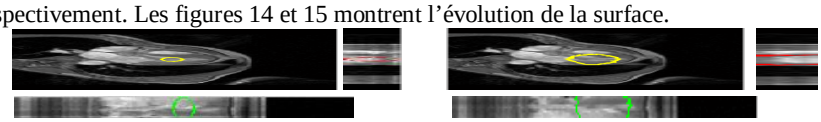
Pour nos tests, nous avons pu téléchargé deux types d'images médicales 3D de format DICOM [19] : la premiére est une image IRM se nommant *« heart »*, de dimension(256X256X16) (Fig. 13.a), et la deuxiéme est une image échographique de dimension(120X128X8) (Fig. 13.b).



(a) (b)

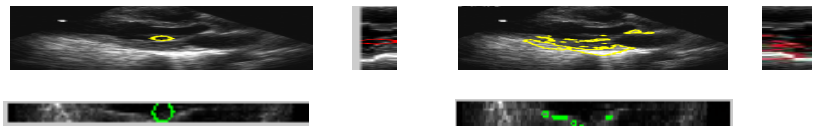
Fig. 13. Images DICOM : coupe centrale, selon les trois plans, :(a) de l' IRM « heart » de dimension (256X256X16) et (b) d'une image échographique de dimension (120X128X8).

Nous avons suivi, initialement l'évolution de la surface en fixant le nombre d'itérations à 20 et 40 pour les deux images IRM *« heart »*, et échographique, respectivement. Les figures 14 et 15 montrent l'évolution de la surface.



(a) (b)

Fig. 14. Test sur l'image IRM « heart » selon les trois plans de la coupe centrale: (a)initialisation de la surface,(b) résultat après 20 itérations.



(a) (b)

Fig. 15. Test sur l'image échographique selon les trois plans de la coupe centrale: (a)initialisation de la surface,(b) résultat après 40 itérations.

Ensuite, nous avons augmenter le nombre d'itérations afin de voir le résultat de ségmentation après 100 itérations et 600 itérations pour les deux images IRM *« heart »*,et échographique, respectivement. Les figures 16 et 17 montrent les résultats pour les deux images.

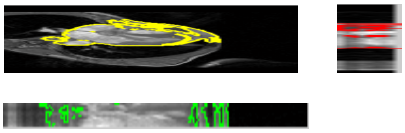


Fig.IV.16 Résultat après 100 itérations

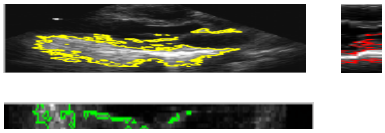


Fig.IV.17 Résultat après 600 itérations

5.4 Interprétation des résultats

Pour l'image « *mri* » de Matlab, nous remarquons que la segmentation est satisfaisante après 100 itérations sauf que, vu la dimension de l'image (128X128X27), le temps de traitement était long, soit huit heures d'exécution (30816s exactement). Pour l'image médicale IRM « *heart »*, de format DICOM, nous remarquons, aussi, que la ségmentation est bonne après 100 itérations, soit après trois heures de traitement (11268s exactement), sachant que sa dimension est de (256X256X16), plus grande que l'image précédente, ce qui nous laisse supposé que le format DICOM est avantageux en traitement d'images 3D. Pour l'image échographique de dimension (120X128X8), la ségmentation est rapide, ce qui nous a poussé à aller jusqu'à 600 itérations, juste pour voir la limite d'évolution de la surface. Sinon à 100 itérations, seulement, la ségmentation donne déja le même résultat.

5 Conclusion

L'objectif de notre travail, était de passer d'un contour actif 2D vers un en 3D, pour une segmentation des images médicales 3D. Vu la diversité des contours actifs, le choix s'est porté sur celui de *Chan* et *Vese*. Bien qu'il souffre du problème du temps de calcul, nous avons procédé par une adoption d'une méthode d'accélération de celui ci. Nous avons exposé, ensuite, les différents tests et résultats sur des images médicales 2D et 3D, afin d'illustrer l'évolution des contours (surfaces). L'algorithme s'est avéré efficace en termes de qualité de segmentation, et a permis des gains de vitesse significatifs en ce qui concerne le temps d'évolution du contour (surface) par rapport au modèle de *Chan et Vese*.

En guise de perspectives, nous pouvons suggérer de compléter le travail par l'ajout des approches d'évaluation de la segmentation, afin d'être plus objectif. Cependant, étudier et tester le maillage explicite représente une perspective intéressante, pour ce travail.

References

1. O. Chilali, K. Hammouche et M. Diaf, « Segmentation d'Images à Bases de Modèles Déformables », JIG'2007, 3émes Journées Internationales sur l'Informatique Graphique, Constantine les 29 et 30 octobre 2007.
2. O. Chilali, M. Diaf et A. Taleb-ahmed, « Application du modèle multiphasé pour la mise en évidence des lésions dans des images IRM », E-Medisys 2008, 2nd International Conference: Medical Systems, Sfax les 29, 30 et 31 octobre 2008.
3. O. Chilali et M. Diaf, « Segmentation d'images à base de modèles de *Chan* et *Vese* », CVA'2007, 2éme Conférence sur la Vision Artificielle, Tizi-Ouzou les 18, 19 et 20 novembre 2007.
4. B. Nazarian , **«** *Imagerie Médicale 3D : Visualisations, segmentations et reconstructions***»,** rapport de laboratoire CNRS , France, 2002.
5. M. Kass, A. Witkin, et D. Terzopoulos. "*Snakes : Active contour models*". International Journal of Computer Vision, 1(4) :321–331, 1988.
6. S. Osher et J. Sethian, « *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations* », Journal of Computational Physics ,79:12-49,1988.
7. L.D. Cohen et I.Cohen, « *Finite Element Methods For Active Contours Model And Balloons For 2D And 3D Image* », IEEE Transaction Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1131-1147, novembre 1993.
8. C. Xu et J.L. Prince, « *Snakes, Shapes And Gradient Vector Flow* », IEEE Transaction on Image Processing, no. 3, pp. 359-369, 1998.
9. V. Caselles, F. Catte, T. Coll, et F. Dibos, "*A geometric model for active contours*", Numerische Mathematik, V.66, pp. 1-31, 1993.
10. V. Caselles, R. Kimmel, et G. Sapiro, "*Geodesic active contours*", Int. J. Comput. Vis., V. 22, no. 1, pp. 61-79, 1997.
11. J.J Rouselle, J. Brilhault, D. Champion et L. Favard, « *Des Contours Actifs Pour Une Biométrie Du Fémur* », ORASIS 2001, Congrès de vision, France, juin 2001.
12. Tony F. Chan et Luminita A. Vese, « Active contours without edges », *Image Processing, IEEE Transactions on, february 2001.*
13. M. Julien, **«***Modèles déformables pour la segmentation et le suivi en imagerie 2D et 3D* **»,** thèse de doctorat, université de tours, France, 2007.
14. A. Dufour, N. Vincent et A. Genovesio, « *3D multi-object segmentation, tracking and visualization in fluorescence microscopy using Active Meshes* », In 2nd International Workshop on Pattern Recognition in Bioinformatics, Singapore, 2007.
15. K. Zhang, L. Zhang, H. Song et W. Zhou, « Active contours with selective local or global segmentation: A new formulation and level set method », *Image and Vision Computing 28 (2010) 668–676, 2010.*
16. S. Yonggang et W.C. Karl, **«** *A fast implementation of the level set method without solving partial differential equations* **»**, Technical Report No. ECE-2005-02, Department of Electrical and Computer Engineering, university of Boston, 2005.
17. S. Hachour, I. Feddag, Y. Iabadene et O. Chilali, « Study of an active contour model: application in real time tracking », ICEEA'10, Interational Conference on Electrical Engeneering, Electronics and Automatic, Béjaia les 01, 02 et 03 Novembre 2010.
18. A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J. Christophe, O. Marin, *et* C. Zimmer, **«** *Segmenting and tracking fluorescent cells in dynamic 3D microscopy with coupled active surfaces* **»**. *IEEE* Transactions on Image Processing, 14(9):1396–1410, 2005.
19. http://www.dclunie.com/medical-image

An Enhanced Bio-Inspired Firefly Algorithm for Remote Sensing Images Segmentation

Beghoura Mohamed Amine[1,1], Fizazi Hadria[1]

[1] SIMPA Laboratory, Computer Science Department, Science Faculty,
University of Science and Technology of Oran Mohamed Boudiaf,
Oran, Algeria.
{amine.beg@gmail.com, hadriafizazi@yahoo.fr}

Abstract. In our work, we applied a new recent bio-inspired optimization algorithm, called the Firefly Algorithm, to solve the problem of remote sensing images segmentation. The Firefly algorithm is inspired from the social behavior of the firefly bugs in the nature. The algorithm is used as an unsupervised classification algorithm which has to divide the images into a set of homogenous clusters. The efficiency of this operation becomes difficult when it comes to the segmentation of complex data such as the remote sensing images. In order to enhance the segmentation results, we have introduced the fuzzy logic and kernel distance to the algorithm in goal to solve the problem of fuzziness and nested groups in the data. The obtained results show the efficiency of our Kernel-based Fuzzy Firefly algorithm.

Keywords: Remote Sensing, Image Segmentation, Firefly Algorithm, Kernel Clustering, Fuzzy Segmentation.

1 Introduction

The image classification is a fundamental operation in the remote sensing process [1]. The task is to automatically extract the geographic characteristics of the images taken by a satellite or an aerial platform. This is by subdividing the image into a set of homogenous regions in respect of the characteristics of that image (e.g., the pixels intensity) [2]. The classification can be supervised and unsupervised [3]. In the supervised classification, a learning step is required to train the algorithms using a prior knowledge on the different classes in image. In satellite images, the classes may be the land covers such as the forest, sand, water, etc. The unsupervised classification is used in case where no prior information is known about the classes in the image. However, the results of the unsupervised classification are sets of pixels. The pixels in the same group show the same spectral characteristics [4]. In this paper we focus on the unsupervised classification of the images.

The biologically inspired algorithms are becoming powerful in the modern numerical optimization [7]. The image segmentation problem can be seen as an optimization problem and be solved using the bio-inspired optimization algorithms [11], where many algorithms have been used in this purpose. For example, Particular

Swarm Optimization (PSO) Algorithm [9], [15] and the genetic algorithms [10]. The bio-inspired approaches have shown a high efficiency in image segmentation compared to the classical classification algorithms (e.g., k-means [8]), which have a high complexity, the classical algorithms have shown many disadvantages due to their sensibility to the initialization and their convergence to the local optimal solutions.

In some problems, the data do not belong to a single group. These kinds of situations cannot be solved with the deterministic methods, which affect each data point into one and only one group. As a solution, fuzzy segmentation methods, especially the fuzzy c-means algorithm (FCM) [5],[12], have been widely used. This is due to the introduction of the fuzziness for belongingness of each image pixel. Unlike the hard segmentation methods, which force pixels to belong exclusively to one class, the fuzzy segmentation methods allow pixels belong to multiple classes with varying degrees of membership that makes the fuzzy segmentation algorithms able to retain more information from the original image than the hard segmentation methods [13].

In this paper, we propose a new kernel-based fuzzy image segmentation approach based on the new recent Firefly Algorithm [14]. The purpose of this approach is to overcome the problem of imprecision and incertitude by introducing the fuzziness logic, to enhance robustness against noise and ability to classify complicated data structure due to the injection of a robust non-Euclidean distance measure, which is obtained through a nonlinear mapping by using Gaussian radial basis function (GRBF). In this paper, we start by formalizing the segmentation problem, we introduce the theory behind the firefly algorithm, and then we present our approach of the image segmentation and the obtained results.

2 Problem Formalization

In the case of image segmentation problem, Let consider $P = \{P_1, P_2, \ldots, P_n\}$ to be a set of n pixels each having d features, where $d = 3$ if we are using the radiometric values of the pixels. These pixels can also be represented by a profile data matrix $Z_{n\times d}$ having n rows of d dimension. The i^{th} row vector Z_i characterizes the ith pixel from the set P and each element $z_{i,j}$ in Z_i corresponds to the j^{th} real value feature ($j = 1, 2, \ldots, d$) of the i^{th} pixel ($i = 1, 2, \ldots, n$). Given such a $Z_{n\times d}$ a segmentation algorithm tries to find out a partition $C = \{C_1, C_2, \ldots, C_k\}$ such that similarity of the pixels in the same cluster C_i is maximal and pixels from different clusters differ as far as possible. Therefore, the problem can be seen as finding an optimal C^* that minimizes or maximizes the function f:

$$\text{Optimize } f(Z_{n\times d}, C) \tag{1}$$

f is an objective function that qualifies the quality of the segmentation based on similarity or dissimilarity measures. The pixels are assigned to the groups having the nearest gravity center C_i. The Euclidean distance is the most known measurement function:

$$d(\vec{Z}_u\vec{Z}_v) = \sqrt{\sum_{i=1}^{d}(z_{u,i} - z_{v,j})^2} = \|\vec{Z}_u - \vec{Z}_v\| \quad (2)$$

Remote sensing images segmentation, especially the images taken by satellites, are confronted to a certain degree of imprecision and incertitude. This is due to the affinity of the image pixels to more than one class of the land cover. In these images, a pixel can represents a geographic area relatively large which also may contain more than one class [20]. This need can be satisfied by using the fuzzy sets instead of deterministic groups of pixels to represent geographic objects with uncertain boundaries [21]. The following figure illustrates the fuzziness problem:

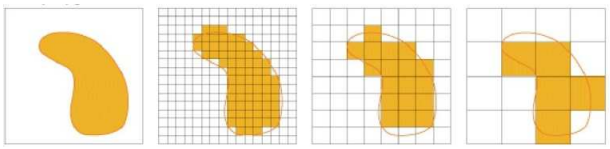


Fig. 1. Imprecision and Incertitude in Remote Sensing Images

The figure 1 (a) contains the object to be taken in image. The figures (b), (c) and (d) illustrate images of different resolutions of the object in (a), where (b) represent a high-resolution image and (d) a low-resolution image. In figure 1 (d), the pixels of the image do not fit the real world object boundaries.

The use of a fuzzy logic to solve problem of imprecision and incertitude is by introducing a membership degree to represent the probability that a pixel belong to a group. This evolves the creation of a partition matrix $\mathbf{U} = [u_{ij}]_{nxk}$, $u_{ij} \in [0,1]$ and $\sum_{j=1}^{k} u_{ij} = 1$, where u_{ij} denotes the grade of membership of the *i*-th pixel to the *j*-th cluster.

$$u_{ij} = \sum_{m=1}^{k} \left(\frac{\|\overrightarrow{z_i} - \overrightarrow{c_j}\|}{\|\overrightarrow{z_i} - \overrightarrow{c_m}\|} \right)^2 \quad (3)$$

A major drawback to the unsupervised classification methods using the Euclidean distance is that they cannot separate non-linearly separable clusters in the input space. The problem of the non-linearly separable cluster is frequent in the remote sensing data classification. It is the main source of misclassification. This is problem can be due to the noise in the data, the similar characteristics of objects and the data complexity (*e.g.*, hyper spectral data).

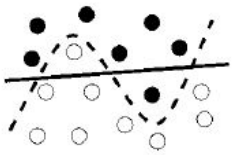


Fig. 2. Non-linear data separation

The approach to solve this problem is to adopt the strategy of nonlinearly transforming the data into a high-dimensional feature space and then performing the clustering within this feature space. A nonlinear mapping function φ is defined as: φ: $z \rightarrow \varphi(z) \in H$, where z is a data vector in the initial feature space, H is transformed feature space with a higher dimension. The distance between any two data points, z_i and z_j, in the new feature space is:

$$\left\|\varphi(z_i) - \varphi(z_j)\right\| \tag{4}$$

Where,

$$\left\|\varphi(z_i) - \varphi(z_j)\right\|^2 = k(z_i, z_i) + k(z_j, z_j) - 2k(z_i, z_j) \tag{5}$$

k is a kernel function. One of most known kernel function is the Gaussian function, it is defined as:

$$k(x, y) = \exp\left(-\left|\frac{|x - y||^2}{\sigma^2}\right.\right) \tag{6}$$

3 The Firefly Algorithm

Fireflies are small beetles. Their name comes from their capability to produce short and rhythmic flashes by a bioluminescence process. This phenomenon allows the fireflies to attract other fireflies (mates) or preys. Following the behavior of the fireflies in the nature, the algorithm was introduced by Xin-She Yang in 2008. It is a population based algorithm that reproduces the mechanism of attraction between fireflies in that population. The algorithm has a lot of similarities with other population based algorithms such as the PSO algorithm, the bees colony algorithm and the bacterial foraging algorithm [17], [18]. The comparison studies done by the author of this algorithm showed the efficiency of the Firefly algorithm on both of the PSO algorithm and the genetic algorithms.

The Firefly algorithm takes in consideration the three following rules [16]:

1) The attraction process occurs between firefly in regardless of their sex.

2) attractiveness is proportional to the brightness of the fireflies. It is defined as the reverse proportion of the their distances between them. For any two flashing fireflies, the less bright one will move towards the brighter one.

3) The brightness of a firefly is determined by the objective function.

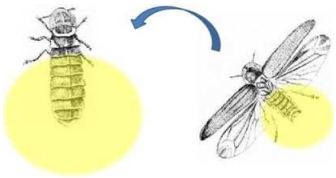


Fig. 3. Attractiveness between fireflies

In the firefly algorithm, the attractiveness function of a firefly is the defined by the following monotonically decreasing function [19]:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad (\mathrm{m} \geq 1) \tag{7}$$

Where, r is the distance between two fireflies, β_0 is the initial attractiveness at $r = 0$, and γ is an absorption coefficient which controls the decrease of the light intensity and which is defined at the initialization of the algorithm. The distance between two fireflies i and j is defined as:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + \left(y_i - y_j\right)^2} \tag{8}$$

The movement of a firefly i at the position x_i toward a brighter firefly j is given by the following equation

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha(rand - 1/2) \tag{9}$$

The first term x_i is the current position of the firefly i, the second term determines the attractiveness of the adjacent fireflies in by the light intensity. The third term is used for the random movement of a firefly. The coefficient $\alpha \in [0, 1]$ is a randomization parameter determined by the problem of interest, while rand is a random number generator uniformly distributed in the space [0, 1]. γ determines the convergence speed of the algorithm towards the optimal solution.

The Firefly algorithm strategy to solve the optimization problems is to move the fireflies of the swarm into the search space in a way where each firefly tries to have

the highest light intensity. Thus, the moves of the firefly will optimize the objective function *f*.

4 Image Segmentation

Before using the firefly algorithm in the segmentation of the images, the algorithm has to be adapted to efficiently solve the problem. First, we define a representation of the fireflies, so that each firefly has a position vector of *d* x *k* dimension, where *d* is the feature space dimension, for example, *d* = 3 in case of using the radiometric values (red, green, blue) of the pixels as features, *k* is the number of clusters. Thus, the position of a firefly *i* is defined as $x_i = \{C_{i0}, C_{i1}, \dots, C_{ik}\}$ where C_{ik} is the center of the k^{th} cluster. After defining the firefly's position vector, we need an evaluation function to define the light intensity of the fireflies depending on their positions. Many validity indexes exist in the literature [22] and which can be more or less convenient to be used as an objective function in the image segmentation problem. A well known and used index is the J_m index [6], it is defined as:

$$J_m(U, C) = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{m} \left\|\vec{z_i} - \vec{c_j}\right\|^2 = f(x_i) \tag{10}$$

Where, u_{ij} is the membership degree of the pixel *i*, represented by z_i, into the cluster *j* which is represented by the center c_j. The parameter *m* defines the clustering fuzziness. The function J_m has to be minimized by the firefly algorithm, a minimal value indicates an optimal segmentation.

After defining the clustering quality measurement function, we introduce the kernel mapping on the J_m index using the Gaussian function. The purpose is to calculate the clustering quality after mapping the data into a high dimensional space:

$$J_m^{\varphi} = 2\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{\varphi}(1 - k(\vec{z_i}, \vec{c_j})) \tag{11}$$

Where

$$u_{ij}^{\varphi} = \frac{\left[1 - k\left(\vec{z_i}, \vec{c_j}\right)\right]^{\frac{1}{1-m}}}{\sum_{m=1}^{k}[1 - k(\vec{z_i}, \vec{c_m})]^{\frac{1}{1-m}}} \tag{12}$$

Thus, we define J_m^{φ} as an objective function $f(x_i)$ to qualify the light intensity of the Fireflies in the population.

The Firefly Algorithm is an iterative algorithm, the algorithm ends when a stop condition is reached. In our case, we define a maximal number of iterations *max-t* as a condition. The following pseudo code describes our Kernel based fuzzy firefly algorithm:

Step 1: Load the Images

Step 2: Preprocessing operations

Step 3: Define the Algorithm parameters:

{*n* population size, *t* number of iterations, *k* number of groups, σ, γ and β_0}

Step 4: (The Firefly Algorithm)

For each firefly *i* at the position x_i do

  For each firefly *j* at the position x_j do

    if $f(x_i) < f(x_j)$

      Move the firefly *i* toward the firefly *j* using (9).

      For each Pixel in the image do

        Calculate the Euclidean distance between the pixel and the centers in x_i

        Assign the pixel to the group having the nearest center

        Calculate the membership degrees of the pixel using the equation (12)

      End For

      Evaluate the new light intensity of the firefly *i* by $f(x_i)$ using the equation (11).

    End if

  End For

End For

Step 5: Repeat 4 until t = t-Max

Step 6: Visualize the results

In the following section, we present the results of images segmentation using the firefly algorithm.

5 Implementation and Results

The new firefly based approach has been applied on two images, a 35x50 noisy synthetic image (Figure 4) which contains three spectral groups (Red, Yellow and Bleu) that can be easily distinguished.

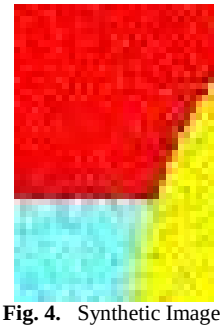


Fig. 4. Synthetic Image

The second image represents the region of Oran (Algeria). The image is a 200x170 taken by the satellite SPOT4 in 2001.

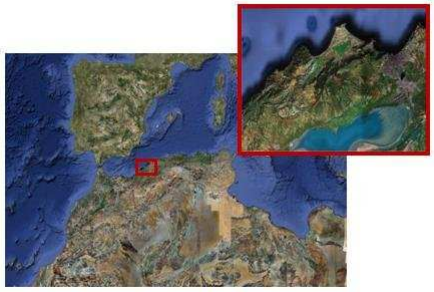


Fig. 5. Region of Oran

A preprocessing step is required before the segmentation in order to make corrections and enhancement on the images. We start our application by loading three images corresponding to the red, the green and the blue channels. In our case, we use the SPOT4 satellite near infrared, red, and green as channels. After the loading, we apply contrast enhancement operations, the obtained images are combined as one three colors image:

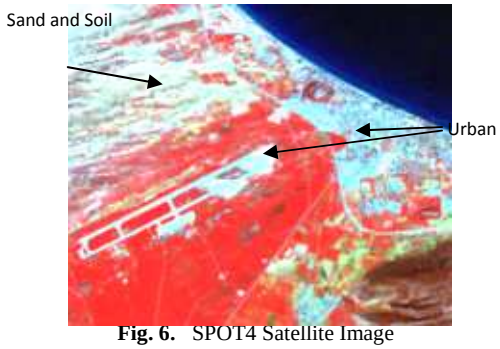


Fig. 6. SPOT4 Satellite Image

The satellite image (figure 6) shows regions with nearly similar radiometric values (i.e., the Urban and the Sand/Soil) which may create confusions in the segmentation results.

After running the algorithm for several times on the test images, the used parameters and the best obtained results are as follow:

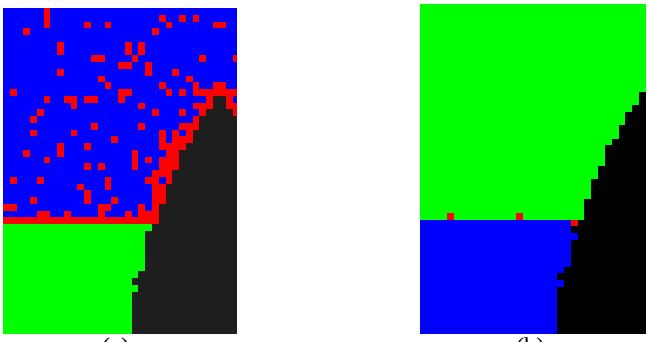
5.1 The Synthetic Image

The synthetic image is partitioned into $k = 4$ groups, the results of the segmentation using the Gaussian Fuzzy Firefly Algorithm (G.F.FFA) are compared with the results of the standard Firefly Algorithm (FFA). Both of the algorithms are experimented using the following parameters:

$$n = 10, \quad t = 20, \gamma = 0.005, \beta_0 = 1 \text{ and } \sigma = 1000.$$

Table 1. The synthetic image segmentation results.

	K groups	Jm index	Processing Time (ms)
Firefly Algorithm	4	1.914×10^6	2427
Gaussian Fuzzy Firefly Algorithm	4	0.0051	4186



(a) (b)

Fig. 7. Visual results of the segmentation of the synthetic Image (a) segmentation by the firefly algorithm (b) segmentation by the Gaussian Fuzzy firefly algorithm.

The visual results show that the segmentation performed by our G.F.FFA is better than the segmentation obtained by the standard Firefly Algorithm. However, the G.F.FFA took a much longer execution time (*TE*) than the standard Firefly Algorithm.

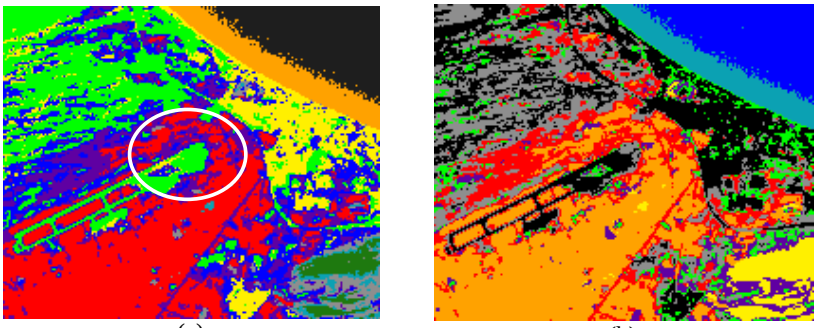
5.2 The Remote Sensing Image

The remote sensing image is partitioned using $k = 10$ groups. Both of the algorithms are experimented using the following parameters:

$$n = 20,\ t = 30,\ \gamma = 0.0005,\ \beta_0 = 1 \text{ and } \sigma = 1000.$$

Table 2. The remote sensing image segmentation results.

	K groups	Jm index	Processing Time (ms)
Firefly Algorithm	10	$3.572 * 10^7$	42231
Gaussian Fuzzy Firefly Algorithm	10	0.055	1964239



(a) (b)

Fig. 8. Visual results of the segmentation of the remote sensing image by (a) Firefly Algorithm (b) Gaussian Fuzzy Firefly Algorithm

In figure 9, the Firefly algorithm failed to separate the urban region (yellow) from the sand/soil land cover (green). However, the Gaussian fuzzy firefly algorithm has well separated the image regions, where the segmentation fit the reality on the land. In the other side, the firefly algorithm took a shorter processing time than the G.F.FFA.

6 Conclusion

In this paper, we proposed a new image segmentation approach based a new recent algorithm that simulates the behavior of fireflies in nature, to solve two image segmentation main problems which are: 1) the imprecision and incertitude. 2) Data noise and complexity. We have applied our approach on a set of images. We have presented its efficiency using a synthetic image to easily distinguish the quality of the segmentation by eyes. Then, we used a SPOT4 satellite image characterized by the complexity of its regions. Compared to the real partition of groups, our experiments showed the efficiency of our approach compared to segmentation using the standard version of the algorithm.

