MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE 8 MAI 1945 – GUELMA, ALGERIE
Faculté des Mathématiques, Informatique et Sciences de la Matière
Département d'informatique



COSI 2011

Proceeding de la 8ème édition du Colloque sur L'Optimisation et les Systèmes d'Information

24 -27 Avril, Guelma, Algérie

Actes du Huitième Colloque sur l'Optimisation et les Systèmes d'Information - COSI'2011

24-27 Avril 2011, Guelma, Algérie

Université de Guelma
Faculté des Mathématiques, Informatique et Sciences de la Matière
Département Informatique

April 6, 2011

Contents

Préface

Ces actes regroupent les articles présentés lors du 8ème Colloque sur l'Optimisation et les Systèmes d'Information (COSI 2011) qui s'est déroulé à Guelma, Algérie, du 24 au 27 Avril 2011.

COSI est la principale manifestation scientifique Algérienne pour les chercheurs qui travaillent sur la conception et le développement d'approches et d'outils pour l'optimisation, la représentation et l'accès à des informations de tout type. Cette manifestation rassemble plusieurs domaines de recherche comme :

- Les bases de données
- La fouille de données
- L'intégration d'information et d'applications
- Les systèmes d'information et applications dédiées
- L'algorithmique discrète
- L'optimisation combinatoire, programmation mathématique, optimisation, …
- La représentation des connaissances et raisonnement
- La programmation par contraintes
- Le traitement d'images et vision artificielle

Le programme de cette édition 2011 comportent à la fois des articles consacrés à des travaux théoriques et à des applications qui font souvent appel à diverses techniques issues des différents thèmes couverts par ce colloque. L'évolution du nombre de papiers soumis témoigne de la pérennité et de l'importance du colloque. Le comité de programme a procédé à l'évaluation de 212 articles soumis et en a sélectionné 37, ce qui donne un taux de sélection de 17.45%.

Le programme du colloque comporte également quatre plénières et trois cours donnés par des chercheurs de renommée internationale.

Cours

Auteur : Yamine Aït Ameur (Ecole Nationale Supérieure de Mécanique et d'Aérotechnique ENSMA, Poitiers, France)
Titre : Formal Verification of processes and data models. Implicit and explicit semantic-based approaches.

Auteur : Pierre Marquis (Université d'Artois, France)
Titre : Reasoning under inconsistency

Auteur : Abdelkader D. Zighed (Université Lumière Lyon 2, Lyon, France)
Titre : Classification and clustering

Conférences plénières

Auteur : Ridha Mahjoub (Université Paris Dauphine, France)
Titre : Approches Polyédrales et conception de réseaux

Auteur : Mahmoud Boufaïda (Université Mentouri, Constantine, Algérie)
Titre : Interopérabilité des systèmes d'information : Méthodes et outils de mise en oeuvre

Auteur : Alain Quilliot (ISIMA, Clermont-Ferrand, France)
Titre : Tendances Nouvelles et Problèmes Emergents en recherche Opérationnelle

Auteur : Djamel Ziou (Université de Sherbrooke, Canada)
Titre : Perception, apprentissage et reconnaissance d'objets

Les précédentes éditions de COSI ont eu lieu à Tizi-Ouzou (2004), Béjaia (2005), Alger (2006), Oran (2007), Tizi-Ouzou (2008), Annaba (2009) et Ouargla (2010).

Nous remercions les auteurs pour leurs excellentes contributions, les auteurs des cours et conférenciers invités, les membres du comité de programme, les membres du comité d'organisation et les sponsors du colloque.

Pr. Mohand-Saïd Hacid

Président du comité de programme

Organisation

Université de Guelma, Algérie

Présidents d'honneur

Professeur Mohamed NEMAMCHA , Recteur de l'Université de Guelma, Algérie

Professeur Athmane MEDDOUR, Doyen de la faculté des Mathématiques, Informatique et Sciences de la matière

Comité d'Organisation

Président

Pr. Hamid SERIDI, Université de Guelma, Algérie

Vice-Président

Pr. Salim HADDADI , Université de Guelma, Algérie

Membres

Karima BENHAMZA, Université de Guelma, Algérie
Yamina BORDJIBA, Université de Guelma, Algérie
Riad BOURBIA, Université de Guelma, Algérie
Karim DJELAILIA, Université de Guelma, Algérie
Brahim FAROU, Université de Guelma, Algérie
Nouredine GOUASMI, Université de Guelma, Algérie
Mourad HADJERIS, Université de Guelma, Algérie
Khaled HALIMI, Université de Guelma, Algérie
Samir HALLACI, Université de Guelma, Algérie
Ali KHEBIZI, Université de Guelma, Algérie
Yacine LAFIFI, Université de Guelma, Algérie
Ali SERIDI, Université de Guelma, Algérie
Ilyes TOUALBIA , Université de Guelma, Algérie
Nawel ZEDADRA, Université de Guelma, Algérie

Comité de Pilotage

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

Comité de Programme

Président

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

Membres

Fateh Ellaggoune, Université de Guelma
Mohamed Tayeb Laskri, Université dAnnaba (Algérie)
Yacine Sam, Université de Tours
Nora Faci, Université Claude Bernard Lyon 1
Riadh Farah, ENSI,Tunis
Babahenini Mohamed Chaouki, Université Mohamed Khider, Biskra
Rachid Nourine, Université d'Oran
Mohand Ouanes, UMM Tizi-Ouzou
Sadok Ben Yahia, Faculté des Sciences de Tunis
Brahim Oukacha, Université Mouloud Mammeri, Tizi-Ouzou
Rokia Missaoui, Universite de Quebec en Outaouais
Emmanuel Trelat, Université d'Orléans
Fatiha Sais, LRI, Université de Paris Sud
Pierre Spiteri, INP, Toulouse
Hayett Merouani, Université Badji Mokhtar, Annaba
Sihem Amer-Yahia, Yahoo! Research
Bornia Tighiouart, Université Badji Mokhtar, Annaba
Belaid Benhamou, LSIS, Marseille
Lakhdar Sais, CRIL
Lhouari Nourine, LIMOS
Mohamed Aidene, Université de Tizi-Ouzou
M. Aider, USTHB (Alger)
H. Ait haddadene, USTHB (Alger)
M. Baiou, Clermont-Ferrand II
Souhila Kaci, CRIL, Université d'Artois, France
H. Belbachir, USTO (Oran)
Nacéra Benamrane, USTO (Oran)
Boualem Benatallah, University of New South Wals (Australie)
Salima Benbernou, Univerdsité Paris Descartes, France
Abdelhafid Berrachedi, USTHB (Alger)
Djamal Rebaïne, Université du Québec à Chicoutimi
Isma Bouchemakh, USTHB (Alger)
Korichi Driss, Université Kasdi Merbah, Ouargla
Djemel Ziou, Université de Sherbrooke, Canada
Michel Habib, Université de Paris VII (France)
Mephu Engelbert, CRIL - Université d'Artois
Youssef Hamadi, Microsoft Research Cambridge
Philippe Mahey, Université de Clermont-Ferrand II
Frédéric Messine, ENSEEIHT, IRIT Toulouse (France)
Laurent Gourvès, Lamsade, France
Jean-Marc Petit, INSA de Lyon (France)
MS. Radjef, Université de Bejaia
Michel Schneider, Université de Clermont-Ferrand II
Bachir Sadi, UMMTO, Tizi-Ouzou
Tatiana Tchemisova, University of Aveiro, Portugal
Farouk Toumani, LIMOS
H. Kheddouci, Université de Lyon I
Rachid Ahmed-Ouamer, UMMTO, Tiszi-Ouzou
Vincent Barra, Clermont-Ferrand II (France)
Mohand Boughanem, IRIT, Toulouse
M.O. Bibi, Université de Béjaia
Hélène Fargier, CNRS, IRIT, Toulouse
Tahar Kechadi, UCD, Irlande
Fouilhoux Pierre, LIP6, Paris
Chaoui Allaoua, Université de Constantine
Marie-Christine Fauvet, Université Joseph Fourier, Grenoble
Allel Hadjali, ENSSAT, Lannion
Alain Leger, France Télécom
Christophe Rey, Université Blaise Pascal, Clermont-Fd
Fethi Rabhi, UNSW, Sydney, Australie
Samir Tata, INT, Paris
Hélène Jaudoin, ENSSAT, Lannion, France

Mahmoud Boufaida, Université Mentouri, Constantine
Ansaf Salleb-Aouissi, Columbia University, USA
Seridi Hassina, Université Badji Mokhtar, Annaba
SERIDI Hamid, Université 8 Mai 1945, Guelma
Haddadi Salim, Université 8 mai 1945, Guelma
Elghazel Haytham, Université Claude Bernard Lyon 1
KHOLLADI Mohamed-Khireddine, Université Mentouri, Constantine
Bilel Derbel, Université des Sciences et Technologies
Abdenour Bouzouane, Université du Québec à Chicoutimi
Ladjel BELLATRECHE, LISI, Université de Poitier, France
Abdessamad Imine, Loria, Université Nancy 2, France
Okba Kazar, Université de Biskra
Mohamed Benmohamed, Université de Constantine
Mohamed Zine Aissaoui, Université 8 Mai 1945, Guelma
Abdelhani BOUKROUCHE, Université 8 Mai 1945, Guelma
Paul Elkhoury, SAP, Germany
Athman Bouguettaya, CSIRO, Australia
Laurent d'Orazio, Université Blaise Pascal Clermont-Fd
Mamadou Kante, Limos
Vincent Limouzy, Limos
Zizette Boufaïda, Université de Constantine
Kamel Adi, Université du Québec en Outaouai, Canada
Nadir Farah, Université d'Annaba
Ahlam Melouah, Université d'Annaba
Nouredine Djedi, Université de Biskra
Yacine Lafifi, Université de Guelma
Smaine Mazouzi, Université de Skikda
Mohamed Yagoubi, Université de Laghouat
Abdelouaheb Moussaoui, Université de Sétif
Azedine Belami, Université de Batna
Sellami Mokhtar, Université d'Annaba
Kamal Melkemi, Université de Biskra
Tarek Khadir, Université d'Annaba

Relecteurs additionnels

Philippe Fournier-Viger
Yehia Taher
Cyril Laitang
Karim TABIA
Dalila Boughaci
Sami Zghal
Ludovic Liétard
Brahim Medjahed
Selma Khouri
Laurent Beaudou
Arnaud Mary
Tlili Yamina
Karen Sauvagnat
Fatiha Boubekeur
Philippe Marthon
Kamel Boukhalfa
Abdelmounaam Rezgui
Ronan Tournier
Damien LOLIVE
Nouredine TAMANI
Rania Khefifi
Grégory SMITS
Elkamel Merah
Raida Elmansouri
Redha Bahri
Kreshnik Musaraj

Théorie des graphes

On graphs vertex critical with respect to b-chromatic number

Mostafa Blidia [*] **Noureddine Ikhlef Eschouf** [†]
Frédéric Maffray [‡]

Abstract

A b-coloring is a proper coloring of the vertices of a graph such that each color class has a vertex that is adjacent to a vertex of every other color. The b-chromatic number $b(G)$ of a graph G is the largest k such that G admits a b-coloring with k colors. A graph G is called *b-critical* if the removal of any vertex of G decreases the b-chromatic number. In this paper, we prove various properties of b-critical graphs. In particular, we characterize b-critical trees.

Keywords: b-coloring, b-crtical graphs.

1 Introduction

Let $G = (V, E)$ be a simple graph with vertex-set V and edge-set E. A coloring of the vertices of G is a mapping $c : V \to \{1, 2, \ldots\}$. For every vertex $v \in V$ the integer $c(v)$ is called the color of v. A coloring is *proper* if any two adjacent vertices have different colors. The *chromatic number* $\chi(G)$ of graph G is the smallest integer k such that G admits a proper coloring using k 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 in all the other $k - 1$ colors classes. We call any such vertex a *b-vertex*. The concept of b-coloring was introduced in [3, 4]. The b-chromatic number $b(G)$

[*]LAMDA-RO Laboratory, Dept of Mathematics, B.P. 270, University of Blida, Algeria.
[†]Dr. Yahia Farés University of Médéa, Algeria.
[‡]C.N.R.S, Laboratoire G-SCOP, 46 Avenue Félix Viallet, 38031 Grenoble Cedex, France.

of a graph G is the largest integer such that G admits a b-coloring with k colors.

More recently, N. Ikhlef eschouf [2] began the study of edge b-critical graphs where he characterized the edge b-critical P_4 sparse graphs and edge b-critical quasi-line graphs. We propose here to study the effect of removing a vertex of a graph G on the b-chromatic number. If x is a vertex of a graph $G = (V, E)$, then $G - x$ is the subgraph of G that results after removing from G the vertex x. A graph G is called *b-critical* if the removal of any vertex of G decreases the b-chromatic number.

For notation and graph theory terminology we in general follow [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). The distance between two vertices x and y, denoted by $d_G(x, y)$ (or $d(x, y)$ if there is no confusion), is the length of a shortest path from x to y. A vertex of degree one is called a pendent vertex or a leaf and its neighbor is called a support vertex. If v is a support vertex, then L_v will denote the set of the leaves attached at v. A rooted tree distinguishes one vertex x called the root. For each vertex $u \neq x$ of T , the parent of u is the neighbor of u on the unique $x - u$ path, while a child of u is any other neighbor of u. For a vertex u in a rooted tree T, we let $C(u)$ and $D(u)$ denote the set of children and descendants, respectively, of u, and we define $D[u] = D(u) \cup \{u\}$. The maximal subtree at u is the subtree of T induced by $D[u]$, and is denoted by T_u. Let $\omega(G)$ denote the size of a maximum clique of G. We let P_k denote the path with k vertices. A vertex of a path P_k distinct from an end-vertex is said to be an internal vertex.

In this paper, we prove various properties of b-critical trees. In particular, we show that if T is a b-critical tree, then $\Delta(T) \leq b(T) \leq \Delta(T) + 1$, where $\Delta(T)$ is the maximum degree in T. We also give a characterization of b-critical trees.

2 Results on b-critical trees

We will use several definition and results due to Irving and Manlove [3].

Remark that if a graph G admits a b-coloring with k colors, then G has at least k vertices of degree at least $k - 1$. Irving and Manlove defined the *m-degree* $m(G)$ of a graph G as the largest integer l such that G has at least l vertices of degree at least $l - 1$. Thus every graph satisfies the following inequality:

Proposition 2.1 *[3]For any graph G, $b(G) \leq m(G)$.*

Definition 2.2 *[3]A vertex v of T such that $d(v) \geq m(T) - 1$ is called a dense vertex of T.*

Definition 2.3 *[3]A tree $T = (V, E)$ is pivoted if T has exactly m dense vertices, and T contains a distinguished vertex v such that:*
(1) *v is not dense.*
(2) *Each dense vertex is adjacent either to v or to a dense vertex adjacent to v.*
(3) *Any dense vertex adjacent to v and to another dense vertex has degree $m - 1$.*

Theorem 2.4 *[3]If $T = (V, E)$ is a pivoted tree, then $b(T) = m(T) - 1$ else $b(T) = m(T)$.*

Definition 2.5 *A graph G is said to be a b-critical graph if for any vertex v, $b(G - v) < b(G)$.*

Definition 2.6 *Let G be a graph and c be a b-coloring of G with $b(G)$ colors. A set S of b-vertices of c is said to be b-system of c if $|S| = b(G)$ and for any two vertices x, y of S, $c(x) \neq c(y)$.*

We begin with some useful observations.

Observation 2.7 *Let T be a b-critical tree and let c be a b-coloring of T with $b(T)$ colors. Let S be the set of all b-vertices of c. Let z be a support vertex. Then the following three properties hold:*
i) $z \in S$. Moreover, $\forall x \in S$, $c(x) \neq c(z)$.
ii) $\forall v \in V \backslash S, N(v) \cap S \neq \emptyset$.
iii) z cannot have two neighbors of the same color such that one of them is a leaf.

Proof. Let $b(T) = k$ and let c be a b-coloring of c with k colors.
i) Otherwise, the removal of any leaf of z do not decrease the b-chromatic number.
ii) Suppose there exists a vertex $u \in V \backslash S$ such that $N(u) \cap S = \emptyset$. Then c remains a b-coloring of $T - u$ with k colors, a contradiction.
iii) Otherwise, the removal of the leaf of the repeated color in $N(z)$ cannot reduce the b-chromatic number. ■

Observation 2.8 *Let $T = (V, E)$ be a b-critical tree. Let c be a b-coloring of T with $b(T)$ colors and let S be the set of all b-vertices of c in T. If S contains two b-vertices x_1, x_2 of the same color, then for $i = 1, 2$, we have:*
i) $N(x_i) \cap S \neq \emptyset$ *and* $N(x_i) \cap (V \backslash S) \neq \emptyset$.
ii) x_i *is not a support vertex.*
iii) Each connected component of $T - x_i$ *contains at least one b-vertex.*
iv) Every neighbor of x_i *is adjacent to at least two b-vertices.*

Proof. Let $k = b(T)$. Let c be a b-coloring of with k colors. Let x_1, x_2 be two b-vertices of the same color. Let u be a neighbor of x_i ($i = 1$ or 2).
i) Without loss of generality, if all neighbors of x_i are b-vertices then c remains a b-coloring of $T - x_j$ for $j \neq i$. If all neighbors of x_i are non-b-vertices then c remains a b-coloring of $T - x_i$. In either case, we have a contradiction. Thus $N(x_i) \cap S \neq \emptyset$ and $N(x_i) \cap (V \backslash S) \neq \emptyset$ for $i = 1, 2$.
ii) $d_T(u) > 1$, otherwise c remains a b-coloring of $T - u$ with k colors, a contradiction.
iii) Suppose there exists a connected component T_i of $T - x_i$ which contains no b-vertex. Since x_i is not a support vertex, it follows that T_i contains some vertex z which is not adjacent to x_i. Hence, c remains a b-coloring of $T - z$ with k colors, a contradiction.
iv) If u is adjacent to x_1 and x_2 then condition (*iv*) holds. So we may suppose that u is not adjacent to one of x_1, x_2, say x_2. Since x_1 is not a support vertex, u is adjacent to at least one vertex besides x_1. If all neighbors of u, other than x_1, are non b-vertices, then c remains a b-coloring of $T - u$ with k colors, a contradiction. ■

Theorem 2.9 *Let $T = (V, E)$ be a b-critical tree, and let c be a b-coloring of T with $b(T)$ colors. Then,*
i) The b-system S of c in T is unique.
ii) $\forall x \in V \setminus S$, $d_T(x) \leq |S| - 2$.
iii) $\forall x \in S$, $|S| - 1 \leq d_T(x) \leq |S|$.