7 References

1. Wilkinsom G G.: "Results and implications of a study of Fifteen years of satellite image classification experiments." IEEE T Geosci Remote Sens, 43(3): 433—440, 2005.
2. Jain, A.K., Murty M.N., and Flynn P.J.: "Data Clustering: A Review, ACM Computing Surveys", Vol 31, No. 3, 264-323, 1999.
3. Björn Waske, Mingmin Chi, Jón Atli Benediktsson, Sebastian van der Linden, Benjamin Koetz: "Algorithms and Applications For Land Cover Classification – A Review" D. Li et al. (eds.), Geospatial Technology for Earth Observation, 2009.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

4. T Lillesand, R Kiefer: "Remote Sensing and Image Interpretation", John Wiley & Sons Publishing, 1994.
5. Pal NR, Bezdek JC (1995) On cluster validity for the fuzzy c-means model. IEEE Trans Fuzzy Syst 3:370–379
6. Kuo-Lung Wu a, Miin-Shen Yang, "A cluster validity index for fuzzy clustering", Pattern Recognition Letters 26 1275–1291, 2005.
7. Carlos M. Fonsecay and Peter J. Flemingz: "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", Genetic Algorithms: Proceedings of the Fifth International Conference (S. Forrest, ed.), San Mateo, CA: Morgan Kaufmann, July, 1993.
8. JA Hartigan, *Clustering Algorithms*, John Wiley & Sons, New York, 1975.
9. J Kennedy, RC Eberhart, Particle Swarm Optimization, Proceedings of the IEEE International Conference on Neural Networks, Vol 4, Perth, Australia, pp 1942-1948, 1995.
10. J. H. Holland. "Adaptation in natural and artificial systems ". Ann Arbor: University of Michigan Press. 1975.
11. Brucker, P.: On the complexity of clustering problems. In: Beckmenn, M., Kunzi, H.P. (eds.) Optimization and Operations Research. Lecture Notes in Economics andMathematical Systems, vol. 157, pp. 45-54. Springer, Berlin (1978).
12. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
13. Zhang, D.Q., Chen, S.C.: A novel kernelized fuzzy C-means algorithm with application in medical image segmentation. Artif. Intell. Med. 32, 37–50 (2004)
14. X.-S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178, 2009.
15. Omran. M, Salman. A, Engelbrecht. A.P, "Image classification using particle swarm optimization", Conference on Simulated Evolution and Learning, vol. 1, pp. 370–374, 2002.
16. X.S. Yang, "Firefly Algorithms for Multimodal Optimisation", Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, vol. 5792, 2009, pp. 169-178.
17. X.-S. Yang, "Firefly Algorithm, Levy Flights and Global Optimization," Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, Petridis), Springer London, 2010, pp. 209-218.
18. S. Lukasik and S. Zak, "Firefly algorithm for con-tinuous constrained optimization tasks" in Proceedings of the International Conference on Computer and Computational Intelligence (ICCCI "09), N. T. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds., vol. 5796 of LNAI, pp. 97-106, Springer, Wroclaw, Poland, October 2009.
19. X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
20. R. Jeansoulin et al. (Eds.): Methods for Handling Imperfect Spatial Info., STUDFUZZ 256, pp. 103–129. 2010.
21. Burrough, P.: Natural objects with indeterminate boundaries. In: Burrough, P.A., Frank, A. (eds.) Geographic Objects with Indeterminate Boundaries, pp. 3–28. Taylor & Francis, London (1996)
22. Kuo-Lung Wu a, Miin-Shen Yang, A cluster validity index for fuzzy clustering, Pattern Recognition Letters 26 1275-1291, 2005.

Détection des Visages par Méthode Hybride: Réseaux de Neurones et Transformé Discrète en Cosinus

Amir Benzaoui[1], Houcine Bourouba
Laboratory of Inverse Problems, Modeling, Information and Systems (PI:MIS)
Department of electronics and telecommunication
University of Guelma, P.O box 401 Guelma, 2400 Algéria
amirbenzaoui@gmail.com

Hayet Farida Merouani
Laboratory of computer research (LRI),
Department of computer sciences
University of Badji Mokhtar, BP.12 Sidi Amar 23000 Annaba, Algeria
hayet.merouani@univ-annaba.org

Résumé. Les performances de la détection des visages dans un système biométrique basé sur le visage ont une importante influence, d'une part le temps de la détection représente 80% du temps total pris par le système et d'autre part une mauvaise détection entraine automatiquement une mauvaise reconnaissance d'identité. Dans ce travail, on cherche à combler la lacune entre les modèles informatisés de la vision et leur homologue humain. Pour ce faire, nous avons proposé une nouvelle méthode de détection dans des cas non contraints, c'est à dire dans des situations où l'éclairage, l'occlusion, la taille et la position du visage ne sont pas contrôlés, cette méthode se base essentiellement sur une technique d'apprentissage automatique en utilisant la décision de plusieurs réseaux de neurones et une technique de compactage d'énergie en utilisant la transformée discrète en cosinus. Un ensemble de visages et de non visages est transformés vers des vecteurs de données qui seront utilisés pour entraîner les réseaux à séparer entre les deux classes tandis que la transformée discrète en cosinus est utilisée pour réduire la dimension des vecteurs, éliminer les redondances d'information et de ne conserver que l'information utile dans un nombre minimum de coefficients. Les résultats expérimentaux ont montré que l'introduction de la transformée discrète en cosinus avec une technique d'apprentissage automatique a donné une amélioration très importante du taux de la reconnaissance et de la qualité de la détection des visages.

Mots clés: Localisation du visage, Détection des visages, Reconnaissance du visage, Les réseaux de neurones, La transformée discrète en cosinus.

1 Introduction

Le problème de la détection des visages est considéré comme la première étape de tout système biométrique qui utilise le visage comme un moyen d'identification. Récemment, plusieurs méthodes de détection ont été proposées, on peut regrouper ces méthodes en deux grandes approches: (1) Les méthodes locales: dans ce cas, l'analyse

du visage humain est donnée par la description individuelle de ses parties et de leurs relations, on peut citer: les méthodes qui se basent sur les caractéristiques invariantes [1], la mise en correspondances [2]…et (2) les méthodes globales: qui se basent sur les techniques d'apprentissage automatiques comme les réseaux de neurones [3], les séparateurs à vaste marge [4],.... Cependant, les systèmes de la vision humaine ont une efficacité impeccable puisqu'elles ont des solutions fournies naturellement par le cerveau humain alors que les modèles informatisés de la vision ne présentent pas encore une grande ressemblance à ces derniers [5].

Dans cet article, on cherche à combler la lacune entre les modèles informatisés de la vision et leurs homologues humains. Pour ce faire, nous avons proposé un système hybride de détection des visages dans une image couleur dans des cas non contraints, qui se base essentiellement sur une technique d'apprentissage automatique en utilisant la décision de plusieurs réseaux de neurones, l'apprentissage s'effectue à partir de deux bases d'images (visage, non visage) qui représentent deux classes à séparer. Ce système se caractérise par un apprentissage au niveau fréquentiel en utilisant la transformée discrète en cosinus (DCT), cette dernière est une technique qui présente une très grande efficacité en terme de compactage d'énergie, elle permet de stocker un maximum de pixels dans un minimum de coefficients localisés en basses fréquences. Nous avons profité l'avantage de cette transformation afin de réduire la dimension des vecteurs et d'éliminer une très grande quantité d'information non utile ce qui permet d'améliorer progressivement la qualité de l'apprentissage.

Cet article est constitué de 5 sections: après une introduction générale à la section I, nous illustrons le principe de compactage de l'énergie par la DCT dans la section II, le fonctionnement interne du système proposé est détaillé dans la section III. Les résultats expérimentaux sont exposés dans la section IV. Nous terminons l'article par une conclusion générale et des perspectives de ce travail; objets de la dernière section.

2 La transformée discrète en cosinus et le compactage d'énergie

Le temps de la décision et la précision dans un système de reconnaissance des formes sont en relation directe avec les vecteurs de donnés utilisés pour effectuer l'apprentissage. La dimension et la corrélation des vecteurs ont un effet direct sur la qualité de l'apprentissage et sur le temps de la décision. En effet, La pertinence de l'information n'est pas aisée à apprécier dans le domaine direct, donc on cherche à effectuer des changements de représentation qui permettent de faciliter la séparation de l'information la plus pertinente. Cette approche est celle des transformations orthogonales, qui remplacent le signal d'origine par sa représentation suivant une base de fonctions, généralement par traitements de blocs.

Plusieurs transformations sont utilisées dans ce genre de problèmes, les plus répandues sont la transformée de Fourier (DFT), la transformée de Karhunen-Loève (KLT) et celle en cosinus discrète (DCT) qui sera utilisée dans la conception de notre système. La transformée de Karhunen-Loève est reconnue par sa précision et son exactitude mais elle présente un inconvénient majeur à cause de la complexité de ses calculs. La transformée de Fourier discrète (DFT) est une méthode très rapide grâce à la simplicité de ses algorithmes mais sa périodicité horizontalement et verticalement

cause une imprécision dans certain cas. La transformée en cosinus discrète est venue pour équilibrer le compromis existant entre la précision et la vitesse, elle a hérité l'exactitude de la KLT et la performance de la DFT grâce à ses algorithmes simples, rapides et efficaces [6].

La transformée en cosinus discrète est une transformation mathématique qui a été introduit par Ahmed, Natarajan et Rao en 1974 [7], qui transforme un ensemble de données d'un domaine spatial vers un spectre fréquentiel [7], elle permet un changement du domaine d'étude tout en gardant exactement la même fonction étudiée. Dans notre cas, on étudie une image, c'est-à-dire une fonction à 2 dimensions: X et Y, indiquant les cordonnées spatiales du pixel et en sortie la valeur du pixel (coefficient) en ce point. La DCT d'un signal discret à deux dimensions est définie comme suit: Soit $f(x,y)$ la fonction décrivant une séquence de longueur *N*N*. La transformée en cosinus discrète 2D de la séquence $f(x,y)$ notée $C(u,v)$ est la suivante [8]:

$$C(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N} \qquad (1)$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } u, v = 0 \\ 1 & \text{sinon} \end{cases}$$

La DCT présente une très grande efficacité en termes de compactage d'énergie, elle permet de stocker un maximum d'informations dans un minimum de coefficients localisés en basses fréquences (coin supérieur gauche) et à chaque fois qu'on redescend vers les hautes fréquences (en bas à droite) il y aura une dégradation d'informations [9]. Ceci permet d'écarter les coefficients ayant des amplitudes relativement petites sans présenter une déformation des caractéristiques de l'image (fig.1).

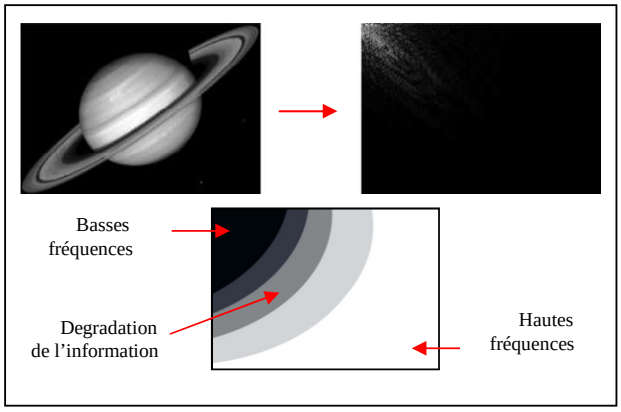


Figure 1 Distribution fréquentielle des coefficients DCT [9].

3 Architecture du système proposé

Tout processus automatique de détection doit prendre en compte plusieurs facteurs qui contribuent à la complexité de sa tâche, car le visage est une entité dynamique qui change constamment sous l'influence de plusieurs facteurs. La structure générale du système de la détection proposée comporte deux phases: une phase d'apprentissage et une phase de détection, cette structure est montrée dans la figure suivante:

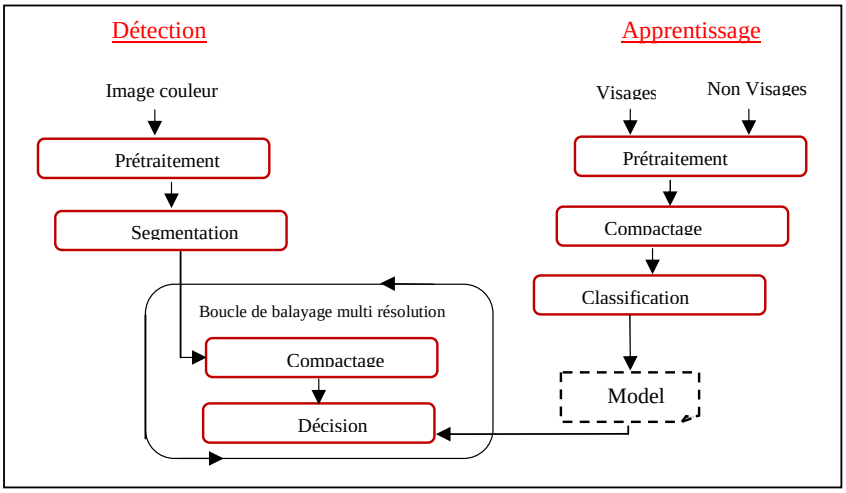


Figure 2 Architecture du système.

3.1 Scenario pour la phase d'apprentissage

Cette phase est nécessaire pour que le système puisse fonctionner. Le système d'apprentissage va avoir en entrée deux bases de données d'images qui représentent les deux classes à séparer. L'une d'entre elles contient des images de visages humains, ces visages sont différents deux à deux en fonction des personnes et de la position du visage (incliné, droit, rotation, …), en plus ces images sont prises dans des conditions variées concernant l'environnement (luminosité, éclairage, …), afin de couvrir au mieux les conditions réelles d'utilisation ultérieures. L'autre base contient des images non visages (objets, partie de visage,…) considérés comme des contres exemples. Le système va analyser le problème, extraire des données, classifier les images et créer un model d'apprentissage (fonction de classification) selon des paramètres qui donneront un meilleur résultat, puis va stocker le model pour qu'il soit utilisé lors de la phase de détection. Nous allons maintenant expliquer le fonctionnement interne de cette phase, en présentant les modules les plus importants qui la composent.

- *Le prétraitement*

Une image utilisée pour l'apprentissage ne peut pas être exploité dans son état brut à cause du bruit (effet de luminance et du background) qui influe négativement sur les performances du système ce qui nécessite des prétraitements bien définis tel que: le passage au niveau gris, la normalisation, l'égalisation de l'histogramme.

- *Le compactage*

Le rôle de ce module est de préparer les données qui sont les images d'entrée pour un éventuel apprentissage, (test et décision). Ce module consiste à récupérer chaque image, après le prétraitement, sous forme d'une matrice de niveaux de gris, puis leurs appliquer la DCT afin d'extraire un vecteur de coefficients qui représente l'information utile de chaque image. Cette opération est effectuée en plusieurs étapes comme le montre la figure suivante:

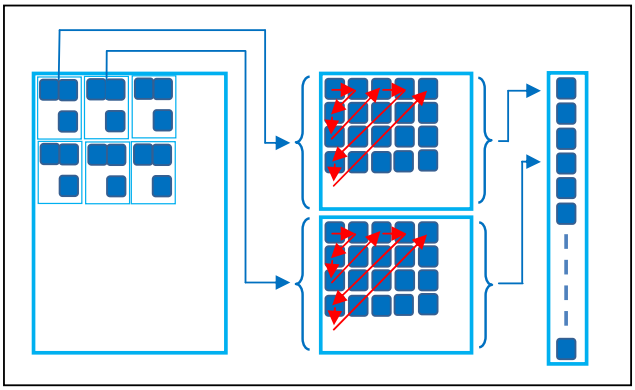


Figure 3 La construction du vecteur des coefficients.

La première étape consiste à découper la matrice de niveau de gris caractérisant l'image en plusieurs blocks de NxN pixels, généralement N égal à 8, cette valeur a été choisie car c'est un bon compromis entre la qualité de l'application de la DCT et la rapidité des calculs. Ensuite, la DCT est appliqué sur chaque block découpé ce qui assure que les amplitudes des composantes de basses fréquences (informations pertinentes de l'image) sont plus élevées que celle des composantes de hautes fréquences (détail de l'image). Ainsi les coefficients de plus petite amplitude peuvent être éliminés et donc l'image est représentée par un petit nombre de coefficients. Finalement, la sélection des coefficients de basses fréquences est effectuée par un balayage en zigzag. La méthode zigzag vient pour permettre la récupération de ces données dans un ordre décroissant d'énergie. Elle consiste à parcourir les éléments de la partie supérieure gauche de la matrice transformée dans un ordre bien précis à partir des fréquences les plus basses vers celles les plus hautes. Nous obtenons à la fin une partie du vecteur de données classées selon la fréquence spatiale.

- *La classification et la décision*

La tâche de la classification est d'employer les vecteurs de coefficients fournis par l'étape précédente (compactage d'énergie) pour assigner les objets d'intérêt à une catégorie ou à une classe (positive ou négative) (fig.4). En d'autre terme, à partir d'un ensemble de vecteurs de coefficients extraits, une méthode d'apprentissage est employée pour former une fonction de classification efficace.

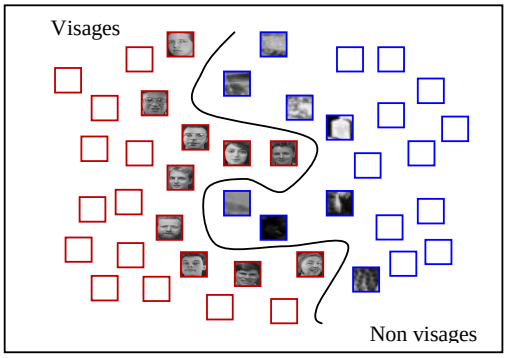


Figure 4 La classification.

Notre système emploie un réseau neurologique artificiel comme classificateur. Selon Haykin [10], un réseau de neurones est un processeur massivement distribue en parallèle qui a une disposition naturelle pour stocker de la connaissance empirique et la rendre disponible à l'usage.

Comme il ya plusieurs types de réseaux de neurones, nous avons choisis le modèle le plus performant des réseaux neurologiques: le perceptron multicouche (MLP), Le MLP apprend de l'échantillon d'entrée et de la valeur de sortie correspondante en changeant les poids synaptiques entre les raccordements. Pour que le MLP analyse le problème donné il faut lui faire un apprentissage, ce dernier agi sur les poids du réseau d'une manière à ce que l'erreur de la détection soit inférieure à un certain seuil. L'algorithme d'apprentissage utilisé dans ce module est le retro-propagation de gradient. Afin de rendre le système plus efficace c.-à-d. réduire le nombre de fausses détections, nous avons entraîné trois réseaux neurones à classifier et ensuite à choisir en fonction de leurs réponses, celles qui doivent être conservées ou éliminées. Les performances de chaque MLP dépendent de l'initialisation des poids, du pas d'apprentissage et du nombre de neurones de la couche cachée, c'est pourquoi nous avons réalisé plusieurs tests avant de préserver une topologie donnant la meilleure performance (l'apprentissage est arrêté lorsque l'erreur quadratique moyenne est ≤ 0.001).

3.2 Scenario pour la phase de détection

Dans ce cas, le système reçoit en entrée une image couleur, l'image sera prétraitée et balayée par un cadre de taille fixe, qui à son tour, passera par la fonction de décision

pour voir si c'est un visage ou non et de l'encadrer si oui. Nous allons maintenant expliquer le fonctionnement interne de cette phase, en présentant les modules les plus importants qui la composent.

- *La segmentation*

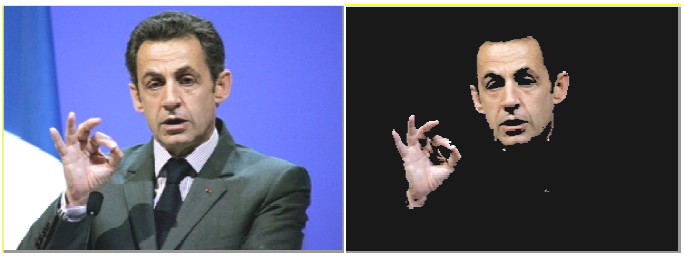
Dans le but d'accélérer le processus de la détection, nous avons utilisé une technique de segmentation par la couleur de la peau humaine, cette technique est largement utilisée et s'est avérée être une caractéristique efficace dans plusieurs applications tel que: la détection de mouvement humain et la vidéosurveillance. Il est avantageux d'utiliser cette information supplémentaire pour isoler les régions susceptibles de contenir des visages. En effet, l'élimination des régions n'ayant pas la couleur de la peau réduit le nombre de cadres qui sont envoyés aux deux modules (compactage et décision) et donc il y aura une optimisation de la recherche et du temps de détection.

Bien que la couleur de la peau puisse largement varier, les récentes recherches montrent que la différence principale est plutôt dans l'intensité que dans la chrominance [11]. Plusieurs espaces de couleur sont utilisées pour étiqueter les pixels comme pixels de couleur de peau: RGB, RGB normalisé, HSV, YCrCb, YIQ[11],... De plus, des méthodes sont proposées pour construire un modèle de couleur de peau : méthode utilisant la tonalité de pixel, méthode non paramétriques basée sur l'histogramme [12], méthode paramétriques utilisant une fonction gaussienne [13].

Pour notre travail, une méthode simple, efficace, robuste et en temps réel pour segmenter les blobs de couleur de peau est le bon choix. Pour ce faire, nous avons choisi une méthode de segmentation par seuillage en utilisant l'espace couleur RGB, ce dernier présente le meilleur taux de précision montré dans [12], le seuillage appliqué sur cet espace s'applique aux trois canaux (Red, Green, Blue), l'ensemble des pixels de l'image est classifié en 2 classes, une classe couleur de la peau et une classe non-couleur de la peau selon la règle de classification suivante[12] :

$$\begin{cases} (\mathrm{R} > 95) \text{ et } (\mathrm{G} > 40) \text{ et } (\mathrm{B} > 20) \text{ et} \\ \left((\mathrm{Max}(\mathrm{R},\mathrm{G},\mathrm{B}) - \mathrm{Min}(\mathrm{R},\mathrm{G},\mathrm{B})) > 15 \right) \text{et} \\ (\mathrm{ABS}(\mathrm{R}-\mathrm{G}) > 15) \text{ et } (\mathrm{R} > G) \text{ et } (\mathrm{R} > B) \end{cases} \qquad (2)$$

On obtiendra en fin de traitement une image où apparaissent seulement les pixels qui vérifient les conditions fixées pour la couleur de peau (fig.5).



(a) (b)

Figure 5 Segmentation de la couleur de la peau humaine.
(a) Image originale. (b) Image segmentée.

- *Le balayage multi résolution*

L'objectif recherché du système étant la détection des visages dans une image. Pour ce faire, une fenêtre est défini de taille 19x19 pixels qui va parcourir dans toutes les régions de couleur de peau dans l'image à la recherche des visages. Cette fenêtre a une taille fixe pour servir de donnée entrante au classificateur. C'est ce qui nous amènera à balayer l'image à plusieurs échelles. Aussi, la détection de visage ne pourra s'opérer que sur des images d'une taille minimale de 19x19 pixels. Le balayage démarrera donc sur une image à sa taille initiale, puis l'image sera successivement réduite à chaque changement d'échelle afin de pouvoir repérer des visages plus ou moins près de cette image (fig.6).

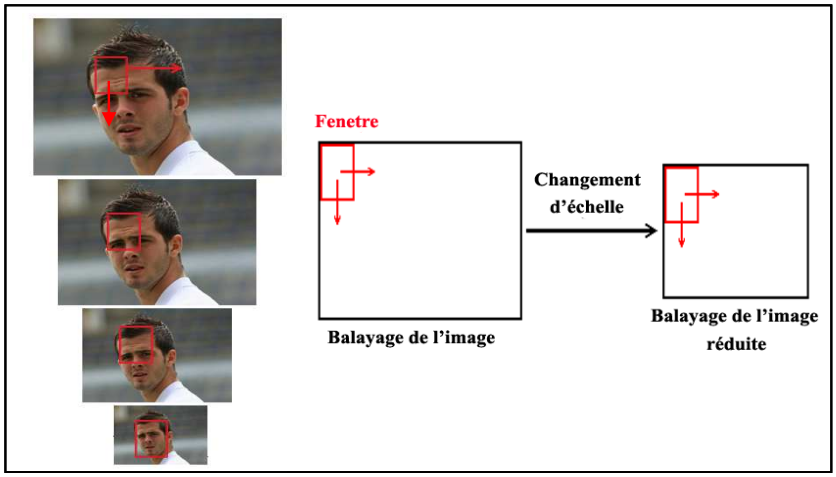


Figure 6 Le balayage multi résolution.

4 Résultats expérimentaux

Pour construire des données d'apprentissage, nous avons utilisé la base de données MIT CBCL [14], cette base est utilisée pour l'apprentissage et le test des réseaux de neurones. Cette base de données se compose de 2 parties : l'ensemble des images d'apprentissage, et l'ensemble des images de test. La première partie se compose de 6977 images (2429 visages, 4548 non visages) et le deuxième partie se compose de 24045 images (472 visages, 23573 non visages). Ces images sont au niveau de gris et ont une taille fixe de 19x19 pixels. Le tableau suivant présente les meilleurs résultats obtenus lors de l'évaluation du système de détection, et cela se fait par des tests successifs effectués selon plusieurs configurations des réseaux de neurones, en comparant aussi entre les deux modes suivants :

- MLP: le classificateur MLP sera utilisé seul sans compactage d'énergie.
- MLP- DCT: le classificateur sera utilisé avec compactage d'énergie en utilisant la DCT.

Table 1 Les meilleurs résultats obtenus.

	Taux de reconnaissance	**Taux de fausses acceptations**
MLP	**55.03**	**9.46**
MLP-DCT	**79.56**	**3.07**

D'après les résultats obtenus, nous déduisons que l'utilisation de la méthode de compactage de l'énergie (DCT) associée au réseau de neurone est indispensable pour améliorer les performances du détecteur. Car présenter des données brutes au classificateur sans aucune transformation n'est pas efficace du point de vue du taux de détection. Donc, la réalisation du classificateur par la méthode MLP-DCT présente un bons taux de détection.

La méthode proposée dans cet article présente plusieurs avantages par rapport aux autres méthodes de détection, elle se caractérise par :

- La rapidité: réduction de l'espace de recherche par l'utilisation d'une technique de segmentation de la couleur de peau humaine.
- La robustesse (en éclairage, position, occlusion, …): grâce à la grande capacité de généralisation des réseaux de neurones.
- La précision: amélioration de la qualité de l'apprentissage par la réduction et la sélection des paramètres les plus pertinents en utilisant une technique de compactage d'énergie.

À partir de l'apprentissage qu'on a réalisé en utilisant l'approche MLP-DCT, Nous avons aussi effectué quelques tests sur des images que nous avons pris par une

caméra personnelle et d'autres télécharger sur le Net. Les tests sont effectués sur des visages qui se présentent dans différentes couleurs et différentes orientations et échelles (fig.7).

Figure 7 Illustration de quelques résultats.

5 Conclusion

Dans cet article, nous avons présenté une méthode de détection de visages dans une image couleur, cette méthode se base sur une technique d'apprentissage automatique en utilisant la décision de plusieurs réseaux de neurones, elle se caractérise aussi par un apprentissage au niveau fréquentiel en utilisant la transformée discrète en cosinus comme une technique de compactage de l'énergie, cette dernière est une transformation orthogonale qui permet un changement du domaine d'étude et qui garde exactement la même fonction étudiée, ce changement du domaine présente un avantage très important, il permet de stocker un maximum d'information dans un minimum de coefficient localisé dans les basses fréquences. Ceci permet d'écarter les coefficients ayant des amplitudes relativement petites sans présenter une déformation des caractéristiques de l'image. Les résultats expérimentaux ont montré que l'introduction de la DCT avec une technique d'apprentissage automatique a donné une amélioration très significative du taux de la détection des visages par rapport aux systèmes qui se basent sur des techniques d'apprentissage seulement.

Une des perspectives de ce travail consiste à appliquer ce système sur d'autres bases de visages présentant de fortes variations dans l'éclairage et dans la pose de la tête ainsi que d'envisager la possibilité de l'employer pour une approche basée sur les éléments caractéristiques du visage.

Références

1. Hsiuao-Ying Chen, Chung-Lin Huang, Chih-Ming Fu: Hybrid-boost learning for multi-pose face detection and facial expression recognition. Pattern Recognition 41 pp. 1173 – 1185 (2008).
2. S. Phimoltares, C. Lursinsap, K. Chamnongthai: Face detection and facial feature localization without considering the appearance of image context. Image and Vision Computing 25 pp. 741–753 (2007).

3. Hyeon Bae, Sungshin Kim: Real time face detection and recognition using hybrid information extracted from face space and facial features. Image and Vision Computing 23 pp. 1181-1191 (2005).
4. Ignas Kukenys, Brendan McCane: Support Vector Machines for Human Face Detection. NZCSRSC 2008, Christchurch New Zealand (April 2008).
5. Z. Li Stan, Anil K. Jain: Handbook of Face Recognition. Springer – First Edition– (2004).
6. SA. Khayem:The Discete Cosine Transform: Theory and Application. Department of Electrical & Computer Engineering Michigan State University (March 2003).
7. Ken Cabeen and Peter Gent: Image Compressing and the Discrete Cosine Transform. Math 45, College of the Redwoods (2009).
8. Dubey R B, Gupta R: High Quality Image Compression. WebmedCentral MISCELLANEOUS (2011).
9. Vijaya Prakash.A.M, K.S.Gurumurthy: A Novel VLSI Architecture for Digital Image Compression Using Discrete Cosine Transform and Quantization. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.9, (September 2010).
10. Simon Haykin: Neural Networks: A Comprehensive Foundation. New York: Macmillan,. Oversized pp 696 including index. Glossy Hardcover (1994).
11. Thanh Phuong Nguyen : Détection de visage dans images de couleur en utilisant une caractéristique invariante. 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT, TUNISIA (2005).
12. Bencheriet Chemesse-Ennehar, Boualleg Abd El halim, Tebbikh Hicham : Segmentation de la Couleur de Peau par Seuillage Selon Différents Espaces de Couleur. 3èmes Journées Internationales sur l'Informatique Graphique (JIG'2007).
13. Thanh Phuong NGUYEN, T. Hoang Lan NGUYEN : Amélioration des performances d'un système de détection de visage. Istitut polytechnique de Hanoi (2007).
14. MIT CBCL Face Data Set, http ://www.ai.mit .edu/projects/ cbcl/ softwar- edatasets/ FaceData2.html.