Proof. Let $b(T) = k$ and S be a b-system of c in T.
i) Suppose on the contrary that S is not unique for c. Then there exist two b-vertices x and y of the same color. By Observation 2.8 (*ii*), x, y are not support vertices. We root T at the vertex x. Every connected component T_i of $T - x$ contains a support vertex z_i. Observation 2.7 (*i*) implies that z_i is the only b-vertex of color $c(z_i)$ in T. Since $d_T(x) \geq k - 1$, T contains at least $k - 1$ support b-vertices. If the number of support vertices is more than $k - 1$,

then we have a support vertex with a repeated color in S, which contradicts Observation 2.7 (i). This implies that each connected component of $T - x$ contains exactly one support vertex. Moreover, $d_T(x) = k-1$. Without loss of generality we may assume that $y \in T_i$. Since y is not a support vertex, $d_T(y) = 2$. Therefore it is clear that $k = 3$. Hence, $d_T(x) = 2$ and every vertex which is not a leaf has degree two. Consequently, T is a path of length at least 7 which is not b-critical, a contradiction. So S is unique for c.

$ii)$ Let $x_1, x_2, ..., x_k$ the b-vertices of c in T of colors $1, 2, ..., k$, respectively. Let u be a non b-vertex of c. By Observation 2.7 (ii) and without loss of generality, we may suppose that $u \in N(x_1)$. We shall show that $d_T(u) \leq k - 2$. Vertex u is adjacent to at most $k - 2$ b-vertices, for otherwise, u would be a b-vertex or there is no available color for it.

Therefore, $V = \bigcup_{i=1}^{k} N[x_i] = N[S]$. So, we claim that:

Claim 1: For $i \neq j$, we have:$|N[x_i] \cap N[x_j]| \begin{cases} \leq 1 \text{ if } x_i \text{ is not adjacent to } x_j \\ = 2 \text{ if } x_i \text{ is adjacent to } x_j \end{cases}$

Proof: If if x_i is adjacent to x_j, it is clear to see that $|N[x_i] \cap N[x_j]| = 2$. If x_i is not adjacent to x_j, then assume that $N[x_i] \cap N[x_j]$ contains at least two vertices. Then it is obvious to see that we have either a cycle C_3 or C_4, which contradicts that T is a tree. So Claim 1 holds.

Claim 2: For $i \geq 1$, $|N(u) \cap N[x_i]| \leq 1$.

Proof: If $i = 1$, then the inegality is obvious. Let a, b be two vertices of $N[x_i]$. Suppose that u is adjacent to a and b. If $x_i = a$, then $(u\ a\ b\ u)$ form a cycle of order 3. If $x_i \neq a, b$, then $(u\ a\ x_i\ b\ u)$ form a cycle of order 4; a contradiction. So Claim 2 holds.

Thus, Observation 2.7 (ii) and Claim 2 imply that,

$$d_T(u) \leq k.$$

Suppose that $l = d_T(u) \geq k - 1$. Let $S_1 = N(u) \cap S$ and $S_2 = S \backslash S_1$. Since $|S_2| \geq 2$, $|S| \geq 3$. If $|S| = 3$, then it is easy to check that T is not b-critical tree. Thus we can assume that $|S| \geq 4$. So we claim that:

Claim 3. For $1 \leq i \leq k$, $d_T(x_i) = k - 1$.

Proof: If $l = k$, then Claim 1-2 imply that for every vertex $v \in V \setminus (S \cup N[u])$, $d_T(v) = 1$, and each vertex of S has exactly one neighbor of degree at least 2 and all other neighbors are leaves. Thus, Observation 2.7 (iii) implies that $d_T(x_i) = k - 1$.

If $l = k-1$, then u is not adjacent to one of $N[x_2], N[x_3]..., N[x_k]$, say $N[x_k]$. By the connectedness of T, $N[x_k]$ is adjacent to one of $N[x_1], N[x_2], ..., N[x_{k-1}]$, say $N[x_t]$ ($t = 1$ or $k-1$). For the same argument cited above, $d_T(x_k) = k-1$ and $d_T(x_i) = k - 1$ for $i \neq t, k$. For the vertex x_t there are three cases to distinguish.

a) $x_t = x_{k-1}$ has one neighbor of degree at least 2 and all other neighbors have degree 1. In this case, $d_T(x_t) = k - 1$.

b) $x_t = x_{k-1}$ has two neighbors, say w_1, w_2 of degree 2 such that one of them, say w_1 is a neighbor of u, and all other neighbors of x_{k-1} have degree 1. Suppose that $c(w_1) = c(w_2)$. Since $|S| \geq 4$, x_{k-1} has at least two neighbors leaves, say w_3, w_4. By Observation 2.7 (iii), $c(w_3) \neq c(w_4)$ and therefore one of them, say w_3, has a color different from $c(u)$. Then we can recolor w_1 with $c(w_3)$. So we have obtained a new b-coloring such that x_{k-1} has two neighbors of the same color where one of them is a leaf. This contradicts the Observation 2.7 (iii) for new b-coloring. This implies that $c(w_1) \neq c(w_2)$. Then Observation 2.7 (iii) implies that $d_T(x_t) = k - 1$.

c) $x_t = x_1$ has two neighbors u, y of degree at least 2 such that y is neighbor of one vertex of $N(x_k)$ and all other neighbors of x_1 have degree 1. Similarly, one can show that $d_T(x_t) = k - 1$. So Claim 3 holds.

Assume that u is adjacent to $r \leq k - 2$ b-vertices and $l - r$ non b-vertices. Let L be the set of vertices of T of degree 1 and let U be the set of vertices of T of degré 2. We distinguish among two cases.

Case 1: $l = k$.

We can see that $U = N(u) \cap N(S_2)$, $V = S \cup L \cup U \cup \{u\}$ and all vertices of $V \setminus (S \cup U \cup \{u\})$ belong to L. We construct a new b-coloring π of T using k colors from the b-coloring c. We first color the vertices of S and $L \cap N(S_1)$, as follows: For $1 \leq i \leq k$, set $\pi(x_i) = c(x_i)$, and for every vertex $y \in L \cap N(S_1)$, assign,

$$\pi(y) = \begin{cases} c(u) \text{ if } c(u) \neq k \text{ and } c(y) = k \\ c(y) \text{ otherwise} \end{cases}$$

We next color the vertices of U and $L \cap N(S_2)$. Let $u_i, z \in N(x_i)$, $(r+1 \leq i \leq k)$, such that $u_i \in U$ and $z \in L \cap N(S_2)$. So, for $r+1 \leq i \leq k$, assign,

$$\pi(u_i) = \begin{cases} i-1 \text{ if } i \neq r+1 \\ 1 \text{ if } i = r+1 \end{cases}, \ \pi(z) = \begin{cases} c(u_{r+1}) \text{ if } i = r+1, \ \ c(z) = 1 \\ c(u_i) \text{ if } i \neq r+1, \ c(z) = i-1 \\ c(z) \text{ otherwise} \end{cases}$$

Finally, set $\pi(u) = k$.

Thus, interchanging the colors of some vertices of T produces a new b-coloring π with k colors such that u and x_k are b-vertices of the same color k, contradicting the uniqueness of S for π. So this case can't occur.

Case 2: $l = k-1$.

Then either u is not adjacent to one of $N[x_1], N[x_2]..., N[x_k]$, say $N[x_k]$, or there exist exactly two vertices of S_2, say x_{k-1}, x_k such that $u \in N(w)$, where $w \in N(x_{k-1}) \cap N(x_k)$. So, there are two cases to consider.

Case 2.1: $u \in N(w)$

Since $l = k-1$, $N(x_i) \cap N(x_j) = \emptyset, (i, j \neq 1, k, k-1)$. By Claims 1-2, $N(x_k) \cap N(x_{k-1}) = \{w\}$. If $|S_2| = 2$, then T is pivoted tree and $m(T) = k$. Thus, $b(T) = k-1$, a contradiction. So, we may suppose that $|S_2| \geq 3$. Therefore $d(w) = 3$ and $V = S \cup L \cup U \cup \{u, w\}$.

We construct a new b-coloring π of T using k colors from the b-coloring c, as follows. We color the vertices of S and $L \cap N(S_1)$ in the same way as the first case. We next color the vertices of U, $L \cap N(S_2)$ and $\{w, u\}$ as follows: Let $u_i, z \in N(x_i)$, $(r+1 \leq i \leq k)$, such that $u_i \in U_i$ and $z \in L \cap N(S_2)$. So, for $r+1 \leq i \leq k-2$, set $\pi(u_i) = i+1$, and for $r+1 \leq i \leq k$, assign,

$$\pi(z) = \begin{cases} c(w) \text{ if } i = k-1, k, \ c(z) = r+1 \\ c(u_i) \text{ if } i \neq k-1, k, \ c(z) = i+1 \\ c(z) \text{ otherwise} \end{cases}$$

Finally, set $\pi(u) = k$ and $\pi(w) = r+1$.

Thus, interchanging the colors of some vertices of T produces a new b-coloring π with k colors such that u and x_k are b-vertices of the same color

k, contradicting the uniqueness of S for π. So this case can't occur.

Case 2.2: u is not adjacent to T_k

Since T is a connected graph without cycle, there exists only one edge between, say $N[x_k]$ and $N[x_{k-1}]$. Let $w_1 \in N[x_{k-1}], w_2 \in N[x_k]$ such that $w_1w_2 \in E$. There are two subcases to consider.

Subcases 1: $w_1 = x_{k-1}$ and $w_2 = x_k$.

If S_2 contains only two vertices, then T is pivoted tree and therefore $m(T) = k$. Thus, $b(T) = k - 1$, a contradiction. So $|S_2| \geq 3$. Then we have, $V = S \cup L \cup U \cup \{u\}$. We construct a new b-coloring π of T using k colors from the b-coloring c, as follows: for every vertex $z \in N(x_k)$, set $\pi(z) = c(z)$. Let $u_i, z \in N(x_i)$, $(r + 1 \leq i \leq k - 1)$, such that $u_i \in U_i$ and $z \in L \cap N(S_2)$. So, for $r + 1 \leq i \leq k - 2$, set $\pi(u_i) = i + 1$, $\pi(u_{k-1}) = r + 1$ and for every vertex $v \in N(x_k) \setminus \{x_{k-1}\}$, assign $\pi(v) = c(v)$. Finally, recolor u with k.

Subcases 2: $w_1 \neq x_{k-1}$ and $w_2 \neq x_k$.

Let $u_{k-1} \in N(u) \cap N(x_{k-1})$. Let $z_0 \in N(x_{k-1})$ and $z_1, z_2 \in N(x_k)$ such that $c(z_0) = c(z_1) = r + 1$ and $c(z_2) = 1$. We construct a new b-coloring π of T using k colors from the b-coloring c, as follows:

a) If $w_1 \neq u_{k-1}$, then $V = S \cup L \cup U \cup \{u\}$. If $c(u_{k-1}) = r + 1$, then assign $\pi(v) = c(v)$ for every vertex $v \in N(\{x_k, x_{k-1}\})$ and color the remaining vertices in the same way as the Case 2.1. If $c(u_{k-1}) \neq r + 1$, then assign $\pi(u_{k-1}) = r + 1$. If $z_0 \neq w_1$ or ($z_0 = w_1$ and $c(w_2) \neq c(u_{k-1})$), then assign $\pi(z_0) = c(u_{k-1})$ and $\pi(v) = c(v)$ for every vertex $v \in N(\{x_k, x_{k-1}\}) \setminus \{u_{k-1}, z_0\}$. If $z_0 = w_1$ and $c(w_2) = c(z_1)$, then set $\pi(z_0) = c(u_{k-1}), \pi(w_2) = c(z_2)$ and $\pi(z_1) = c(w_2)$. Also, for every vertex $v \in N(\{x_k, x_{k-1}\}) \setminus \{u_{k-1}, z_0, z_1, z_2\}$, assign $\pi(v) = c(v)$. Finally, recolor the remaining vertices in the same way as the Case 2.1.

b) If $w_1 = u_{k-1}$, then $d_T(w_1) = 3$ and $V = S \cup L \cup U \cup \{u, w_1\}$. If $c(w_1) = r+1$, then for every vertex $v \in N(x_k)$ assign $\pi(v) = c(v)$ and recolor the remaining vertices in the same way as the Case 2.1. If $w_1 \neq z_0$, then assign $\pi(w_1) = r+1$ and $\pi(z_0) = c(w_1)$. If $w_2 = z_1$, then set $\pi(w_2) = 1$ and $\pi(z_2) = \pi(w_2)$ and for every veretx $v \in N(\{x_{k-1}, x_k\}) \setminus \{u_{k-1}, z_0, z_1, z_2\}$, reolor the remaining vertices in the same way as the Case 2.1.

In either subcases we have obtained a new b-coloring π with k colors from

the b-coloring c such that u and x_k are b-vertices of the same color k, contradicting the uniqueness of S for π. So this case can't occur.

This implies that $d_T(u) \leq k-2 = |S|-2$.

iii) The lower bound is trivial. Let us now prove the upper bound. Let x be a b-vertex and let $T_1, T_2, ..., T_p$ connected components of $T-x$. Suppose that $d_T(x) \geq k+1$. Then $p \geq k+1$. Therefore $N(x)$ contains at least one leaf, say u, for otherwise, the uniqueness of S and Observation 2.8 imply that $|S| \geq d_T(x)+1 \geq k+2$, a contradiction. Let $N^r(x) \subset N(x)$ denote the set of neighbors of x of color $r, 1 \leq r \leq k$. Let l be the color of u. Then $\left|N^l(x)\right| = 1$, otherwise, c remains a b-coloring of $T-u$ with k colors, a contradiction. Since $d_T(x) \geq k+1$, there is a color $t \neq l$ such that $\left|N^t(x)\right| \geq 2$. We distinguish among two cases.

Case 1: $\left|N^t(x)\right| \geq 3$.

Then $N(x)$ contains at least three vertices, say x_1, x_2, x_3 of color t. Without loss of generality, we may suppose that $x_i \in T_i, i = 1, 2, 3$. Observation 2.8 and the uniqueness of S imply that one of T_1, T_2, T_3, say T_1, does not contain any b-vertex of color t, l.

Case 2: $\forall r, 1 \leq r \leq k$, $r \neq c(x), |N^r(x)| \leq 2$.

Since $d_T(x) \geq k+1$, there are two colors that appear exactly twice in $N(x)$. Without loss of generality, we may suppose that $x_1, x_2 \in N^t(x)$ and $x_3, x_4 \in N^h(x)$, $h \neq t, l$. Also we may suppose that $x_i \in T_i, i = 1, 2, 3, 4$. Since S is unique, by pigeonhole principle, we can see that there exists a connected component T_s, $1 \leq s \leq 4$, that contains no b-vertex colored t, l (or h, l). Without loss of generality, we may suppose that T_1 contains no b-vertex colored t, l.

In either case, recolor the vertex set $V(T_1) \cup \{u\}$ as follows. We exchange the color t by l and conversely. Then u would be a vertex with a repeated color in $N(x)$. Hence, c remains a b-coloring of $T-u$ with k colors, a contradiction. So $d_T(u) \leq k-2$ ■

As an immediate consequence of Theorem 2.9, we have the following corollary.

Corollary 2.10 *If T is a b-critical tree, then $\Delta(T) \leq b(T) \leq \Delta(T)+1$*

3 Characterization of b-critical trees

In this section, we shall give a characterization of b-critical trees. This amounts to characterize the b-critical trees having a b-chromatic number equal to $\Delta(T)$ or $\Delta(T)+1$.

3.1 b-critical tree with $b(G) = \Delta(T)$.

In order to characterize the b-critical trees T with $b(T) = \Delta(T)$, we define a family $\mathcal{T}_1$ as follows:
Definition: Class $\mathcal{T}_1$. *A tree T is in class $\mathcal{T}_1$ if, and only if, for some integers k and p with $k \geq 4$ and $2 \leq p \leq k-2$, the vertex-set of T can be partitioned into four sets $\{v\}$, D_1, D_2, X with the following properties:*

- *$|D_1| = p$, and every vertex of D_1 is adjacent to v;*
- *$|D_2| = k - p$, and every vertex of D_2 has a neighbor in D_1;*
- *Every vertex of X has a neighbor in $D_1 \cup D_2$;*
- *There is a vertex $w \in D_1$ such that w has a neighbor in D_2, w has degree k, and every vertex of $D_1 \cup D_2 \setminus \{w\}$ has degree $k-1$.*

Note that there is no other edge than those mentioned in the definition, because T is a tree. Moreover, it is easy to see that the definition implies that $|X| = k^2 - 3k + p + 1$. So T has $k^2 - 2k + p + 2$ vertices. Also, $\Delta(T) = k$, $m(T) = k$, the dense vertices are the vertices in $D_1 \cup D_2$, and $b(T) = k$.

Class $\mathcal{T}_1$ may contain several non-isomorphic members with the same value of k and p, depending on the adjacency between D_1 and D_2.

Then we have the following result.

Lemma 3.1 *If $T \in \mathcal{T}_1$ then T is b-critical tree.*

Proof. Let $T = (V, E)$ be a tree of $\mathcal{T}_1$. By definition of T, $b(T) = k$ and $m(T) = k$. Let $I = \{u \in V(T) : d_T(u) = k-1\}$. Let $y \in V(T)$. If $y \in N[I] \cup \{w\}$, then $b(T-y) \leq m(T-y) \leq m(T) - 1 = k-1$. If $y \in N(w)$, then $T-y$ is a pivoted tree. Therefore by Theorem 2.4, $b(T-y) = m(T)-1 = k-1$. Thus T is a b-critical tree. ■

Root T at a vertex x of maximum degree. Let $C(x) = \{x_1, x_2, ..., x_{\Delta(T)}\}$ be the set of children of x. Let $A(x) = \{v \in C(x) : v \text{ is } b\text{-vertex}\}$. Let L_x be the set of leaves attached at x. Let $B(x) = C(x) \setminus (A(x) \cup L_x)$. Let $T - x = T_1 + T_2 + ... + T_{\Delta(T)}$ such that $x_i \in V(T_i), 1 \leq i \leq \Delta(T)$. We let $D(x_i)$ denote the set of descendants of x_i.