Fouille de données et optimisation

Modeling and Optimization in Logistic and Transport of the Fuel Distribution

Abdelaziz Benantar and Rachid Ouafi

USTHB, Faculty of Mathematics Operational Research Department, LAID3 Laboratory,
BP 32 16111 El Alia, Bab-Ezzouar, Algiers Algeria
a_benantar@yahoo.fr
rouafi@usthb.dz

Abstract. This paper reports the experience of solving the fuel distribution problem for the biggest fuel company in Algeria. The company serves their customers with fleet of vehicles which have several compartments, each compartment being dedicated to one product. Also, each customer has his own time period and only during this period the service can be accepted. The demand of each customer can be contain one or more products. The objective is to determine a set of routes that respect all capacity and time constraints, serve all customers and minimize the number of vehicles and total distance traveled. According to this real world problem, a mathematical model is established for the Multi-compartment vehicle routing problem with time windows (MCVRPTW). A Tabu search method is developed to solve this problem. The proposed method has been tested on instances obtained by adding compartments to Solomon's VRPTW instances and real data provided by the company.

Keywords: Combinatorial Optimization, Transportation, Logistic, Tabu Search, VRPTW, Fuel Delivery.

1 Introduction

The fuel distribution VRP problem is a complex combination optimization problem. It consists to determine with which transportation asset and at what time each of the requirements will be satisfied. A typical requirement is composed by fuel type (Gasoline, petroleum and Kerosene), geographical position of depots and customers and required delivery expressed by customers. All of the above determines the requirements that the dispatching system should output a set of instructions telling drivers what to deliver, when and where. An efficient solution is one that enables products to be delivered when and where required at least cost.

In research literature, only very little attention, has been paid to the fuel distribution problem that allow transportation of several types of fuel on the

same vehicle, but in different compartments. The only published papers concern this problem, e.g., Avella et al.[1], Brown and Graves[2], Brown et al. [3] and Van der Bruggen et al. [4], El Fallahi et al. [5], Cornillier et al.[6]. Our research has been motivated by a demand for methods and software handling compartments and time windows from the distribution of fuel to service stations where the vehicles are divided into a number of compartments in order to transport the different fuel types.

The problem addressed in this paper reports the experience of solving the fuel distribution problem for the biggest fuel company in Algeria, which has sixty (60) depots located in Ten (10) Districts, all over the country, that deliver several fuel type to more than two thousand (2000) customers (gas stations) using around 800 vehicles, not necessary belong to the company, that deliver more than ten (10) million cubic meters at year. The work that this distribution carries out can be shown as follows: each depot receives the fuel from refinery; store it to be delivered by a fleet of vehicles to customers, which locate in the same district as the depot. Before every delivery, each customer must call the scheduler at the plant where it belongs, to order the fuel for the next day, with the corresponding temporal restrictions. For the customer, this demands calendar becomes a contractual agreement that fixes the quantities ordered of each product, dates and times during which it must be served. The depot uses these agreements to develop the supply plan, which is not imposed in the proposed version. Moreover, it's from this latter plan and considering these demands calendars of customers that the distribution plan can be established. In the following, we qualify our problem as a variant of vehicle routing problem called the Multi-compartment Vehicle Routing Problem with Time Windows (MCVRPTW).

The rest of this paper is organized as follows. In section 2, we describe the mathematical model of the fuel distribution problem. Section 3, present the Tabu Search Algorithm and its components. Then, we report computational results for the real data and Solomon's VRPTW instances test sets in section 4. Finally, we present our conclusions and discus some further research in 5.

2 Mathematical Model

In order to obtain the most cost/time efficient manner of transporting fuel to customers, we will need to solve the Multi-compartment Vehicle Routing Problem with Time Windows (MCVRPTW). The model satisfies: (1) Single depot: every vehicle starts and returns to this depot. (2) Vehicles have several compartments: each compartment is dedicated to one product. (3) All products delivered on a route must be assigned to compartments of the vehicle, (4) Consider the vehicle and compartment capacities constraints, (5) Consider the possibility to bring the different products ordered by a customer using several vehicles, (6) when a vehicle arrives at a customer location before its earliest service time, the vehicle must wait until the service is possible. If the vehicle arrives late, a penalty for lateness is incurred. The problem considers as the primary objective

the minimization of the number of vehicles and secondarily the minimization of total distance traveled of all routes.

To simplify the problem, we define the sets, parameters and variables used in the problem formulation as follows:
N: Number of customers.
$V = N \cup \{0\}$ Where 0 represents the depot.
K: Number of vehicles that can be used.
P: Number of products.
Q: Number of compartments.
c_{qk}: Capacity of compartment q for the vehicle k.
C_k: Capacity of vehicle k.
d_{ip} : Demand of customer i for the product p.
s_i: Service time at customer i.
t_{ijk}: Travel time of vehicle k from customer i to customer j.
a_i : Earliest start time for servicing the customer i.
b_i : Latest start time for servicing the customer i.
d_{ij} : Distance between the customer i and customer j.
C_{ijk} : Cost spent by the vehicle k from customer i to customer j.
x_{ijk} : Equals 1, if i precede j in the route of vehicle k, 0 otherwise.
y_{ikp} : The 0-1 variables take value 1 if and only if customer i receives product p from vehicle k.
z_{pqk} : The binary variables indicate whether product p is assigned to compartment q on vehicle k.
s_i : Specifies the arrival time at i when serviced by vehicle k. In case of vehicle k doesn't service customer i. s_i has no meaning and consequently its value is considered irrelevant.

With this notation, constraints and the objective function of the mathematical model can then be formulated as follows:

2.1 Constraints and Interpretations

2.1.1. The Delivery Constraints

All customers must be served at most once by each route, i.e:

$$\sum_{i \in V} x_{ijk} \leq 1, \forall j \in N, \forall k \in K \tag{1}$$

The continuity of each route is ensured as follows:

$$\sum_{i \in V} x_{ijk} = \sum_{i \in V} x_{jik}, \forall j \in N, \forall k \in K \tag{2}$$

2.1.2. The Vehicle and Compartment Constraints

A vehicle can only be loaded up to its capacity, i.e:

$$\sum_{p \in P} \sum_{i \in N} d_{ip} \sum_{j \in V} x_{ijk} \leq C, \forall k \in K \tag{3}$$

The products loaded into compartments on a given vehicle do not exceed the compartments capacities:

$$\sum_{i \in N} d_{ip} y_{ikp} \leq \sum_{q \in Q} c_q z_{pqk}, \forall k \in K, \forall p \in P \tag{4}$$

Each compartment is dedicated at most to one product by each route:

$$\sum_{p \in P} z_{pqk} \leq 1, \forall k \in K, \forall q \in Q \tag{5}$$

Each product ordered by customer is brought by one single vehicle:

$$\sum_{k \in K} y_{ikp} = 1, \forall i \in N, \forall p \in P \tag{6}$$

If the customer i is not visited by vehicle k, we set the variable to zero for each product p, as follows:

$$y_{ikp} \leq \sum_{j \in V} x_{ijk}, \forall i \in N, \forall k \in K, \forall p \in P \tag{7}$$

2.1.3. The Scheduling Routes Constraints

The relationship between the vehicle departure time from a customer and its immediate successor is established as follows:

$$x_{ijk}(s_{ik} + t_i + t_{ij} - s_{jk}) \leq 0, \forall i \in V, \forall j \in V, \forall k \in K \tag{8}$$

This constraint can be linearized as:

$$s_{ik} + t_i + t_{ij} - M_{ij}(1 - x_{ijk}) \leq s_{jk}, \forall i \in V, \forall j \in V, \forall k \in K \tag{9}$$

The large constants M_{ij} ($M_{ij} > 0$) can be replaced by $max\{b_i + t_i + t_{ij} - a_j, 0\}, i, j \in N$.When $max\{b_i + t_i + t_{ij} - a_j, 0\} = 0$, these constraints are satisfied for all values of s_{ik}, s_{jk} and x_{ijk}.

2.1.4. The Time Window Constraints

The time window constraints are formulated as follows:

$$a_i(\sum_{j \in V} x_{ijk}) \leq s_{ik} \leq b_i(\sum_{j \in V} x_{ijk}), \forall i \in N, \forall k \in K \tag{10}$$

$$a_0 \leq s_{0k} \leq b_0, \forall k \in K \tag{11}$$

2.1.5. The Integrity and non Negativity Constraints

Constraints (12) - (15), define the decision variables, which are all binary except for the variables s_{ik}.

$$x_{ijk} \in \{0, 1\}, \forall i \in V, \forall j \in V, \forall k \in K \tag{12}$$

$$y_{ikp} \in \{0, 1\}, \forall i \in N, \forall k \in K, \forall p \in P, d_{ip} \neq 0 \tag{13}$$

$$z_{pqk} \in \{0, 1\}, \forall p \in P, \forall q \in Q, \forall k \in K \tag{14}$$

$$s_{ik} \geq 0, \forall i \in N, \forall k \in K \tag{15}$$

2.2 Objective Function

The proposed model minimizes the sum of the travel costs over all routes, as follows:

$$min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \tag{16}$$

Subject to: (1)→(15)

3 Design of Algorithm

Due to the complexity of solving the MCVRPTW, the Tabu search algorithm is proposed to tackle the fuel distribution problem in practice.

3.1 The Construction Routes Heuristic

The proposed Tabu search starts from one initial feasible solution and improve it. This solution is built by using the construction routes heuristic inspired by the nearest neighborhood principle. The principle behind this heuristic is that, every route is started by finding the un-routed customer that is closest to the depot. The closeness relation tries to take both temporal and geographical closeness of the customers into account. At every subsequent iteration the customer closest to the last customer added to the route is considered for insertion to the end of the route presently generated. Before adding a customer to the route, the heuristic checks the constraints of vehicle capacity and the requirement of time window. When the search fails a new route is started. Note that the insertion costs are evaluated in terms of distance and service time.

3.2 The Neighborhood Structure

After obtaining an initial feasible solution, the neighborhood search procedure is proposed to improve it by exploring the solutions space. At each iteration, the best non-tabu move in the neighborhood $N(S)$ is selected to move from the incumbent solution S to the best solution S^*. Neighborhood search improvement procedure applies $2-Opt^*$ and $Or-Opt$ methods with slight modifications. In Section 3.2.1, we describe the neighbor lists that take into account the time windows, and in Section 3.2.2, the details of the neighborhoods are described.

3.2.1. Neighbor List. We consider a neighbor list for each customer i, which is a set of customers preferable to visit immediately after i. The algorithm computes these values once at the beginning and stores the best $N(i)$ customers as a neighbor list of i.

3.2.2. The Neighborhood. We use the $2-Opt^{**}$ and $Or-Opt$ neighborhoods with slight modifications, wherein we restrict the $2-Opt^{*}$ neighborhood by using the neighbor lists. The $2-Opt^{*}$ neighborhood was proposed in [7]. A $2-Opt^{*}$ operation removes two edges from two different routes (one from each) to divide each route into two parts and exchanges the second parts of the two routes. The $2-Opt^{*}$ operation always changes the assignment of customers to vehicles. We also use an intra-route neighborhood to improve individual routes, which is a variant of the $Or-Opt$ neighborhood. An intra-route operation removes a continuous segment of customers of length $L_{segment}$ (a parameter) and inserts it into another position of the same route, where the position is limited within $L_{insertion}$ (a parameter) from the original position. Sometimes, solutions generation violate the capacity and time constraints. In the aim of to enlarge the search space by visiting these infeasible solutions, the capacity and time violations are multiplied by two coefficients and the penalized term added to the objective function. The resulting new evaluation function $F'(S)$ is inspired by the one proposed by Gendreau et al. for the VRP [8]:

$$F'(S) = F(S) + \alpha C(S) + \beta T(S). \tag{17}$$

Where $F(S)$ is the routing cost of solution S, $C(S)$ and $T(S)$ are the total over-capacity and overtime of all routes respectively and α, β are two penalty factors. Initially set equal to 1, these parameters are periodically divided by $1+\rho$ ($\rho \in]0,1[$) if all previous ϕ solutions were feasible, or multiplied by $1+\rho$ ($\rho \in]0,1[$) if they were all infeasible. This way of proceeding produces a mix of feasible and infeasible solutions which acts as a diversification strategy.

3.3 The Tabu List

Tabu list is one of the key factors that determine the quality of a Tabu search algorithm. The most popular Tabu list is constructed by those recently visited solutions, or the moves to a solution. If the size of the Tabu list is too large, it will spend more time to compare with the current solution one by one, but if the size of the Tabu list is too small, it will be very hard to escape from local optima.

In this paper, the Tabu list is implemented as an upper triangular matrix L of $K \times K$ dimensions where each element $L(k,k^{'}), (k,k^{'} = 1,...,K, k < k^{'})$ is associated to pair of routes k and $k^{'}$. Each element $L(k,k^{'})$ contains a set of attributes able to characterize the solution and also records the iteration in which the arc (i,j) has been removed from the route k to the route $k^{'}$. An arc removed at iteration t is forbidden to be reinserted in the solution until iteration $t+\theta$. When an exchange between the routes k and $k^{'}$ is accepted, we just change information corresponding to the line k and the column $k^{'}$. Thus, we avoid calculating information above the other pairs of routes which not contain neither k nor $k^{'}$.

The size of Tabu list θ takes its values in $[\theta_{min}, \theta_{max}]$ starting from θ_{init}. The parameter θ is updated according to the quality of the solutions obtained during

the recent moves. After each improvement of the current objective function, θ is updated as $\theta = max(\theta - 1, \theta_{min})$, with the aim of intensifying the search around this solution. Otherwise, after ϕ_{LT} consecutive moves deteriorating the value of the visited solutions, the size of the tabu list is updated as $\theta = min(\theta + 1, \theta_{max})$.

3.4 Diversification and Intensification

To make the search strategy fully effective in our Tabu search method, we have included the diversification and intensification features. The purpose of diversification is to widen the set of solutions considered during the search. Intensification consists of accentuating the search in promising regions of the solution space. Three mechanisms are used for diversifying or intensifying the search of a Tabu search method. The first, described in Section 3.2.2, consists of biasing the evaluation of infeasible moves by adding to the objective function a penalized term. The second mechanism consists of conducting a limited search on a small number of starting solutions. The full search then proceeds starting from the most promising solution. So, after γ_{max} iterations without improving the best solution found or after ϑ_{max} iterations since the last restart, the search is started with an empty tabu list. Note that the maximal number of restarts is fixed as η_{max}. Finally, the search is accentuated around the best known solution by increasing or decreasing the size of the tabu list as explained in Section 3.3.

4 Computational Results

The proposed Tabu search approach is coded in C++ and has been tested on 2.0 GHz PCs with 2 GB memory running under windows XP. To assess the effectiveness of our Tabu search algorithm we considered two alternatives solutions approaches for the proposed testing:

Firstly, we have tested the behavior of our algorithm for solving standard Solomon's VRPTW instances, i.e. we have interpreted VRPTW data sets as MC-VRPTW instances with one product type and one compartment which is a valid input for our algorithm. Hence, we considered 30 Solomon's VRPTW instances with different characteristics. The instances are divided into 06 groups (test-sets) denoted R1, R2, C1, C2, RC1 and RC2. Each of the test sets contains 05 instances. Each instance has 100 customers. All these instances may be downloaded from [16]. Table 1 compares the results obtained by the proposed Tabu search algorithm with the best known Solomon's VRPTW solutions. Considering as the primary objective the minimization of the number of vehicles and secondarily the minimization of the total traveled distance, Table 1 shows that we are able to produce solutions with a deviation of distance and number of vehicles from the best-known Solomon's VRPTW solutions around -2.97(%) and -3.06 (%) respectively. This is quite remarkable because our algorithm is not specially designed for the VRPTW and the best known VRPTW solutions reported in literature were obtained by using various metaheuristics and settings of parameters.

Table 1. Results compared to best-known Solomon's VRPTW solutions

Problem	Optimal solution		Our solution	
	NV	Total Distance	NV	Total Distance
R101	19	1645.79	19	1672.91
R103	13	1292.68	13	1317.35
R105	14	1377.11	15	1388.95
R107	10	1104.66	10	1213.33
R109	11	1194.73	11	1240.96
C101	10	828.94	10	839.87
C102	10	828.94	10	840.01
C103	10	828.06	10	838.07
C104	10	824.78	10	831.29
C105	10	828.94	10	856.45
RC101	14	1696.94	14	**1692.15**
RC102	12	1554.75	13	1627.46
RC103	11	1261.67	11	1324.93
RC104	10	1135.48	**9**	1260.97
RC105	13	1629.44	13	1724.93
R202	3	1191.70	3	1198.12
R204	2	825.52	2	**823.41**
R206	3	906.14	3	930.13
R208	2	726.75	2	754.33
R210	3	939.34	3	940.19
C201	3	591.56	4	791.61
C202	3	591.56	4	593.38
C203	3	591.17	4	610.36
C204	3	590.60	4	613.54
C205	3	588.88	4	592.15
RC201	4	1406.91	4	1408.78
RC202	3	1367.09	3	1370.53
RC203	3	1049.62	3	1051.28
RC204	3	798.41	3	800.03
RC205	4	1297.19	5	1310.33
Deviation from optimality			**-3.06**	**-2.97**

Secondly, we considered the actual data sets provided by the fuel logistic company. The tests were based on the orders delivered by the customers of Algiers's depot on thirty consecutive working days. The distance data needed in this paper are calculated by Djikstra Algorithm. The time window of each

customer is at in a certain interval in [7, 12], [12, 17], [17, 23] in the day. It's assumed that the service time at each customer is proportional to the amount of his demand. The average speed of vehicles is 60km/h.

The plans produced by the human operator in the company were also obtained for each of these thirty days. These plans (referred to as manual plans) contained the orders, the routing sequences and the vehicle identities. The manual plans were not as detailed as the algorithm's plans, in that the arrival, departure and waiting times at each call were in the algorithmic schedules but not in the manual ones. The solutions given by our algorithm depend on a few parameters used by our approach. These parameters values are chosen after some tests. The algorithm automatically hired vehicles if the company had insufficient but none were needed during the thirty days, either in the manual or in the algorithmic solutions.

Table 2 resumes the comparison between the Tabu search algorithm and manual solutions over a testing period of thirty days. It can seen from the table 2 that the method solutions are better than the manual solutions. The method allows a reduction of 17.68 (%) in cost of the traveled distance. We expect to obtain even better results in high demand periods, when the human operator is under pressure because of urgent deliveries and a much greater number of orders have to be delivered.

Table 2. Algorithmic and manual solutions

Day	customers	Vehicles used		Distance (km)		Cost (u)	
		I	II	I	II	I	II
1	75	12	09	1876	1655	178204	157252
2	68	8	07	1489	1398	141494	132766
3	88	14	11	1510	1416	143469	134499
4	91	14	12	1488	1361	141316	129305
5	56	08	06	999	843	94863	80091
6	61	13	09	1123	1068	106690	101415
7	19	05	03	530	498	50390	47292
8	71	13	09	1240	1107	117832	105137
9	64	13	10	1102	1001	104672	95125
10	59	10	08	1047	1002	99459	95163
11	95	18	15	1829	1724	173767	163782
12	78	17	13	1687	1541	160281	146387
13	73	14	12	1364	1227	129546	116547
14	23	06	04	459	305	43642	28964
15	58	09	08	994	805	94451	76468
16	62	07	05	1250	1032	118734	98007
17	51	08	06	1066	941	101228	89354
18	49	05	05	1113	1092	105772	103777
19	69	10	08	845	695	80277	66035
20	81	11	08	1131	961	107463	91326
21	15	04	03	476	339	45202	32158
22	104	17	13	1818	1469	172733	139522
23	98	13	10	1772	1443	168353	137124
24	84	10	07	1371	1005	130268	95454
25	92	14	12	1000	807	95012	76710
26	93	14	11	2042	1532	194004	145519
27	81	08	06	1932	1646	183584	156390
28	20	06	03	721	513	68488	48713
29	130	21	17	2407	1900	228653	180532
30	123	18	14	2455	1955	233241	185693
Average	**71**	**11**	**09**	**1338**	**1143**	**127103**	**108550**
Reduction						**17.09** (%)	

I: Human operator, II: Approach Solution

5 Conclusion and Further Research

In this paper, we propose an efficient algorithm for the fuel distribution problem. Under this scenario, a variant of the vehicle routing problem with time windows constrained by the composition vehicles fleet (vehicles have several compartments) is studied. The proposed methodology is based on a tabu search algorithm. Two sets of problems are used to test the algorithm. One obtained by adapting thirty known Solomon's VRPTW instances and the other by using real data provided by the fuel logistic company.

As no published method is available for comparison, the performance of our method is evaluated in two perspectives. Firstly, we showed that the method could be used to give good solutions for the standard VRPTW problem. Secondly, it is evaluated through real instances and compared with the plan made by the human operator. The results showed that it can be considered as an advanced logistic system for the fuel distribution. It is compact, fast and easy to use.

The possible extensions of this paper would be to (1) study other heuristics algorithms for comparing the solutions quality for this problem and their computational times (2) propose the parallel methods in the aim to centralize the fuel distribution management on all Districts.

References

1. Avella, P., Boccia, M., Sforza, A.: Solving a Fuel Delivery Problem by Heuristic and Exact Approaches. European Journal of Operational Research 2004; 152:170-9
2. Brown, GG., Graves, WG.: Real-time Dispatch of Petroleum Tank Trucks. Management Science 1981;27:19-31
3. Brown, GG., Ellis, CJ., Graves, WG., Ronen, D.: Real-time, Wide Area Dispatch of Mobil Tank Trucks. Interfaces 1987;17:107-20
4. Van der Brugen, L., Gruson, R., Salomon, M.: Reconsidering the Distribution of Gasoline Products for a Large Oil Company. European Journal of Operation Research 1995;81:460-73
5. El Fallahi, A., Prins, C., Wolfler Calvo, R.: A Memetic Algorithm and a Tabu Search for the Multicompartment Vehicle Routing Problem. Computers and Operations Research 2008; 35(5):1725-1741
6. Cornillier, F., Boctor, F.F., Renaud, J.: Heuristics for the Multi-Depot Petrol Station Replenishment Problem with Time Windows. Computers and Operations Research 2012; 220:361-369
7. Potvin, J.Y., Kervahut, T., Garcia, B.L., Rousseau, J.M.: The Vehicle Routing Problem with Time Windows part I: Tabu Search. INFORMS Journal on Computing 8(2),158-164 (1996)
8. Gendreau, M., Hertz, A, Laporte, G.: A Tabu Search Heuristic for the Vehicle Routing Problem. Management Science 1994;40:1276-90
9. Braysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows Part I: Route Construction and Local Search Algorithms. Transportation Science. 39 (2005) 104-118
10. Braysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows Part II: Metaheuristics. Transportation Science, 39 (2005) 119-139

11. Chajakis, E., Guignard, M.: Scheduling Deliveries in Vehicles with Multiple Compartments. Journal of Global Optimization 2003; 26(1):43-78
12. Cornillier, F., Boctor, F., Laporte, G., Renaud, J.: A Heuristic for the Multi-Period Petrol Station Replenishment Problem. European Journal of Operational Research 2008; 191(2):295-305
13. Derigs et al.: Vehicle Routing with Compartments: Applications, Modeling and Heuristics. Springer-Verlag 2010, available online 11 February 2010
14. Gendreau, M., Potvin, J-Y., Braysy, O., Hasle, G., Lkketangen, A.: Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. Springer Science + Business Media, LLC 2008
15. Nasser, A. El-Sherbeny.: Vehicle Routing with Time Windows: An Overview of Exact, Heuristic and Metaheuristic Methods [J]. Journal of King Saud University (Science) Available online 7April 2010
16. http://www.sintef.no/Projectweb/TOP/Problems/VRPTW/Solomon-benchmark/, 2011

Problème de Détermination du Gagnant dynamique : modèle mathématique et approche de résolution

Larbi Asli et Méziane Aïder

LAID3, Faculty of Mathematics, USTHB,
BP 32 El Alia, 16111 Algiers, Algeria
{aslilarbi@yahoo.fr,m-aider@usthb.dz}
http://www.usthb.dz/perso/math/maider

Résumé Dans cet article, nous proposons un modèle mathématique pour le problème de détermination du gagnant dynamique (PDG), pour une enchère combinatoire en ligne. Ce modèle permet d'implémenter, sur le web, une enchère combinatoire suivant le mécanisme anglais. Nous proposons un algorithme de résolution qui donne à chaque instant la liste des gagnants temporaires, les articles vendus ainsi que le gain temporaire. Une expérimentation numérique utilisant cet algorithme sur des données simulées donne lieu à des résultats satisfaisants.

Mots Clés. Enchère combinatoire anglaise, liste des gagnants temporaires, gagnant final, période d'exercice, offre temporaire.

1 Introduction

Le concept d'enchère est devenu à la portée de la plupart des citoyens. Toutefois, en pensant à une enchère, nous imaginons immédiatement une grande salle pleine de spécialistes et d'intéressés dans une maison de ventes aux enchères telles que Sotheby ou Christie. Le courtier présente l'article à vendre, généralement une œuvre d'art, et annonce un prix initial. Les enchérisseurs indiquent leur volonté d'acheter l'article pendant que le prix augmente. Aujourd'hui l'enchère ne se rapporte plus à un temps ou à un espace. L'arrivée d'internet a permis le transfert de la maison des ventes aux enchères en ligne. Les enchères en ligne sont devenues de plus en plus populaires, et actuellement, ebay est au moins aussi bien connu que Sotheby ou Christie. La variété de produits ou services mis en vente pour l'enchère est incroyable : on peut acheter presque n'importe quoi, des œufs de dinosaure à la propriété d'immobiliers, et des diamants aux vêtements usagés [14].

Les enchères constituent d'importants mécanismes permettant d'évaluer certains produits ou services pour lesquels il est a priori impossible d'attribuer une estimation (du prix). De nombreux mécanismes ont été développés pour décrire le processus d'attribution des prix. Nous pouvons citer les quatre mécanismes

de base : l'enchère anglaise (le prix du produit débute bas et augmente jusqu'à ce qu'il n'y ait plus d'offres émises), l'enchère hollandaise (le prix du produit débute haut et décroit jusqu'à ce qu'un acheteur se propose d'acquérir le produit), l'enchère du meilleur prix (les mises sont secrètes et le gagnant est celui qui propose la plus haute mise), l'enchère Vickery (les mises sont secrètes et le gagnant est celui qui propose la plus haute mise, mais il payera la deuxième plus haute mise). Ces différents mécanismes appartiennent à deux types d'enchères : orales et écrites. Ils sont destinés à vendre ou acheter soit un seul article ou service à la fois, ou bien plusieurs en même temps. Ce dernier s'intitule enchère de plusieurs articles ou bien enchère combinatoire. Ce type d'enchères est très populaire car il permet de vendre ou d'acheter une combinaison d'articles par une seule soumission. Dans cet article, nous nous intéressons à ce dernier type d'enchères.

2 Les caractères distinctifs des mécanismes d'enchère

Une enchère se distingue par de nombreuses caractéristiques, parmi lesquelles nous pouvons citer :

- la procédure de communication : elle permet de distinguer entre les enchères orales et les enchères écrites ;
- le mécanisme d'enchère : permet aux agents qui veulent acquérir le bien de savoir de quelle manière construire leurs offres ;
- le mécanisme d'attribution : détermine la manière dont les vainqueurs sont désignés ;
- la règle de paiement : permet d'établir le prix dont devra s'acquitter le gagnant (premier prix ou deuxième prix ?) ;
- le caractère commun ou non de la valeur du bien ;
- les stratégies des agents.

3 Enchères combinatoires

Les enchères combinatoires se définissent comme étant un ensemble d'objets soumis à la vente face à plusieurs acheteurs. Chaque acheteur, pour des raisons de complémentarité entre les objets, désire acheter un sous-ensemble d'objets qui lui est propre, et pour lequel il fournit une estimation. Les conflits entre les acheteurs naissent des éventuelles intersections entre des sous-ensembles d'objets convoités. Le vendeur doit alors résoudre un problème d'optimisation combinatoire NP-dur pour réaliser la vente qui lui rapportera le plus [2].

Les enchères combinatoires ont plusieurs applications, notamment en économie, en théorie des jeux et pour l'allocation des ressources dans les systèmes multi-agents [13], [11], [10]. Elles permettent une allocation meilleure des objets selon les besoins spécifiques des enchérisseurs. La difficulté essentielle dans la modéli-

sation et la formulation de ce type de problème d'enchères réside dans le nombre de combinaisons d'objets possibles [2].

4 Travaux antérieurs

De nombreuses méthodes ont été proposées pour résoudre le problème de la détermination du gagnant [10]. Ces méthodes peuvent être réparties en deux catégories :

- les méthodes exactes, qui sont généralement basées sur l'algorithme par séparation et évaluation "Branch-and-Bound" ;
- les méthodes non exactes, souvent dites, du rest à tort, méthodes approchées, qui sont basées sur les heuristiques ou les métaheuristiques.

Les algorithmes qui ont été développés pour la résolution du PDG consistent essentiellement en l'adaptation de schémas généraux de résolution de problèmes d'optimisation. Nous relevons :

- Adaptation de la séparation et évaluation :
  - Branch-on-Items (BoI) [11] : la séparation se fait en introduisant un nouvel article ;
  - Branch-on-Bids (BoB [11] et CABOB [12]) : la séparation s'effectue en ajoutant une nouvelle offre ;
  - Combinatorial Auction Structural Search (CASS) [4] ;
  - Combinatorial Auctions Multi-Unit Search (CAMUS) [7] ;
- Autres méthodes exactes :
  - Programmation dynamique [9] ;
  - Programmation linéaire continue et en nombres entiers [1] ;
  - Programmation par Contraintes [6] ;
- Méthodes approchées :
  - Recherche locale [2] ;
  - Méthodes mémétiques [3] ;
  - Méthodes hybrides (généralement, hybridation d'une métaheuristique avec le "branch and bound") [5].

5 Concepts et définitions

Le problème de détermination du gagnant dans les enchères combinatoires peut s'énoncer de la manière suivante :

Un ensemble M d'articles notés $i, i = 1, \ldots, m$ sont mis à la vente aux enchères et un ensemble de n enchérisseurs notés $E_j, j = 1, \ldots, n$ soumettent, chacun, une offre. L'offre S_j soumise par l'enchérisseur E_j consiste en un $(m+1)$-vecteur $(A_j, c_j) = (a_{1j}, a_{2j}, \ldots, a_{mj}; c_j)$ comprenant, pour chaque article i, la quantité a_{ij} demandée ainsi que le montant c_j proposé pour cette offre.

5.1 Définitions

Définition 1 La période d'exercice, notée T, représente la période de temps définie par l'administrateur de l'enchère (administrateur de site internet) et durant laquelle les enchérisseurs peuvent lancer leurs offres. En général, cette période est supérieure à 24 heures. De plus, cette période est supposée être discrétisée, et ne sont pris en compte que l'instant initial $t_0 = 0$ et les instants $t_l, l = 1, 2, \ldots$ où une offre est soumise.

Définition 2 L'offre temporaire (à l'instant t_l), notée S_j, représente l'offre lancée par l'enchérisseur j à un instant t_l. Cette offre n'est pas définitive et peut être mise à jour durant la période d'exercice.

Définition 3 La liste des gagnants temporaires (à l'instant t_l), notés $LTG(l)$, représente la liste des enchérisseurs gagnants à l'instant t_l. Cette liste n'est pas définitive et peut évoluer durant la période d'exercice, en fonction des futures soumissions.

Définition 4 Deux offres sont en conflit si elles ne peuvent être retenues simultanément, autrement dit pour au moins un article, la somme des nombre d'exemplaires contenus dans ces deux offres est supérieure au nombre d'exemplaires disponibles. Le graphe de conflit [2] est le graphe dont l'ensemble des sommets représente les offres et deux sommets sont reliés si et seulement si les offres qu'ils représentent sont en conflit.

5.2 Concepts de base

Outre les notions déjà introduites, nous utilisons également les concepts et notations ci-après :

T **:** période d'exercice ;

t_l **:** instant où un enchérisseur E_j lance une offre ;

S_j **:** $(n+1)$-vecteur (A_j, c_j), représentant l'offre temporaire de l'enchérisseur E_j,

- A_j **:** n-vecteur des nombres d'exemplaires des articles i contenus dans S_j ;
- c_j **:** montant de l'offre S_j ;

β_i **:** nombre d'exemplaires de l'article i mis en enchère ;

LTG_l **:** liste temporaire des gagnants à l'instant t_l ;

LFG **:** liste finale des gagnants ;

$NEVT$ **:** nouvel événement ;

LEC_j **:** liste des enchérisseurs en conflit avec E_j ;

x_j **:** variable binaire valant 1 si l'offre j est acceptée 0 autrement ;

Z_k **:** gain temporaire à l'instant t_k.

6 Modèle mathématique

Le modèle mathématique associé au problème de détermination du gagnant se compose des éléments suivants :

Variables de décision

A chaque enchérisseur $E_j, j = 1, \ldots, n$ est associée une variable x_j telle que :

$$x_j = \begin{cases} 1, & \text{si l'offre de } E_j \text{ est acceptée}; \\ 0, & \text{autrement.} \end{cases}$$

Contraintes

Le problème est soumis à la contrainte de disponibilité des articles et donc il ne peut être vendu plus que le nombre d'exemplaires disponibles de chaque article $i, i = 1, \ldots, m$. Ceci se traduit par la relation suivante qui doit être satisfaite par les variables de décision :

$$\sum_{j=1}^{n} a_{ij} x_j \leq \beta_i \quad i = 1 \ldots m$$

Fonction objectif

Il s'agit de maximiser le gain total des ventes, soit :

$$\max Z = \sum_{j=1}^{k} c_j x_j.$$

Au départ du processus, à l'instant $t_0 = 0$ du début du temps d'exercice, le problème ne contient que le nombre m d'articles mis en enchères. Le modèle mathématique se construit et se complète au fur et à mesure. Ainsi, à un instant t_l, lorsqu'une k-ème nouvelle offre est soumise, nous aurons un problème PDG_k à k variables :

$$(PDG_k) \begin{cases} \max Z = \displaystyle\sum_{j=1}^{k} c_j x_j \\ \displaystyle\sum_{j=1}^{k} a_{ij} x_j \leq \beta_i \qquad i = 1, \ldots, m \\ x_j \in \{0, 1\} \; j = 1, \ldots, k \end{cases}$$

A l'achèvement de la période d'exercice, le modèle sera final et définitif et s'écrira comme suit :

$$(PDG) \begin{cases} \max Z = \displaystyle\sum_{j=1}^{n} c_j x_j \\ \displaystyle\sum_{j=1}^{n} a_{ij} x_j \leq \beta_i \qquad i = 1, \ldots, m \\ x_j \in \{0, 1\} \; j = 1, \ldots, n \end{cases}$$

7 Modélisation et Algorithme de résolution

En raison de leur aspect statique, la plupart des modèles mathématiques d'enchères combinatoires associés au problème de détermination du gagnant ne permettent pas à l'enchérisseur de renouveler son offre. Cependant, dans la réalité, les enchérisseurs sont en rude concurrence tant qu'il reste du temps pour émettre des soumissions. Nous proposons une formulation sous la forme d'un modèle dynamique pour une enchère combinatoire qui permet aux enchérisseurs de renouveler leurs soumissions jusqu'à la fin du temps d'exercice en se basant sur le mécanisme de l'enchère anglaise.

L'idée de base consiste à implémenter l'énchère combinatoire sur le web. Avant d'entamer la procédure, le propriètaire d'un bien doit définir pour chacun des m types d'articles ou services mis en enchère, le nombre β_i pour $i = 1, \ldots, m$ d'unités de l'article i disponibles (destinés à la vente), ainsi que la période d'exercice T durant laquelle les enchérisseurs peuvent lancer leurs offres.

Dés l'ouverture de la période d'exercice, les enchérisseurs lancent leurs soumissions. Le premier offrant E_1 lance sa soumission S_1 à l'instant t_1. Elle sera représentée par un vecteur de dimension m où chaque composante contient le nombre d'unités demandé de l'article ou service i, ainsi que le montant c_1 attribué à sa soumission. E_1 sera considéré comme gagnant temporaire. Puis à l'instant t_2, un deuxième soumissionaire E_2 lance sa soumission S_2. Si S_2 n'est pas en conflit avec S_1, E_2 sera lui aussi un gagnant temporaire, autrement on vérifie si $c_2 > c_1$, alors E_2 remplacera E_1 dans la liste des gagnants temporaires, et ainsi de suite.

À l'instant t_k un enchérisseur E_j lance soit une mise à jour d'une offre qu'il a prélablement faite, soit une nouvelle offre :

1. **L'offre de E_j est une mise à jour d'une offre qu'il a déjà faite :**
   On teste si l'enchérisseur E_j est dans la liste des gagnants temporaires ($E_j \in LTG$?), alors si sa nouvelle offre n'est pas en conflit avec les offres des gagnants temporaires, alors E_j reste gagnant. Sinon, on résout le problème temporaire PDG_k en tenant compte de la nouvelle offre de E_j (et en ignorant son ancienne offre).
   Dans le cas où l'enchérisseur E_j n'est pas dans la liste des gagnants temporaires ($E_j \notin LTG$), deux cas sont envisageables. Si l'offre de E_j n'est pas en conflit avec les offres des gagnants temporaires, alors il sera lui aussi gagnant temporaire. Si l'offre de E_j est en conflit avec les offres des gagnants temporaires, alors on considère E_j comme une nouvelle offre (et on supprime son ancienne offre).
2. **L'offre de E_j est une nouvelle offre :**
   Si l'offre de E_j est en conflit avec un seul gagnant et que son offre est meilleure (de valeur supérieure), alors il le remplace dans LTG. S'il est en conflit avec plus d'un gagnant, alors deux cas peuvent être envisagés :
   - si le montant de son offre est supérieure à la somme des montants des offres des gagnants temporaires avec lesquels il est en conflit, alors on met

à jour la liste des gagnants temporaires LTG en enlevant toutes ces offres de LTG et en y mettant E_j.

- si le montant de son offre est inférieure à la somme des montants des offres des gagnants temporaires avec lesquels il est en conflit, alors on résout le problème temporaire PDG_k.

Au fur et à mesure que le temps passe, la liste des gagnants temporaires change jusqu'à l'écoulement du temps d'exercice ($t > T$). La séance se clôturera et la liste des gagnants temporaires deviendra finale.

Le principe de l'algorithme, que nous présentons ci-après sur un exemple numérique, est décrit dans Algorithme 1.

Présentation de l'algorithme sur un exemple pratique

Un fournisseur d'outils informatiques veut vider son stock. Il propose aux enchères l'ensemble des articles qu'il possède qui sont repartis comme suit :

Désignation	**Quantité**
Imprimante	130
Ècran TFT	350
Unité centrale	348
Imprimante multifonction	92

Table 1. Données de l'exemple

Déroulement de l'algorithme

- **À l'instant t_1, l'enchérisseur E_1 lance une offre $S_1 = (A_1; c_1)$ avec $A_1 = (20, 150, 75, 0), c_1 = 182500$. E_1 devient gagnant temporaire $LTG = \{E_1\}$.**
- **À l'instant t_2 un deuxième enchérisseur E_2 lance son offre $S_2 = (A_2, c_2)$, avec $A_2 = (130, 0, 0, 92)$ et $c_2 = 108840$ se revendique. E_2 est en conflit avec E_1, et de plus $c_1 > c_2$. Donc il n'y a pas de changement sur $LTG = \{E_1\}$.**
- **Puis l'enchérisseur E_3 lance son offre $A_3 = (70, 100, 150, 20)$ et $c_3 = 274200$, qui n'est pas en conflit avec E_1. Donc $LTG = \{E_1, E_3\}$.**
- **L'enchérisseur E_4 entre alors en concurrence en soumettant son offre $A_4 = (20, 55, 65, 30)$ et $c_4 = 133750$, mais sans influence sur la situation puisqu'il est en conflit avec LTG et $c_4 < c_1$ et $c_4 < c_3$.**
- **À l'instant t_5, E_2 met à jour son offre en proposant $A_2 = (40, 0, 0, 40)$ et $c_2 = 37600$, qui n'est pas en conflit avec LTG. Donc on met à jour $LTG = \{E_1, E_2, E_3\}$.**
- **Par suite à l'instant t_6 l'enchérisseur E_5 lance une nouvelle offre $S_5 = (A_5, c_5)$ avec A_5=(10,10,10,0) et $c_5 = 21600$. Cette offre est en conflit avec LTG et $c_5 < c_j \quad j \in LTG$.**

Algorithme 1 PDGL.

Entrées: β, T.
Sorties: La liste LFG des offres gagnates.

Initialisation : $LTG = \emptyset$, $LFG = \emptyset$, $Z = 0$, $t = 0$, $LEC_j = \emptyset$, pour $j = 1, \ldots, n$;
Tantque $t < T$ **Faire**
  NEVT
  Si NEVT = nouvelle offre S_j **Alors**
    Calculer LEC_j
    Si $LEC_j = \emptyset$ **Alors**
      $LTG = LTG \cup \{E_j\}$
    Sinon
      Si $|LEC_j| = 1$ ($LEC_j = \{e_i\}$) **Alors**
        Si $c_j > c_i$ **Alors**
          $LTG = LTG \cup \{E_j\} \setminus \{E_i\}$
        Finsi
      Sinon
        $|LEC_j| > 1$ ($LEC_j = \{e_i\}$)
        Si $\sum_{\{i/E_i \in LEC\}} c_i < c_j$ **Alors**
          $LTG = LTG \setminus LEC$
        Sinon
          Résoudre le problème temporaire (PDG_k)
        Finsi
      Finsi
    Finsi
  Sinon
    (NEVT = mise à jour S_j)
    Si $E_j \in LTG$ **Alors**
      Calculer LEC
      Si $E_j \cap LEC = \emptyset$ **Alors**
        E_j reste gagnant
      Sinon
        Résoudre le Problème temporaire PDG_k
      Finsi
    Sinon
      Calculer LEC_j
      Si $LEC_j = \emptyset$ **Alors**
        $LTG = LTG \cup \{E_j\}$
      Sinon
        Aller à l'étape 9 :
      Finsi
    Finsi
  Finsi
Fin Tantque

. **À l'instant t_7, E_4 lance une mise à jour S_4 avec $A_4 = (0, 50, 50, 30)$ et $c_4 = 106000$. Son offre ne présente aucun conflit, donc la liste des gagnants temporaire devient $LTG = \{E_1, E_2, E_3, E_4\}$.**

- **À l'instant t_8, une nouvelle offre S_6 est présentée par l'enchérisseur E_6, avec $A_6 = (20, 150, 75, 0)$ et $c_6 = 205750$. E_6 remplacera E_1 dans LTG, parce que $c_6 > c_1$.**
- **Puis, à l'instant t_9, E_5 met à jour son offre. Il devient gagnant parce qu'il ne présente aucun conflit.**
- **E_1 exclut E_6 de la liste temporaire des gagnants en mettant à jour à son offre.**
- **E_6 réagit de son côté en faisant une mise à jour qui le laisse remplacer E_4 dans LTG.**
- **Une nouvelle offre est présentée par E_7, $A_7 = (0, 1, 1, 1); c_7 = 3000$ mais sans aucun effet.**
- **E_4 réagit à nouveau en mettant à jour son offre $A_4 = (60, 0, 15, 30); c_4 = 39850$. Ceci le conduit à remplacer E_2 dans la liste temporaire des gagnants.**

A cet instant le temps attribué à l'exercice arrive à terme et la liste des gagnants temporaires $LTG = \{E_1, E_4, E_3, E_5, E_6\}$ devient finale. Ces étapes sont résumées dans Table 2.

Instant	**Evénement**	**Offre S_j**	**Montant**	**LTG**
t_1	Nouvelle offre	$E_1 = (20, 150, 75, 0)$	182500	$\{E_1\}$
t_2	Nouvelle offre	$E_2 = (130, 0, 0, 92)$	108840	$\{E_1\}$
t_3	Nouvelle offre	$E_3 = (70, 100, 150, 20)$	274200	$\{E_1, E_3\}$
t_4	Nouvelle offre	$E_4 = (20, 55, 65, 30)$	133750	$\{E_1, E_3\}$
t_5	Mise à jour	$E_2 = (40, 0, 0, 40)$	37600	$\{E_1, E_2, E_3\}$
t_6	Nouvelle offre	$E_5 = (10, 10, 10, 0)$	21600	$\{E_1, E_2, E_3\}$
t_7	Mise à jour	$E_4 = (0, 50, 50, 30)$	106000	$\{E_1, E_2, E_3, E_4\}$
t_8	Nouvelle offre	$E_6 = (20, 150, 75, 0)$	205750	$\{E_2, E_3, E_4, E_6\}$
t_9	Mise à jour	$E_5 = (0, 0, 50, 2)$	60820	$\{E_2, E_3, E_4, E_6, E_5\}$
t_{10}	Mise à jour	$E_1 = (20, 150, 75, 0)$	206000	$\{E_1, E_2, E_3, E_4, E_5\}$
t_{11}	Mise à jour	$E_6 = (0, 50, 50, 30)$	116000	$\{E_1, E_2, E_3, E_5, E_6\}$
t_{12}	Nouvelle offre	$E_7 = (0, 1, 1, 1)$	3000	$\{E_1, E_2, E_3, E_5, E_6\}$
t_{13}	Mise à jour	$E_4 = (60, 0, 15, 30)$	39850	$\{E_1, E_4, E_3, E_5, E_6\}$

Table 2. *Exemple exécuté par Algorithme 1*

8 Conclusion

Dans ce travail, nous avons étudié un problème d'enchères combinatoires PDG "Problème de Détermination de Gagnant" en ligne. Nous avons proposé un nouveau modèle mathématique, basé sur le mécanisme d'enchère anglaise, qui se construit au fur et à mesure que les enchérisseurs lancent leurs offres durant la période d'exercice. Puis nous avons développé une heuristique qui donne la liste des gagnants à n'importe quel instant de l'exercice. Un exemple didactique illustre l'exécution de l'algorithme développé.

Références

[1] A. Anderson, M. Tenhunen and F. Ygge. Integer programming for combinatorial auction winner determination. In Proceedings of 4th International Conference on Multi-Agent Systems, IEEE Computer Society Press, July, pp. 39-46 (2000).

[2] D. Boughaci, B. Benhamou, H. Drias. Une recherche locale stochastique pour le problème de la détermination du gagnant dans les enchères combinatoires. In Actes des Quatrièmes Journées Francophones de Programmation par Contraintes, Nantes, 4-6 juin 2008 (JFPC 2008), pp. 59-68 (2008).

[3] D. Boughaci, B. Benhamou, H. Drias. Memetic Algorithms for the Optimal Winner Determination Problems in Combinatorial Auctions, In Journal of SoftComputing (to appear).

[4] Y. Fujishima, K. Leyton-Brown, Y. Shoham. Taming the computational complexity of combinatorial auctions : optimal and approximate approaches. In Sixteenth International Joint Conference on Artificial Intelligence, pp. 548-553 (1999).

[5] Y. Guo Y, A. Lim, B. Rodrigues and Y. Zhu. Heuristics for a bidding problem. Computers and Operations Research Vol 33, Issue 8, August 2006, pp. 2179-2188.

[6] A. Holland, B. O'sullivan. Towards Fast Vickrey Pricing using Constraint Programming. Artificial Intelligence Review, Vol 21, no 3-4 / June, pp. 335-352 (2004).

[7] K. Leyton-Brown, M. Tennenholtz and Y. Shoham. An Algorithm for Multi-Unit Combinatorial Auctions. In Proceedings of the 17th National Conference on Arti

cial Intelligence, Austin, Games-2000, Bilbao, and ISMP-2000, Atlanta (2000).

[8] N. Nisan. Bidding and allocation in combinatorial auctions. In Proceedings of the ACM Conference on Electronic Commerce (EC-00), Minneapolis : ACM SIGecom, ACM Press, October, 2000, pp. 1-12. (2000).

[9] M. H. Rothkopf, A. Pekee and M. Ronald. Computationally manageable combinatorial auctions. Management Science, Vol. 44, No. 8, pp.1131-1147 (1998).

[10] T. Sandholm. Optimal Winner Determination Algorithms. In Cramton T. et al. (ed.), Combinatorial Auctions, MIT Cambridge (2006).

[11] T. Sandholm, S. Suri. Improved Optimal Algorithm for Combinatorial Auctions and Generalizations Generalizations. In : Proceedings of the 17th national conference on artificial intelligence, pp. 90-97 (2000).

[12] T. Sandholm, S. Suri, A. Gilpin, D. Levine. CABOB : A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), p. 1102-1108, Seattle, WA, August, 2001.

[13] S. de Vries, R. Vohra. Combinatorial Auctions : A survey. In INFORMS Journal of Computing, Vol 15, pp. 284-309 (2003).

[14] Riikka-Leena Leskelä. Bidder Support in Iterative, Multiple-Unit Combinatorial Auctions, Doctoral Dissertation Helsinki University of Technology, Series 2009/12 (2009).

INTERPRETATION DES IMAGES MAMMOGRAPHIQUES PAR LA METHODE DE K-MEANS ET SEARCH HARMONY

Hamida Samiha RAHLI et Nacéra BENAMRANE
Laboratoire Imagerie et Vision Artificielle
Département d'Informatique
Faculté des Sciences, USTO-MB
B.P 1505 El'mnaouer 31000, Oran, Algérie
Hamida_rahli@yahoo.fr nabenamrane@yahoo.com

Résumé:

Dans cet article, nous proposons une approche d'interprétation des images de mammographie pour une description précise des anomalies mammaires qui permet de prévoir la gravité de la tumeur afin d'évaluer le risque en termes de malignité et bénignité. Cette approche est composée de trois étapes, la première concerne la segmentation d'images en régions par fusion. La seconde étape est l'extraction des caractéristiques photométriques, géométriques et de texture, de chaque région. La dernière étape est la détection, qui est basée sur une approche hybride combinant les algorithmes K-means et Search Harmony. L'approche proposée a été testée sur des images de mammographie et les résultats obtenus sont encourageants.
Mots-clés. Interprétation, Spécification, K-means, Search Harmony, Images de mammographie.

Introduction:

Le cancer du sein est l'un des cancers les plus répandus chez les femmes. C'est une tumeur maligne du sein qui se développe au niveau de la glande mammaire (adénocarcinome). Les principaux facteurs de risque de survenue d'un cancer du sein sont l'âge (75% des cancers du sein surviennent après 50 ans), les antécédents familiaux, l'obésité, l'alimentation, le surpoids, une ménopause précoce ou une grossesse tardive.

Il n'existe pas encore de moyen d'éviter son apparition. Cependant, un cancer du sein dépisté tôt est plus facile à traiter, engendre moins de séquelles et permet de retrouver une bonne qualité de vie après le traitement. Le diagnostic précoce d'un cancer du sein, dans l'état actuel des connaissances en médecine, ne peut se faire que grâce à la mammographie de dépistage. L'espoir lié à cette démarche est non seulement de diminuer la mortalité due à cette maladie, mais également de rendre les traitements des cancers dépistés moins lourds.

Les radiologues spécialistes détectent autour de 70% de cas de cancer du sein à cause de la difficulté de l'examen de mammographie qui s'accroît avec le type de

tissu du sein examiné, les conditions de réalisation, le nombre de cliché disponibles. A ce titre, plusieurs recherches ont été menées ces dernières années afin de développer des outils d'aide au diagnostic (CAD *Computer Assisted Detection*) de cette maladie qui a pour but l'interprétation des images de mammographies qui donne une description précise des anomalies, notamment des commentaires particuliers sur la présence de masses, avec une mention précise de leur taille, de leur densité, de leur morphologie et de leur contour. Avec l'exploitation de ces différentes caractéristiques, le chercheur peut spécifier et prévoir la gravité de la tumeur pour évaluer le risque en termes de malignité/bénignité.

Dans cet article, nous proposons une approche pour le développement d'un système d'interprétation des images de mammographie basé sur les deux algotthmes K-means et Search Harmony.

1 L'algorithme de K-means :

K-means est l'un des algorithmes d'apprentissage non supervisé les plus simples qui permettent de résoudre le problème de classification du fait de sa simplicité de mise en œuvre. K-means permet de classer ou de regrouper des objets en fonction des attributs/caractéristiques en nombre K de classe, dans laquelle les objets à l'intérieur de chaque classe sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres classes. Chaque classe de la partition est définie par ses objets et son centroïde.

L'objectif est alors de minimiser la somme de l'inertie intra-classe sur l'ensemble des classes. L'algorithme procède en deux étapes : dans la première phase, on réassigne tous les objets au centroïde le plus proche, et dans la deuxième phase, on recalcule les centroïdes des classes qui ont été modifiés. Pour mesurer la proximité entre un centroïde et un objet, on calculera une distance entre ces deux vecteurs. On pourra utiliser, par exemple, la distance euclidienne.

Voici l'algorithme de K-moyenne :

L'algorithme des k-moyennes clustérise les données en k groupes où k est prédéfini

- Étape 1: choisir les centres des clusters (par exemple au hasard).
- Étape 2: affecter des instances aux clusters en se basant sur leur distance au centre des clusters.
- Étape 3: centroïdes des clusters sont calculés.
- Etape 4: aller à l'étape 1 jusqu'à convergence.

2 Algorithme recherche d'harmonie (harmony search algorithm)

En écoutant et en appréciant une pièce de musique classique, on s'est demandé s'il pleut y avoir un lien entre le fait de jouer de la musique et celui de trouver une

solution optimale à un problème complexe tel que la conception d'un réseau d'eau potable ou d'un autre problème d'ingénierie moderne ? Ce lien est représenté par un algorithme nommé « Harmony Search ALgorithm (HSA) » développé par Geem et al [].

Depuis son apparition en 2001, HSA a été appliquée à plusieurs problèmes d'optimisation, entre autres : l'optimisation de fonctions, l'optimisation de l'ingénierie, conception de réseaux de distribution d'eau, le problème de transport, la classification par l'algorithme des K plus proches voisins et l'optimisation de la production dans les réseaux électriques.

L'harmony search (HS) est un algorithme méta-heuristique, imitant le processus d'improvisation des musiciens. Dans le processus, chaque musicien joue une note pour trouver une meilleure harmonie tout ensemble. De même que, dans les processus d'optimisation, chaque variable de décision prend une valeur de façon à avoir une meilleure combinaison (vecteur) de l'ensemble

La modélisation du processus d'improvisation d'un ensemble de musiciens se base sur les règles qui dirigent chaque interprète dans l'élaboration de l'harmonie. En effet, lorsqu'un musicien improvise un ton, le plus souvent, il suit l'une des trois règles :

a. Jouer un ton de sa mémoire,
b. Jouer un ton adjacent au ton de sa mémoire,
c. Jouer un ton totalement aléatoire dans l'ensemble des sons possibles.

Par analogie, quand HSA affecte une valeur à une variable de décision, il suit l'une des trois règles ;

d. Le choix d'une valeur quelconque de la mémoire des harmonies (Harmony Memory :HM) : « considération de la mémoire »,
e. Le choix d'une valeur adjacente à la valeur de HM : « ajustement du ton »,
f. Le choix d'une valeur totalement aléatoire dans l'intervalle des valeurs possibles : « randomisation ».

Ces trois règles de HSA sont dirigées en utilisant deux paramètres, qui sont :

- Le taux de considération de la mémoire HM (Harmony Memory Considering Rate : HMCR).
- Le taux d'ajustement du ton (Pitch Adjusting Rate :PAR).

2.1 Structure générale de l'algorithme :

a) Initialiser le problème et les paramètres de l'algorithme :

Il faut noter que le problème d'optimisation spécifiquement résolu par HSA est sous la forme :

$$\begin{cases} Min\ f(x) \\ avec\ x^t = (x_1, x_2, .., x_i, .., x_N) \\ et\ x_i^{Min} \leq x_i \leq x_i^{Max}\ \forall i = 1..N \end{cases}$$

$f(x)$ est la fonction objective, x est le vecteur solution du problème, x_i est la valeur possible de la ième variable de décision. Cette valeur bornée par x_i^{Min} et x_i^{Max} . N est le nombre de variables de décision.

Dans cette étape, les paramètres suivants sont définis :

- Le taux de considération de la mémoire d'harmonie : Harmony Memory Considering Rate (HMCR).

- ❖ Le taux d'ajustement du ton : Pitch Adjusting Rate(PAR).
- ❖ Le critère qui n'est autre que le nombre d'itération de l'algorithme.
- ❖ La taille du vecteur mémoire contenant les solutions candidates et la valeur de la fonction objective correspondante. Ce paramètre, appelé HMS (Harmony Memory Size), représente la modélisation de la capacité (taille) de la mémoire des musiciens.

b) Initialiser la mémoire de l'harmonie

Dans l'étape 2, la matrice HM est remplie par des vecteurs solutions générés aléatoirement jusqu'à atteindre la taille maximale HMS.

$$\mathbf{HM} = \begin{bmatrix} x_1^1 & \cdots & x_n^1 & | & f(\mathbf{x}^1) \\ \vdots & \ddots & \vdots & | & \vdots \\ x_1^{hms} & \cdots & x_n^{hms} & | & f(\mathbf{x}^{hms}) \end{bmatrix}.$$

c) Improviser une nouvelle harmonie :

Un nouvel vecteur d'harmonie, x'=(x'$_{1,}$ x'$_{2,...,}$ x'$_{n).}$, est généré sur la base de trois règles : (1) la considération de la mémoire, (2) l'ajustement par un pas et (3) la sélection aléatoire. La génération d'une nouvelle harmonie est appelée improvisation.