Observation 3.2 *Let $k = b(T)$. Let T be a b-critical tree. Let c be a b-coloring of T with k colors and let S be a b-system of c . Let $x \in S$ and let $x_1 \in C(x)$ be a non b-vertex such that $d_T(x_1) \geq 2$. Let $x_{\Delta-1}, x_\Delta \in S$ such that $c(x_1) \neq c(x_{\Delta-1}), c(x_\Delta)$. If both x_Δ and $x_{\Delta-1}$ belong (or both at once do not belong) to $D(x_1)$, then interchanging the colors $c(x_{\Delta-1})$ and $c(x_\Delta)$ in $G[D(x_1)]$ produce a new b-coloring π of T with k colors.*

Lemma 3.3 *Let x be a vertex of maximum degree of a b-critical rooted tree T with the root x such that $b(T) = \Delta(T)$. If L_x and $B(x)$ are nonempty sets, then for any b-coloring of T with $\Delta(T)$ colors, all colors of $B(x)$ are distinct.*

Proof. Let $k = b(T)$. Let c be a b-coloring of T with k colors and let S be a b-system of c. Let x be a vertex of maximum degree. By Theorem 2.9 (ii) and (iii), x is a b-vertex. Since $k = \Delta(T)$, it follows that $C(x)$ contains exactly two vertices of the same color. Suppose there exists two vertices of $B(x)$, say x_1, x_2 of the same color t. Since $k = \Delta(T)$, the colors of $C(x)\backslash\{x_1, x_2)$ are distinct. Hence, every vertex of L_x has a color different to t. Let $S_{T_i} = \{v \in D(x_i) \cap S : x_i \in B(x)\}$. Then by Observation 2.7 (i), $|S_{T_i}| \geq 1$. On the other hand, for $i = 1, 2$, x_i is adjacent to at least one b-vertex other than x colored $r_i \neq t$, otherwise, c remains a b-coloring of $T - x_i$ with k colors. By Theorem 2.9 (i), a b-vertex of color t cannot be in one of S_{T_1}, S_{T_2}, say S_{T_1}. Since L_x is a nonempty set, then it contains at least one vertex, say x_h, of color $h \neq t$. If the color t is in S_{T_2}, then $|S_{T_2}| \geq 2$ and S_{T_1} cannot contain all b-vertices of all the colors of L_x, Indeed, it is clear to see that $|S_{T_2}| \geq 2$. It therefore suffices to show the second part. Suppose on the contrary that all colors of L_x appear in S_{T_1} (i.e., S_{T_1} contain all b-vertices of all the colors of L_x). Then

$$k = |S_{T_1}| + |S_{T_2}| + \sum_{i=3}^{|B(x)|} |S_{T_i}| + |A(x)| + |\{x\}|$$
$$k \geq |L_x| + 2 + (|B(x)| - 2) + |A(x)| + 1 = \Delta(T) + 1 = k + 1$$

a contradiction. So there is a color of L_x, say, h, which does not appear in S_{T_1}. Thus S_{T_1} contains no b-vertex of colors t, h. If the color t is not in S_{T_2}, then t is not in $S_{T_1} \cup S_{T_1}$. b-vertex colored h, cannot be in one of S_{T_1}, S_{T_2}, say S_{T_1}. Thus S_{T_1} contains no b-vertex of colors t, h. In either case, interchanging the colors t and h in $G[D[x_1]]$ produce a new b-coloring π of T with k colors such that x_h would a leaf of repeated color in $C(x)$. Hence,

π remains a b-coloring of $T - x_h$ with k colors, a contradiction. So all colors of $B(x)$ are distinct. ■

We let $D(B - x_1)$ denote the set of descendants of $B(x)\backslash\{x_1\}$.

Lemma 3.4 *Let x be a vertex of maximum degree of a b-critical rooted tree T with the root x such that $b(T) = \Delta(T)$. Let x_Δ, x_1 be two vertices of $A(x)$ and $B(x)$, respectively, of the same color t. If $D(B - x_1)$ contains a b-vertex, say z, of color r, then $B(x)$ cannot contain a vertex of color r.*

Proof. Let $k = b(T)$. Let c a b-coloring of T with k colors. Let S be a b-system of c. Let $z \in D(B - x_1)$ be a b-vertex of color r. Since S is unique, $r \neq t$. Suppose on the contrary that there is a vertex colored r in $B(x)\backslash\{x_1\}$. Since $D[x_1]$ does not contain a b-vertex of color r or t, Observation 3.2, implies that interchanging the colors t and r in $G[D[x_1]]$ produce a new b-coloring π with k colors such that $B(x)$ contains two vertices of color r, a contradiction to Lemma 3.3. ■

Lemma 3.5 *Let x be a vertex of maximum degree of a b-critical rooted tree T with the root x such that $b(T) = \Delta(T)$. Then the following three properties hold:*
i) $|L_x| \geq 1$ *(x is a support vertex).*
ii) $|B(x)| = 1$.
iii) $|A(x)| \geq 1$.

Proof. Let $k = b(T)$ and let T be a b-critical tree with $k = \Delta(T)$. Let c be a b-coloring of T with k colors and let S be a b-system of c. Root T at a vertex x of maximum degree. By Theorem 2.9 (ii) and (iii), x is a b-vertex.

$i)$ Suppose that x is not a support vertex. Then Observation 2.7 (i) implies that any connected component T_i of $T - x$ contains at least one b-vertex. Let $S_{T_i} = V(T_i) \cap S$. Therefore $k \geq |\{x\}| + \sum_{i=1}^{\Delta(T)} |S_{T_i}| \geq \Delta(T) + 1$, a contradiction.

$ii)$ Since $k = \Delta(T)$, it follows that $C(x)$ contains two vertices, say x_1, x_Δ, of the same color t. For the same argument, Observation 2.7 (iii) and Theorem 2.9 (i) imply that $B(x) \neq \emptyset$. Suppose that $|B(x)| \geq 2$. By Lemmas 3.3, 3.4 and Observation 2.7 (iii), one of x_1, x_Δ, say x_1, belongs to $B(x)$, and therefore $x_\Delta \in A(x)$. We shall show that $D(B - x_1)$ contains no b-vertex. Suppose on the contrary that there is a b-vertex, say z, of color r in $D(B - x_1)$. Then by Lemma 3.4, L_x contains a vertex, say x_r, of color r.

Since $D[x_1]$ contains no b-vertex of color t or r, Observation 3.2 implies that there is a new b-coloring π with k colors such that x_1 has color r. Therefore x_r is a vertex with a repeated color in $C(x)$. Thus π remains a b-coloring of $T - x_r$ with k colors, a contradiction. So $D(B - x_1)$ contains no b-vertex. By Observation 2.7 (ii), $D(B - x_1) = \emptyset$. Also by the partition of $C(x)$, $B(x)\backslash\{x_1\} = \emptyset$. Consequently, $B(x) = \{x_1\}$.

$iii)$ Suppose that $A(x) = \emptyset$. Then Lemma 3.3 and Observation 2.7 (iii) imply that all vertices of $C(x)$ are distinct. But then $k = \Delta(T) + 1$, a contradiction. ■

Since $k = \Delta(T)$, $C(x)$ contains two vertices of the same color. By Lemma 3.3 and 3.4, one of them, say x_1, is the only vertex of $B(x)$ and therefore the other, say x_Δ belongs to $A(x)$. We are now in a position to characterize the b-critical trees having a b-chromatic number equal to $\Delta(T)$.

Theorem 3.6 *Let $T = (V, E)$ be a tree with $b(T) = \Delta(T)$. Then T is b-critical if and only if $T \in \mathcal{T}_1$.*

Proof. If $T \in \mathcal{T}_1$, then by Lemma 3.1, T is b-critical. To prove the converse, let $k = b(T)$ and let T be a tree with $k = \Delta(T)$. Let c be a b-coloring of T with k colors and let S be a b-system of c. Root T at a vertex x of maximum degree. By Theorem 2.9, x is a b-vertex. Since $k = \Delta(T)$, $C(x)$ contains two vertices of the same color. By Lemmas 3.3, 3.5 and Observation 2.7 (iii), one of them, say x_1, is the only vertex of $B(x)$ and therefore the other vertex, say x_Δ belongs to $A(x)$. Let t be the color of x_1 and x_Δ. We let $D(A)$ denote the set of descendants of $A(x)$.

Claim 1. $D(A)$ is an independent set

Proof: We shall show that $D(A)$ contains no b-vertex. Suppose that $D(A)$ contains a b-vertex z of color r. Theorem 2.9 (i) implies that $A(x)$ cannot contain a vertex of color r. Therefore $r \neq t$. For the same argument $D(x_1)$ cannot contain b-vertices colored t or r. Since x is a b-vertex, L_x contains a vertex, say x_r, of color r. By Observation 3.2, interchanging the colors t and r in $G(D[x_1])$ produce a new b-coloring of T with k colors such that x_r would a leaf of repeated color in $C(x)$. Hence, π remains a b-coloring of $T - x_r$ with k colors, a contradiction. Thus $D(A)$ contains no b-vertex. Then Observation 2.7 (ii), implies that $D(A)$ is an independent set. So Claim 1 holds.

Claim 2. *The children of x_1 are b-vertices.*

Proof: Suppose on the contrary that there exists a child of x_1, say u, which is not b-vertex. By Observation 2.7 (ii), u is adjacent to a some b-vertex z of color, say r. Theorem 2.9 (i) implies that $A(x)$ cannot contain a vertex of color r. Hence, $r \neq t$. Since x is a b-vertex, L_x contains a vertex, say x_r, of color r. We let $D(u)$ denote the set of descendants of u. It is clear to see that $c(u) \neq r, t$. Since $D(x_1)\backslash D[u]$ contains no b-vertex of color r or t, by Observation 3.2, interchanging the colors t and r in $G[D[x_1]\backslash D[u]]$ produce a new b-coloring π of T with k colors such that x_r is vertex with a repeated color in $C(x)$. Thus π remains a b-coloring of $T - x_r$ with k colors, a contradiction. So Claim 2 holds.

If a and b are two vertices of $V(T)$, then $d(a, b)$ denote the distance between a and b in T.

Claim 3. $|S \cap D(x_1)| = |L_x|$.

Proof: Let $x_h \in L_x$ a vertex of color h. Then $D(x_1)$ contains a b-vertex of color h, otherwise, $A(x)$ contains a b-vertex of color h and therefore x_h would become a vertex of repeated color in $C(x)$, a contradiction. Let $z \in D(x_1)$ a b-vertex of color r. Then by Theorem 2.9 (i), $A(x)$ cannot contain a vertex of color r. Hence, $r \neq t$. Thus, L_x contains a vertex of color r. So Claim 3 holds.

Claim 4. *For each b-vertex $z \in D(x_1), d(x_1, z) \leq 2$.*

Proof: Suppose there exists a b-vertex $z \in D(x_1)$ such that $d(x_1, z) \geq 3$. By Theorem 2.9 (i), $A(x)$ cannot contain a vertex of color $c(z)$. Let v be a child of x_1. By Claim 2, v is a b-vertex. Since $d(x_1, v) = 1$, $v \neq z$. So $c(v) \neq t, c(z)$.

Let $C'(v) = \{u \in C(v) : \text{color of } u \text{ appears in } L_x\}$ and $C''(v) = C(v)\backslash C'(v)$. The vertices of $C'(v)$ cannot be all b-vertices, otherwise, Claim 3 implies that z is a child of v and therefore $d(x_1, z) = 2$, a contradiction. So $C'(v)$ contains at least one non b-vertex. We distinguish among two cases.

Case 1: $C'(v)$ contains at least one leaf, say v_0.

We recolor x_1 with $c(v_0)$ and v_0 with t. We obtain a new b-coloring π of T with k colors such that $C(x)$ contains a leaf, say x_r, colored $c(x_r)$, where

$c(x_r)$ is repeated twice in $C(x)$. So π remains a b-coloring of $T - x_r$ with k colors, a contradiction.

Case 2: $C'(v)$ contains no leaf.

Let $D(C')$ denote the set of descendants of $C'(v)$. Let T' denote the induced subgraph of $D(C') \cup C'(v) \cup \{v\}$. Since $C'(v)$ contains no leaf, any connected component T'_i of $T' - v$ contains exactly one b-vertex. Indeed, Observation 2.7 (ii) implies that any component contains at least one b-vertex. If T' contains more than $|L_x|$ b-vertices, then $k \geq |L_x| + 1 + |\{x\}| + |A(x)| = |C(x)| + 1 = \Delta(T) + 1$, a contradiction. So any connected component T'_i of $T' - v$ contains exactly one b-vertex. Therefore, Claim 3 implies that all vertices of $C''(v)$ are leaves. On the other hand, Observation 2.7 (ii) implies that for each b-vertex $z_i \in T'_i$, $d(v, z_i) = 2$ or 3. In particlar, $d(v, z) = 2$ or 3. Moreover, all vertices of $C(z)$ are leaves. Let v' be a vertex of $C(v)$ such that $z \in D(v')$. Then v' is a non b-vertex that belongs to $C'(v)$. Since $c(v')$ is a color that appears in L_x, then $c(v') \neq t, c(x)$. Let v'' the parent of z. It is clear to see that $d_T(v') = d_T(v'') = 2$. There are two subcases to consider

Sub**case 1:** $d(v, z) = 2$.

In this case $v'' = v'$. So one can recolor x_1 with $c(v')$ and v' with t. We obtain a new b-coloring π of T with k colors such that $C(x)$ contains a leaf, say x_r, colored $c(x_r)$, where $c(x_r)$ is repeated twice in $C(x)$.

Subcase 2: $d(v, z) = 3$.

Since all vertices of $C(z)$ are leaves, there is a leaf, say $v''' \in C(z)$, such that $c(v''') \neq c(v')$ appears in L_x. If $c(v'') = t$, then recolor v'' with $c(v''')$ and v''' with $c(v'')$. Now one can recolor x_1 with $c(v')$ and v' with t. We obtain a new b-coloring π of T with k colors such that $C(x)$ contains a leaf, say x_r, colored $c(x_r)$, where $c(x_r)$ is repeated twice in $C(x)$.

In either subcase π remains a b-coloring of $T - x_r$ with k colors, a contradiction. So Claim 4 holds.

Claim 5. $\forall v \in S \backslash \{x\}, d_T(v) = \Delta(T) - 1$.

Proof: If $v \in A(x)$, then by Claim 1, every vertex of $N(v)$ is a leaf. Hence,

by Observation 2.7 (*iii*), $d_T(v) = \Delta(T) - 1$. If $v \in D(x_1)$ is a b-vertex such that $d(v, x_1) = 2$, then all children of v are leaves, otherwise $D(x_1)$ contains a b-vertex which is at distance three of x_1, which contradicts Claim 3. For the same argument cited above $d_T(v) = \Delta(T) - 1$. If $d(v, x_1) = 1$ (i.e. v is a child of x_1), then by Theorem 2.9 (*iii*), $d_T(v) \leq \Delta(T)$. If $d_T(v) = \Delta(T)$, then v serves the same role as x with $B(v) = \{x_1\}$. Therefore, $A(v)$ contains a b-vertex $x'_\Delta \neq x_\Delta$ with the same color as x_1. This contradicts the uniqueness of S. Then $d_T(v) = \Delta(T) - 1$. So Claim 5 holds.

Using Lemmas 3.5 and Claims $1 - 5$ we can deduce that $T \in \mathcal{T}_1$. This completes the proof of Theorem 3.6. ■

3.2 b-critical tree with $b(G) = \Delta(T) + 1$

Let $k = \Delta(T) + 1$. For the purpose of characterizing b-critical trees with $b(T) = k$ we define the family $\mathcal{T}_2$ of all trees $T = T_k$ that can be obtained from a sequence $T_1, T_2, ..., T_k$ of trees, where $T_1 \notin \mathcal{T}_2$ is a star of order k, and, T_{i+1} (not in $\mathcal{T}_2$), $1 \leq i \leq k-2$, can be obtained recursively from T_i by one of the three operations listed below.

Operation O_1 : Identify the center of a star of order $k-1$ with one leaf of a support vertex of degree $k-1$ of T_i.

Operation O_2 : Attach a star of order $k-1$ of center x by joining x to any vertex u of T_i such that $1 \leq d_{T_i}(u) \leq k-3$.

Operation O_3 : Attach a star of order k by joining one of its leaves to any vertex u of T_i such that $1 \leq d_{T_i}(u) \leq k-3$.

Let $\mathcal{T}_2^*$ be a subfamily of $\mathcal{T}_2$ defined as follows: Let $T_1 = K_{1,k-1}$ and let $u_1, u_2, .., u_{k-1}$ be the leaves of T_1. Let $T_2, T_3, .., T_k$ be stars of order $k-1$. For $j = 2, .., l$ $(3 \leq l \leq k-1)$, identify the center of T_j with one leaf u_j of T_1 (first operation). For $j = l+1, .., k$ attach the star T_j by joining its center to the leaf u_1 of T_1 (second operation).

It is easy to see that the resulting tree is pivoted. Also, $\Delta(T) = k-1$, $m(T) = k$, and $b(T) = k-1$.

Observation 3.7 *If $T \in \mathcal{T}_2^*$, then T is non b-critical tree.*

Proof. If $T \in \mathcal{T}_2^*$, then T is pivoted. Therefore $b(T) = \Delta(T) < \Delta(T) + 1$. Thus T is not b-critical. ■

Also, it is easy to verify the following observation

Observation 3.8 *If T is a pivoted tree of $\mathcal{T}_2$, then $T \in \mathcal{T}_2^*$.*

Lemma 3.9 *If $T \in \mathcal{T}_2 \backslash \mathcal{T}_2^*$, then T is b-critical with $b(T) = \Delta(T) + 1$.*

Proof. If $T \in \mathcal{T}_2 \backslash \mathcal{T}_2^*$, then by Observation 3.8, T is a non pivoted tree. This implies that $b(T) = m(T)$ and $\Delta(T) = m(T) - 1$. Thus $b(T) = \Delta(T) + 1$. By definition of $\mathcal{T}_2 \backslash \mathcal{T}_2^*$ we have, $m(T - w) \leq m(T) - 1$ for every vertex $w \in V(T)$. Then $b(T - w) \leq \Delta(T)$. Thus T is b-critical. ■

Theorem 3.10 *Let $T = (V, E)$ be a tree with $b(T) = \Delta(T) + 1$. Then T is b-critical if and only if $T \in \mathcal{T}_2 \backslash \mathcal{T}_2^*$*

Proof. Let $k = \Delta(T) + 1$. Lemma 3.9 implies the sufficiency. To prove the necessity, let T be a b-critical tree with $b(T) = k$. We first shall show that T belongs to $\mathcal{T}_2$. Since $b(T) = k$, Theorem 2.9 implies that T has exactly k vertices of degree $k - 1$. Let $S = \{x_1, x_2, ..., x_k\}$ be a b-system of rooted tree T at x_1 such that $d(x_1, x_2) \leq d(x_1, x_3) \leq ...d(x_1, x_k)$. Let T_1 be the subgraph of T induced by $N[x_1]$. Then T_1 is a star of center x_1 and order k. Also, it is easy to see that $C[x_i], i \geq 2$, is a star of center x_i of order $k - 1$. Let $S_i = \{x_1, x_2, ..., x_i\}, 2 \leq i \leq k$, be a subset of S. For $i \geq 2$, let T_i be the subgraph of T induced by $(N(S_i) \backslash S) \cup S_i$. Let $x_r \in S_i$ such that $d(x_r, x_{i+1}) = \min\{d(x_{i+1}, y) : y \in S_i\}$. Since T is a tree, there exists only one path connecting x_r to x_{i+1}. Let P be the path connecting x_r to x_{i+1}. The choice of x_r implies that any internal vertex of P is a non b-vertex. So we claim that

The length of P is no more than 3

Suppose on the contrary that the length of P is more than 4. Let $u \in V(P)$ such that $u \notin N[\{x_{i+1}, x_r\}]$. The choice of x_r and x_{i+1} implies that u has no neighbor in S. Thus $b(T-u) \geq b(T)$, a contradiction. So $d(x_{i+1}, x_r) \leq 3$.