Dans la considération de la mémoire, la valeur de la première variable de décision(x'$_1$) pour le nouvel vecteur(x'$_1$-x'$^{HMS}_1$) est choisie aléatoirement de la mémoire HM. Les valeurs des autres variables de décision (x'$_{2,}$ x'$_{3,...,}$ x'$_{n)}$ sont choisies de la même manière. Le HMCR, qui varie entre 0 et 1, est le taux de choisir une valeur parmi les valeurs stockées historiquement dans HM, alors que (1− *HMCR*) est le taux de sélectionner aléatoirement une valeur parmi les valeurs possible.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} \text{ avec la probabilité } HMCR \\ x_i' \in X_i \text{ avec la probabilté } (1 - HMCR) \end{cases}$$

Par exemple, un HMCR de 0.95 indique que l'algorithme HS choisira une valeur de la variable de décision parmi les valeurs historiquement stockées dans la HM avec une probabilité de 95% ou parmi les autres valeurs possibles avec une probabilité de (100–95)%. Chaque élément obtenu par la considération de la mémoire est analysé pour déterminer s'il doit être ajusté par un pas ou non. Cette opération utilise le paramètre PAR, qui est le taux d'ajustement par un pas, l'ajustement se fait comme suit :

$$\text{Décision d'ajustement pour } x_i' \leftarrow \begin{cases} \text{Oui avec la probabilité } PAR, \\ \text{Non avec la probabilté } (1 - PAR). \end{cases}$$

Dans le cas ou la probabilité est égale à (1 - PAR) la valeur de x'$_i$ est laissée sans ajustement. Si la décision de l'ajustement par un pas pour x'$_i$ est OUI, x'$_i$ est substitué comme suit :

$$x_i' \leftarrow x_i' + rand() * bw_i$$

Où *bw* est la longueur de l'intervalle des valeurs possibles, *rand*( ) est un nombre aléatoire entre 0 et 1.

d) Mis à jour de la mémoire de l'harmonie :

Si le nouvel vecteur d'harmonie, $x'=(x'_{1,}\ x'_{2,\ldots,}\ x'_{n)}$ est meilleur que la mauvaise harmonie dans la HM, jugé sur l'évaluation du nouveau vecteur par la fonction objectif, alors on substitut la mauvaise harmonie dans HM par la nouvelle harmonie.

Dans l'étape 3, La considération de HM, l'ajustement par un pas ou la sélection aléatoire sont appliqués séquentiellement pour chaque variable du nouvel vecteur harmonie.

e) Verification du critère d'arrêt:

Si le critère d'arrêt (nombre maximum d'improvisations) est satisfait, le calcul est terminé. Autrement, l'étape 3 et 4 sont répétées.

3 Approche proposée :

Notre approche est basée sur une hybridation de la méthode K-means et la méthode d'optimisation Search Harmony afin de détecter les tumeurs.

3.1 Segmentation et Extraction des caractéristiques :

En effet après une segmentation de l'image en régions basée sur une technique de fusion, nous nous intéressons à extraire les caractéristiques de chaque région. L'espace de caractéristique est très grand et complexe en raison de la grande diversité des tissus normaux et la variété des anomalies. L'espace de caractéristiques peut être divisé en trois sous-espaces: les caractéristiques photométrique comme le niveau de gris moyen et la variance, les caractéristiques géométriques telle que la surface et pour les caractéristiques de texture, il s'agit d'obtenir la matrice de moyennes d'espace du second ordre, appelée matrice de cooccurence. Cette matrice contient une masse très importante d'informations difficilement manipulable. C'est pour cela qu'elle n'est pas utilisée directement mais à travers des mesures dites indices de textures. En 1973, Haralick et al. ont proposé quatorze indices. Nous présentons ci-dessous les caractéristiques utilisées :

Le niveau de gris moyen.

Il représente la moyenne des intensités de tous les pixels de la région.

$$NGM = \sum_{1}^{S} I(i) / S(R)$$

où I(i) est l'intensité du pixel i et S(R), la surface de la région.

La variance.

Cet attribut caractérise la variation des niveaux de gris dans une région.

$$V = \sum_{1}^{S} \left(I(i) - NGM\right)^{2} / S(R)$$

Homogénéité.

Cet indice est d'autant plus élevé que l'on retrouve souvent le même couple de pixels.

$$\left(1/N^{2}\right) \sum_{i} \sum_{j} \left(MAT(i, j)\right)^{2}$$

Entropie.

L'entropie fournit un indicateur de désordre que peut présenter une texture. Elle est faible si on a souvent le même couple de pixels et forte si chaque couple est peu représenté.

$$1 - \frac{1}{N.\ln(N)} \sum_{i} \sum_{j} MAT(i, j).\ln(MAT(i, j)).1_{MAT(i, j)}$$

Avec $1_{MAT(i,j)}$ = 1 si Mat(i,j)≠0 et 0 sinon.

N : somme des éléments de la matrice de cooccurrence.

Contraste.

Le contraste est élevé quand on passe d'un pixel très clair à un pixel plus foncé et inversement.

$$\frac{1}{N(D-1)^{2}} \sum_{k=0}^{D-1} k^{2} \sum_{|i-j|=k} MAT(i, j)$$

3.2 Détection par K-means-harmony Search :

La structure de notre approche de détection des tumeurs s'inspire de l'approche du médecin lors de l'examen radiologique. Cette approche hybride K-means qui une

méthode de classification très connue et une méthode d'optimisation harmony Search pour optimiser la détection.

Dans les paragraphes suivants, nous détaillerons notre approche étape par étape.

a) Initialisation de la mémoire d'harmonie :

Elle doit être initialisée de manière aléatoire, chaque ligne de la mémoire correspond à une classe spécifique des régions dans laquelle la valeur de l'élément i dans chaque ligne est choisie au hasard de la distribution uniforme sur l'ensemble {1,2} et indique le numéro de la classe, on attribut n régions aléatoirement à chaque classe.

On a un ensemble de régions $R_i = \{R_1, R_2, .., R_n\}$, la matrice $M = [a_{ij}] = \begin{cases} 1 \; si \; la \; jéme \; région \; appartienne \; à \; la \; kième \; classe \\ 0 \; sinon \end{cases}$

b) Calcul de la probabilité d'appartenance des régions :

Pour chaque région $R_i = \{R_1, R_2, .., R_n\}$, la probabilité d'appartenance de chaque régions dans chaque classe (Tumeur/Non Tumeur) est calculée.

$$\sum_{i=1}^{k} ((D_{max} - D(R_j - R_j, C_k))$$

$D_{max} = max_i \; \{D(R_j, C_k), R_j$ est la région récemment improvisée.

c) Calcul de la fonction de fitness :

La fonction de fitness est la distance minimale des centroïdes des classes.

$$f = \frac{\sum_{i=1}^{k} \left\{ \frac{\sum_{j=1}^{n} D(R_j, C_k)}{N_i} \right\}}{K}$$

K : Nombre de classe pour notre cas est égal à 2.

N_i : Nombre de régions dans la classe i $N_i = \sum_{i=1}^{n} a_{ij}$.

d) Improvisation d'une nouvelle harmonie et mise à jour de la HM:

On a pris HMCR (harmony memory considering rate) (HMCR=0.95) et PAR (pitch adjusting rate) (PAR = 0.3).

Algorithme de l'improvisation et de mise à jour des HM :

```
Tant que(i<=nombre de critère maximal) faire
  Si(rand ∈ (0,1)<=HMCR) alors
    Choisir une valeur de la HM pour la valeur de
niveau de gris
    Choisir une valeur de la HM pour la valeur de la
surface
Si(rand ∈ (0,1)<=PAR) alors
Ajuster la valeur de niveau de gris
Ajuster la valeur de la surface
```

$$val_{new} = val_{old} + rand \in (0,1) * (val_{max} - val_{min}).$$

```
Fsi
Else
Choisir une variable aléatoire ε (0,1).
```
$val_{new} = val_{min} + rand \in (0,1) * (val_{max} val_{min}).$
```
 Fsi
Fintantque
```

On réitère ce processus par le nombre d'itération, et on calcule à chaque fois ces paramètres,
On obtient cette matrice qui contient la valeur de la fonction de fitness, la matrice de M qui contient la probabilté d'appartenance pour chaque région et les valeurs de C1,C2 qui ont été ajustées.

f1	M1	C1	C2
f2	M2	C1	C2
.	.	.	.
.	.	.	.
.	.	.	.
f30	M30	C1	C2

A la fin de ce processus, on prend la valeur minimale de f, et on affiche les régions de la matrice M qui contient les régions appartenant à la classe 1 (Tumeur).

4 Résultats :

L'approche proposée a été testée sur des images de mammographie de taille 256x256. La figure 1montre quelques valeurs des caractéristiques des régions.

La Region :	La Surface (Pixels) :	Le Niveau Du Gris Moyen :	La Varia	Homogeneite :	Heterogeneite :	Contraste :	ENTROPIE :
N°: 1 -->	2329 -->	77 -->	207,51	0,04 -->	109,03 -->	277,03 -->	57,01
N°: 2 -->	15967 -->	125 -->	42,97	0,45 -->	1487,58 -->	418,81 -->	268,44
N°: 3 -->	20910 -->	132 -->	39,35	0,73 -->	3383,03 -->	538,26 -->	436,59
N°: 4 -->	24 -->	156 -->	79,88	0,65 -->	3383,04 -->	538,94 -->	436,60
N°: 5 -->	2 -->	82 -->	3444,50	0,72 -->	3383,03 -->	538,52 -->	436,60
N°: 6 -->	13 -->	94 -->	71,62	0,04 -->	112,89 -->	280,18 -->	58,88
N°: 7 -->	2 -->	82 -->	3362,00	0,73 -->	3383,03 -->	538,46 -->	436,60
N°: 8 -->	4 -->	126 -->	1325,00	0,68 -->	3383,04 -->	538,84 -->	436,60
N°: 9 -->	4 -->	103 -->	902,00	0,94 -->	3363,74 -->	497,55 -->	423,93
N°: 10 -->	8 -->	90 -->	152,25	--> 0,04 -->	112,89 -->	280,18 -->	58,88
N°: 11 -->	1149 -->	36 -->	71,29	--> 0,06 -->	89,14 -->	163,56 -->	46,16
N°: 12 -->	6 -->	86 -->	266,50	--> 0,06 -->	137,71 -->	306,70 -->	71,24
N°: 13 -->	27 -->	129 -->	47,56	--> 0,94 -->	3376,96 -->	523,37 -->	432,04
N°: 14 -->	102 -->	135 -->	11,91	--> 0,93 -->	3384,14 -->	529,20 -->	435,06
N°: 15 -->	2 -->	50 -->	1300,50	--> 0,06 -->	135,55 -->	304,96 -->	70,12
N°: 16 -->	2 -->	68 -->	2312,00	--> 0,93 -->	3319,78 -->	496,28 -->	412,43
N°: 17 -->	11 -->	123 -->	161,36	--> 0,95 -->	3358,13 -->	507,68 -->	422,57
N°: 18 -->	121 -->	131 -->	23,93	--> 0,92 -->	3392,82 -->	532,39 -->	436,53
N°: 19 -->	4 -->	102 -->	901,50	--> 0,94 -->	3351,91 -->	502,12 -->	418,83

Fig. 1**.** Caractéristiques photométriques, géométriques et de texture

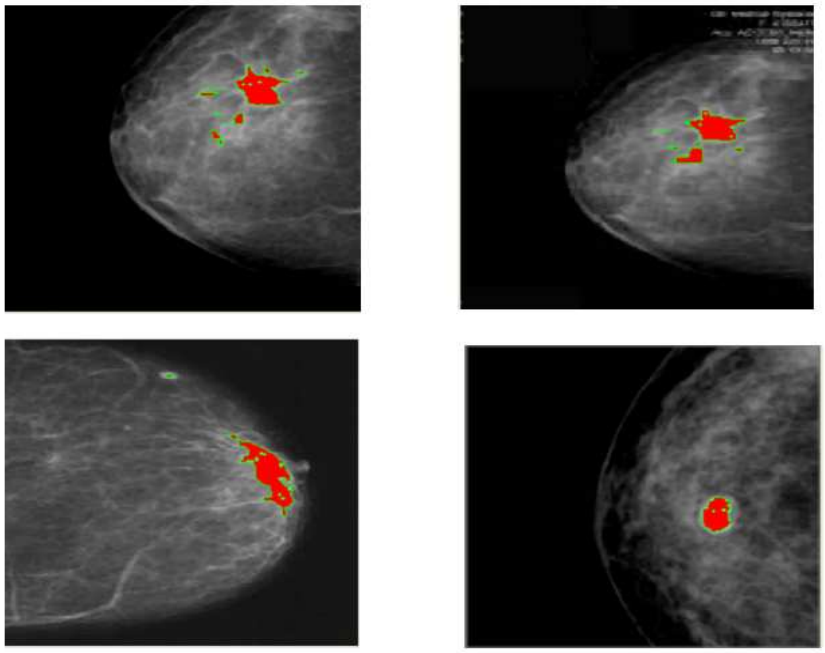


Fig. 2. Résultats de détection obtenus

La figure 2 montre les résultats obtenus de la détection des tumeurs obtenus sur des images de mammographie de taille 256*256 ayant subis un prétraitement.

Conclusion :

Dans cet article nous avons proposé une approche d'interprétation des images de mammographie permettant de détecter les tumeurs basée sur l'hybridation de deux algorithmes : « K-moyenne et Search Harmony ». Cette approche a été testée sur des images de mammographie et les résultats obtenus sont intéressants et peuvent être améliorés en prenant en considération d'autres critères de fusion et d'indices de texture.

Références bibliographiques :

[1] Osama Moh'd Alia, Rajeswari Mandava, Mohd Ezane Aziz "A Hybrid Harmony Search Algorithm to MRI Brain Segmentation » Proc. 9th IEEE Int. Conf. on Cognitive Informatics (ICCI'10), 2010.

[2]Corinne VACHIER «Extraction de caractéristiques, segmentation d'image et morphologie mathématique » Thèse pour obtenir le grade de Docteur en morphologie

mathématique, 18 Décembre 1995, Ecole Nationale Supérieure des Mines de Paris, France.

[3] Giap Nguyen « Extraction de zones d'intérêts dans une image de textures » Rapport de stage, 30 Aout 2009, Laboratoire Informatique, Image et Interaction (L3I) Université de La Rochelle, France.

[4] Alfonso Rojas Domínguez, Asoke K. Nandi "Toward breast cancer diagnosis based on automated segmentation of masses in mammograms", Pattern Recognition, Volume 42, issue 6, June 2009.

[5] Rabi Narayan Panda, Dr. Bijay Ketan Panigrahi, Dr. Manas Ranjan Patro "Feature Extraction for Classification of Microcalcifications and Mass Lesions in Mammograms", International Journal of Computer Science and Network Security (IJCSNS), Volume 9, No 5, May 2009.

[6] Claudio Marrocco, Mario Molinara, Ciro D'Elia, Francesco Tortorella "A computer-aided detection system for clustered microcalcifications", Artificial Intelligence in Medicine, Volume 50, Issue 1, Pages 23-32,September 2010.

[7] HANAE Naoum, "approche neuro-mimétique au service du dépistage du cancer du sein", Université du Québec à Montréal, Mémoire présenté comme exigence partielle en informatique, Octobre 2009.

[9] K.Thangavel, M.Karnan "Computer Aided Diagnosis in Digital Mammograms: Detection of Microcalcifications by Meta Heuristic Algorithms" International Journal on Graphics Vision and Image Processing, 5(5):31-61, 2005.

[10] H. D. Cheng, Xiaopeng Cai, Xiaowei Chen, Liming Hu and Xueling Lou "Computer-aided detection and classification of microcalcifications in mammograms: a survey", Pattern Recognition, Volume 36, issue 12, Pages 2967-2991, December 2003.

Complete and incomplete approaches for graph mining*

Amina Kemmar [1], Yahia Lebbah [1], Mohamed Ouali [1], and Samir Loudni [2]

University of Oran, Es-Senia, Lab. LITIO,
B.P. 1524 EL-M'Naouar, Oran, Algeria
[2] University of Caen - Campus II, Department of Computer Science, France
{kemmami,ylebbah}@yahoo.fr,mohammed.ouali@gmail.com,samir.loudni@unicaen.fr

Abstract. In this paper, we revisit approaches for graph mining where a set of simple encodings are proposed. Complete approaches are those using an encoding allowing to get all the frequent subgraphs. Whereas incomplete approaches do not guarantee to find all the frequent subgraphs. Our objective is also to highlight the critical points in the process of extracting the frequent subgraphs with complete and incomplete approaches. Current canonical encodings have a complexity which is of exponential nature, motivating this paper to propose a relaxation of canonicity of the encoding leading to complete and incomplete encodings with a linear complexity. These techniques are implemented within our graph miner GGM (Generic Graph Miner) and then evaluated on a set of graph databases, showing the behavior of both complete and incomplete approaches.

Keywords: Graph mining, frequent subgraph, pattern discovery, graph isomorphism.

1 Introduction

Graph-mining represents the set of techniques used to extract interesting and previously unknown information about the relational aspect from data sets that are represented with graphs. It becomes an important research field with the increasing demand on the analysis of large amounts of structured and semi-structured data. The most natural form of knowledge that can be extracted from graphs is also a graph. The combinations of theses graphs are called graph patterns and are required in a number of different application domains including semantic web [1], VLSI reverse engineering [7], chemical compound classification [2], computer vision and video indexing. The most important problems studied in the graph-mining are the patterns matching, the clustering and the frequent subgraph discovery problem which makes the subject of this paper.

We revisit some approaches for graph mining where a set of simple encodings are proposed. Complete approaches are those using an encoding enabling to get

* This work is supported by TASSILI research program 11MDU839 (France, Algeria).

all the frequent subgraphs. Whereas incomplete approaches do not guarantee to find all the frequent subgraphs. Our objective is also to highlight the critical points in the process of extracting the frequent subgraphs. The introduced techniques are implemented within GGM (generic graph miner). We provide an experimentation with GGM showing the behavior of complete and incomplete approaches.

It is not proven if the canonical encoding of graphs is in the class of NP-complete problems, nor in polynomial class. This is also verified in practice, since that all the current canonical encodings have complexities which are of exponential nature. This motivates deeply our work on proposing a relaxation of the canonicity of the encoding, leading us to what we qualified in this paper *complete* and *incomplete* encodings with low polynomial complexities. For instance, let us take one database of our experimentations, with 187 nodes and 180 edges to find all the graphs having a frequence greater than 204: our incomplete algorithm has performances which are close to those of the state of the art graph miner Gaston [5].

The following section 2 introduces preliminaries on graph mining and the current approaches to solve frequent subgraph discovery problem. Section 3 explains our graph mining algorithm GGM. Experimental results of GGM are given in section 4. Section 5 concludes the paper and addresses some perspectives.

2 Frequent subgraph discovery problem

An undirected graph $G = (V, E)$ is made of the set of vertices V and the set of edges $E \subseteq V \times V$. Each edge (v_1, v_2) is an unordered pair of vertices. We will assume that the graph is labeled with vertex labels L_V and edge labels L_E; the same label can be assigned to many vertices (or edges) in the same graph. The size of a graph $G = (V, E)$ is defined to be equal to $|E|$.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection $\psi : V_1 \rightarrow V_2$ such that for every $u_1, v_1 \in V_1$, $(u_1, v_1) \in E_1$ if and only if $(\psi(u_1), \psi(v_1)) \in E_2$; ψ is called an isomorphism. Since our graphs are labeled, this mapping must also preserve the labels on the vertices and edges. Given two graphs $G_s = (V_s, E_s)$ and $G = (V, E)$, G_s is a subgraph of G (a) if and only if $V_s \subseteq V$, $E_s \subseteq E$, or (b) if G_s is isomorphic to a subgraph of G. Let be some two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the subgraph isomorphism problem consists to find an isomorphism between G_2 and a subgraph of G_1; in some sense, we have to determine whether G_2 is included or not in G_1.

The canonical label of a graph $G = (V, E)$, denoted by $ca(G)$, is defined as the unique code that is invariant on the ordering of the vertices and edges in the graph. Straightforwardly, two graphs having the same canonical label are isomorphic. Canonical labeling and graph isomorphism are not known to be in NP-complete problems.

The frequent subgraph discovery problem is defined as follows:

Definition 1 (Frequent Subgraph discovery). *Given a database $\mathcal{G}$ which contains a collection of graphs. The frequency of a graph G in $\mathcal{G}$ is defined by*

$freq(G, \mathcal{G}) = \#\{G' \in \mathcal{G} | G \subseteq G'\}$. *The support of a graph is defined by*

$$support(G, \mathcal{G}) = freq(G, \mathcal{G})/|\mathcal{G}|.$$

The frequent subgraph discovery problem consists to find all connected undirected graphs F *that are subgraphs of at least* $minsup|\mathcal{G}|$ *graphs of* $\mathcal{G}$*:*

$$F = \{G \in \mathcal{G} | support(G, \mathcal{G}) \geq minsup\},$$

for some predefined minimum support threshold minsup *that is specified by the user.*

Generally, we can distinguish between the methods of discovering frequent subgraphs according to the way the three following problems are handled:

Candidates generation problem This is the first step in the frequent subgraph discovery process which depends on the search strategy. It can be done with breadth first or depth first strategies. With breadth first strategy, all k-candidates (i.e., having k edges) are generated together, then $(k+1)$-candidates and so on; making the memory consumption huge [4][3]. But with a depth approach, the k-candidates are iteratively generated, one by one. At any stage of the depth process, only one k-candidate is generated, then we proceed to find its extensions to its related $(k+1)$-candidates, and this process is repeated with each $(k+1)$-candidate. This depth strategy enables to find quickly first frequent graphs. The algorithms of such type are numerous. For example, the Gspan [6] algorithm encodes the generated graphs using the canonical DFS code which makes the test of isomorphism more efficient. Another one is the GASTON [5] tool which divides the search process in three steps: finding frequent paths, then transforming these paths into trees, and finally transforming these trees into cyclic graphs. This last approach allows to use the existing efficient algorithms for discovering frequent paths and frequent trees.

Subgraph encoding problem When some new candidate is produced, we should verify that it has been already generated. This can be resolved by testing if this new candidate is isomorphic to one of the already generated subgraphs. The canonical DFS code [6] is usually used to encode the generated frequent subgraphs. By this way, verifying that the new candidate is isomorphic to one of the already generated candidates is equivalent to testing if its encoding is equal to the encoding of some already generated candidate.

Frequency computation problem If some new candidate is declared to be not isomorphic to any of the already produced candidates, we should compute its frequency. It could be done by finding all the graphs of the database which contain this new candidate.

In the following section, we present a new algorithm GGM - Generic Graph Miner - for finding connected frequent subgraphs in a graphs database. We propose also some simple encodings to handle efficiently the frequency counting problem.

3 GGM, a generic graph miner

GGM finds frequent subgraphs, parameterized with some encoding strategies detailed in section 3.3. It is generic, because we aim to make the key steps of GGM easily parameterized.

Algorithm 1 GGM($\mathcal{G}$, f_{min})

Require: $\mathcal{G}$ represents the graph dataset and f_{min} the minimum frequency threshold.
Ensure: $\mathcal{F}$ is the set of frequent subgraphs in $\mathcal{G}$.
1: $\mathcal{EL} \leftarrow$ all frequent edge labels in $\mathcal{G}$
2: $\mathcal{N} \leftarrow$ all frequent node labels in $\mathcal{G}$
3: $\mathcal{P} \leftarrow$ Generate-Paths($\mathcal{N}$, $\mathcal{G}$, f_{min})
4: $\mathcal{T} \leftarrow$ Generate-Trees($\mathcal{P}$, $\mathcal{G}$, f_{min})
5: $\mathcal{C} \leftarrow$ Generate-Cyclic-Graphs($\mathcal{P} \cup \mathcal{T}$, $\mathcal{G}$, f_{min})
6: $\mathcal{F} \leftarrow \mathcal{P} \cup \mathcal{T} \cup \mathcal{C}$
7: RETURN $\mathcal{F}$

The general structure of the algorithm is illustrated in algorithm 1. The algorithm initializes the frequent subgraphs with all frequent edges and nodes within the graph database $\mathcal{G}$. Then, the algorithm proceeds with three separated steps:

1. enumerating frequent paths from the frequent nodes,
2. generating the frequent trees from the frequent paths by keeping the same extremities of each initial path,
3. extending the frequent paths and trees by adding an edge between two existing nodes to obtain cyclic graphs.

This approach is inspired from GASTON [5] in which these three steps are repeated for each discovered subgraph. In other words, GASTON loops on the above three steps, whereas in our approach, they are executed one time only.

3.1 Candidates generation

Path enumeration A path is encoded by the concatenation of node and edge labels according to their appearance in the path. The problem of such encoding is that a path can have two orientations. This problem can be resolved by using the approach used by the Gaston tool [5]. A new path is obtained by adding a frequent edge to both extremities if this path is asymmetric, to one extremity otherwise. This extension is done with *EXTEND-NODE* algorithm 2. More precisely, *EXTEND-NODE* allows to add a new node to some current graph, this node will be connected with the node *node* given as argument. If the code of the current candidate does not exist in the set *codes*, the following step computes its frequency.

Tree and cyclic graph enumeration The frequent trees are generated from the frequent paths by adding a frequent edge to each node in the path except the extremities. These last ones will never be extended in any of the

Algorithm 2 EXTEND-NODE($\mathcal{F}, \mathcal{EL}, \mathcal{CD}, F, Node$)

Require: $\mathcal{F}$ represents the frequent subgraphs, $\mathcal{EL}$ is the set of edge labels, $\mathcal{CD}$ represents the set of codes of all the generated graphs, F is the graph to extend and $Node$ represents the node to extend.
Ensure: $\mathcal{F}$ is the set of frequent subgraphs in $\mathcal{G}$.

```
1: for each edgelabel adjecent to Node in EL do
2:    {Generate the graph candidate C from F}
3:    C ← F + (Node, edgelabel)
4:    if Code(C) doesn't exist then
5:       {Calculate the frequency}
6:       SET-FREQUENCY(C, Node, edgelabel)
7:       if C.freq >= f_min then
8:          F ← F ∪ {C}
9:       end if
10:      CD ← CD ∪ {Code(C)}
11:   end if
12: end for
13: RETURN F
```

following generation steps. By using these frequent subgraphs, we enumerate the frequent cyclic subgraphs by adding a frequent edges between two nodes. In the next step, we should verify if the obtained subgraph has been already generated.

3.2 Frequency counting

The strategy adopted to calculate the frequency of a given candidate C having $(k+1)$ edges generated from the frequent subgraph F having k edges is tightly related to the process of generating new candidates. As explained in section 3.1, roughly speaking, from the frequent subgraph F, the candidates generation process tries to add some edge to obtain the new candidate C. The number of successfull extensions of the current instances of F with the considered edge is straightforwardly the frequency of C. This approach is much more efficient than finding from scratch the number of instances of C in the database which is equivalent to solving the subgraph isomorphism of C in the database. Its inconvenience is its memory consumption particularly at the beginning of the algorithm.

3.3 Graph encoding

The canonical labeling is used to check whether a particular candidate subgraph has already been generated or not. However, developing algorithms that can efficiently compute the canonical labeling is critical to ensure that the mining algorithm can scale to very large graph datasets. There exists different ways to assign a code to a given graph, but it must uniquely identify the graph such that if two graphs are isomorphic, they will be assigned the same code. Such

encoding is called a *canonical encoding*. It is not proven if the canonical encoding of graphs is in the class of NP-complete problems, nor in polynomial class. This is also verified in practice, since that all the current canonical encodings have complexities which are of exponential nature. The first way to obtain such canonical encoding is to convert the adjacency matrix representation into a linear sequence of symbols. According to the order of the nodes in the original matrix, a new code is obtained. However, the canonical labeling represents the minimum or the maximum of these codes. This canonical labeling is used in [4]. Another canonical labeling used in [6] is based on the DFS [6] tree to attribute a code to a graph. Then, the minimum of these codes (minimum DFS) will be considered as the canonical label. But computing such canonical encodings is very costly, particularly on large graphs. Moreover, all the manipulated subgraphs during graph mining should be encoded. Thus having a quick canonical encoding is a key issue.

Definition 2 (Canonical encoding). *Let f be an encoding function, G_1 and G_2 are two graphs. f is canonical if :*

$$G_1 \text{ and } G_2 \text{ are isomorphic iff } f(G_1) = f(G_2).$$

For more details on canonical encodings, see the references [4] and [6].

The idea of our encoding is to use a non-canonical encoding, resulting in two kinds of encodings : complete and incomplete.

Definition 3 (Complete and incomplete encodings). *Let f be an encoding function. For any two distinct non-isomorphic graphs G_1 and G_2, f is complete if $f(G_1) \neq f(G_2)$. Otherwise, f is said to be incomplete.*

The complete approach allows to find all frequent subgraphs when the same subgraph can be generated repetitively, because with this encoding, two isomorphic graphs can lead to different codes. Whereas, using the incomplete one, two not isomorphic graphs can have the same encoding, which cause the elimination of a frequent graph not yet generated.

In the following, we propose two encodings:

DFS based complete encoding This encoding is a relaxation of that defined in [6]. Such encoding is processed by taking only one walk through a depth first search (and not the minimum as in [6]). It is straightforward that this encoding is complete, and the same graph can be generated several times as illustrated in Figure 1 which shows that two isomorphic graphs can have different codes. For the graph (a) in Figure 1, there exists several DFS codes. Two of them, which are based on the DFS trees in Figure 1(b)-(c) are listed in Table 1.

Since this encoding visits only once the edges, it is then straightforward that its worst complexity is $O(m)$, where m is the number of edges.

Edge sequence based incomplete encoding Given a graph G and an edge $e_{ij} = (v_i, v_j) \in G$, where $deg(v_i) \leq deg(v_j)$, the edge e_{ij} is represented by the 5-tuple: $(deg(v_i), deg(v_j), l_v(v_i), l_e(v_i, v_j), l_j(v_j))$,

edge	0	1	2	3	4	5
Fig 1.(b)	$(1, 2, X, s, Y)$	$(2, 3, Y, t, Y)$	$(3, 1, X, s, X)$	$(3, 4, X, q, Z)$	$(4, 2, Z, t, Y)$	$(2, 5, Y, r, Z)$
Fig 1.(c)	$(1, 2, Y, s, X)$	$(2, 3, X, s, X)$	$(3, 1, X, t, Y)$	$(3, 4, X, q, Z)$	$(4, 1, Z, t, Y)$	$(1, 5, Y, r, Z)$

Table 1. DFS codes for Figure 1 (b)-(c)

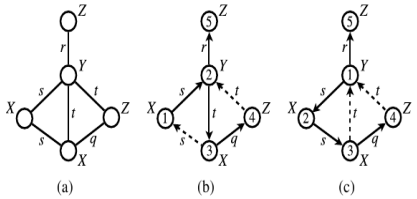


Fig. 1. Different DFS trees associated to the labeled graph (a)

where $deg(v)$ is the degree of v in the graph G, $l_v(v)$ and $l_e(e)$ are the labels of the vertex v and the edge e respectively.
Given a graph G, we denote SEQ-DEG(G) the sequence of its edge codes :

$$SEQ - DEG(G) = code(e_1)code(e_2)...code(e_{|E|})$$

where $e_i <_l e_{i+1}$, the relation $<_l$ defines a lexicographic order between edges (e.g. $(2, 3, X, s, Y) < (2, 3, Z, s, Y)$). The code associated to the graph (a) of Figure 1 is:

$$(1, 4, Z, r, Y)(2, 3, X, s, X)(2, 3, Z, q, X)(2, 4, X, s, Y)(2, 4, Z, t, Y)(3, 4, X, t, Y).$$

Enumerating the edges is done with O(m), but sorting lexicographically the edges requires $O(m\ log(m))$ which is the worst case complexity of sorting algorithms. Thus, in final, the complexity of this encoding is $O(m\ log(m))$.
This encoding is not complete because we can find examples of non-isomorphic graphs having the same code, one of these examples is illustrated in Figure 2. Note that finding such a graphs is quite non trivial task. We will see that this is confirmed by the reasonable performance of this encoding in our experimentations.

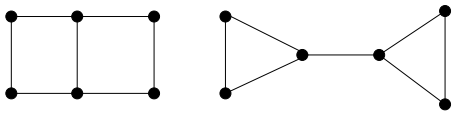


Fig. 2. Example of two non-isomorphic unlabeled graphs having the same encoding SEQ-DEG

4 Experimental Results

We performed a set of experiments to evaluate the performance of our algorithm GGM on two kinds of graph databases. The first database of large graphs contains some molecular structures of chemical compounds (PTE [1]). The second databases of small graphs are extracted from the last one (PTE1,PTE2,PTE3). The characteristics of these datasets are illustrated in Table 2. All experiments were done on 2.4Ghz Intel Core 2 Duo T8300 machines with 2GB main memory, running the Linux operating system.

Name	PTE1	PTE2	PTE3	PTE
Number of graphs	1	5	20	340
Number of nodes	8	98	519	9189
Number of edges	7	102	530	9317
Average number of nodes	8	19	25	27
Average number of edges	7	20	26	27
Number of node labels	2	10	10	66
Number of edge labels	1	3	4	4

Table 2. Caracteristics of graph datasets used in the experiments

	PTE1	PTE1	PTE3
MinFreq	1	3	8
Gaston	0,00	0,00	0,00
GGM SEQ-DEG	0,02	0,13	0,17
GGM DFS	0,22	3,42	3,94

Table 3. Runtimes in second of Gaston and GGM on simple graph datasets

MinSup % = MinFreq	20% = 68			50% = 170			60% = 204		
Algorithm	GASTON	SEQ-DEG	DFS	GASTON	SEQ-DEG	DFS	GASTON	SEQ-DEG	DFS
#freq. paths	53	53	-	17	17	17	9	9	9
#freq. trees	124	97	-	15	14	66	2	2	6
#freq. cyclic graphs	13	12	-	2	2	10	0	0	0
# Total	190	162	-	34	33	93	11	11	15
runtime (s)	0,06	2,29	>2000	0,02	0,60	4,29	0,01	0,26	0,68

Table 4. Results of Gaston and GGM (with the DFS encoding and SEQ-DEG) on the graph dataset PTE. MinSup represents the minimum support threshold and MinFreq the minimum frequency.

We have done a comparison between our algorithm and the Gaston tool. Table 3 (resp. Table 4) shows the results of our algorithm with the first database (resp. second database).

The minimum frequency was set to different MinFreq expressed by MinSup (i.e. minimum support). For instance, for the PTE database which contains 340 graphs, the number of graphs that are subgraphs of at least $60\% = \frac{204}{340}$ graphs of PTE, is given by $\#Total = \#freq.paths + \#freq.trees + \#freq.cyclic\ graphs$. From this frequency, we see that there is no frequent cyclic graphs (i.e. #freq. cyclic graphs = 0).

[1] The Predictive Toxicology dataset (PTE) can be downloaded from `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/PTE`.

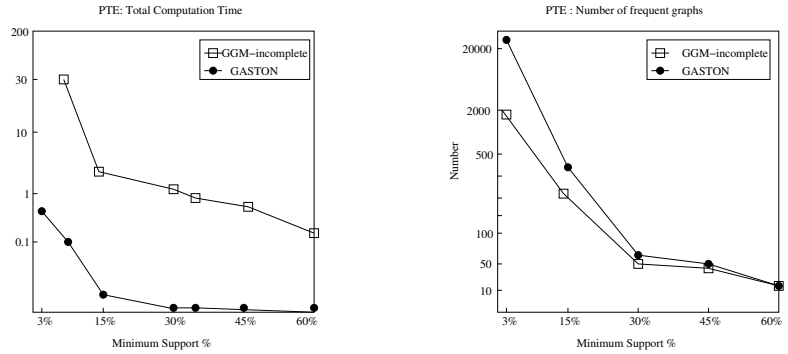


Fig. 3. Results of Gaston and GGM-incomplete on the graph dataset PTE

Concerning the results on both databases, the complete encoding is usually less performant than the incomplete one. This is explained by the fact that the complete encoding handles larger set of candidates than the incomplete one. For the incomplete encoding, the number of frequent subgraphs discovered by GGM is not too far from that of Gaston if the minimum freqeuncy is larger than 30% (Figure 3). We notice also that from the frequency of 170 graphs, the result is the same.

The execution time of Gaston is better than GGM-incomplete. We point out that Gaston is recognized as one of the efficient graph miners in the literature. It is not obvious to verify if the good performance of Gaston is due to its efficient data structures or its graph encoding? Thus, the experimental perspective of this paper is to integrate our incomplete encoding into Gaston in order to make clearer the contribution of an incomplete encoding in graph mining.

5 Conclusion

In this paper, we have presented algorithm *GGM* for the frequent subgraph discovery problem in a graph datasets. We pointed out the key points in the graph mining process. We combined several strategies inspired from existing algorithms to implement *GGM*. The two important points in the process of discovery are the generation of new candidates and the frequency counting. Our experimentations show the effectiveness of the incomplete approach compared to the complete one. It shows the importance of handling a reasonable amount of candidates, which is the case in incomplete approaches, and not forthe complete ones. The main perspective is to improve our encodings, and find a way to combine complete and incomplete approaches to improve the whole efficiency of GGM.

The graphs of our benchmarking are not dense: it is advised to experiment dense graphs, where canonical encodings are of exponential nature. We have also to integrate the incomplete and complete encodings into the open source

Gaston miner, in order to compare more easily the introduced encodings within the same miner.

Acknowledgements

The authors would like to acknowledge reviewers for helpful comments.

References

1. Bettina Berendt, Andreas Hotho, and Stum Gerd. Towards semantic web mining. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 264–278, London, UK, UK, 2002. Springer-Verlag.
2. Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Trans. on Knowl. and Data Eng.*, 17:1036–1050, August 2005.
3. Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 13–23, London, UK, 2000. Springer-Verlag.
4. Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 313–320, Washington, DC, USA, 2001. IEEE Computer Society.
5. Siegfried Nijssen and Joost N. Kok. The gaston tool for frequent subgraph mining. *Electron. Notes Theor. Comput. Sci.*, 127:77–87, March 2005.
6. Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. *Order A Journal On The Theory Of Ordered Sets And Its Applications*, 02:721–724, 2002.
7. Ken'ichi Yoshida and Hiroshi Motoda. Clip: concept learning from inference patterns. *Artif. Intell.*, 75:63–92, May 1995.

Bases de données et Systèmes d'Information

The Bag of Similarity Scores: A New Conceptual Representation for Software Entities

Mostefai Abdelkader[1] Malki Mimoun[2] Sidi Mohamed Benslimane[2]

[1]Saida University & EEDIS Laboratory, Djilali Liabes university SBA, Algeria.

[2] EEDIS Laboratory, Djilali Liabes university SBA, Algeria.

[1] {mostefaia_aek , malki_m , sidim_2 }@yahoo.fr

[2] {malki_m , sidim_2 }@yahoo.fr

Abstract. Selecting a good representation of the entities to be clustered along with good clustering algorithms is of paramount importance [12] for the effectiveness of any process of software architecture recovery.

This paper investigates the use of the Bag of Similarity Scores (i.e., BoSS) representation for legacy software clustering. We introduce a new approach to create conceptual representation of software entities and apply it in an agglomerative Hierarchical clustering process to recover the legacy software architecture of medium-sized legacy bank software. The experiment demonstrates the effectiveness of the representation.

Keywords: Legacy software, software, Clustering, architecture recovery, entity representation, BoSS model

1 Introduction

Software systems must evolve to meet new requirements (i.e., domain business or technical requirements) or over time it becomes less satisfactory [3]. Reverse engineering with its program understanding tools is the key of success of any process of software evolution. Reverse engineering is an important theme in the software evolution (i.e., maintenance) domain. The aims of reverse engineering are to produce higher-level, more abstract, software models from the source code. Reverse engineering is needed when an understanding of the architecture and behaviour of software system is needed, where only the reliable information about the software system is the code source, due to the unavailability of software documents (e.g. design documents) or are inconsistent with respect to source code, which mean that had not been updated when source code is updated, this software system are commonly denoted as legacy software systems. Recovering the legacy software architecture is one of the typical tasks that practitioners are involved in, when evolving a software system. Software Architecture recovery can be casted as software clustering problem [1][2]. Software clustering has received a lot of interest of researchers in the software evolution domain , software clustering is taken primarily as technique for understanding legacy software systems. The aims of software **clustering** process is to divide legacy software to a set of subsystems, each

subsystem is a group of entities that implement a specific topic (i.e., functionality, concept). The result of the process is a set of cluster, thus entities within a cluster have similar characteristics or features, and are dissimilar from entities in other clusters. One of the three fundamental questions to be answered when we are involved in a clustering activity is [3]: What are the entities to be clustered? And how are described. For software architecture recovery, Entities are class, methods, files, procedures, functions, packages and so on, where the representation is carried by choosing set features that represent the entity. In [12], Anquetil et al, confirm the importance of a proper description scheme of the entities being clustered, A good description schemes (i.e. representation or model) of the entities to be clustered is done by choosing a set of features that represent the entities. Selecting a good features along with good clustering algorithms is of paramount importance [12][5] for the effectiveness any process of software architecture recovery.
Software is developed to solve a domain specific problem and engineered as set of collaborating modules(i.e. class, functions, methods..) that work on a set of data to deliver a set of services to end users. Each service or functionality delivered to end users is built around a subset of the software data, often called a view. Due to the lack or the inconsistent of the legacy software documentation as explained before. One consistent source of information about data manipulated by the software is the source code this information is encoded in identifiers names. From data point of view, domain ontologies [22] can be taken as conceptual model of some domain where data are grouped into interlinked concepts.

Traditionally all approaches proposed in the software clustering domain are bottom-up approach , exploiting the source code of the legacy software (formal or informal information) to build representation of entities to be clustered, to the best of our knowledge a top-down approach that take an extern source of knowledge to derive features and guide the clustering process have never been explored .

in [6] Anquetil et al, give two argument on why this approach is a difficult which make it unpopular, the two arguments are, the first one is the absence of domain knowledge and the second is, each given solution will be domain specific . In nowadays, ontology [13] is fully axiomatized theory about the domain and a lot of ontologies are built for different domains [14]. Which mean that a great part of knowledge about the domain are encoded in domain ontologies, which eliminate the first argument stated in [6], this new fact can heavily easiest the top-down approach.

Taking this fact into account, this paper proposes the bag of similarity scores (i.e., BoSS) model, a new representation of software entities based on domain ontology. The key idea is that the semantic of an entity is determined by the concepts that it manipulates, this model is used in an agglomerative hierarchical clustering process of bank management software and shown a promising result. The model is well adapted for conceptual clustering and the recovering of legacy software design model (i.e. class model) where entities that work on the same concepts from ontology point of view are semantically similar and grouped together, with this model we can instantiate a hierarchical clustering process to recover the legacy software functionalities and browse this functionalities according to functional view, which

mean deriving set of entities that work on a given set of concepts, this set delimit the set of data that a functionality work on .

The structure of the remainder of this paper is as follows:

In Section 2, presents the related work. In Section 3 we explain what is ontology. In Section 4, we present the BoSS (Bag of Similarity Scores) model. In section 5, we report the promising results shown when using our model in agglomerative hierarchical clustering process on a medium sized c software and Conclusion and future work is presented in section 6.

2 Related Works

Software architecture recovery is one typical task of the program understanding activities. The most proposed approaches formulate the problem as clustering problem. For a software clustering process to be effective, it is important to choose entities and features carefully. Entities are usually files, functions, classes or methods. Many features can be used to describe an entity but it is important to select the good features. In the literature features are divided into two classes:

2.1 Structural Based Features

The features used to describe entities are references from an entity to other program components, these features are commonly called formal features [6], a feature is formal if it impacts the behaviour of the software. Formal features are extracted from the code e.g., functions called by an entity, global variables, user defined types or macros referred to by an entity and files included by an entity. In this category Wiggerts [4] survey and analyse of the most used clustering algorithms in the software clustering domain. Sartipi and Kontogiannis [15] propose to recover the cohesive subsystems , a process composed of three phases, first phase relations between C programs are extracted then these relationships are used to build an attributed relational graph, while in the third phase the graph is manually or automatically partitioned using data mining techniques. In [16] The Bunch clustering tool is presented, source code is represented as graph and several heuristics are used to navigate through the search space of all possible graph partitions, a proposed fitness function is used to measure the quality of each partition . In [1] Maqbool et al, survey hierarchical clustering research in the context of software architecture recovery and remodularization and present an analysis of two proposed clustering algorithms, they report the experimental assessment on some large leagacy software . The use of genetic algorithms in software clustering is reported in [17]. In [6] the authors study three aspects of the clustering activity: Abstract descriptions chosen for the entities to cluster, metrics computing coupling between the entities and clustering algorithms. The result reports the importance of a proper description scheme of the entities being clustered.

2.2 Content Based

Derived from the lexicon contents of the entity [6] e.g. identifiers names and comments. A feature related to the lexicon of an entity is called informal feature [6]. In this category, in [20] Corazza et al, investigate the effectiveness of exploiting

lexical information for software system clustering. The investigation has been conducted on a dataset of 13 open source Java software systems, the paper provide a good revue of the most proposed approaches in the software clustering domain. Kuhn et al. in [18] use Latent semantic indexing techniques approach to group software artifacts. The work Scanniello et al in [28] also fall in this category, they propose to automate the partitioning of a given software system into subsystems, first analyzes of software entities is done and then they use Latent Semantic Indexing to compute dissimilarity between these entities. Finally, software entities are grouped using iteratively the k-means clustering algorithm. An empirical experiment is conducted on three open source software systems implemented using the programming languages Java and C/C++. For the approaches that use combination of structural based and content based features, we report the work of Adritsos and Tzerpos in [2] where LIMBO, a hierarchical algorithm for software clusteringis presented, in this approach formal and informal features are taken to reduce the complexity of a software system by decomposing it into clusters. In [19], the authors investigate the combined use of semantic and structural information of programs to support the comprehension tasks involved in the maintenance and reengineering of software systems. Semantic refers to the domain specific issues (both problem and development domains) of a software system where structural, refers to issues such as the actual syntactic structure of the program along with the control and data flow that it represents. Components are clustered togother according to Latent Semantic Indexing tool.

3 What is Ontology

Ontology play an important role in all software engineering phases [22] and are used in many fields: knowledge representation, knowledge engineering, qualitative modeling, language engineering, database design, information retrieval and extraction, and knowledge management and organization [22]. Many definitions of the concept of ontology can be found in the Artificial Intelligence domain and in computing in general. The most widely cited one is: “Ontology is a specification of a conceptualization” [21]. Uschold in [4], state that “ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms”, from this the main ontology component are concepts and relations that hold between concepts . In this paper we are interested in concepts that exist in a given domain and their description in terms of attributes , this concepts are derived from the domain ontology. As an example, from the domain ontology of the bank management, we can extract account as conept and identifier, name,amount as attributes . For more detail see[14].

4 The Bag of Similarity Scores Model

The key idea is that in the software domain, software are engineered as a set of collaborating entities that work on a set of data represented by meaningful programming construct (i.e. variable name identifiers, records name identifiers, comments) in the source code level and stands to some concept in solution domain. The semantic of an entity is determined by the concepts that manipulates, so, in the software domain, entities that work on the same set of concepts are conceptually similar.

Our aims is to build a conceptual representation of entities, that serves as basis for computing the conceptual similarity(i.e. semantic) to achieve a conceptual clustering and browsing of legacy software. Assuming that a good source code naming strategy is followed in the implementation phase, and the presence of the domain ontology, given an entity E, a concept c and a similarity measure, we score, according to a similarity metric, the relevance of the concept c to the entity E .

The proposed conceptual representation BoSS (i.e., Bag of similarity Scores) model is a variant of vector space model and provide conceptual representation of an entity. Under this model, entities are represented as real vectors ***d*** indexed over a fixed vocabulary ***V*** . The vocabulary consists of the domain ontology concepts, this concept is numbered from 1 to N and N is total number of concepts in the domain ontology. If we have N concepts in the domain ontology, an entity E_i is represented by an N-dimensional vector and write as $d_j = (w_{1,j}, w_{2,j}, ..., w_{Nj})$, the value w_{ij}/ j<=N is the similarity score between entity ***i*** and the concept ***J*** of the domain ontology.

4.1 Calculating the Similarity Score between an Entity and a Concept

To find similarity between entities, various similarity or distance measures have been used. In our context, in order to score each entity to different concepts, an appropriate similarity or distance measure is needed. The similarity measure computes the similarity between two entities based on the scores of their features (i.e., features selected to represent the entities). A similarity measures always yield a value in the interval [0..1], being close to 1, mean that the two entities are more similar. Often distance measures are commonly used to measures dissimilarity, in this case similarity measure is computed as follow: sim(e1,e2)=1-distance(e1,e2), distances most used are Euclidian distance and Manhattan distance [7]. In order to understand the proposed similarity measures we use an example of two entities represented by a set of features f1,f2,f3,f4, see Table 1, this table is the features matrix table of two entities E1 and E2, where 1 denote that the entity possess the feature, 0 denote the absence of feature in the entity .

	F1	F2	F3	F4
E1	1	0	1	0
E2	1	1	0	1

Table 1. Features matrix for E1 and E2

Suppose that a is the number of features that are present ‘1’ in both entities E1 and E2 , b represents features that are present in E1 but absent in E2 , c represents features that are not present in E1 and present in E2 , and d represents the number of features that are absent ‘0’ in both entities. n = a+b+c+d is the total number of features. The table 2 summarize the set of the popular similarity measures for binary features for details see[8] . In the literature a,b,c and d are calculated using table 3.

Similarity name	Mathematical representation
Jaccard	a/(a + b + c)
Euclide distance	√(b+c)
Cosine distance	a/(√(a+b)(a+c)) 2

Table 2. Some well know similarity measures for binary features.

		E1	
E2		1(presence)	0 abscence
	1(presence	a	b
	0 abscence	c	d

Table 3. Contingency table used to calculate a,b,c and d

For non binary features table 4 list some well know similarity measures, where Ma is the sum of features that are present '1' in both entities E1 and E2 , Mb represents sum of features that are present in E1 but absent in E2 , Mc represents sum of features that are not present in E1 and present in E2 .

Ellenberg	0.5*Ma/(0,5 *Ma +Mb +Mc)
Unbiased Ellenberg	0,5 *Ma/(0,5 *Ma + b + c)

Table 4. Some well know similarity measures for non-binary features.

Our approach is to use the cosine metric to score the similarity between an entity E and an ontology concept C , the following section gives the detail of the approach .

A cosine approach to Score the similarity between an Entity and an ontology concept

Our idea is to look to a concept as query and entity as textual document and to use one of the IR (i.e., information retrieval) tool [7](e.g. bag of terms, latent semantic indexing) to index the document and to score the similarity between the entity and the concept, in these model the cosine similarity [7] are the most used metric to score similarity between a query and document . The steps need to concretise this idea are:

Query Formulation.

Each concept in the ontology is mapped to query. This query is the union of the all attributes concepts and its names.

Corpus Creation.

Source code is analysed and entities (i.e. procedures and functions) composing the legacy software are extracted to be used as documents (i.e. each module is a document), each future document will be composed only from comments and identifiers names (the rest is discarded) and a corpus is created as set of documents each document represent an entity composing the legacy software.

Indexing.

Before this task start each document (i.e. entity) is pre-processed so identifiers terms are split and stemmed then the corpus is indexed using the vector space model (other IR technique can be used) and a vector is generated for each document. In this model In this model document (dj) and queries (q) are represented as a vector of terms: $d_j = (w_{1,j}, w_{2,j}, ..., w_{tj})$, $q = (w_{1,q}, w_{2,q}, ..., w_{t,q})$ where w is the weight attributed to term in the document, absence of term in document is denoted by the weight 0. tf-idf (term frequency-inverse document frequency)is the most common approach used to compute this weight [29],where the term specific weights in the document vectors are products of local and global parameters. The weight vector for document d is $V_d = [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T$, where $w_{t,d} = tf_{t,d} \cdot \frac{|D|}{||(d' \in D | t \in d')||}$ and $tf_{t,d}$ is term frequency of term t in document d (a local parameter) , $\log \frac{|D|}{||(d' \in D | t \in d')||}$ is inverse document frequency (a global parameter). | D | is the total number of documents in the document set, $|(d' \in D | t \in d')|$ is the number of documents containing the term t.

Computing the weight between entity E and concept C.

We calculate the similarity measure for query representing a concept C and the entity E composing the corpus using the cosine similarity [7]. Cosine similarity is the common used similarity measure to calculate the relevance ranking of document to a query [29] . Using the cosine, the similarity between document dj and query q can be calculated as: $sim(d_{j,q}) = \frac{d_{j,q}}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$

5 Case Study

In order to evaluate the effectiveness of our model, we setup the following process, see fig.1, to conduct a hierarchical clustering on a medium-sized c bank management legacy software, composed of 4 files downloaded from http://www.planet-source-code.com. In this process entities to be clustered are functions or procedures composing the software and represented using the BoSS model.

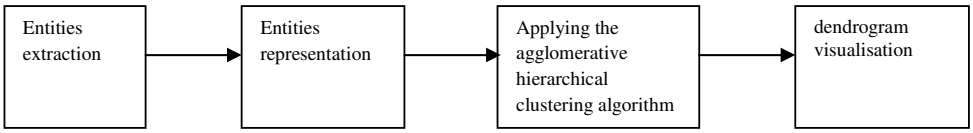


Fig. 1. The architecture recovery process

5.1 Entities Extraction

In this step, The source code is parsed and all functions and procedures are extracted and used as entities composing the legacy software, we use the SRCML tool [23] to accomplish this task .

5.2 Entities Representation

In this second step we extract all concepts composing the bank domain ontology. Then using the BoSS model we build the a vector that represent each entity as discussed before , in this experiment we adopt the second approach (the cosine similarity on textual representation of the entity) to compute the similarity weights . The result of this step is the feature entity matrix, where each row is an entity and each column is a concept. The cellule formed by the intersection of row i and column j is the similarity weight computed previously between entity ei and concept cj. This matrix serves as input for the agglomerative hierarchical clustering algorithm. Fig 2 shows the entity features matrix computed for the bank management software. Bank, transaction, customer, account and currency are domain ontology concepts. Attributes of each concept are not shown. But as example customer has name, surname, address as attributes.

	Bank	transaction	customer	account	currency
det_ac	0,11	0,00	0,00	0,00	0,00
chk_bank	0,10	0,00	0,00	0,00	0,00
view_transactionfra	0,08	0,00	0,00	0,02	0,00
view_transactiontoa	0,08	0,00	0,00	0,02	0,00
chk_name	0,08	0,00	0,00	0,03	0,00
…..					
view_transactionfras	0,05	0,00	0,00	0,00	0,00
deposit	0,00	0,03	0,00	0,04	0,00
withdraw	0,00	0,00	0,00	0,04	0,00
vldt_wamount	0,00	0,02	0,00	0,02	0,00
set_min_bal	0,00	0,00	0,00	0,00	0,00
view_transactions	0,00	0,00	0,00	0,00	0,00
vldt_damount	0,00	0,00	0,00	0,00	0,00

Fig. 2. The entity features matrix for bank software(partial).

5.3 Applying the Agglomerative Hierarchical Clustering Algorithm :

Agglomerative Hierarchical clustering algorithms use a bottom-up approach to decompose set of entities into clusters. The idea behind Agglomerative Hierarchical Clustering is a simple one. We start by assigning each object in a cluster of its own and then repeatedly, using a similarity or a distance metric, merge the closest pair of clusters until we end up with just one cluster containing everything. the legacy software is presented as a nested decomposition or hierarchy , the clustering decomposition of modules into sub modules is useful, especially for understanding

large system [10]. The result of the clustering algorithm can be visualized through a dendrogram, which is a tree like structure representing clusters formed at different stages of algorithm. A cut through the tree determines a set of clusters of the system. Since entities are represented using our BoSS model, conceptual browsing of the decompositions is allowable, which mean that developers can easily, derive which entities work on a given a set of concepts, this approach is very useful when searching entities that implement some functionality. The basic algorithm of Agglomerative Hierarchical clustering is given in Figure 3[11].

1. Assign each object to its own single-object cluster. Calculate the distance between each pair of clusters.
2. Choose the closest pair of clusters and merge them into a single cluster (so reducing the total number of clusters by one).
3. Calculate the distance between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all the objects are in a single cluster.

Fig. 3. The basic algorithm of Agglomerative Hierarchical clustering

The most used agglomerative hierarchical algorithms are Complete Linkage (CL), Single Linkage (SL), Weighted Average Linkage (WAL) and Unweighted Average Linkage (UWAL). these algorithms compute similarity between the newly formed cluster result of merging of two cluster and the other clusters/entities. These algorithms yield different scores of similarity for the same two entities [12]. A distance matrix that store similarity value between different clusters in each step have to be maintained and updated after each iteration, for the first iteration, this matrix maintain the similarity value between each pair of entities and is calculated from the entity features matrix.

5.4 Generating the Dendrogram :

The result of the clustering algorithm can be visualized through a dendrogram , which is a tree like structure representing clusters formed at different stages of algorithm. A cut through the tree determines a set of clusters of the system, fig 4 shows the dendrogram result of clustering the bank software using an agglomerative hierarchical clustering algorithm using the Euclidian distance and an average link strategy. Cutting the dendrogram at a certain cut point will result in distinct trees. Each distinct tree correspond to a cluster or a subsystem . In our case the result of cutting at some point, yield a set of cluster where each cluster is composed of set of entities that work on the same concepts. For example, a cut in the 0.09, dissimilarity level, produce three clusters that are show in fig 5. As an interpretation, the cluster 1 is subsystem that manage transactions, cluster2 is subsystem that manage accounts and cluster 3 is subsystem that manage currency . which are very consistent to some with the implemented functionality of the software.