Let T_{i+1} be the subgraph of T induced by $N(S_{i+1}) \backslash S) \cup (S_{i+1})$. There are three cases to consider.

Case 1: $d(x_{i+1}, x_r) = 1$. Then $P = x_{i+1} - x_r$, that is, x_r is adjacent to x_{i+1}. Since $C[x_{i+1}]$ is a star of order $k - 1$, it follows that T_{i+1} is obtained from T_i by the first operation.

Case 2: $d(x_{i+1}, x_r) = 2$. Then $P = x_r - a - x_{i+1}$ where a is a non b-vertex belonging to $V(T_i)$ or $V(T \backslash T_i)$. Thus T_{i+1} is obtained from T_i by the second

operation.

Case 3: $d(x_{i+1}, x_r) = 3$. Then $P = x_r - a - b - x_{i+1}$ where a, b are two non b-vertices such that $a \in V(T_i)$ and $b \in V(T \backslash T_i)$. Thus T_{i+1} is obtained from T_i by the third operation.

Then $T = T_k$ is a tree obtained after $k-1$ steps by one of the three operations O_1, O_2 or O_3, from a star of order k. So $T \in \mathcal{T}_2$. On the other hand, by Observation 3.7, $T \notin \mathcal{T}_2^*$. This achieves the proof. ■

References

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

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

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

[4] 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.

Relation entre les nombres de domination forte et faible dans les graphes

Razika Boutrig et Mustapha Chellali
Laboratoire de LAMDA-RO, Département de Mathématiques
Université de Blida
B.P. 270, Blida, Algérie.
e-mail: m_chellali@yahoo.com

Résumé. Soit $G = (V, E)$ un graphe simple. Un sous ensemble S de V est dit dominant de G si tout sommet de $V - S$ est adjacent à au moins un sommet de S. Un ensemble $D \subseteq V$ est un dominant faible (resp, dominant fort) si chaque sommet $v \in V - D$ est adjacent à un sommet $u \in D$, où $\deg(v) \geq \deg(u)$ (resp, $\deg(v) \leq \deg(u)$). Le cardinal minimum d'un ensemble dominant faible (resp, ensemble dominant fort) de G est appelé le nombre de domination faible (resp, le nombre de domination forte) noté $\gamma_w(G)$ (resp, $\gamma_s(G)$) de G. Dans cet article, on montre que si G est un graphe connexe d'ordre $n \geq 3$, alors $\gamma_w(G) + t\gamma_s(G) \leq n$, où $t = \frac{3}{\Delta+1}$ pour tout graphe G, $t = \frac{3}{5}$ si G est un graphe bloc et $t = \frac{2}{3}$ si G est un graphe sans griffes.
Mots-clés: domination faible, domination forte.

1 Introduction

Soit $G = (V, E)$ un graphe simple. Le voisinage ouvert de $v \in V$ est $N(v) = \{u \in V \mid uv \in E\}$ et le voisinage fermé de v est défini par $N[v] = N(v) \cup \{v\}$. Pour un sous-ensemble $S \subseteq V$, le voisinage ouvert de S est $N(S) = \cup_{v \in S} N(v)$, le voisinage fermé de S est $N[S] = N(S) \cup S$ et $G[S]$ est le sous graphe induit par S. Si v est un sommet de V, alors le degré de v, noté par $\deg(v)$, est le cardinal de son voisinage ouvert. Une étoile subdivisée SS_q est un arbre obtenu à partir d'une étoile $K_{1,q}$ par la subdivision de chaque arête une seule fois. Une griffe est le graphe biparti complet $K_{1,3}$. Un graphe est sans griffes, s'il ne contient pas de sous graphe $K_{1,3}$. Un graphe bloc G est un graphe dont tous les blocs de G sont des cliques. Il est bien connu que les graphes blocs sont des graphes triangulés sans $K_4 - \{e\}$.

Dans [5], Sampathkumar et Pushpa Latha ont introduit le concept de la domination faible et forte dans les graphes. Un ensemble $D \subseteq V$ est un dominant faible (wd-ensemble) si chaque sommet $v \in V - D$ est adjacent à un sommet $u \in D$, où $\deg(v) \geq \deg(u)$. L'ensemble D est un dominant fort (sd-ensemble) si chaque sommet $v \in V - D$ est adjacent à un sommet $u \in D$, où $\deg(u) \geq \deg(v)$. Le cardinal minimum d'un ensemble dominant faible (resp, dominant fort) de G est appelé le nombre de domination faible (resp, forte) noté $\gamma_w(G)$ (resp,

$\gamma_s(G)$) de G. Si D est un sd-ensemble de taille $\gamma_s(G)$, alors on dit que D est un $\gamma_s(G)$-ensemble. La domination forte et faible ont éte étudiées par exemple dans [1, 2, 3, 4].

Dans leur papier introduisant la domination forte et faible dans les graphes, Sampathkumar et Pushpa Latha ont montré qu'un graphe G d'ordre n satisfait $\gamma_w(G)+\gamma_s(G) \leq n$ si G est un graphe d-équilibré (G admet un sd-ensemble D_1 et un wd-ensemble D_2 tel que $D_1 \cap D_2 = \emptyset$). Cependant qu' il existe des graphes G pour lesquels $\gamma_w(G)+\gamma_s(G) > n$. Par exemple si G est une étoile subdivisée SS_q avec $q \geq 3$, alors $\gamma_w(SS_q) = \gamma_s(SS_q) = q+1 = \frac{n+1}{2}$.

Dans cet article, nous montrons le résultat suivant.

Theorem 1. *Soient G un graphe connexe d'ordre $n \geq 3$ et de degré maximum Δ. Alors $\gamma_w(G) + \frac{3}{\Delta+1}\gamma_s(G) \leq n$. Par ailleurs:*

i) Si G est un graphe sans griffes, alors $\gamma_w(G) + \frac{3}{5}\gamma_s(G) \leq n$, et
ii) Si G est un graphe bloc, alors $\gamma_w(G) + \frac{2}{3}\gamma_s(G) \leq \frac{3n-1}{3}$.

2 Preuve du Theorème 1

Commençons par donner les deux lemmes suivants, utiles pour la suite.

Lemma 2. *Si G est un graphe non trivial, alors il existe un $\gamma_s(G)$-ensemble D tel que pour tout sommet $x \in D$ ayant au moins un voisin dans $V \setminus D$, il existe un sommet $y \in V - D$ adjacent à x et $\deg(x) \geq \deg(y)$.*

Preuve. Parmi tous $\gamma_s(G)$-ensembles D, on choisit l'un qui satisfait $\sum_{u \in D} \deg(u)$ est maximum. Il est clair que le résultat est valide si $|V| = 2$. Soit $|V| \geq 3$ et supposons que D contient un sommet x tel que $N(x) \cap (V - D) \neq \emptyset$ et $\deg(y) \geq \deg(x)$ pour tout $y \in N(x) \cap (V - D)$. Alors $\{y\} \cup D - \{x\} = D'$ est un $\gamma_s(G)$-ensemble tel que $\sum_{u \in D'} \deg(u) > \sum_{u \in D} \deg(u)$, contradiction avec le choix de D. □

Lemma 3. *Soit B un ensemble indépendant d'un graphe connexe G tel que $\deg(x) \geq 3$ pour tout $x \in B$. Alors:*

i) Si G est un graphe sans griffes, alors $3|B| \leq 2|N(B)|$.

ii) Si G est un graphe bloc, alors $2|B| + 1 \leq |N(B)|$.

Preuve. (i)- Soit E' l'ensemble des arêtes entre B et $N(B)$. Alors puisque $\deg(x) \geq 3$ pour tout $x \in B$, $3|B| \leq |E'|$. Aussi puisque G est sans griffes et B est indépendant, donc chaque sommet de G a au plus deux voisins dans B, ce qui implique que $|E'| \leq 2|N(B)|$. Par conséquent $3|B| \leq |E'| \leq 2|N(B)|$.

(*ii*) Supposons maintenant que G est un graphe bloc et soit $A = N(B)$. Considérons le graphe $G[(B, A)]$ induit par les sommets de B et A. Sans perte

de généralité, on suppose que $G[(B,A)]$ est connexe, sinon on peut répéter la procédure décrite ci-dessous pour chaque composante connexe. Soient v_1, v_2, ..., v_t les sommets de B et $A_1, A_2, ..., A_t$ les sous ensembles de A ordonnés comme suit: $A_1 = N(v_1) \cap A$ et pour $2 \leq k \leq t$, v_k est le sommet de B adjacent à un sommet de $\underset{j=1}{\overset{k-1}{\cup}} A_j$ et $A_k = N(v_k) \cap \left(A - \underset{j=1}{\overset{k-1}{\cup}} A_j \right)$. Puisque tout sommet de B est de degré au moins trois, on a $|A_1| \geq 3$. Aussi, puisque $G[(B,A)]$ est un graphe bloc connexe, alors chaque sommet v_k, pour $k \geq 2$, possède exactement un seul voisin dans $\underset{j=1}{\overset{k-1}{\cup}} A_j$ sinon on aura un cycle de longueur ≥ 4 ou bien $K_4 - \{e\}$. Doù $|A_k| \geq 2$ pour $2 \leq k \leq t$. Par conséquent, $|N(B)| = |A| = |A_1| + |A_2| + ... + |A_t| \geq 3 + 2(t-1) = 2|B| + 1$. □

Avant de donner la preuve de notre résultat, on rappelle les valeurs exactes de $\gamma_s(G)$ et $\gamma_w(G)$ pour les chaînes P_n et les cycles C_n.

Observation 4. *1) Pour tout cycle C_n, $\gamma_w(C_n) = \gamma_s(C_n) = \lceil \frac{n}{3} \rceil$.*

2) pour toute chaîne P_n d'ordre $n \geq 2$ on a

$$\gamma_s(P_n) = \lceil \tfrac{n}{3} \rceil \quad \textit{et } \gamma_w(P_n) = \begin{cases} \lceil \frac{n}{3} \rceil & \textit{si } n \equiv 1 (\mathrm{mod}\, 3) \\ \lceil \frac{n}{3} \rceil + 1 & \textit{sinon} \end{cases}$$

Voici maintenant la preuve du Theorème 1

Preuve du Theorème 1. Il est clair que si $n \geq 3$, alors $\Delta \geq 2$. Si $\Delta = 2$, alors G est ou bien un cycle C_n ou une chaîne P_n et par l'Observation 4, le resultat est vérifié. Supposons maintenant que $\Delta \geq 3$ et soit D un $\gamma_s(G)$-ensemble satisfaisant la proprieté du Lemme 2. Soient $A = \{x \in D : N(x) \cap (V - D) \neq \emptyset\}$ et $B = D - A$. D'aprés le choix fait sur D, $V - D$ faiblement domine A. Si $B = \emptyset$, alors $A = D$ et d'où $\gamma_w(G) \leq |V - D| = n - \gamma_s(G)$. Donc le résultat est aussi valide pour (i) et (ii) dans le cas où G est un graphe sans griffes ou bien un graphe bloc, respectivement. On peut supposer maintenant que $B \neq \emptyset$. Si B contient deux sommets adjacents u et v, alors l'un de $D - \{u\}$ ou $D - \{v\}$ est un dominant fort de G, d'où la contradiction. Par conséquent, B est un ensemble indépendant. Notons que chaque sommet de D est de degré au moins deux sinon $n = 2$ ou bien G n'est pas connexe. Aussi puisque $N(B) \subseteq A$, alors $\deg(u) \geq 3$ pour tout $u \in B$ car autrement $D - \{u\}$ est un sd-ensemble de G, et donc contradiction. Comme $V - D$ faiblement domine A, alors $(V - D) \cup B$ faiblement domine G, et d'où

$$\gamma_w(G) \leq |(V - D) \cup B| = n - |D| + |B|. \tag{1}$$

Il nous reste à examiner la relation entre $|B|$ et $|D|$ dans le cas où G est un graphe quelconque, graphe sans griffes et graphe bloc. Notons que $|D| = |B| + |A| \geq |B| + |N(B)|$. Soit $E(B, N(B))$ l'ensemble des arêtes entre B et $N(B)$. Comme $\deg(u) \geq 3$ pour tout $u \in B$ et $N(B) \subset D$, alors $3|B| \leq |E(B, N(B))|$. Ainsi tout sommet $y \in N(B)$ est de degré au plus $\Delta - 1$ autrement $D - N(y) \cap B$ devient un sd-ensemble de G, d'où la contradiction. Il s'ensuit que

chaque sommet de $N(B)$ a au plus $\Delta-2$ voisins dans B et d'où $|E(B,N(B))| \leq (\Delta-2)|N(B)|$. Ce qui implique que $3|B| \leq |E(B,N(B))| \leq (\Delta-2)|N(B)|$ et donc $|N(B)| \geq \frac{3}{\Delta-2}|B|$. Comme $|D| \geq |B|+|N(B)|$, on obtient $|B| \leq \frac{\Delta-2}{\Delta+1}|D|$. Par substitution dans (1), on obtient $\gamma_w(G) \leq n - \frac{3}{\Delta+1}|D|$, d'où le résultat. Le détail des calculs est omis.

En utilisant le Lemme 3, on peut améliorer ce résultat pour les graphes sans griffes et les graphes blocs. Et nous trouverons (i) et (ii), respectivement. □

Comme la classe des graphes blocs contient la classe des arbres. On énonce le corollaire suivant comme conséquence du Theorème 1.

Corollary 5. *Si T est un arbre d'ordre $n \geq 3$, alors $\gamma_w(T) + \frac{2}{3}\gamma_s(T) \leq \frac{3n-1}{3}$.*

References

[1] J. H. Hattingh and M. A. Henning, On strong domination in graphs. *J. Combin. Math. Combin. Comput.* 26 (1998) 73–92.

[2] J. H. Hattingh and R. C. Laskar, On weak domination in graphs. *Ars Combinatoria* 49 (1998).

[3] D. Rautenbach, Bounds on the weak domination number. *Austral. J. Combin.* 18 (1998) 245–251.

[4] D. Rautenbach, Bounds on the strong domination number. *Discrete Mathematics,* 215 (2000) 201–212.

[5] E. Sampathkumar and L. Pushpa Latha, Strong, weak domination and domination balance in graphs. Discrete Math. 161 (1996) 235-242.

Calcul d'invariant dans les ordres
amilioration d'une borne

TALEM Djamel[1] et SADI Bachir[2]

Departement de Mathématiques, Universite de Mouloud MAMMERI,Tizi Ouzou,Algeri

[1]vouleze@yahoo.fr
[2]sadibach@yahoo.fr

Résumé. Etant donné une suite d'ordres $P = D^0(P), D(P), D^2(P), \ldots$, avec $D^i(P)$ est l'ordre strict défini sur l'ensemble des antichaînes maximales de $D^{i-1}(P)$, où $D^i(P) = D(D(...D(P))...)$ i fois. En 1994 T.Y.Kong et P.Ribemboim (voir [3]) ont montré que la suite construite converge vers un ordre total, et le nombre d'iterations au bout desquels on aura un ordre total est majoré par deux fois le nombre d'éléments de la plus longue chaîne de P moins un.
Dans cet article nous donnons le nombre d'itèrations exacte pour un ordre d'intervalles, et nous améliorons la borne supérieure de convergence dans le cas d'un ordre quelconque.

Mots-clé: Calcul d'invariant dans les ordres, Ensemble partiellement ordonné, Chaîne, Antichaîne

1 Introduction

Etant donné un ordre P. Sur l'ensemble des antichaînes maximales de P, on définit un nouvel ordre noté $D(P)$ comme suit: A, B deux antichaînes maximales de P, $A < B$ si et seulement si, $\forall a \in A, \exists b \in B$ tel que $a < b$. De la même façon on définit $D^2(P)$ sur l'ensemble des antichaînes maximales de $D(P)$, et ainsi de suite. A la fin, on aura construit une suite d'ordres $P, D(P), D^2(P), \ldots$, où $D^i(P) = D(D(...D(P))...)$ i fois.

En 1994 T.Y.Kong et P.Ribemboim (voir [3]) ont montré que la suite construite convérge vers un ordre total, c'est à dire, il existe un entier naturel i tel que $D^i(P)$ est un ordre total. Plus précisément, si on pose $cdev(P)$ égal au plus petit entier naturel i pour lequel $D^i(P)$ est un ordre total, alors $cdev(P) \leq 2d(P) - 1$, où $d(P)$ est le nombre d'éléments de la plus longue chaîne de P.

Le problème demeure toujours dans la connaissance de la valeur exacte de $cdev(P)$ ou l'amélioration de sa borne supérieure sans passer par le calcul de la suite $P, D(P), D^2(P), \ldots$.

En 2006, B.Sadi (voir [1]) a calculé la valeur de l'invariant pour quelques classes particulières d'ordres, et il a introduit un nouveau paramètre qui a permis l'amélioration de la borne supérieure de l'invariant dans le cas d'un ordre quelconque.

Dans cet articl nous nous intéressons à une approche du paramètre *cdev* dans le cas d'un ordre quelconque, et nous donnons le nombre d'iterations exacte pour un ordre d'intervalles. Tout cela se fait en introduisant de nouveaux paramètres caractérisant un ordre et influant sur le nombre d'étapes au bout desquelles la suite d'ordres converge vers un ordre total.

2 Notations et résultats préliminaires

2.1 Ensemble ordonné

Un ensemble **ordonné** est un couple $(X, \leq_p)$ où X est un ensemble non vide et $\leq_p$ est une relation d'ordre définie sur X, (i.e réflexive, anti-symétrique et transitive).

On note $P = (X, \leq_p)$ ou P, l'ensemble ordonné ou **ordre** P. Deux éléments x et y de X sont dits **comparables** si $x \leq_P y$ ou $y \leq_P x$ avec l'intérprétation usuelle, sinon ils sont dits **incomparables**, ils sont alors notés $x \| y$.

Un couple $(x, y) \in X \times X$ est une **couverture** et on écrit $x <\cdot\ y$ si $x <_P y$ et il n'existe pas z tel que $x <_P z <_P y$. On dit aussi y **couvre** x (y est un **succésseur immédiat** de x) ou x **couvert** par y (x est un **prédécésseur immédiat** de y).

Un couple $(x, y) \in X \times X$ est un **saut** si x et y sont incomparables.
L'ensemble des éléments n'ayant aucun prédécèsseur (resp. succésseur) sera noté $Min(P)$ (resp.$Max(P)$).

Une **chaîne** de P est un sous-ensemble d'éléments de X deux à deux comparables. Elle a pour **longueur** le nombre de ses éléments. On note $d(P)$ la longueur de la plus grande chaîne de P. La **hauteur** de P est la quantité $h(P) = d(P) - 1$.

Une **antichaîne** est un sous-ensemble d'éléments de X deux à deux incomparables. Une chaîne (resp. antichaîne) de P est dite **maximale** si elle n'est pas strictement incluse dans une autre chaîne (resp. antichaîne).

Un ordre **total** (ou une chaîne) est un ensemble d'éléments deux à deux comparables. $Q = (X, \leq_Q)$ est une **extension** de $P = (X, \leq_P)$ si $x \leq_P y$ implique $x \leq_Q y$. De plus, si Q est un ordre total, on dit que Q est une extension linéaire de P.

2.2 L'ordre $D(P)$

Soit P un ensemble partiellement ordonné. On note $D(P)$ l'**ordre strict** défini sur les antichaînes maximales de P par: A, B deux antichaînes maximales de P, $A < B$ si et seulement si $\forall a \in A$, $\exists b \in B$ tel que $a < b$. D'une manière générale, $D^i(P)$ est l'ordre strict défini sur l'ensemble des antichaînes maximales de $D^{i-1}(P)$. Nous allons voir que $D(P)$ est un ordre où deux antichaînes maximales ayant une intersection non vide ne peuvent pas être comparables.

Remarque 1. Deux antichaînes maximales A, B de P sont incomparables dans $D(P)$ si et seulement si $A \cap B \neq \emptyset$ ou $\exists a_1$, $a_2 \in A$, $\exists b_1$, $b_2 \in B$ tels que: $a_1 < b_1$ et $b_2 < a_2$.

Exemple 1. 1. Cet exemple donne le diagramme de $D(P)$ obtenu à partir de celui de P.

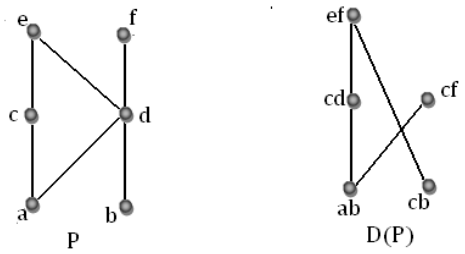


Fig. 1.

2. La figure ci-dessous montre comment se forme la suite P, $D(P)$, $D^2(P)$,... $D^i(P)$, où $D^i(P)$ est un ordre total, et qui est la limite de cette suite d'ordres.

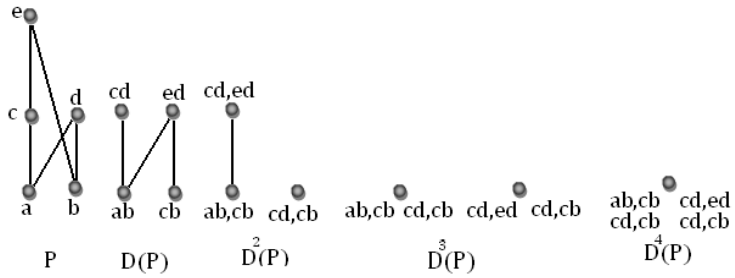


Fig. 2. *$D^4(P)$ est un ordre total, $cdev(P) = 4$.*

2.3 L'inclinaison d'un ordre

Soit $P = (X, \leq_P)$ un ensemble ordonné. On pose $Min(P) = \{x \in P / Pred(x) = \emptyset\}$, où $Pred(x)$ est l'ensemble des prédécesseurs de x, excepté x. Soit l'application *rang*, notée rg, définie sur P à valeurs dans N par:

Pour $x \in P$, rg(x)=$\begin{cases} 0 & \text{si } x \in Min(P), \\ max\{rg(y)/y \in Pred(x)\} + 1 & \text{si non.} \end{cases}$

Définition 1. *Pour $0 \leq i \leq k$, $N_i = \{x \in P / rg(x) = i\}$ est le $(i+1) - ième$**niveau** de P. N_i est dit **complet** si, N_i est une antichaîne maximale de P, c'est à dire, $N_i \in D(P)$. Sinon il est dit incomplet.*

Remarque 2. Le niveau N_i est incomplet, s'il existe $x_j \in N_j$ avec $j < i$, qui soit incomparable aux éléments de N_i.

Définition 2. *On appelle $cdev(P)$ et on lit: "chaîne-déviation de P" le plus petit entier naturel i pour lequel $D^i(P)$ est une chaîne.*

Remarque 3. Dans la suite on suppose que les ordres considérés sont finis.

Définition 3. *Soit A une antichaîne maximale de P. On appelle inclinaison de A, la quantité $I(A) = \max_{x \in A, y \in A}\{|rg(y) - rg(x)|\}$.*

Définition 4. *Soient deux antichaînes maximales A, B de P telles que, $I(A) = |rg(a_2) - rg(a_1)|$ et $I(B) = |rg(b_2) - rg(b_1)|$. On dit que A **croise** B, s'il existe $i \in \{0, 1, ..., d(P)\}$ tel que, $rg(a_1) \leq i \leq rg(a_2)$, $rg(b_1) \leq i \leq rg(b_2)$.*

Remarque 4. A ne croise pas B s'il existe i tel que: $\forall a \in A$ et $\forall b \in B$, $rg(a) \leq i < b$. Dans ce cas on a forcement $A < B$.

On pose $A(P) = \{A \in D(P) / I(A) \geq 2\}$. $F(P)$ est un sous ensemble de $A(P)$ dont les éléments ne se croisent pas deux à deux.

L'**inclinaison** de $F(P)$ est la quantité: $I(F(P)) = \sum_{A \in F(P)} I(A)$.
Soit $F_0(P)$ un sous ensemble de $A(P)$ vérifiant: $I(F_0(P)) = \max I(F(P))$.

Définition 5. *On appelle **inclinaison** d'un ordre P la quantité*

$I(P)$=$\begin{cases} 0 & si\ \forall A \in D(P),\ I(A) = 0, \\ 1 & si\ A(P) = \emptyset\ et\ \exists A \in D(P),\ I(A) = 1 \\ I(F_0(P)) & si A(P) \neq \emptyset \end{cases}$

Définition 6. *On note par I_n la classe d'ordres P telle que $I(P) = n$.*

2.4 Quelques classes d'ordres

a) $C_2 + C_2$ est une somme disjointe de deux chaînes à deux éléments.

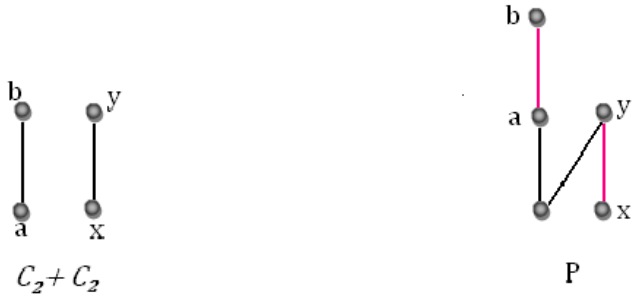


Fig. 3. (a, b) et (x, y) constituent un $C_2 + C_2$. P contient un $C_2 + C_2$ défini par (a, b) et (x, y). On dit aussi que P contient une subdivision isomorphe à $C_2 + C_2$.

b) **Ordre faible**.
Un ordre est dit faible si les seules antichaînes maximales de cet ordre sont ses niveaux.

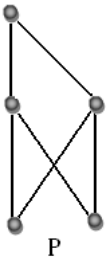


Fig. 4. Ordre faible

d) **Ordre de hauteur 1**.
On appelle ordre de hauteur 1 tout ordre biparti.

Définition 7. *A un niveau N_i on associe M_i^t, $t \geq 1$, un sous-ensemble de $D(P)$ tel que $M_i^t = \{A \in D(P)/I(A) = t, A \cap N_i \neq \emptyset, et A \cap N_{i+t} \neq \emptyset\}$. Pour $t = 1$, on ecrit $M_i = \{A \in D(P)/I(A) = 1, A \cap N_i \neq \emptyset, et A \cap N_{i+1} \neq \emptyset\}$.*

3 Calcul de l'invariant pour un ordre d'inclinaison 1

3.1 Ordre d'inclinaison 1

Soit $P = (X, \leq_p)$ un ordre dans I_1. $N_0, N_1, \ldots, N_k$ sont ses niveaux.

$V_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$, avec $0 < i_1 < i_n < k$, est une suite des éléments de X vérifiant:

1. Pour $j = i_1, ..., i_{n-1}$, $rg(x_{i_{j+1}}) = rg(x_{i_j}) + 1$.
2. Pour $j = i_1, ..., i_{n-1}$, $(x_{i_j}, x_{i_{j+1}})$ est un saut.
3. Il existe $x_{i_0} \in N_{i_1 - 1}$ et $x_{i_{n+1}} \in N_{i_n + 1}$ tels que (x_{i_0}, x_{i_1}) et $(x_{i_n}, x_{i_{n+1}})$ sont des sauts.

x_{i_1} (resp.x_{i_n}) est l'éxtrémité initiale (resp. terminale) de V_i.
V_i est maximale si elle l'est pour 1), 2), 3).

A l'ordre P, on associe l'ensemble $U^0(P)$ des éléments de X défini par l'algorithme suivant:

i) Sur l'ensemble de toutes les suites vérifiant 1), 2), 3), et ayant leurs extrémités initiales dans N_1, soit $V_0 = \{x_1, x_2, \ldots, x_s\}$ avec $x_i \in N_i$ pour $i = 1, \ldots, s$ tel que $|V_0|$ est maximum sur cet ensemble.

ii) Sur l'ensemble de toutes les suites vérifiant 1), 2), 3), et ayant leurs extrémités initiales dans N_{s+2}, soit $V_1 = \{x_{s+2}, x_{s+3}, \ldots, x_r\}$ avec $x_j \in N_j$ pour $j = s + 2, \ldots, r$ tel que $|V_1|$ est maximum.

On continue de la même manière jusqu'à ce qu'on parcourt tous les niveaux de P. Ainsi on construit $V_0, V_1, \ldots, V_m$. Après, on pose $U(P) = \bigcup_{i=0}^{i=m} V_i$, et
$V_m = \{x_{m_1}, x_{m_2}, ..., x_{m_t}\}$. Ensuite:

1. On pose $U^0(P) = U(P)$, si:
   - le dernier niveau de P est complet.
   - Le dernier niveau de P est incomplet, $x_{m_t} \in N_{k-1}$ et $\forall A_{k-2} \in M_{k-2}$, $\exists A_{k-1} \in M_{k-1}$ tel que, $A_{k-2} < A_{k-1}$.

2. On pose $U^0(P) = U(P) \cup \{x_k\}$, avec x_k est un élément quelconque dans N_k, si:
   - Le dernier niveau de P n'est pas complet, $rg(x_{m_t}) \leq k - 2$.
   - Le dernier niveau de P n'est pas complet, $rg(x_{m_t}) = k - 1$ et qu'il existe $A_{k-2} \in M_{k-2}$ incomparable aux éléments de M_{k-1}.

Remarque 5. Dans i) de l'algorithme précédent, s'il n'existe aucune suite vérifiant 1), 2) et 3) et ayant son extrémité initiale dans N_1. On considère l'ensemble des suites vérifiant 1), 2), 3), et ayant leurs extrémités initiales dans N_2. Ainsi de suite.

Exemple 2. .

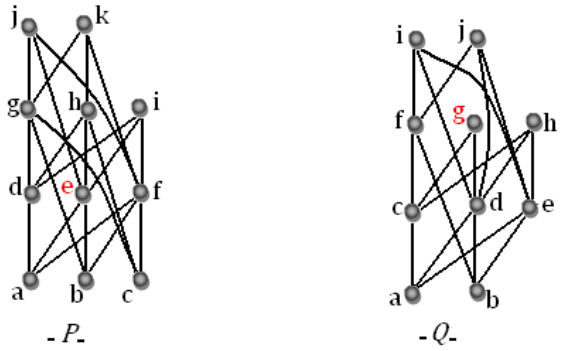


Fig. 5.

$V_1(P) = \{e\}$, $U(P) = \{e\}$ *et* $V_1(Q) = \{g\}$, $U(Q) = \{g\}$. *D'après 2) de la définition précedente,* $U^0(P) = \{e, k\}$,*et* $U^0(Q) = \{g, j\}$.

Proposition 1. *Soit P un ordre dans I_1 ne contenant pas de sous-ordre faible induit par l'union des niveaux consécutifs. $N_0, N_1, ..., N_k$ sont ses niveaux qui sont tous complets. Alors:*

1. *Il y a autant de niveaux dans $D(P)$ que dans P.*
2. *Pour tout, $i = 0, 1, 2, \ldots, k$, $rg(N_i) = i$.*
3. *Pour tout $A \in M_i$, $rg(A) = i$ dans $D(P)$.*
4. *Pour $i = 0, 1, ..., k$, $N_i^1 = \{N_i\} \cup M_i$ est un niveau de $D(P)$.*
5. *Au moins le dernier niveau de $D(P)$ n'est pas complet.*

Preuve. 1. Pour $i = 0, \ldots, k$, N_i est une antichaîne maximale, alors N_i est un élément de $D(P)$. De plus, on a $N_0 < N_1 < \ldots < N_k$, donc $C = \{N_0, N_1, \ldots, N_k\}$ est une chaîne dans $D(P)$. Ainsi $d(D(P)) \geq k + 1 = d(P)$. Et d'aprés [3], on a $d(D(P)) \leq d(P)$, d'où $d(D(P)) = k + 1 = d(P)$.

2. De la chaîne $C = \{N_0, N_1, \ldots, N_k\}$ on déduit que pour $i = 0, 1, 2, \ldots, k$, $rg(N_i) = i$ [3].
3. $A_0 \in M_0$, donc $\exists x \in A_0 \cap N_0$, donc $rg(A_0) \leq rg(x) = 0$ [3], c'est à dire $rg(A_0) = 0$. Pour $i = 1, 2, \ldots, k-1$ $A_i \in M_i$, donc $\exists x \in N_i \cap A_i$, donc $rg(A_i) \leq rg(x) = i$ (voir[3]). .... (1)
   D'autre part $A_i > N_{i-1}$, donc $rg(A_i) > rg(N_{i-1}) = i - 1$. .... (2)
   De (1) et (2), on a $rg(A_i) = i$, pour $i = 1, 2, \ldots, k - 1$.

4. Pour 0, 1, ...,k $N_i^1 \supseteq \{N_i\} \cup M_i^1$. Evident.
   Soit $i \in \{0, 1, ..., k\}$ tel que $A \in (\{N_i\} \cup M_i^1)^C$, alors $A \neq N_i$ et A n'appartient pas à M_i^1. Comme P est d'inclinaison 1, il existe $i_0 \neq i$ tel que $A = N_{i_0}$ ou $A \in M_{i_0}^1$, donc $rg(A) = i_0 \neq i$. Par conséquent $A \in (N_i^1)^C$. D'où pour $i = 0, 1, ..., k$ $N_i^1 = \{N_i\} \cup M_i^1$.
5. $N_k^1 = \{N_k\}$ est le dernier niveau de $D(P)$. P sans sous-ordre faible induit par l'union des niveaux consécutifs, alors $\exists A_{k-1} \in M_{k-1}$ tel que A_{k-1} est incomparable à N_k. Donc A_{k-1} est incomparable aux éléments de $N_k^1 = \{N_k\}$. Ce qui veut dire que N_k^1 n'est pas complet.

Proposition 2. *[4] Soient P un ordre dans I_1 et $N_0, N_1, \ldots, N_k$ ses niveaux qui sont tous complets. Alors:*
$|U^0(P)| = |U^0(D(P))| - 1$.

Proposition 3. *[4] Soit P un ordre dans I_1. $N_0, N_1, \ldots, N_k$ sont les niveaux de P qui sont tous complets, excepté N_k. Alors: $|U^0(P)| = |U^0(D(P))| + 1$.*

Définition 8. *On dit qu'un ordre P dans I_1 atteint la borne superieure si $cdev(P) = 2d(P) - 1$.*

Proposition 4. *[4] Une condition nécessaire et suffisante pour qu'un ordre P vérifie $Cdev(P) = 2d(P) - 1$, est que les trois conditions suivantes soient vérifiées:*
1. *$I(P) = 1$.*
2. *$M_i \neq \emptyset$, pour tout i.*
3. *$U^0(P) = \emptyset$.*

Exemple 3. .

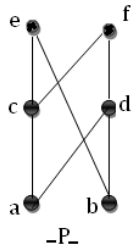


Fig. 6. $cdev(P) = 5 = 2d(P) - 1$

.

3.2 Ordre d'intervalles dans I_1

Au cours du calcul de $D^i(P)$ pour un ordre quelconque, il existe un entier naturel $i_0 \in \{1, 2, ..., cdev(P)\}$ tel que $\forall i \geq i_0, D^i(P)$ est un ordre d'intervalles. Ainsi la connaissance de $cdev(P)$ pour cette classe d'ordres

pourrait permettre d'accélerer le Calcul de *cedev* dans le cas d'un ordre quelconque. Nous allons voir que dans un ordre d'intervalles d'inclinaison quelconque, deux antichaînes maximales seront comparables si et seulement si elles ont une intersection vide.

Définition 9. *Un ordre est dit d'intervalles s'il ne contient pas une subdivision isomorphe à $C_2 + C_2$.*

Proposition 5. *Soit P un ordre d'intervalles d'inclinaison quelconque. Deux antichines maximales A, B de P sont incomparables dans $D(P)$ si et seulement si $A \cap B \neq \emptyset$*

Preuve. En effet, de la remarque 1, s'il a_1, $a_2 \in A$, et $\exists b_1$, $b_2 \in B$ tels que: $a_1 < b_1$ et $b_2 < a_2$, alors (a_1, b_1), (b_2, a_2) constituent un $C_2 + C_2$.