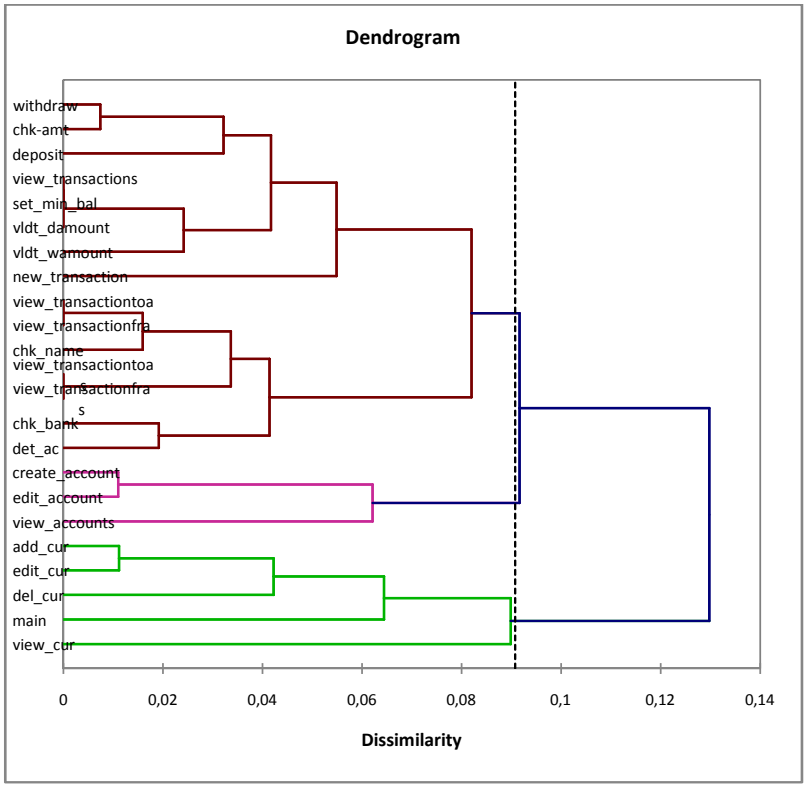


Fig. 4. The dendrogram result of applying the agglomerative hierarchical clustering algorithm.

Entities	Classe
det_ac ,chk_bank ,view_transactionfra, view_transactiontoa, chk_name, view_transactionfras, view_transactiontoas, new_transaction chk-amt, deposit, withdraw, vldt_wamount, set_min_bal, view_transactions, vldt_damount	1
view_accounts, edit_account, create_account,	2
del_cur, view_cur, edit_cur,add_cur, ,main	3

Fig. 5. The bank software decomposition.

5.5 Validation

In order to evaluate the decomposition result of applying our representation and the agglomerative clustering algorithms, [24]propose a metric like MoJo that measure the quality of the decomposition given an authoritative decomposition(i.e., an expert

decomposition). The MoJo distance is the minimum between the number of operations (i.e.move or join) needed to transform a partition A to B and number of operations needed to transform B to A(i.e.MoJo(A,B)=min(mno(A,B),mno(B,A))).

Where the quality $Q(M) = \left(1 - \frac{\text{Mojo(A,B)}}{\text{n}}\right) \times 100\%$, n is the number of entities to be clustered. The value of Q computed for our decomposition is Q=(1-(4/23))*100=82.7%, which is a very interesting value and demonstrate the effectiveness of our representation, this value can be interpreted as follow: Our decomposition is not distant from the expert one.

6 Conclusion and Future Work

Software evolution is about changing software to meet new needs, understanding software systems precede any change. The out of date or lack of documentation makes it a expensive activity [3]. Reverse engineering aims at abstracting from source code the relevant information about it. Software architecture recovery is one typical task of the reverse engineering activities, usually casted as software clustering problem where the most proposed algorithms in the clustering field are used in this process. Algorithms alone are not sufficient for the effectiveness of this process. Selecting a good representation of the entities to be clustered along with good clustering algorithms is of paramount importance [12] for the effectiveness of any process of software architecture recovery.

In this paper the Bag of Similarity Scores model is proposed, this new model aim at capturing the semantic of entity, So similar entities from the conceptual point of view are grouped together when using a clustering process to recover the legacy software architecture.

In order to demonstrate the effectiveness of our model, we have evaluated it by using it in clustering process of a medium-sized software, the reported result show that with this model, the clusters produced by the clustering process are not distant from the one identified by an expert.

In the future we plan to validate our model by varying the clustering algorithms and using large legacy software.

7 References

[1] O. Maqbool and H. A. Babri, "Hierarchical clustering for software architecture recovery," IEEE Trans. Software Eng., vol. 33, no. 11, pp. 759 – 780, November 2007.

[2] P. Andritsos and V. Tzerpos, "Information theoretic software clustering," IEEE Trans. Software Eng., vol. 31, no. 2, pp. 150 – 165, February 2005.

[3] Lehman, M.M.: On understanding laws, evolution and conservation in the large program life cycle. Systems and Software **1**(3) (1980) 213–221

[4] T. A. Wiggerts, "Using clustering algorithms in legacy systems remodularization," in WCRE '97: Proceedings of the Fourth Working Conference on Reverse Engineering (WCRE '97). Washington, DC, USA: IEEE Computer Society, 1997, p.33.

[5] Jain, A. K. and Dubes, R. C. 1988. *Algorithms for Clustering Data*, Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ

[6] Nicolas Anquetil, Timothy Lethbridge Extracting **Concepts** from File Names; a New File **Clustering** Criterion: In: *Proc. Int'l Conf. on Software Engineering*.IEEE (April 1998) , p. 84--93.

[7] Christopher D. Manning. Prabhakar Raghavan. Hinrich Schütze Introduction information retrieval : Cambridge University Press New York, NY, USA 2008 ISBN:0521865719 9780521865715

[8] S.-S. Chot, S.-H. Cha, and C. C. Tappert, "A survey of Binary similarity and distance measures," Journal of Systemics, Cybernetics and Informatics, vol. 8, no. 1, pp. 43 – 48, 2010

[9] Levenstein, I.V.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory (1966)

[10] R. Koschke and D. Simon, "Hierarchical Reflection Models," *In Proc. of the 10th Working Conf. Reverse Engineering*, 2003, pp. 36-45.

[11] David J. Hand, Heikki Mannila and Padhraic Smyth *principles of data mining:* ISBN-13: 978-0-262-08290-7,2010

[12] N. Anquetil, C. Fourier, and T. C. Lethbridge, "Experiments with hierarchical clustering algorithms as software remodularization methods," Proc. Working Conf. Reverse Eng., 1999.

[13] R.A. Falbo, C.S. Menezes, and A.R.C. Rocha, "A Systematic Approach for Building Ontologies", in Proceedings of the IBERAMIA'98, Lisbon, Portugal, 1998.

[14] Staab, Steffen; Studer, Rudi (Eds.) Handbook on Ontologies, 2nd ed., 2009, ISBN 978-3-540-70999-2 springer

[15] K. Sartipi and K. Kontogiannis, "A user-assisted approach to component clustering," Journal of Software Maintenance, vol. 15, no. 4, pp. 265– 295, 2003.

[16] B. S. Mitchell and S. Mancoridis, "On the automatic modularization of software systems using the bunch tool," IEEE Trans. Softw. Eng., vol. 32, no. 3, pp. 193–208, 2006.

[17] D. Doval, S. Mancoridis, and B. S. Mitchell, "Automatic clustering of software systems using a genetic algorithm," in STEP '99: Proceedings of the Software Technology and Engineering Practice. Washington, DC, USA: IEEE Computer Society, 1999, p. 73.

[18] A. Kuhn, S. Ducasse, and T. Gˆırba, "Semantic clustering: Identifying topics in source code," Information & Software Technology, vol. 49, no. 3, pp. 230–243, 2007.

[19] J. I. Maletic and A. Marcus, "Supporting program comprehension using semantic and structural information," in ICSE, 2001, pp. 103–112.

[20] Rashid Naseem_, Onaiza Maqbooly, Siraj Muhammad :Investigating the use of Lexical Information for Software System Clustering,In the 15th European Conference on Software Maintenance and Reengineering,2011,p35-44

[21] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Presented at the Padua workshop on Formal Ontology, March **1993.**

[22] Calero, Coral; Ruiz, Francisco; Piattini, Mario Ontologies for Software Engineering and Software Technology, ISBN 978-3-642-07087-7 springer 2006.

[23] srcml : www.sdml.info/projects/**srcml .**

[24] Tzerpos, V.; Holt, R.C.; Toronto Univ., Ont. MoJo: a distance metric for software clusterings: in Sixth Working Conference on Reverse Engineering **,1999.**

Cadre Méthodologique pour l'Urbanisation des Systèmes d'Information dans une Approche Orientée Services selon la Démarche PRAXEME

Amel Boussis, Fahima Nader
LMCS (Laboratoire de Méthodes de Conception de Systèmes)
E.S.I (Ecole nationale Supérieure d'Informatique) ex (I.N.I)
Alger, Algérie
amelboussis@yahoo.fr
f_nader@esi.dz

Résumé. Cet article aborde la problématique de l'urbanisation des systèmes d'information. Le développement du système d'information d'une entreprise est certainement une tâche complexe. D'où le choix pour les organisations d'opter pour une démarche d'urbanisation de leur SI. Dans nos travaux nous nous intéressons à une démarche complète de refonte de SI. Cette démarche est orientée services (Service Oriented Architecture) et basée sur une cartographie puis une orchestration des processus métiers relatifs au SI.

Mots clés: Urbanisation, Processus Métiers, SOA, Praxème

1 Introduction

Tout le monde est d'accord pour affirmer que le système d'information est aujourd'hui au centre du fonctionnement d'une entreprise ou d'une organisation. Son fonctionnement et son efficacité sont donc d'une extrême importance.

Un système d'information (SI) est un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures permettant d'acquérir, de traiter, de stocker, communiquer des informations dans les entreprises. Au cours de la vie de l'entreprise et de son évolution, le système d'information est amené à se modifier tant au niveau de sa structure que de son fonctionnement.

C'est en réponse à cette évolution permanente que l'idée d'urbanisme a été intégrée au sein des entreprises modernes. Le principe de base de l'urbanisme au niveau informatique est, au travers de règles et de principes fondamentaux, de suivre ces évolutions ainsi que ses impacts sur l'ensemble du système.

Le développement du système d'information d'une entreprise est certainement une tâche complexe. D'où le choix pour les organisations d'opter pour une démarche d'urbanisation de leur SI. Une telle démarche devient nécessaire lorsque l'organisation a accumulé un grand nombre de projets étalés sur plusieurs années. L'urbanisation du système d'information a pour but de répondre à plusieurs objectifs : le rationaliser, lui permettre d'être modulaire et le rendre plus innovant. Il s'agit néanmoins d'un concept visant à le simplifier, pour utiliser un terme de vulgarisation.

Le présent papier est structuré de la manière suivante : la deuxième section présente le principe d'urbanisation des SI ainsi qu'un état de l'art des principaux travaux réalisés. La troisième section décrit les éléments de base de la démarche d'urbanisation des SI. La quatrième section présente le contexte d'application. La cinquième section résume l'approche proposée, l'architecture du système est présentée en sixième section. Pour conclure, les orientations futures sont exposées

2 Principe d'Urbanisation des SI

Le Club Urba-EA[1] propose la définition suivante:

«Urbaniser, c'est organiser la transformation progressive et continue du système d'information visant à le simplifier, à optimiser sa valeur ajoutée et à le rendre plus réactif et flexible vis à vis des évolutions stratégiques de l'entreprise, tout en s'appuyant sur les opportunités technologiques du marché. L'urbanisme définit des règles ainsi qu'un cadre cohérent, stable et modulaire, auquel les différentes parties prenantes se réfèrent pour toute décision d'investissement dans le système d'information. »

La cartographie est l'ensemble des études et des opérations scientifiques, artistiques et techniques, intervenant à partir des résultats d'observations ou de l'exploitation d'une documentation, en vue de l'élaboration et de l'établissement de cartes, plans et autres modèles d'expression, ainsi que de leur utilisation [2].

« Les cartographies sont au cœur de la démarche à suivre pour un projet d'urbanisation de système d'information. On distingue même quatre types de cartographies (cartographie métier, cartographie fonctionnelle, cartographie applicative et cartographie technique) qui peuvent être réalisés pour décrire le système existant et/ou le système cible. Comme pour la cité, la cartographie d'un système d'information est à la fois :

- Scientifique: ne repose-t-elle pas sur un méta-modèle?
- Artistique : il s'agit là aussi de communiquer et, partant de là, l'esthétique est aussi un moyen de faciliter la communication.
- Technique: la réalisation s'appuie sur un certain nombre de techniques. » [1].

[1] www.urba-ea.org

L'urbanisation du SI a été étudiée par de nombreux auteurs [1][13][14]. Les travaux de ces auteurs complètent les travaux relatifs à l'architecture d'entreprise [15-19]. Tous ces auteurs utilisent des métaphores pour fonder la notion d'architecture d'entreprise et d'urbanisation des SI. En particulier, la métaphore de la cité est utilisée comme fondement de l'urbanisation des SI. Ainsi le SI est considéré comme la base de la cité de l'information peuplée par des applications qui doivent cohabiter et échanger conformément à un ensemble de protocoles et de règles de gouvernance. Ainsi, à l'instar d'une cité dont l'évolution progressive et harmonieuse permet l'intégration de différentes contraintes issues de l'environnement, un système d'information urbanisé est suffisamment agile pour évoluer et intégrer les changements organisationnels et technologiques nécessaires à la survie d'une organisation.

3 Eléments de Base de la Démarche d'Urbanisation du SI

3.1 Cartographie des processus

Avant de s'attacher à améliorer l'efficacité d'une organisation, il convient d'abord de la connaître, donc d'établir au préalable une cartographie des processus composants cet organisme de façon à en comprendre le fonctionnement. Selon [11] : « La cartographie des processus d'une entreprise ou d'une organisation est une façon graphique de restituer l'identification des processus et leur interaction. »

D'après [6], L'élaboration d'une cartographie des processus et de maîtrise des interfaces répond parfaitement aux exigences de la version 2000 de la norme ISO et elle permet d'apporter des solutions à de nombreuses questions. Elle est à la base de l'identification des processus importants, elle est utile pour préparer les programmes d'audits internes, elle aide à la mise en place des dispositifs de mesure et de surveillance des processus et elle peut servir à mettre en œuvre les programmes d'amélioration.

Pour établir la cartographie, il est utile de procéder comme suit : [7]

- Présenter la cartographie des processus de réalisation et la cartographie des processus de pilotage (ce sont les cartographies déjà établies).
- Etablir la cartographie des processus de support.
- Définir les flux d'interface entre ces trois cartographies : les liens externes entre les trois catégories de processus.

3.2 Modèles d'entreprise

L'entreprise est une structure complexe. Afin de mieux comprendre le fonctionnement, l'organisation, les ressources et les informations échangées dans une entreprise, on a besoin de représentations abstraites mais manipulables : des modèles.

Modéliser, c'est représenter la « réalité » d'un objet ou d'un système [3]. Un modèle d'entreprise sert à représenter différentes vues et dimensions de celle-ci [4]. Plusieurs outils, langages et standards pour modéliser certaines vues de l'entreprise sont apparus. Le langage UML est utilisé aujourd'hui pour modéliser certaines vues de l'entreprise. UML n'est pas une méthode dans la mesure où elle ne présente aucune démarche. A ce titre UML est un formalisme de modélisation objet.

3.3 Une architecture logique orientée services (SOA)

La notion de SOA (Service Oriented Architecture) définit un style d'architecture reposant sur l'assemblage de services proposés par les applications. Dans ce style d'architecture, les différents composants logiciels sont connectés par un couplage lâche (les services sont indépendants l'un de l'autre afin de pouvoir changer facilement l'ordre de leur orchestration pour former le processus).

Un « service », au sens de la SOA, est une connexion à une application, offrant l'accès à certaines de ses fonctionnalités. Les fonctions proposées par un service peuvent être des traitements, des recherches d'informations, etc. par exemple, une application de gestion de clientèle peut offrir un service retournant les coordonnées (adresse, tél,…) d'un client.

Dans une architecture SOA on s'intéresse néanmoins à plusieurs aspects différents de la conception d'un SI. Le projet PIM4SOA [8] définit quatre vues afin de définir une architecture SOA :

La vue informationnelle : l'information est liée aux messages et aux objets métiers échangés entre les services.

La vue processus : les processus décrivent l'enchaînement et la coordination des services en termes d'interactions et des flux de contrôle de processus.

La vue services : les services présentent une abstraction et une encapsulation des fonctionnalités fournies par une entité autonome.

La vue qualité de service (QoS) : on s'intéresse à d'autres aspects non fonctionnels tels que : la sécurité et la performance des services.

Ces vues concernent une architecture logique. L'implémentation d'une solution d'urbanisation nécessite de s'appuyer sur une plate-forme technique. Un modèle relatif à une architecture technique doit être utilisé. Cette architecture doit constituer un canevas technologique sur lequel projeter le modèle logique.

3.4 Une architecture technique orientée ESB

L'architecture orientée services (SOA) est implémentée à l'aide d'un bus ESB (Entreprise Service Bus). Cette technologie de plate-forme d'intégration est aujourd'hui développée dans le cadre de la communauté ObjectWeb au travers du projet Petals[2]. Dans [9] on définit un bus ESB comme une plate-forme permettant de gérer l'utilisation conjointe d'applications mises en commun par les partenaires.

[2] http:// petals.objectweb.org

Le pilotage des processus associés à ce partenariat peut par ailleurs être assuré par l'intermédiaire du bus et de ses outils de gestion de workflows. Le bus joue finalement le rôle de médiateur entre les partenaires en assurant les fonctions de connexions et de gestion des accès. L'ESB est principalement un outil d'échange asynchrone disposant d'interfaces standardisées ou intégrées. Il peut aussi offrir des services à valeur ajoutée, activés par des événements. Actuellement, l'enjeu est de construire des profiles UML en guise d'architecture technique.

4 Cas d'Application

La caisse nationale des assurances sociales (CNAS), a été Créée par le Décret exécutif n° 92-07 du 04 Janvier 1992 portant réorganisation du système de sécurité sociale.

La CNAS ambitionne de moderniser progressivement son S.I. En effet, toute la logistique matérielle, les moyens humains, les programmes de formation, les règles, procédures et règlements, en un mot tout le système d'information a été mobilisé pour garantir la qualité de la prestation fournie à l'encontre des populations d'assurés sociaux et de leurs ayants-droit, qui doit être l'essence même de l'existence d'une caisse d'assurance.

Il se trouve que le SI actuel de la CNAS, se caractérise par la disponibilité de certaines informations utiles, actualisées à travers le portail Web, mais ne renseigne pas sur ses activités métiers. Un cloisonnement fonctionnel ralentissant toute prise de décision a été constaté sur les applications existantes.

Diriger l'activité en se focalisant sur les processus métiers de bout en bout, suppose une approche transversale qui dépasse les frontières des départements. Ces processus impliquent plusieurs acteurs et systèmes, qui sont en réalité réparties entre différentes zones fonctionnelles, mais qui interagissent souvent dans les mêmes procédures appartenant à la chaine de valeur de l'entreprise. Ce mode de management est, ce que la CNAS veut entreprendre, un mode qui met en avant l'idée que la CNAS pourrait être une entreprise orientée service.

Notons l'existence de deux principaux macro processus métiers constituant les deux branches fonctionnelles de l'organisme à savoir :

- Le recouvrement via les cotisations de la population des employeurs.
- Les prestations par le remboursement de la population des assurés sociaux.

Ce dernier est divisé en deux processus métiers :

- Remboursement des frais médicaux.
- Remboursement des arrêts de travail.

Ces deux processus métiers représentent l'activité principale des structures de paiement de la CNAS. Ils comprennent les étapes suivantes : contrôle des droits aux prestations, liquidation du dossier, validation et paiement du dossier.

Cependant, repenser l'architecture du SI doit se faire de manière progressive. Le processus métier «remboursement des frais médicaux » a été pris comme processus métier pilote car toute l'organisation qui l'accompagne est un élément majeur dans le bon fonctionnement de la caisse. C'est aussi un indicateur fondamental de la qualité des prestations dispensées.

5 Notre Démarche d'Urbanisation

Nous adoptons la méthode PRAXEME[3] dans notre démarche d'urbanisation. Praxème est une méthodologie d'entreprise, qui se veut publique et open source, de conception ou refonte de SI, couvrant l'ensemble des aspects de l'entreprise de la stratégie au développement du logiciel.
Pour représenter l'entreprise et embrasser l'ensemble des angles d'appréciation, la méthode repose sur un schéma identifiant et articulant 8 aspects.

L'aspect ou facette est une vue du système, où le système est vu selon un type de préoccupation particulier. L'aspect, tout en étant une composante du système, a donc une nature relative: il est lié à un point de vue, un type de préoccupation, une spécialisation.

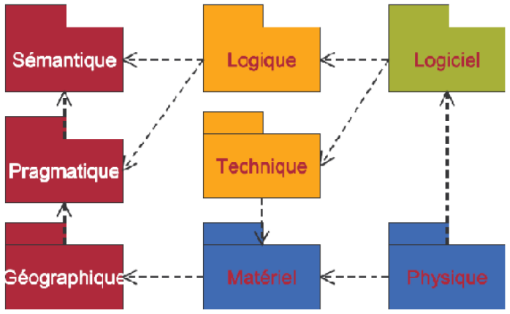


Fig. 1 : Topologie du Système Entreprise

5.1 Etape 1 : (Aspect Sémantique)

Dans cette étape, les objets et les concepts au cœur du système sont décrits. Nous exprimons dans cette phase tout ce qu'il y a de plus stable dans notre SI. On y retrouve la classe sémantique (classe d'objets), les propriétés informatives (attributs), et actives (opérations ou méthodes), tel que : la classe Assuré dans notre cas. Le modèle métier sémantique est élaboré en termes de packages, modélisé à travers les diagrammes de : domaine, classe, transition et d'états d'UML. Un référentiel de données est conçu, relatif aux métiers étudiés.

[3] www.praxeme.org

5.2 Etape 2 : (Aspect Pragmatique)

Au niveau de cette 2ème étape, les acteurs du SI sont délimités ainsi que leurs types et fonctions. La modélisation pragmatique va délimiter le style de gestion, la structure de commandement et d'opération en entités organisationnelles, les processus métiers et les profiles utilisateurs, tel que le processus métier étudié dans notre cas.

En d'autres termes, du point de vue de l'utilisation, il s'agit de définir les domaines fonctionnels et du point de vue de l'organisation, il s'agit d'identifier les processus métiers.

5.3 Etape 3 : (Aspect Logique)

Le résultat des deux étapes précédentes est projeté sur un modèle SOA. Ce dernier est compatible avec un profile UML pour SOA d'IBM [12]. Le modèle SOA est composé de trois couches :

- **La couche SOA métier** : est celle des services métiers composant le processus « Remboursement des frais médicaux »,
- **La couche logique SOA applicative** : constituée de bus de service ESB dont le registre de services permet aux services d'être publiés, recherchés et invoqués.
- **La couche SOA composant** : invoque les services composants concernés, qui exécutent au niveau des serveurs dans lesquels ils se trouvent, les méthodes implémentées par les objets regroupés dans les composants.

Il s'agit d'une transcription des descriptions sémantiques et pragmatiques, transcription guidée par les règles de structuration architecturale.

5.4 Etape 4 : (Aspect Technique)

Dans cette étape on passe d'un modèle logique SOA à un modèle technique ESB. Pour cela on a besoin de définir un profile UML pour USB. Ce profile va contenir les éléments de base qui définissent un ESB tel que : « Bus », « Message XML », « Annuaire de services », « Adresse d'un service », etc. l'aspect matériel est étroitement lié à l'aspect technique car le choix de l'architecture matérielle utilisée est fixé à ce niveau.

5.5 Etape 5 : (Aspect Physique)

Dans cette étape on s'intéresse particulièrement à une transformation d'un modèle à un texte. Le but est de générer à partir du modèle ESB défini des représentations textuelles nécessaires au paramétrage et à l'implémentation de la plateforme ESB. On distingue une représentation BPEL (Business Process Execution Language) pour l'orchestration des services, une représentation XML pour présenter les messages échangés entre les web services et leur structure, une représentation WSDL (Web Services Description Language) pour décrire les web services, leurs adresses, etc.

Et de ce fait, fixer les règles de localisation des composants logiciels en l'occurrence les services web sur l'architecture matérielle et couvrant ainsi l'aspect géographique.

5 Architecture du Système

L'architecture du système d'information proposée ci-dessous (Fig.2) est une architecture en couches, conforme à l'architecture de référence en SOA d'IBM [12]. Elle permet de distinguer le métier de l'applicatif, dont l'architecture applicative est une architecture SOA, implémentée par le bus de service (ESB) et les services web packagés.

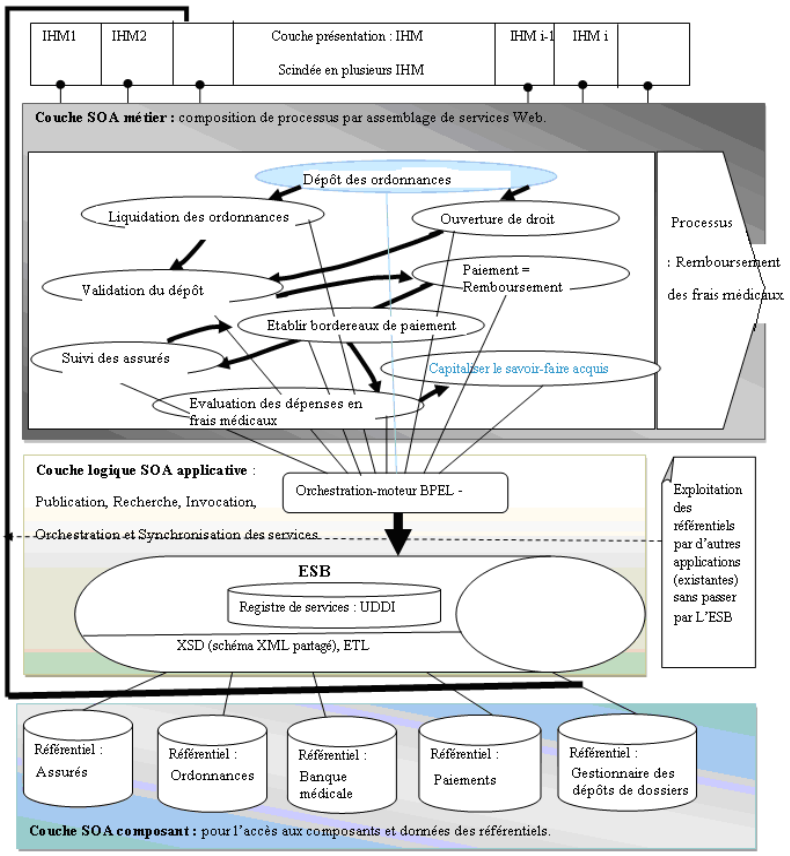


Fig. 2. Architecture du système d'information à base de SOA.

La couche métier est celle des services métiers composant notre processus « remboursement des frais médicaux », les services métiers invoquent les services déployés dans le registre de services via une orchestration d'appels par le biais du moteur de processus « BPEL».

Une fois un service est appelé, il invoque à son tour les services composants concernés, qui exécutent au niveau des serveurs dans lesquels se trouvent, les méthodes implémentées par les objets regroupés dans les composants. L'exécution est totalement distribuée, la synchronisation est assurée par le bus de services qui est doté de la fonction de transport.

L'architecture proposée (Fig.2), apporte une amélioration considérable aux systèmes d'information de la caisse pour supporter le processus métier relatif au remboursement en :

- *favorisant son agilité:* car elle permet de structurer d'une manière dynamique le système d'information relatif. En effet, toute modification (ajouter ou enlever une étape, modifier l'ordre de communication des étapes) qui peut être apportée sur le processus, à l'avenir, est facilement maitrisable, grâce au principe de couplage faible associé aux services et la séparation entre le métier et l'applicatif.
- *améliorant son accessibilité* : faciliter la communication entre la caisse et d'autres entreprises ou partenaires tels que les professionnels de santé, grâce à l'utilisation de services Web. Et permet ainsi d'assurer la pérennité de la solution métier mise en œuvre.

De ce fait, la solution offre un gain de temps appréciable pour toute extension (i e. extension du processus lui-même ou généralisation de SOA sur d'autres processus)

6 Conclusion et Perspectives

Nous avons présenté dans cet article une démarche d'urbanisation d'un système d'information. Cette démarche est orientée « services » (SOA). Nous nous sommes inspirés de la méthodologie Praxème avec ses différents aspects pour arrêter les étapes relatives à notre approche. Ceci a été validé par l'application de la démarche au cas pratique du système d'information relatif au domaine des prestations en assurances sociales, plus précisément au processus métier : remboursement des frais médicaux aux assurés sociaux.

Ce travail du point de vue méthodologique, constituera une première pierre dans l'édifice de la mise en ouvre progressive d'une entreprises orientée services, telle que la CNAS.