Proposition 6. *[4] Soient P un ordre d'intervalles dans I_1 et $N_0, N_1, ..., N_k$ ses niveaux. Alors:*
Il existe un ordre d'intervalles Q ayant au moins les k premiers niveaux complets, et vérifiant: $d(Q) = d(P)$, $U^0(Q) = U^0(P)$, et $cdev(Q) = cdev(P)$.

Exemple 4. .

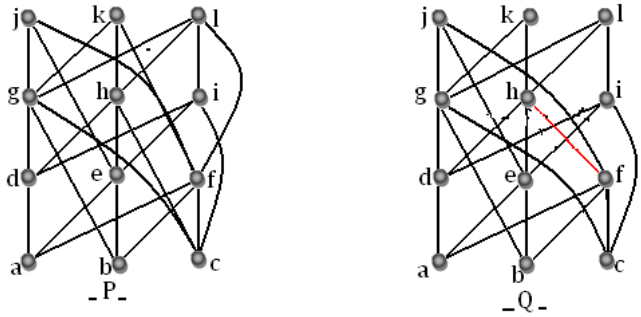


Fig. 7.

P et Q sont deux ordres d'intervalles dans I_1. Le niveau $N_2 = \{ghi\}$ n'est pas complet dans P. Q est obtenu à partir de P en rajoutant la couverture (f, h). Ainsi on peut facilement vérifier que, $d(Q) = d(P) = 4, U^0(Q) = U^0(P) = \{e\}$. Les niveaux de Q sont complets, et $cdev(Q) = cdev(P) = 6$.

Théorème 1. *[4] Soit P un ordre d'intervalles dans I_1 ne contenant pas un ordre faible défini par l'union des niveaux consécutifs comme sous-ensemble de P, alors $cdev(P) = 2d(P) - |U^0(P)| - 1$.*

Exemple 5. .

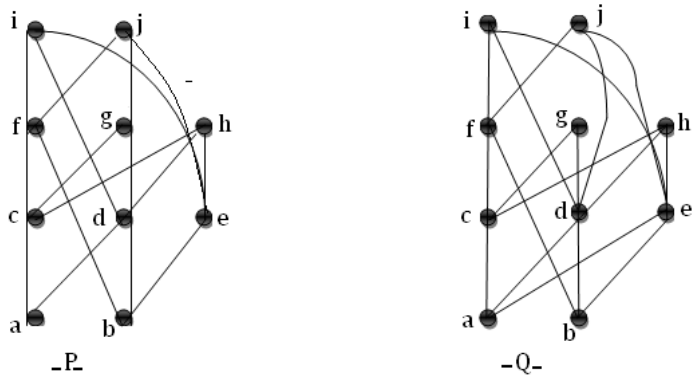


Fig. 8.
$U^0(P) = \{g\}$, $cdev(P) = 2 \times 4 - 1 - 1 = 6$.
$U^0(Q) = \{g, n\}$, $cdev(Q) = 2 \times 4 - 1 - 2 = 5$.

4 Calcul de l'invariant pour ordres d'inclinaison quelconque

Proposition 7. *Si P est un ordre d'intervalles dans I_n, $n \geq 1$, alors $D(P)$ est un ordre d'intervalles dans I_1.*

Preuve. On raisonne par l'absurde.
On suppose que $D(P)$ n'appartient pas à I_1. Soit $D(P) \in I_2$, donc il existe A_0, A_1, A_2, A_3 dans $D(P)$ vérifiant $A_1 <_\bullet A_2 <_\bullet A_3$, $rg(A_1) = rg(A_0)$ et $A_0 \cap A_3 \neq \emptyset$. A_0, A_1, A_2, A_3 sont des antichaînes maximales de P.
Soient $a_1 \in A_0 \cap A_1$, $a_3 \in A_0 \cap A_3$, ainsi (a_1, a_3) est un saut dans P.
$A_1 <_\bullet A_2 \Longrightarrow \exists x \in A_2$ tel que $a_1 <_\bullet x$ ... (1).
$A_2 <_\bullet A_3 \Longrightarrow \exists y \in A_2$ tel que $y <_\bullet a_3$ ... (2).
De (1) et (2), il vient que (a_1, x), (y, a_3) sont des couvertures, et (a_1, a_3), (x, y) sont des sauts, ainsi $(a_1, x), (y, a_3)$ est un $C_2 + C_2$ dans P. C'est absurde, car P est un ordre d'intervalles.

Proposition 8. *Si P est un ordre d'intervalles dans I_n, $n > 1$, alors: $cdev(P) = 2d(D(P)) - |U^0(D(P))|$.*

Preuve. On a, $cdev(P) = 1 + cdev(D(P)) = 1 + 2d(D(P)) - |U^0(D(P))| - 1 = 2d(D(P)) - |U^0(D(P))|$.

Lemme 1. *Soient P un ordre quelconque ayant tous ses niveaux complets, et $F_0(P) = \{A_1, A_2, ..., A_n\}$. Alors: $d(D^2(P)) \leq d(P) - \sum_{i=1}^{n} I(A_i) + n - 1 = d(P) - I(P) + n - 1$.*

Preuve. Par définition, les éléments de $F_0(P)$ ne se croisent pas deux à deux. Ainsi dans $D(P)$, à tout A_i, $i = 1, ..., n$, correspondent $I(A_i) - 1$

niveaux de $D(P)$ incomplets. Le dernier niveau de $D(P)$ est aussi incomplet (proposition 1), donc on a, au moins $(\sum_{i=1}^{n} I(A_i)-n+1)$ niveaux incomplets dans $D(P)$ qui vont tous être suprimés dans le passage à $D^2(P)$, ce qui veut dire que $D^2(P)$ possède au plus $d(P)-(\sum_{i=1}^{n} I(A_i)-n+1)$ niveaux, c'est-á-dire, $d(D^2(P)) \leq d(P)-\sum_{i=1}^{n} I(A_i)+n-1 = d(P)-I(P)+n-1$.

Remarque 6. Dans le theorème ci-dessus, la démonstration se fait en se mettant dans le cas le plus défavorable, c'est-á-dire, on imagine un ordre auquel va correspondre le maximum possible d'itérations pour atteindre l'ordre total.

Théorème 2. *Pour tout ordre P, on a:*
1) $cdev(P) \leq 2d(P)-|U^0(P)|-1$, si $I(P)=1$.
2) $cdev(P) \leq 2d(P)-2I(P)+2|F_0(P)|-1$, si $I(P) \geq 2$.

Preuve. 1) $I(P)=1$.
Si $|U^0(P)|=0$, $cdev(P) \leq 2d(P)-1$ (Voir [3]).
Supposons que, $|U^0(P)| \neq 0$ et les niveaux de P sont tous complets.
Dans $D(P)$, on a $d(D(P))=d(P)$ (proposition 1). $|U^0(D(P))|=|U^0(P)|+1$ (proposition 2). Au moins le dernier niveau de $D(P)$ est incomplet (proposition 1).
Supposons que le dernier niveau de $D(P)$ est l'unique niveau incomplet.
Dans $D^2(P)$, on a $d(D^2(P))=d(D(P))-1=d(P)-1$, car le dernier niveau va disparaitre.
$|U^0(D^2(P))|=|U^0D(P)|-1=|U^0(P)|$ (proposition 3).
De plus $D^2(P)$ est un ordre d'intervalles (voir[3]), donc $cdev(D^2(P))=2d(D^2(P))-|U^0D^2(P)|-1=2d(P)-|U^0(P)|-3$ (théorème 1). D'autre part, $cdev(P)=2+cdev(D^2(P))=2d(P)-|U^0(P)|-1$. Ainsi, la valeur de $cdev(P)$ ne peut pas dépasser $2d(P)-|U^0(P)|-1$; par conséquent, $cdev(P) \leq 2d(P)-|U^0(P)|-1$.

2) D'après le lemme précédent, $d(D^2(P)) \leq d(P)-I(P)+|F_0(P)|-1$.
D'autre part, $cdev(D^2(P)) \leq 2d(D^2(P))-1$ (Voir [3]), et $cdev(P)=2+cdev(D^2(P)) \leq 2+2d(D^2(P))-1 \leq 2+2d(P)-2I(P)+2|F_0(P)|-2-1$, c'est-à-dire,
$cdev(P) \leq 2d(P)-2I(P)+2|F_0(P)|-1$.

References

[1] B. SADI, Suite d'ensembles partiellement ordonnés, *ARIMA*, 4, 2006.
[2] Claude Berge, Graphes et hypergraphes ed.Dunod, Paris,
[3] T.Y.KongetP.Ribemboim, Channing of partially ordred sets, *C.R.Acad.Sci. Paris, t. 319, Série I,p.533-537,1994.*
[4] DJ. TALEM et B. SADI, Calcul dinvariant dans les ensembles partiellement ordonnés, *COSI'2009. Annaba 25 - 27 Mai, p.87-96*

Requêtes flexibles et recherche d'information

Prise en compte des liens pour la sélection d'éléments pertinents dans les documents XML

Samia Iltache[1], Mohand boughanem[2]

[1]Univeristé M'hamed Boughara de Boumerdes, 35000 Boumerdes Algérie.
s_iltache@hotmail.com
[2]IRIT – SIG-RI, 118 Route de Narbonne, 31062 Toulouse Cedex 4. bougha@irit.fr

Résumé. La plupart des méthodes exploitant les documents XML ont pour but de retrouver les unités d'information les plus pertinentes répondant à une requête utilisateur en se basant sur les liens structurels contenus dans ces documents. Les liens de références ont été largement utilisés dans la recherche d'information sur le web. Nous pensons que les liens de référence reliant les documents XML, à l'instar des liens hypertextes, possèdent également une richesse qui doit être prise en compte pour mesurer la pertinence d'un élément vis-à-vis d'une requête. Dans cet article, nous proposons de rajouter une autre dimension aux mesures prenant en compte les liens structurels pour évaluer la pertinence d'un élément vis-à-vis d'une requête. Il s'agit de l'information apportée par les liens *Xlink* et *Xpointer*. Pour ce faire, nous présentons une adaptation, que nous appelons *EléméntRank*, de l'algorithme *PageRank* aux documents XML et nous montrons comment transformer les liens *Xlink* en plusieurs liens *Xpointer* afin d'adapter le principe de *PageRank* à une nouvelle unité d'information représentée par un élément. Nous retenons principalement de *PageRank* le principe qui considère qu'une page web référencée par plusieurs pages est une bonne page et nous l'adaptons à une granularité plus fine.

Mots clés: recherche d'information, XML, Liens Xlink, liens Xpointer.
Keywords: information retrieval, XML, links XLink, links XPointer

1 Introduction

L'information structurelle contenue dans les documents XML est exploitée par les systèmes de recherche d'information afin de traiter l'information avec une granularité plus fine que le document. La réponse fournie à l'utilisateur ne se résume plus à un document entier mais à des parties de document apportant une information pertinente à un besoin utilisateur. Ces modèles doivent mesurer la pertinence de ces unités d'information en fonction deux dimensions représentées par l'*exhaustivité* et la *spécificité* [1]. L'*exhaustivité* mesure à quel point l'unité d'information traite du sujet de la requête utilisateur et la *spécificité* mesure à quel point l'unité d'information se focalise sur le besoin de l'utilisateur. Certaines approches calculent le score des nœuds en propageant les termes dans l'arbre du document [2], ou propagent le poids

des termes [3]. D'autres approchent déduisent le score des nœuds en fonction de celui des nœuds feuilles [4]**.** Alors que les techniques classiques de recherche d'information étaient basées sur l'analyse du contenu des documents, la naissance du web a poussé la communauté de la recherche d'information à réfléchir sur de nouveaux algorithmes prenant en compte le contenu des documents mais aussi la structure des liens hypertextes du web.

Notre problématique concerne la prise en compte des liens de référence *XLink* et *XPointer* pouvant exister entre documents XML. Cette nouvelle dimension doit s'ajouter aux mesures d'*exhaustivité* et de *spécificité* citées ci-dessus pour évaluer la pertinence d'un élément vis-à-vis d'une requête. Nous considérons que ces liens thématiques possèdent une richesse qui doit être prise en compte comme source d'évidence pour mesurer la pertinence d'un élément.

2 Etat de l'art

Les modèles de recherche d'information adaptés au web prennent en compte la combinaison de la structure des documents et des liens hypertextes afin de retrouver les documents qui correspondent le mieux au besoin utilisateur. Différents modèles ont été développés afin d'utiliser les liens dans le processus de recherche d'information. L'algorithme *PageRank* créé par Page et Brin [5] est basé sur la connectivité du graphe du web et permet de donner un rang à chaque page web indépendamment des requêtes utilisateurs. Le rang d'une page web peut être défini comme étant le nombre de pages web pointant vers elle. Il est fondé sur l'hypothèse: "*une page référencée par un grand nombre de pages est une bonne page*". D'autres approches (*l'algorithme HITS* et l'approche de *propagation de pertinence*) affectent pour chaque requête utilisateur, un rang aux pages Web liées à la requête. Le principe de l'algorithme *HITS* [6] est de construire un sous graphe du web et d'ordonner les pages web appartenant à ce sous graphe en leur affectant un rang. Contrairement à *PageRank*, ce sous graphe contient les pages web en relation avec la requête utilisateur. La propagation de la pertinence [7] se base sur le principe qu'*un document référencé par un grand nombre de documents pertinents est un bon document*. Ainsi, la valeur de pertinence d'une page est modifiée en fonction de la pertinence des pages auxquelles elle est liée.

Peu de travaux ont été proposés pour l'exploitation des liens de référence reliant les documents XML. L'un des premiers travaux, appelé XRANK, est proposé par lin et al [8]. Les auteurs définissent une collection d'hyperliens des documents XML comme étant un graphe $G = (N, CE, HE)$ où $N = NE \cup NV$ avec *NE* représentant l'ensemble des éléments et *NV* l'ensemble de leur valeurs (attribut, nom de balises), *CE* l'ensemble des liens structurels et *HE* l'ensemble des liens de référence, puis calculent le score d'un élément en fonctions des scores obtenus en prenant en compte les liens relatifs aux trois ensembles CE, HE et CE^{-1} (liens de l'ensemble CE pris dans le sens inverse). Benny et al [9] appliquent l'algorithme HITS sur les Top-N documents renvoyés pour filtrer les résultats renvoyés à l'utilisateur. La méthode que nous proposons est construite sur une adaptation de l'algorithme *PageRank*.

3 Notre approche

Un document XML est représenté sous forme d'un arbre contenant un nœud racine, des nœuds internes représentant les éléments ou les attributs et des nœuds feuilles contenant les valeurs des éléments ou des attributs. Nous faisons référence à un nœud par un identificateur : n_i pour désigner le ième nœud et f_i pour désigner le ième nœud feuille. Un Exemple d'arbre XML est donné par la figure 1.

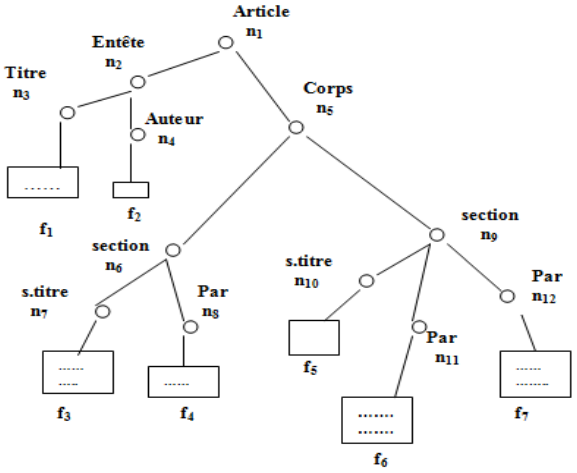


Fig. 1. exemple d'arbre XML.

3.1 ElémentRank

Les éléments appartenant aux documents XML peuvent être liés entre eux par plusieurs types de liens:

- Des liens hiérarchiques permettant de représenter la structure interne d'un document XML.
- Des liens de référence permettant à un élément de référencer un ou plusieurs éléments appartenant au même document ou à des documents différents. Ces liens sont matérialisés grâce à *Xpointer*.
- Des liens de référence permettant à un élément de référencer des documents entiers grâce à *Xlink*.

Nous pensons que prendre en compte les liens de référence liant des documents XML peut apporter une information supplémentaire dans le calcul de la pertinence d'un élément vis-à-vis d'une requête à l'instar des liens entre documents HTML dans le web. Nous citons en l'occurrence l'algorithme *PageRank* qui considère qu'une page web référencée par plusieurs pages est une bonne page. Par analogie à *PageRank*, on

dira *qu'un nœud référencé par plusieurs autres nœuds est un nœud important*. Nous représenterons ainsi chaque nœud d'un arbre XML par une valeur calculée sur la base des liens du graphe XML indépendamment des requêtes utilisateur. Cette valeur représente l'*ElémentRank* d'un élément noté ***ER***. Nous pouvons définir le graphe XML comme étant un graphe orienté dont les nœuds et les arcs sont représentés respectivement par tous les éléments des documents XML et les différents liens structurels et les liens de référence reliant les éléments entre eux. La figure 2 représente un exemple de graphe XML.

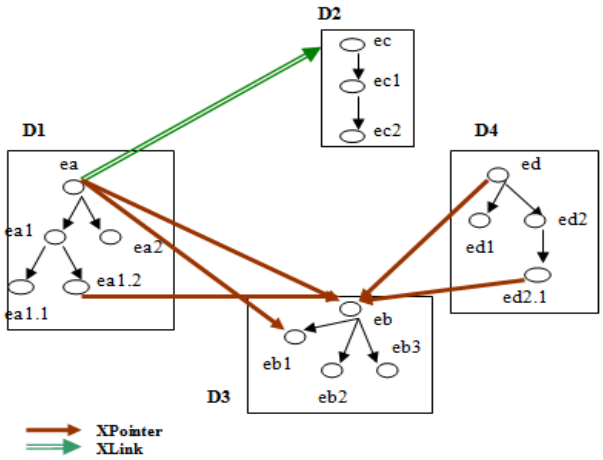


Fig. 2. Exemple de graphe XML

3.1.1 ElémentRank initial des nœuds

Si on considère que pour accéder à un élément, nous devons d'abord accéder au document qui le contient, l'*ElémentRank* initial de chaque nœud est calculé comme suit :

$$ER(ed) = p(ed/d) \times p(d) \quad \textbf{(1)}$$

Où $p(ed/d)$ est la probabilité de sélectionner un élément quelconque ed appartenant au document d et $p(d)$ est la probabilité de sélectionner un document quelconque d dans la collection. On suppose que les documents ont tous la même chance d'être sélectionné. La probabilité *p(d)* est donnée par l'équation 2.

$$P(d) = \frac{1}{nbdoc} \quad \textbf{(2)}$$

De même que pour les documents, on suppose que chaque élément *ed* a une même probabilité d'être sélectionné à l'intérieur du document où il apparaît. Cette probabilité est calculée par l'équation 3.

$$P(ed/d) = \frac{1}{nbe(d)} \quad (3)$$

ed représente un élément quelconque appartenant au document *d* et *nbe(d)* est le nombre d'éléments appartenant au document *d*.

Ainsi l'*ElémentRank* initial de chaque nœud est donné par l'équation 4

$$ER(ed) = \frac{1}{nbdoc \times nbe(d)} \quad (4)$$

Pour illustrer le calcul initial de l'*ElémentRank* d'un nœud, nous considérons une collection composée de 4 documents. Nous avons alors *P(d)=1/4* pour tous les documents XML de la collection. L'*ElémentRank* initial de quelques éléments représentés dans la figure 2 est calculé comme suit :

$$ER(ea) = \frac{1}{4*5} = 0{,}05$$

$$ER(ed) = \frac{1}{4*4} = 0{,}062$$

$$ER(eb) = \frac{1}{4*4} = 0{,}062$$

$$ER(ec1) = \frac{1}{4*3} = 0{,}083$$

3.1.2 Propagation de l'ElémentRank

L'ensemble des nœuds appartenant aux différents documents XML de la collection forme un graphe orienté dont les arcs sont représentés par les différents liens reliant ces nœuds. L'*ElémentRank* d'un nœud permet de donner un rang à chaque élément du graphe indépendamment des requêtes utilisateur. A l'instar de *PageRank*, chaque élément distribue uniformément son *ElémentRank* à tous les éléments du graphe vers lesquels il pointe. Les liens XML sont, comme nous l'avons cité plus haut, de types différents, par conséquent nous devons prendre en compte ce facteur dans les traitements.

1ère étape : traitement du lien Xlink

Un élément *e* qui fait référence à un document *d* par un lien *XLink* (lien thématique) suppose que l'auteur juge important le document entier et que tous ses éléments apportent une information supplémentaire ou complémentaire pour cet élément *e*.

L'élément *e* partagera donc son *ElémentRank* uniformément entre tous les éléments appartenant au document *d*. Etant donné qu'un lien *Xlink* relie un élément à un document, nous considérons alors que tout lien *XLink* reliant un élément *e* à un document *d* sera matérialisé par plusieurs liens de type *XPointer* entre l'élément *e* et tous les éléments appartenant au document *d*. Le lien *XLink* $ea \rightarrow d2$ de la figure 2 est transformé comme suit : Le lien ea → d2 sera remplacé comme le montre la figure 3 par les liens $\begin{cases} ea \rightarrow ec \\ ea \rightarrow ec1 \\ ea \rightarrow ec2 \end{cases}$

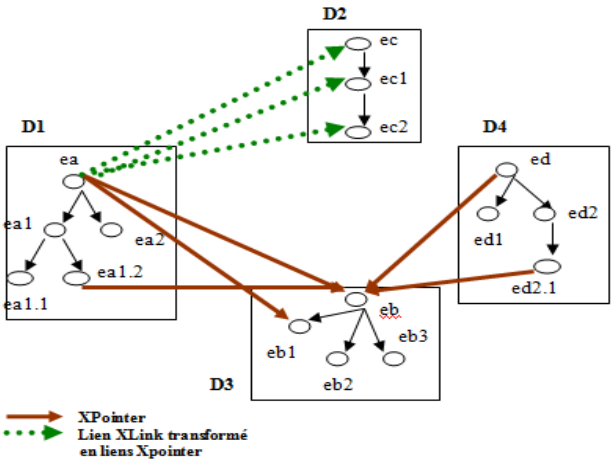


Fig. 3. Transformation des liens XLink

2ème étape : traitement des liens hiérarchiques et des liens *Xpointer*

Un auteur, en inscrivant des liens d'un élément *e* vers d'autres éléments, donne une certaine crédibilité (son *ElémentRank*) à ces éléments. Ces éléments sont susceptibles d'apporter une information supplémentaire ou complémentaire à *e*. Pour le calcul de l'*ElémentRank* de chaque nœud, nous devons définir quels types de liens prendre en compte.

Cas 1 : Prise en compte des liens de référence *Xlink* et *Xpointer*.

Dans un premier temps nous ne considérons que les liens de référence. A partir d'un élément *e*, un utilisateur peut visiter des éléments en suivant les liens de référence. L'élément *e* distribuera son *ElémentRank ER(e)* uniformément à tous les nœuds auxquels il fait référence par un lien *XPointer* ou *XLink.* En se basant sur la formule du calcul du *PageRank* définie par Brin **[Brin et al, 1998]**, l'*ElémentRank* de chaque nœud sera calculé comme suit :

$$ER(e_i) = 1 - d + d * \sum_{e_{rj} \to e_i} \frac{ER(e_{rj})}{Ne_{rj} + \sum_{d_k} nbed_k} \quad (5)$$

Où e_{rj} est un élément pointant vers l'élément e_i par un lien *XPointer*, Ne_{rj} représentent tous les éléments pointés par e_{rj} en utilisant des liens de type *XPointer*, d_k sont les documents pointés par e_{rj} par des liens de type *XLink*, $nbed_k$ est le nombre d'éléments appartenant au document d_k, d est un paramètre prenant ses valeurs dans l'intervalle [0,1] et permettant de faire converger l'algorithme de manière plus ou moins rapide.

La première partie de l'équation 5 indique qu'à tout moment l'utilisateur peut interrompre sa progression dans la navigation en suivant les liens (deuxième partie de l'équation 5) et reprendre le processus à partir d'un élément quelconque appartenant à un document choisi de manière aléatoire. Le calcul de l'*ElémentRank* de chaque nœud s'effectue de manière itérative.

Cas 2 : Prise en compte des liens hiérarchiques et des liens de référence dans le calcul de l'EléméntRank

Une autre approche consiste à inclure les liens de type hiérarchique. Un utilisateur peut à partir d'un élément *e* visiter un nœud référencé par *e* en suivant un lien de référence soit visiter un de ses éléments enfants en suivant le lien hiérarchique qui les relie. Ainsi nous ne distinguons pas entre les liens hiérarchiques et les liens de référence. L'*élémentRank ER(e)* sera distribué uniformément à tous les liens sortants de *e*. Nous considérons ainsi que les liens *ea→ec1*, *ea→eb*, *ea→ea2* de la figure 3 ont la même importance. L'équation 5 devient :

$$ER(e_i) = 1 - d + d * \sum_{e_j \to e_i} \frac{ER(e_j)}{Ne_j + \sum_{d_k} nbed_k} \quad (6)$$

Les équations 5 et 6 sont identiques, la seule différence réside dans $(e_j \to e_i)$. e_j représente un élément pointant vers l'élément e_i par un lien hiérarchique ou par un lien *XPointer*, Ne_j représentent tous les éléments pointés par e_j par un lien hiérarchique où par un lien *XPointer*.
En considérant le cas 2 dans lequel les liens structurels et les liens de référence sont pris en compte, le calcul de l'*ElémentRank* de quelques éléments de la figure 3 est comme suit :

$$ER(ea1.1) = 1 - d + d * (\frac{ER(ea1)}{2})$$

$$ER(ea1) = 1 - d + d \times (\frac{ER(ea)}{Nea + \sum_{d_k} nbed_k}) = 1 - d + d \times (\frac{ER(ea)}{Nea + (3)})$$

$$= 1 - d + d \times \frac{ER(ea)}{4+3} = 1 - d + d \times \frac{ER(ea)}{7}$$

$$ER(eb) = 1 - d + d(\frac{ER(ea)}{7} + \frac{ER(ea1.2)}{1} + \frac{ER(ed)}{3} + \frac{ER(ed2.1)}{1})$$

Nea est représenté par (*ea1,ea2,eb,eb1*)
Il ya un document pointé par *ea* : c'est le document *D2*
$nbed_k$ est le nombre d'éléments appartenant à *D2*. Il est égal à 3 (*ec,ec1,ec2*)

Le calcul de l'*élémentRank* de chaque élément se fait par itération jusqu'à convergence. Pour ce calcul nous avons utilisé l'équation 6. La valeur 0,1 attribuée au paramètre *d* permet une convergence au bout de la troisième itération. Pour ce calcul, on considère que la collection est composée de 4 documents. Chaque document XML a une probabilité *P(d) =1/4* d'être sélectionné par un utilisateur. Nous donnons le calcul de l'*élémentRank* de l'élément *eb* à l'étape 1 et à l'étape 2 de l'itération :

Etape1 : $$ER(eb) = 1 - d + d(\frac{ER(ea)}{7} + \frac{ER(ea1.2)}{1} + \frac{ER(ed)}{3} + \frac{ER(ed2.1)}{1})$$

$$= 0,9 + 0,1(\frac{0,05}{7} + \frac{0,05}{1} + \frac{0,06}{3} + \frac{0,06}{1}) = 0,91$$

Etape2 : $$ER(eb) = 0,9 + 0,1(\frac{0,90}{7} + \frac{0,90}{1} + \frac{0,90}{3} + \frac{0,91}{1}) = 1,12$$

3.2 Pertinence d'un élément vis-à-vis d'une requête

Un élément appartenant à un document XML est évalué selon deux dimensions. Une première dimension dépend de la requête utilisateur. Elle est calculée sur la base du contenu et de la structure des documents XML. La deuxième dimension, indépendante des requêtes, indique l'*ElémentRank* de cet élément basé sur la connectivité du graphe des documents XML. Le score final d'un élément *e* vis-à-vis d'une requête utilisateur est obtenu par la combinaison des valeurs apportées par ces deux dimensions comme le montre l'équation 7.

$$Score_{final}(e,q) = \alpha\, score(e,q) + (1-\alpha)\, ER(e) \quad (7)$$

Où *score(e,q)* représente la pertinence d'un nœud *e* relativement à la requête *q*, calculée sur la base des liens structurels contenus dans un document XML. $\alpha \in]0,1]$ est un paramètre permettant de déterminer l'importance d'un élément en fonction des différents liens auxquels il participe. Sa valeur sera fixée par expérimentation. A l'issue de cette étape, le système retournera à l'utilisateur une liste contenant les éléments triés par ordre décroissant de leur score.

4 Illustration du calcul de l'*ElémentRank* pour l'évaluation du score d'un nœud.

Nous considérons pour ce calcul les documents *article1.xml(d1)* et *article2.xml(d2)*, extraits d'un document de cisco academie et la requête *Q(paquet, routeur)* contenant deux termes. Les arbres des documents *article1.xml* et *article2.xml* sont donnés respectivement par la figure 4 et la figure 5.

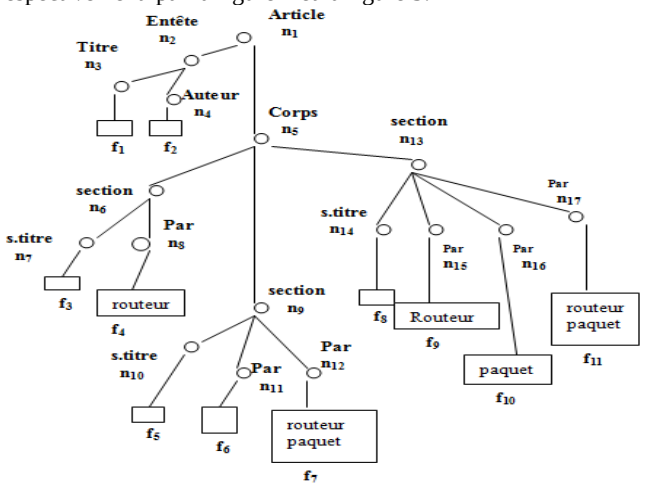


Fig. 4. Arbre du document *article1.XML*

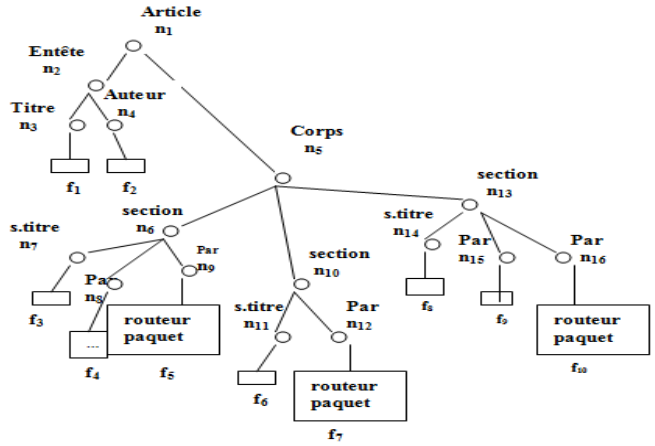


Fig. 5. Arbre du document *article2.XML*.

Nous considérons maintenant le graphe XML représenté par tous les nœuds des documents *article1.xml* et *article2.xml*, par les liens hiérarchiques reliant les nœuds dans chaque document et par les liens de référence suivants définis par *XPointer* :

$n_8^{d2} \rightarrow n_{12}^{d1}$

$n_9^{d2} \rightarrow n_{12}^{d1}$

$n_{15}^{d2} \rightarrow n_{12}^{d1}$

Calculons l'*ElémentRank* des nœuds du graphe XML. Nous considérons dans le premier cas uniquement les liens de référence et nous utiliserons l'équation 5 puis dans le deuxième cas nous prenons en compte les liens de référence et les liens hiérarchiques et nous appliquerons l'équation 6.

Cas1 : Prise en compte des liens de référence dans le calcul de l'EléméntRank.

Le tableau 1 donne deux listes, contenant chacune les nœuds triés par ordre décroissant des scores. Les deux listes sont basées sur le calcul *Rsv (Q, noeud)* en prenant en compte le contenu et les liens structurels. La deuxième liste intègre les liens de référence.

Table 1. Comparaison des résultats obtenus avec et sans prise en compte des liens de référence.

Nœuds résultat triés par ordre décroissant des scores	
Sans l'*ElémentRank*	Avec l'*ElémentRank*
- n_{12}^{d2}	- n_{12}^{d1}
- n_{16}^{d2}, n_{12}^{d1}	- n_{12}^{d2}
- n_{17}^{d1}, n_9^{d2}	- n_{16}^{d2}
- n_{13}^{d1}	- n_{17}^{d1}, n_9^{d2}
- n_8^{d1}	- n_{13}^{d1}
	- n_8^{d1}

Nous constatons que l'ordre des nœuds a changé. Le score initial du nœud n_{12}^{d1}, basé sur le contenu et les liens structurels est inférieur à celui du nœud n_{12}^{d2}. Les références au nœud n_{12}^{d1} définies plus haut ont permis d'augmenter son score et lui attribuent ainsi un meilleur classement dans la liste résultat retournée à l'utilisateur. Les liens de références *Xlink* et *Xpointer* véhiculent une information traduisant le lien thématique reliant les éléments entre eux. Par analogie au principe du *PageRank*, un élément pointé par plusieurs éléments est un élément important. De ce fait, il voit son *ElémentRank* augmenter. Nous utilisons cet *ElémentRank* comme critère permettant de calculer l'importance d'un nœud relativement à une requête. La pertinence d'un

élément ne dépend plus uniquement de la répartition des termes de la requête dans les différents éléments d'un arbre XML et des liens hiérarchiques définissant sa structure mais, désormais, elle dépend également de son *ElémentRank*. La liste résultat, représentée dans le tableau 1, obtenue sur la base de ces deux critères est donc meilleure et répond mieux au besoin des utilisateurs.

Cas2 : Prise en compte des liens hiérarchiques et des liens de référence dans le calcul de l'ElementRank

Le tableau 2 donne deux listes, contenant chacune les nœuds triés par ordre décroissant des scores. Les deux listes sont basées sur le calcul *Rsv (Q, noeud)* basé sur le contenu et les liens structurels. La deuxième liste tient en compte également des liens de référence et des liens hiérarchiques dans le calcul de l'ElémentRank.

Table 2. Comparaison des résultats obtenus avec et sans prise en compte des liens de référence et des liens hiérarchiques.

Nœuds résultat triés par ordre décroissant des scores	
Sans l'*ElémentRank*	Avec l'*ElémentRank*
- n_{12}^{d2}	- n_{12}^{d1}
- n_{16}^{d2} , n_{12}^{d1}	- n_{12}^{d2}
- n_{17}^{d1} , n_{9}^{d2}	- n_{16}^{d2}
- n_{13}^{d1}	- n_{9}^{d2}
- n_{8}^{d1}	- n_{13}^{d1}
	- n_{17}^{d1}
	- n_{8}^{d1}

Nous constatons que lors de la prise en compte des liens hiérarchiques dans le calcul de l'*ElémentRank*, les nœuds enfants voient leur *ElémentRank* diminuer par rapport à celui de leur nœud parent et celui leur nœud ancêtre si le nombre de leur frère est grand. C'est le cas du nœud n_{17}^{d1}. *ER*(n_{17}^{d1}) est inférieur à *ER*(n_{13}^{d1}) et il est inférieur à *ER*(n_{5}^{d1}). Le nœud n_{13}^{d1} obtient ainsi un meilleur classement que le nœud n_{17}^{d1} dans la liste résultat retournée à l'utilisateur comme le montre le tableau 2. Ce qui représente une perte au niveau de la spécificité des nœuds vis-à-vis d'une requête. Nous remarquons également que les nœuds n_{17}^{d1} et n_{9}^{d2} initialement avaient le même score mais avec la prise en compte des liens hiérarchiques le score du nœud n_{9}^{d2} est plus important que ce lui du nœud n_{17}^{d1}. Leur *ElémentRank* est différent car ces nœuds n'ont pas le même nombre de frères (n_{17}^{d1} possède 3 frères et n_{9}^{d2} possède 2 frères).

La valeur de l'ElémentRank d'un nœud calculée en prenant en compte les liens structurels dépend du nombre de frères qu'il possède et de sa position dans la

structure hiérarchique du document XML. Il semble donc que la prise en compte des liens hiérarchiques dans le calcul de l'*ElémentRank* pourrait biaiser sa valeur.

5. Conclusion

La structure des documents XML ne se limite plus à une structure hiérarchique. La pertinence d'un élément ne dépend plus uniquement de la répartition des termes de la requête dans les différents nœuds d'un arbre XML et de ses liens structurels. Les liens de référence *Xlink* et *Xpointer* véhiculent une information permettant de définir d'une autre manière le calcul de la pertinence d'un élément. Dans cet article, nous avons intégré les liens de référence comme un autre critère permettant la recherche des éléments les plus exhaustifs et les plus spécifiques. Ce critère tient compte de la connectivité du graphe XML pour calculer l'*ElementRank* de tout élément du graphe. L'*ElementRank* modifie l'importance d'un élément. La combinaison des valeurs apportées par les liens structurels et les liens de référence est calculée par le système afin de retourner à l'utilisateur les éléments les plus pertinents relativement à son besoin.
Il serait intéressant de mesurer l'*ElémentRank* d'un élément en considérant uniquement les nœuds importants le référençant. Ce calcul pourrait également se faire en utilisant les liens entrants et les liens sortants d'un élément. Une implémentation et des tests sur une collection de documents permettraient de mesurer l'apport de l'*ElementRank* sur la sélection d'éléments pertinents vis-à-vis des méthodes classiques et des méthodes intégrant les liens XML.