Les apports d'application de SOA dans cette caisse, se sont manifestés explicitement via la flexibilité de ses processus métiers et l'ouverture de son système d'information sur le monde extérieur à travers les services web exposés, afin qu'il puisse être interactif sur le web et permettre ainsi aux entreprises, employeurs et les assurés sociaux d'être contributeurs.

Dans un futur proche, nous comptons généraliser l'application de SOA dans la caisse, en la propageant sur d'autres processus, suivant la démarche présentée. Enfin le rapprochement des deux approches SOA et Web 2.0, pourrait être un prolongement utile et donnera une forte valeur ajoutée pour tout système d'information orientés services.

7 Références

1. C.Longépé, « Le projet d'urbanisation du système d'information », 3e édition, Dunod 2006.

2. F.Choay, P. Merlin, « Dictionnaire de l'urbanisme et de l'aménagement », PUF 1996.

3. Actes de l'école de printemps de modélisation d'entreprise. GDR-MACS – Albi. 28-30 Mai 2002

4. H.Pingaud. Modélisation du système d'information et de l'entreprise : convergence ou complémentarité ? Actes de la 2ème Ecole de Modélisation d'Entreprise, Nimes, 2004.

5. Ahmed ALAMI's WEBLOG, L'Architecture Orientée Services et J2EE. La modélisation de la démarche d'urbanisation. *http://soaj2ee.blogspirit.com/urbanisme* consulté en Décembre 2010

6. Y. Mougin, « *La cartographie des processus* » 2éme édition, Éditions d'Organisation, 2004.

7. D. Bouami, F.Ouzennou, « *Approche processus-Identification des processus* », École d'Ingénieurs Mohammadia Rabat, CPI'2007 – Rabat, Maroc.

8. G. Benguria, X. Larrucea, B. Elveseater, T. Neple, A. Beardsmore, M. Friess. A Platform Independent Model for Service Oriented Architectures. 2ème conference internationale sur l'intéropérabilité, IESA'06- Bordeau- Mars 2006.

9. D. A. Chappell. Entreprise Service Bus. O'Reilly, 2004.

10. http://fr.wikipedia.org/wiki/Architecture_orient%C3%A9e_services consulté en Janvier 2011.

11. H.Brandenburg , JP.Wojtyna, « *L'approche processus mode d'emploi* » Éditions d'Organisation, 2003.

12. Dodani, M.H: SOA 2006 : State of the Art, journal of object technology, vol 5, No. 8, November-December 2006.

13. Jean G., Urbanisation du business et des SI, Paris, Editions Hermès, 2000.

14. Sassoon J., *Urbanisation des SI*, Paris, Editions Hermès, 1998.

15. Zachman, J.A., A Framework for Information Systems Architecture, IBM Systemps Journal, Vol. 26, 1987, pp. 276-292.

16. Zachman, J.A., and Sowa, J., Extending and Framework for IS Architecture, IBM Systemps Journal, vol. 31, 1992, pp. 590-616.

17. Fayad, M., Henry D., Bougali, D., Enterprises Frameworks, Software Practice and Experience, vol. 32, 2002, pp. 735-786.

18. Kaisler, S.H., Armour F., Valivullah M., Enterprise Architecting : Critical P, in the Proceedings of the 38th HICSS, Hawaï, IEEE, 2005.

19. Maier, M.W., Rechtin, E., The Art of Systems Architecturing, CRC Press, 2000.

A new RTT based mechanism against Wormhole attack in wireless sensor networks

BRAHIM Nacéra[1,1] and KECHAR Bouabdallah[1]

[1]Computer science department, sciences faculty, Oran university, Oran, Algeria
{brna2000, bkechar2000}@yahoo.fr

Abstract. As the applications of wireless sensor networks (WSNs) diversify, providing secure communication is a critical requirement. Wormhole attack is one of the serious attacks, which forms a serious threat in the networks especially against ad-hoc wireless routing protocols.
In this paper, we describe Wormhole attack, its establishment and its impact on routing protocol such as AODV. Wormhole attack is difficult to detect and prevent, as it can work without need to compromise nodes or break the encryption system used in the network. We present some wormhole attack detection approaches that are divided into two classes, centralized approaches that require a central entity and decentralized ones. The first approaches' class contains Statistical Wormhole Detection, which detects the increase in the number of the neighbors and Wormhole Detection with multi-dimensional Scaling, which uses the network visualization and finally Detection and elimination of Wormhole by the graph theoretical approach. The second one contains packet leashes approach which limits packet's life time, and time of flight approach [4], which is similar to the "temporal packet leashes"[2].
Finally, we propose a new Round Trip Time (RTT) based mechanism for detecting wormhole attacks in wireless multi-hop networks during route setup procedure (Route Discovery). Unlike many techniques proposed in literature, it does not use any special location hardware as GPS or global clock synchronization.

Keywords: Wormhole attack, wireless sensor network, security, AODV

1 Introduction

A wireless ad hoc sensor network [6] [9] consists of a large number of autonomous sensor nodes that monitor the environment across a geographical area and a few base stations that collect the sensor readings. Its nodes can make local processing of data which they get. Sensors have wireless communication capability without need to any pre-existent infrastructure. They are usually battery powered and limited in computing and communication resources, they are typically used out in an open, uncontrolled environment, often in hostile territories [9].

Like all other computer systems, Wireless sensor networks are vulnerable. Security mechanisms of wired networks and even ad hoc networks cannot be applied on this kind of networks because of the sever resources constraints due to their limitations on energy [6], processing capacity and communication band width.
In this paper we try to describe one of the most dangerous attacks on wireless sensor network and present different methods proposed to defend it, included our proposed method.

2 Wormhole attack

Wormhole is an out of band link with low latency between two remote nodes. The attacker establishes initially a direct low-latency communication link, called Wormhole, between two legitimate nodes, the origin and destination points, that have not a one hop link between them. So the goal of attacker is to convince two distant nodes that are direct neighbors [1].

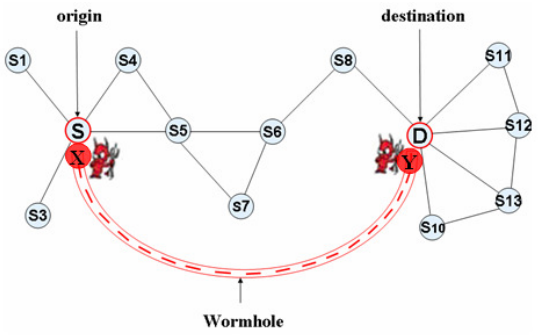


Fig. 1. Wormhole attack.

Then he listens to traffic illegally at one end of the link, which is the origin point, and forwards it through the tunnel to the other end, which is the destination point without adding its address in the packet's header. This has the result of creating an inexistent link between the two legitimate remote nodes under the power of the attacker, which is virtually invisible.

The Wormhole link can be established using a wired connection, optic, a long-range, out-of-band wireless directional transmission [9] [10].
Wormhole devices and links deployed by the attacker are not part of the network, so they don't require a valid ID of the network.
By this attack the adversary seeks traffic forwarding and not its content, therefore he does not need to break the cryptographic system used by the network's nodes[8].
Then the properties of integrity, confidentiality and authentication are preserved. That makes the attack Wormhole invisible to upper layers.

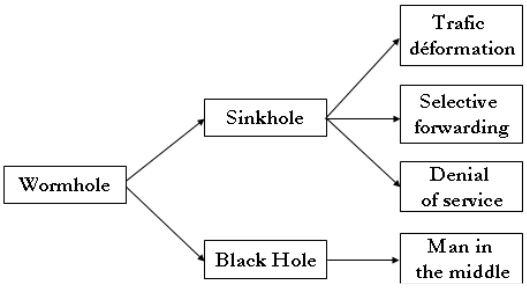


Fig. 2. Attacks due to Wormhole attack

By successfully Wormhole Attack, the malicious node will have total control of traffic, so it will benefit from this path to execute Sinkhole attack, which seeks to attract to itself the data from all its neighbours. Thus, it can apply "Traffic deformation" by manipulating or corrupting the contents of packets or blocking certain types of traffic "selective forwarding". By a switch on both ON and OFF states, the attacker could cause a broke of route, which leads to a DoS attack "Denial of Service"[10]. The attacker can launch Black Hole attack by announcing as having the shortest route. And thereafter, it can monitor and analyze traffic from all its neighbors "Man in the Middle".

3 How is AODV protocol vulnerable

3.1 Impact of Wormhole attack on the Neighborhood discovery (HELLO)

When a protocol reactive like AODV is used, nodes periodically broadcast Hello messages indicating their presence in the network.
See figure 1, if node 'S' broadcasts a Hello message then the malicious node, which is in its neighborhood, intercept it and carries it through the Wormhole tunnel to the other end where is his accomplice which rebroadcasts it. That Hello arrives at the node 'D' which believes that 'S' is its direct neighbor.

3.2 Impact of Wormhole attack on the Rout Request (RREQ)

If a node in the network wishes to communicate with another, it floods the network with a RREQ message requesting a route to its destination, which is the shortest path. One of the RREQ's copies passes through the Wormhole, and arriving before the others it obligates the destination to select its path (the path that the RREQ travelled to reach the destination).
In the case of AODV protocol the intermediate nodes update, the entries of the previous nodes, in their routing table. These entries are kept while the route is valid, during this period if this node wishes to communicate with one of the nodes

correspondents to one of these entries then it will be obligated to pass through Wormhole as the shortest path. And as the effect of wormhole propagates on the network

3.3 Impact of Wormhole attack on the Route Reply (RREP)

The RREP is formed just after receiving the first RREQ arrived at the destination. If RREQ reached the destination through a Wormhole link then the last one is included in the path of the RREP.
The passage of the RREP by the intermediate nodes encourages them to updates entries correspondent to the source or destination node passing by the Wormhole link.

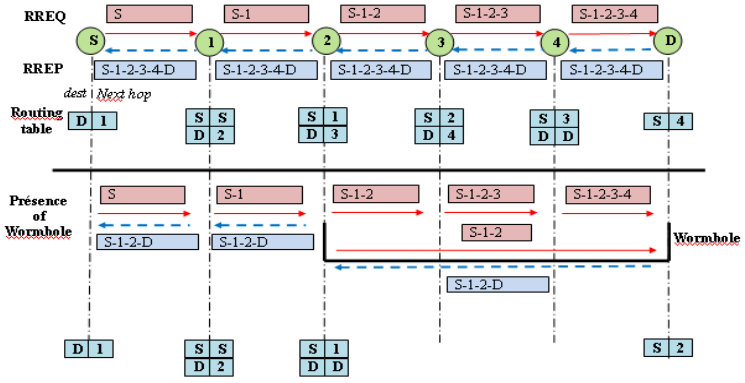


Fig. 3. Impact of Wormhole attack on RREQ and RREP of AODV protocol

4 Wormhole attack detection

Cryptographic mechanisms witch ensure confidentiality, integrity, authentication, and non-repudiation cannot detect or prevent the Wormhole attack. Because last one is not set up to create or modify traffic, but just to replay the packets generated by the legitimate nodes in another part of the network. As soon as this attack was born and classified as a dangerous attack, many researchers are working to reach a solution. All the solutions are based on this principle: «To detect Wormhole attack, some mechanism is required to ensure that any transmission received by a node s is really from a valid neighbor of s and which is situated in the covered area of s. » [3]. Several approaches were proposed, they are divided in two classes:

4.1 Centralized approaches

In this type of approaches, the collected neighbourhood information of every node are sent to a central entity, which takes care to build the total model of the network and try then to detect the inconsistencies which indicate the presence of Wormhole.

Statistical Wormhole Detection. Suppose a network consists of n static nodes placed randomly in a flat area of size s with fixed and same communication range r for all nodes [3]. This mechanism detects the increase in the number of the neighbours of the sensors, which is due to the new links created by the wormhole in the network. The probability of two nodes being neighbours is:

$$q = \frac{r^2\pi}{s}. \qquad (1)$$

The probability that a node have exactly k neighbours is:

$$p(k) = \frac{(n-1)}{k}.q^k.(1-q)^{n-1-k}. \qquad (2)$$

Where $0 \leq k \lhd n$

Let us partition the set {0, 1, 2 . . .}
Where $(B_1 \cup B_2 \cup \cdots \cup B_m) = \{0, 1, 2 \ldots\}$

Such that $(e(i) = n \sum_{k \in B_i} p(k)) > 5.$

Then $\chi 2$ is computed using the following formula:

$$x^2 = \sum_{\forall i} \frac{r(i)-e(i)}{e(i)}. \qquad (3)$$

Where r(i) is the real number of nodes with number of neighbours in Bi. If $\chi 2$ is below the threshold that corresponds to a given significance level, then the hypothesis is accepted, and no wormhole is indicated. Otherwise the hypothesis is rejected, and a wormhole is indicated [5]. The major disadvantage of this method is that it detects the presence of a link Wormhole without locate it (and specify compromised nodes) [3].

Wormhole Detection with multi-dimensional Scaling. This approach uses the network visualisation to detect wormhole attack in stationeries wireless sensor networks [3]. It is based on augmenting connectivity information and distances estimations between neighbour nodes (witch allow to detect that the neighbour node is not situated in the communication area covered by the current node). Every node measures the distance which separates it from each of his neighbours

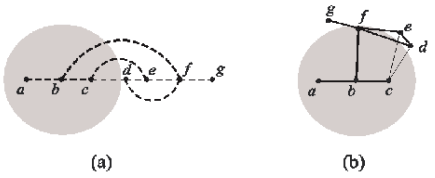


Fig. 4. Wormhole detection with multi dimensional scaling

Part (a) shows the real placement of the network's nodes. The circle represents the communication range of node "b ". The link between "f" and "b" is a Wormhole.

Part (b) shows the virtual plan reconstituted from the measures of inaccurate distance of the neighbour nodes.

The inconvenience of this mechanism is that it is not applicable to the mobile wireless sensor networks.

Detection and elimination of Wormhole by the graph theoretical . Each ad hoc network can be represented by a geometric graph defined as follows [1]:

G(V, r) Where V: summits Set, and $V \subset Rd$;

r: communication range, And $\|i-j\| \leq r / \; i, j \in V$ ...

Entries in the connectivity matrix are noted by: $e(i,j) = \begin{cases} 1 \; si \; \|i-j\| \leq r \\ 0 \; si \; \|i-j\| > r \end{cases}$ (4)

The existence of the Wormhole attack violates this model, because two nodes which are not neighbours can establish a path of a single jump using of the Wormhole link. So in logical graph $\tilde{G}(V, E_{\tilde{G}})$ we can find $e(i,j) = 1 \; pour \; \|i-j\| > r$

Let us note by: C_{X} : The connectivity matrix of the graph X;

$C_X(i)$: The connectivity vector corresponding to a node i;

$C_X(i,j)$: The connectivity of nodes i and j.

The detection of the Wormhole links established by the attacker is made by a XOR operation between the matrix of connectivity of the geometrical graph C_G and that of the logical graph $C_{\tilde{G}}$. Wormhole link is detected as follow:

If $(C_G \oplus C_{\tilde{G}})(i,j) = 1$ $\Rightarrow$ The link between node i and node j is a Wormhole.

4.2 Decentralized approaches

The advantage of this approach is that it does not require the presence of a central entity in the network. Every node builds the model of its neighborhood by using the data collected locally.

Based on localization. This type of approaches is based on limitation of the distance that each packet can cross between every two successive nodes.

Geographical packet leashes is based on geographical information. To build geographic leashes, it is required to equip all nodes with GPS (Global Positioning System). At sending the sender add it position Ps, and the time of transmission of the first bit Ts, in the packet. The receiver then compares them with his position Pr and the time of receiving of the first bit Tr, by calculating the real distance which separates him from the sender as follows[2]:

$$d' \leq d + 2\Delta p + 2v_{Max}(tr - ts + \Delta t) \quad (5)$$

Where $d = \|\vec{p}_r - \vec{p}_s\|$;

Δp: $positioning \; \max error$;

Δt: $time \max error$;

v_{max}: $\max nodes \; speed$;

t_s, t_r: $transmission \; and \; reception \; time$.

Based on time. This type of approaches is used for limitation the distance to across between each two successive nodes by the limitation of the time of flight between them.

Temporal packet leashes are implemented with a packet expiration time. The building of the temporal leashes requires a narrow synchronization of the clocks of all the nodes where the difference between clocks of any two nodes of the network is Δt witch has to be known by all the nodes of the network and generally has to be on order of some microseconds or even hundreds of nanoseconds. During the sending, the sender inserts into the packet the transmission time of the first bit of the packet t_s. When the packet arrives at receiver, the last one compares t_s with reception time of the first bit t_r, So the receiver can calculate the distance crossed by the packet. The main inconvenience of this solution is that it requires a mechanism of synchronization between all the nodes because of the use of clock.

Time of flight [4] is similar to that of the "temporal packet leashes"[2], It is based on the estimation of **RTT** [1]. A packet is accepted if it verifies the following condition:

$$RTT < (2 * R)/v + \Delta t \qquad (6)$$

Where RTT: Round Trip Time;
V: Speed of light.
R: radio transmission range.
Δt : packet processing time.

The advantage that brings the use of the RTT is the elimination of the need of the synchronization required by temporal packet leashes because the node uses only its own clock.

5 Our contribution

In this section our wormhole detection mechanism will be discussed in detail.
The detection process is based on computing distance between every two successive nodes along the established path by *RTT* (Round Trip Time).
We consider the time between a node sending the RREQ & receiving RREP as RTT. Travel time (Tp) is considered as the time between a node receiving the RREQ & sending RREP. To avoid confusion in this paper we use RREQ, RREP respectively to refer to the Route Request, Route Reply. And we use Rrep and Rreq respectively to refer to RREP and RREQ receiving time.
Our mechanism consists of two phases: detection and isolation.

5.1 Detection phase

When a source node needs to transmit some data, it initiates a route discovery by broadcasting RREQ, and it saves its sending time Treq. When receiving RREQ, each Intermediate node on his turn forwards it and saves its sending time Treq[7]. When destination node is reached it generates a RREP. As it travel the reverse route to reach

[1] RTT: (Round Trip Time) which is the time which passes between the sending of a packet and the reception of its acquittal Ack.

its destination, each intermediate node 'i' saves its receiving Time Rrepi, and calculates RTT and Travel Time as:

$$RTTi = R_{Repi} - T_{Reqi} \quad (7)$$

$$TPi = T_{Repi} - R_{Reqi} \quad (8)$$

RTT includes the period of travel time which is the time for RREQ to reach destination and forward Rrep by node (i+1). The latter is the precedent node in the reverse route. $RTT_i = Dprop_i + TP_{(i+1)}$ (9)

Where: TP_i : Travel time calculated by node$_i$;

$Dprop_i$: Propagation delay calculated by node$_i$;

So propagation delay is calculated as : $Dprop_i = RTT_i - TP_{(i+1)}$ (10)

The distance between node i and node (i+1) is

$$d_{i(i+1)} = Dprop_i * \frac{C}{2}. \quad (11)$$

Where: $d_{i\ (i+1)}$: Distance between node$_i$ and node$_{(i+1)}$. And c is Light speed, ($c = 299\,792\,458$ m/s).

If $d_{i,(i+1)} > R$ then node$_{(i+1)}$ and node$_i$ are fake neighbors. This means that this message had passed by wormhole link to reach node$_i$. So last one modifies TP and wormhole fields of RREP and forwards it to node$_{(i+1)}$;

Otherwise node$_i$ modifies TP field of RREP and forward it to node$_{(i-1)}$;

5.2 isolation phase

If a link wormhole is detected by node$_i$ then this one:

- Sends a warning-hello to node$_{(i+1)}$ to inform it that they are fake neighbors;
- Marks node$_{(i+1)}$ in his neighbor list as fake neighbor;
- Marks RREP to indicate that it passed by a wormhole link and transmits it subsequently to the next node in the reverse route not to establish route but to detect other wormhole tunnels if any.

Receiving a warning-hello from the node$_i$, node$_{(i+1)}$ marks the node$_i$ as a false neighbor. This operation is used to prevent any subsequent direct communication between nodes i and (i+1).

Any intermediate node crossed by RREP tests the value of Wormhole field (which indicates the passage through a wormhole tunnel). if its value is TRUE then it removes the entry corresponding to the destination in its routing table.

When the source receives a RREP having passed through a wormhole link, then it:

- Deletes the route to destination D in its routing table;
- Drop RREP;
- Initiates another route discovery.

In route discovery initiated subsequently, all received RREQ from a false neighbor will be dropped. Which prevent any RREQ to pass through a wormhole tunnel. And like that the tunnel is avoided.

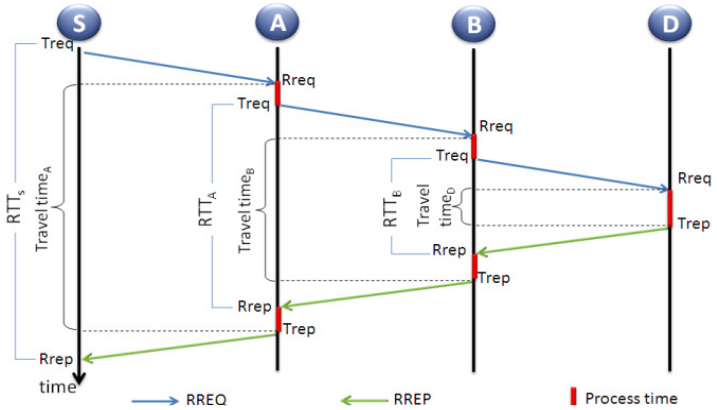


Fig. 5. Unfolding of route discovery in time.

As an example, consider Figure 5. To establish a route from S to D, a RREQ is broadcasted by S and forwarded by A,B,C to reach destination D. all nodes save sending Time of RREQ. Whereas: T_{Reqs},T_{ReqA},T_{ReqB} are times the nodes S, A, B forward RREQ. When receiving RREQ, D generates RREP and calculates its Travel Time TP_D :

$$TP_D = T_{RepD} - R_{ReqD}. \quad (12)$$

Adds it into RREP and sends it to its neighbor. When receiving RREP, node calculates RTT, propagation delay and distance respectively based on equations (7), (8), (10), and (11). If a wormhole link is established between nodes D and B Then the RREP passed by nodes D, B, A and finally it reaches the source node S. The table bellow illustrates intermediates nodes computing:

Node B	Node A	Node S source
$RTT_B = R_{RepB} - T_{ReqB}$ $Dprop_B = RTT_B - TP_D$ $TP_B = T_{RepB} - R_{ReqB}$ $d_{B,D} = Dprop_B * \frac{c}{2}$	$RTT_A = R_{RepA} - T_{ReqA}$ $Dprop_A = RTT_A - TP_B$ $TP_A = T_{RepA} - R_{ReqA}$ $d_{A,B} = Dprop_A * c/2$	$RTT_S = R_{RepS} - T_{ReqS}$ $Dprop_S = RTT_S - TP_A$ $d_{S,B} = Dprop_S * c/2$

Table 1. calculations performed by the nodes (RTT, Dprop, TP, d).

6 Simulation

In this section, the proposed mechanism is simulated using network simulator (ns2). In this experiment, the network includes **8** nodes deployed in a 1000 meters × 1000 meters field and the transmission range is defined 250 meters. There is no movement of nodes and the background traffic is generated randomly by a random generator provided by ns2. The CBR connection with 4 packets per second are created and the

size of the packet is 512 bytes. In the simulation, two legitimate nodes are compromised to establish Wormhole tunnel.

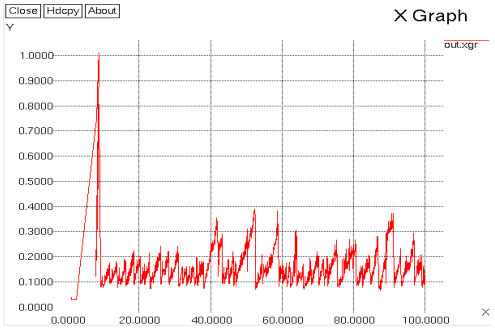


Fig. 6. End to end delay using AODV protocole.

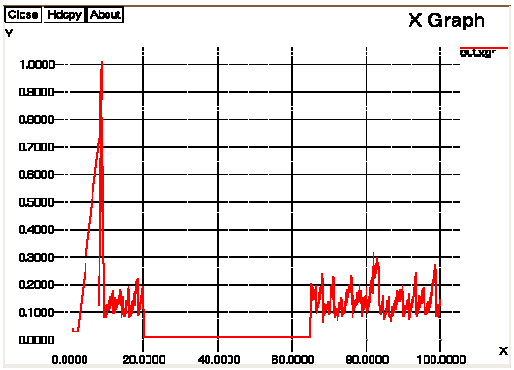


Fig. 7. End to end delay using AODV protocol with tunnel Wormhole activated from second 20 to second 60.

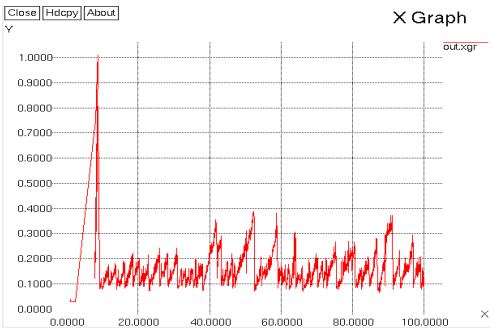


Fig. 8. End to end delay using AODVSEC protocol with tunnel Wormhole activated from second 20 to second 60.

Figure 6 illustrates end to end delay using AODV. In figure 7 we observe wormhole effect on end to end delay because wormhole route have low latency. But in figure 8 which represents end to end delay using AODVSec in the presence of wormhole attack, end to end delay is not changed.

7 Conclusion

In this paper, we have studied the problem of wormhole detection & isolation in wireless sensor networks. Several solutions are proposed into two different classes: *Centralised approaches* like Statistical Wormhole Detection which, Wormhole Detection with multi-dimensional Scaling, and the detection and elimination of Wormhole by the graph theoretical. The *decentralized approaches* class regroups packet leaches with its two variants geographical and temporal leashes, time of flight approach. And have proposed our RTT based mechanism; witch focuses on securing route discovery.

Comparing with other solutions Our's presents the following benefits:

- It secures route discovery, on one side: we do not have to frequently check for wormhole which causes a lot of bandwidth and resource consuming, on the other side, the wormhole will be identified before it can do any harm to the network because wormhole attacks have to interfere in the route setup before they can cause any damage.
- It is based on RTT, so it doesn't require any supplement special location hardware like GPS nor global clock synchronization (each node use its own clock to calculate RTT).

8 Perspective

Our work is being completed, now our solution is implemented in the AODV protocol on our simulation platform NS2. In perspective, we will focus our research on integrating it in the AOMDV protocol to benefit of the multipath routing.

References

1. RadhaPoovendranandLoukasLazos, **"**A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks". *Wireless Netw (2007)*, C _Springer Science+Business Media, LLC 2007, 8 May 2006.
2. Y.-C. Hu, A. Perrig, and D. B. Johnson**,** "Packet leashes: a defense against wormhole attacks in wireless networks",*INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communication Societies,Vol. 3*,IEEE, March 30 April 3rd 2003**,** pp. 1976-1986.
3. L. Butty´an and J.P. Hubaux, SECURITY AND COOPERATION IN WIRELESS NETWORKS Thwarting malicious and selfish behavior in the age of ubiquitous computing,Lausanne – Budapest, 2005 – 2006.
4. M. A. Gorlatova, "Review of Existing Wormhole Attack Discovery Techniques". Defense R&D Canada √ Ottawa CONTRACT REPORT DRDC Ottawa CR 2006-165. August 2006.
5. N. Song, L. Qian and X. Li, "Wormhole Attack Detection in Wireless Ad Hoc Networks: a Statistical Analysis Approach", *Parallel and Distributed Processing Symposium, 2005, proceedings of, 19th IEEE International IPDPS'05,*_c Springer-Verlag, Berlin Heidelberg, 04-08 April 2005, pp. 128-141.
6. I.F. Akyildiz,W. Su, Y. Sankarasubramaniam and E. Cayirci. "Wireless sensor networks: a survey. *Computer* Networks",*Computer Networks (Elsevier)***,** 2002,pp.*393-422*.
7. H.Pucha, D. Hu Y.C. Koutsonikolas, S.M. Das, "On optimal ttl sequence-based route discovery in manets ». Distributed Computing Systems Workshops,, 2005, p.923.
8. W. wang, B. Bhargava, Y. Lu, and X. Wu, "Defending against Wormhole Attacks in Mobile Ad Hoc Networks", *under review at Wiley Journal Wireless Communication and Mobile Computing 2005,*2005,pp. 1–21.
9. A.k. Pathan, "Security of Self-Organizing Networks: MANET, WSN, WMN, VANET", *Journal of High Speed Networks (JHSN)*, Auerbach Publications, London, 2011.
10. W. Sharif, C. Leckie "New Variants of Wormhole Attacks for Sensor Networks" ,*Telecommunication Networks and Applications Conference, ATNAC*, IEEE, 2006.

Avec le soutien de

Université Abou Bekr Belkaid (Tlemcen)
&
Agence Nationale pour le Développement de la Recherche Universitaire

---

Autres Sponsors