5. Bibliographie

1. Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval.Technical report, FERMI ESPRIT BRA 8134, University of Glasgow, 1996.
2. H.cui, J-R.Wen, J-R.Chua, "Hierarchical indexing and flexible element retrieval for structured document", april 2003.
3. N. Fuhr and K. Grossjohann. XIRQL : a query language for information retrieval in XML documents. In Proceedings of SIGIR 2001, Toronto, Canada, 2003.
4. Karen Sauvagnat. Modèle flexible pour la recherche d'Information dans des corpus de documents semi structurés. Thèse de doctorat, IRIT, Université Paul Sabatier de Toulouse, 2005.
5. S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW8, 1998, 107-117.
6. Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environnement. Journal of the ACM, vol. 46, Septembre 1999, pp.604-632.
7. J. Savoy, J. Picard: Retrieval effectiveness on the web. Information Processing & Management, vol. 37, 2001, 543-569.
8. Lin Guo Feng Shao Chavdar Botev Jayavel Shanmugasundaram, XRANK: Ranked Keyword Search over XML Documents, 2003.
9. Benny Kimelfeld, Eitan Kovacs, Yehoshua Sagiv, Dan Yahav, Using language models and the Hits Algorithm for XML Retrieval, In INEX 2006, pp 253-260, Heidelberg 2007.

Exploitation des liens dans la recherche d'informations dans les documents XML

Samia Berchiche-Fellag[1], Mohand Boughanem[2]

[1] Université Mouloud Mammeri de Tizi-Ouzou, 15000 Tizi-Ouzou, Algérie.
samfellag@yahoo.fr
[2] IRIT - SIG-RI, 118 Route de Narbonne, 31 062 Toulouse Cedex 4. bougha@irit.fr

Résumé. L'exploitation des liens dans la recherche d'information structurée(RIS) est importante pour deux raisons : en premier elle permet d'améliorer l'ordonnancement ou le ranking des éléments pertinents déjà retrouvés en second elle permet de retrouver des éléments inaccessibles par les méthodes classiques de recherche. Dans ce papier, nous proposons d'exploiter le contenu informationnel du lien, ainsi que le contenu de la balise titre des documents en attribuant un score à chacune de ses représentations et montrer ainsi que la prise en compte de ces deux paramètres est importante pour retrouver les éléments pertinents.

Mots clés : Recherche d'information structurée(RIS), XML, lien, balise titre

KeyWords: Structured information retrieval, XML, Link, title tag

1 Introduction

La recherche d'informations sur le web est basée sur les liens hypertextes. Ces derniers permettent de lier les pages web entre elles, ce qui octroie aux utilisateurs la possibilité de cliquer sur ces liens pour naviguer d'une page à une autre.
Un lien d'une page *p* vers une page *p'* peut être perçu comme une approbation de *p* pour le contenu de *p'*. En se basant sur cette idée, plusieurs travaux ont été réalisés notamment dans la RI sur le web, pour montrer que l'exploitation des liens permet non seulement d'améliorer le score des pages web restituées par les moteurs de recherche mais aussi de retrouver des pages pertinentes inaccessibles par la recherche classique.

Les deux algorithmes les plus connus sont PageRank[1] proposé par Brin & Page et HITS[2] de Jon kleinberg. La popularité de ces deux algorithmes et le succès phénoménal du moteur de recherche Google, qui utilise PageRank, ont engendré un grand nombre d'algorithmes de recherche qui exploitent l'analyse des liens comme Tophits[3], trafficrank[4], Trustrank[5] et Pathrank[6].
Comme pour la RI sur le web, la communauté (RIS) ou RI dans les documents XML s'intéresse à l'exploitation des liens dans ses dits documents. Dans la RIS, le granule

d'information restitué suite à une requête donnée étant un élément de document et la pertinence dans le cadre de la RIS est évaluée selon deux grandeurs : l'*exhaustivité* et *la spécificité.* La *spécificité* mesure si tout le contenu du granule d'information concerne la requête. L'*exhaustivité* mesure si toutes les informations requises dans la requête sont présentes dans le granule d'information.

La problématique qui en découle est posée selon deux aspects :

- Peut on adapter les algorithmes de la RI sur le web à la RIS ?
- Quels sont les paramètres à prendre en compte pour exploiter efficacement les liens ?

Pour tenter de résoudre cette problématique, nous proposons d'une part d'adapter l'idée de Savoy[11] à la RIS et particulièrement d'autre part d'exploiter l'information textuelle portée par le lien ainsi que l'information portée par la balise titre du document XML.

L'objectif essentiel de ce papier est de montrer que la prise en compte conjointe des informations des paramètres titre et lien est primordial pour l'ordonnancement des résultats restitués en réponse à une requête donnée.

Pour bien illustrer notre travail, cet article est structuré comme suit : en section 2, nous présentons un bref aperçu sur les travaux relatifs à l'exploitation des liens dans les documents XML, nous détaillons en section 3 notre approche, s'en suivra un exemple illustratif en section 4, et nous clôturons par une conclusion et des perspectives à entreprendre pour enrichir ce travail.

2 Etat de l'art

Peu de travaux ont été proposés pour l'exploitation des liens en RIS, les quelques travaux qui s'y sont reportés s'inspirent globalement des algorithmes Hits et PageRank. XRank proposé par [7] inspiré de PageRank et adapté à une granularité plus fine que le document à savoir l'élément, est l'un des premiers travaux permettant l'exploitation des liens pour le réordonnancement des résultats dans les documents XML. Sa méthode consiste à calculer le score d'un élément en fonction de trois scores relatifs aux ensembles CE, HE, CE^{-1} (CE : liens hiérarchiques entre nœuds, HE : liens Xlink entre nœuds et CE^{-1}: le même ensemble CE sauf que le sens des liens est inversé). Les auteurs dans [8, 9] exploitent les liens pour le reranking des résultats en estimant deux paramètres « local indegree » qui représente le nombre de liens de la collection entrants à un article et « global indegree » qui représente le nombre de liens entrants à un article à partir des documents renvoyés comme résultats à une requête donnée. Les auteurs dans [10] utilisent le modèle de langage et l'algorithme Hits en appliquant deux méthodes d'évaluation pour le filtrage des documents et le classement des éléments. La première méthode est une interpolation linéaire des modèles de langage, à savoir, le corpus, le document et l'élément. La deuxième méthode est basée sur l'application de l'algorithme Hits en combinaison avec l'approche du modèle de langage.

La méthode que nous proposons est basée sur l'activation propagée proposée par Savoy [11]. Savoy propose de transmettre une fraction de score de chacune des pages extraites du Web vers ses voisines (les pages liées directement). Un modèle de

recherche est utilisé pour obtenir une liste triée de pages Web, et en se basant sur cette liste, les scores des documents seront propagés selon les hyperliens sans tenir compte de leur orientation (entrants ou sortants).

3 Approche proposée

Deux types de liens sont à considérer dans les documents XML, les liens intra documentaires et les liens extra documentaires. Les liens intra documentaires sont représentés par XPOINTER[12], et les liens extra documentaires sont référencés par XLINK[13] voir exemple de documents avec Xlink et Xpointer ci dessous. On se propose d'étudier dans ce papier les liens XLINK.

Exemple 1. Document XML avec XLINK

```
  <?xml version="1.0" encoding="utf-8" ?>
  <article>
 <titre id="2305">Physique nucléaire</titre>
 <body>: le
  <collectionlink  xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="6147.xml">noyau atomique</collectionlink>
<section>
  <titre>Cohésion du noyau</titre>
<p>
  A l'intérieur du noyau, les
  <collectionlink xmlns:xlink="http://www.w3.org/1999/xlink"  xlink:type="simple"
xlink:href="6718.xml">nucléon</collectionlink> s sont soumis à: l'
  <collectionlink  xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="41704.xml">interaction forte</collectionlink>
  <collectionlink  xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="14449.xml">Loi de Coulomb</collectionlink>
...
</p>
...
</section>...
...
</body>
...
</article>
```

Exemple 2. Document XML avec XPOINTER

```
  <?xml version="1.0" encoding="utf-8" ?> <article>
 <titre id="2305">Physique nucléaire</titre>
 <body>: le
  <collectionlink  xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="6147.xml">noyau atomique</collectionlink>
<section>
  <titre>Cohésion du noyau</titre>
<p>
```

A l'intérieur du noyau, les
<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="6718.xml#xpointer(id('intéraction magnétique'))"
...
</p>
...
</section>
...
</body>
...
</article>

L'exploitation des liens dans notre approche est réalisée lors de la phase de recherche. Rappelons qu'en RIS, le granule d'information considéré n'est plus le document entier mais une partie de celui-ci, en l'occurrence l'élément.

Soit C un corpus de documents XML. Un document d dans C est un ensemble d'éléments e. Un élément e est un sous arbre XML représenté par un ensemble de termes t_i pondérés par un poids p_i obtenus par la propagation de termes bien distribués dans les éléments enfants de e et respectant la contrainte suivante: le poids moyen de t_i dans ses éléments enfants doit être compris entre le poids moyen et le poids maximal de tous les termes de ses éléments enfants, pour plus de détails voir l'article[14].

Nous présentons dans ce qui suit notre approche pour la prise en compte des liens dans l'interrogation des documents XML.

Soit q une requête utilisateur représentée par un ensemble de termes éventuellement pondérés $q = \{(t_1,p_{q1}), \ldots\ldots ,(t_n,p_{qn})\}$ avec t_k un terme de la requête et p_{qk} le poids de t_k dans la requête q et n le nombre de termes dans la requête.

La recherche d'éléments réponses se fait en deux phases : la première phase consiste à retrouver les éléments réponses à la requête q donnée sans prise en compte des liens en interrogeant le corpus C, la seconde phase consiste à exploiter les liens pour réordonnancer les éléments trouvés et pour retrouver des éléments non trouvés initialement.

Nous allons mettre l'accent sur cette deuxième phase et montrer ainsi que la prise en compte de l'information portée aussi bien par le lien que par le titre est importante pour une restitution correcte des résultats.

Suite à l'interrogation du corpus C à travers une requête q, un ensemble E d'éléments réponses est retourné dans lequel chaque élément e est représenté par un score de pertinence appelé $\mathbf{Score_{elt}}$[14]. Les éléments réponses de l'ensemble E sont obtenus **sans** prise en compte des liens.

Nous allons à présent exploiter les liens portés par chaque élément.

Un élément e peut référencer plusieurs documents de C à travers des liens, ces liens peuvent se rapporter à la requête comme ils peuvent ne pas s'y rapporter, il serait judicieux de pouvoir différencier entre ces deux types de liens.

A cet effet, nous proposons d'exploiter le contenu informationnel du lien et le contenu de la balise titre du document cible, l'intuition que nous avons suivie est « *si des termes de la requête sont présents dans le lien, le document référencé se rapporte forcément à la requête, et si de plus les termes de la requête se retrouve dans son titre*

alors ce document ne peut être que pertinent pour la requête et de ce fait ces éléments le sont aussi ».

Nous définissons à cet effet deux scores, $\mathbf{score_{Link}}$ qui mesure le contenu informationnel du lien et $\mathbf{score_{titre}}$ qui mesure l'information portée par la balise titre du document référencé,

$$\mathbf{score_{link}} = \frac{\sum_{i=1}^{n} \mathbf{tf_{qi}}}{\sum_{j=1}^{m} \mathbf{tf_{linkj}}} . \qquad (1)$$

avec tf_q : fréquence du terme de la requête figurant dans le lien
tf_{link} : fréquence du terme dans le lien
n : nombre de termes de la requête figurant dans le lien
m : nombre de termes total du lien et $n <= m$

$$\mathbf{score_{titre}} = \frac{\sum_{i=1}^{n} \mathbf{tf_{qi}}}{\sum_{j=1}^{m} \mathbf{tf_{titrej}}} . \qquad (2)$$

avec tf_q : fréquence du terme de la requête figurant dans le titre
tf_{titre} : fréquence du terme dans le titre du document référencé
n : nombre de termes de la requête figurant dans le titre
m : nombre de termes total du titre et $n <= m$

Les liens exploités dans notre cas sont les liens **entrants** vers un document, l'intuition qui nous a guidée est: *le concepteur d'un document se rapportant à un sujet donné ne fait référence à un autre document que s'il le considère comme pertinent sur ce même sujet*.

Chaque document cible aura pour score :

$$\mathbf{Score_{doc}} = \alpha \mathbf{Score_{titre}} + \beta \sum_{i=1}^{k} \mathbf{Score_{linki}} + \gamma \sum_{i=1}^{k} \mathbf{Score_{refi}} . \qquad (3)$$

Avec $score_{ref}$ étant le score de la source qui a référencé le document, c'est le score de l'élément si c'est la première référence c'est le score du document par la suite.
k : le nombre de liens entrants vers le document.
Les paramètres α, β et γ indiquent l'importance accordée à chaque score et prennent leurs valeurs dans l'intervalle [0,1]. Les valeurs finales de ses paramètres vont être fixées par expérimentation.

Le résultat obtenu à l'issue de cette étape est un ensemble *D* de documents ordonnés par ordre décroissant des scores.

Comme rappelé précédemment, l'unité d'information à restituer dans la RIS étant l'élément il est impératif d'identifier à partir de *D* les éléments pertinents, comme chaque document du corpus est représenté par un ensemble d'éléments, l'identification des éléments devient aisée. Néanmoins, le score sera calculé comme suit :

$$\mathbf{Score}_{\mathbf{elt}} = \mathbf{Score}_{\mathbf{eltinitial}} + \mathbf{Score}_{\mathbf{doc}} \quad (4)$$

Le $score_{eltinitial}$ est le score initial de l'élément calculé dans la première phase de recherche.

Algorithme de recherche

Données : *C* = {*d* / *d*= ensemble d'élément *e*}, *e*={(t_1,p_1), …… ,(t_n,p_n)/ *n* :nombre de terme représentants l'élément *e*}, *q* = {(t_1,p_{q1}), …… ,(t_n,p_{qm})}/ *m* : nombre de termes de la requête}
Résultats : Ensemble *E* d'éléments réponses

1. Pour chaque *e* d'un document *d* du corpus *C*
   1.1 calculer le $score_{elt}$ de *e* avec la requête *q*
   1.2 sauvegarder dans *E* par ordre décroissant du $score_{elt}$
2. Pour chaque *e* de l'ensemble *E*
   2.1 retrouver tous les liens sortants vers les documents de *C*
   2.2 pour chaque document cible jusqu'à 5 itérations
      2.2.1 calculer $score_{titre}$ avec la formule *(1)*
      2.2.2 calculer $score_{link}$ de tous les liens entrants vers ce dernier avec la formule *(2)*
      2.2.3 calculer $score_{doc}$ avec la formule *(3)*
      2.2.4 Sauvegarder chaque document dans *D* par ordre décroissant des scores.
3. Pour chaque document *d* de *D*
   3.1 identifier les éléments *e*
   3.2 pour chaque élément *e*
      3.2.1 calculer $score_{elt}$ avec la formule *(4)*
      3.2.2 sauvegarder dans *E* par ordre décroissant des scores.

4 Exemple illustratif

Pour des simplifications de calcul, nous avons considérés que la balise titre et le link contiennent chacun au maximum trois termes de fréquence 1 chacun.
Soit *q* une requête contenant trois termes pondérés à 1 chacun, $q = \{t_1,1), (t_2,1), (t_3,1)\}$ et soit *E* l'ensemble d'éléments réponses à *q*, $E = \{e_1, e_2, e_A, e_3, \ldots, e_z / scoree_A =$

7} et soit à considérer les liens référencés par l'élément e_A qui est un élément du document *H*.
Le titre = ϕ signifie que le titre ne contient aucun terme de la requête.

Figure 1. Graphe représentant les liens entrants et sortants de documents XML liés.

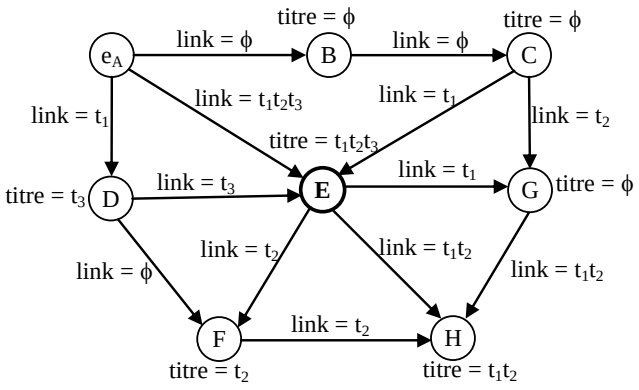


Tableau 2. Résultats des scores des documents après exploitation des liens.

Documents	**Score** $\alpha = 0,6$, $\beta = 0,8$ et $\gamma = 0,1$	**Score avec** $\alpha = 0$, $\beta = 0$ et $\gamma = 0,1$
e_A	7	7
B	0,7	0,7
C	0,07	0,07
D	1,16	0,7
E	**2,75**	**0,77**
F	0,86	0,15
G	0,82	0,08
H	2,17	0,10
Score final de e_A	7+2,17=9,17	7+0,10=7,10

L'ensemble *D* obtenu pour les valeurs de $\alpha = 0,6$, $\beta = 0,8$ et $\gamma = 0,1$ est : ***D={E, H, D, F, G, B, C}***. Ces résultats semblent cohérents, car on constate que l'intuition qui nous a guidée est confirmée, à savoir qu'un document référencé par un document pertinent, est aussi pertinent puisque ce dernier lui propage son score. Nous constatons aussi que le l'information porté par le lien et le titre est très importante lorsqu'on constate que les documents B et C sont mal classés car n'ayant aucune information concernant les termes de la requête aussi bien dans le link que dans le titre, les documents E et H se voient placés en tête de liste car leurs titre et leur lien sont porteurs d'informations pertinentes.

Suite à l'évaluation avec plusieurs valeurs de γ, dont un exemple a été présenté ici pour les valeurs de $\alpha = 0$, $\beta = 0$ et $\gamma = 0,1$ et qui a comme résultat l'ensemble ***D= {E, D, B, F, H, G, C}***, nous déduisons que la non prise en compte des paramètres titre et link, donne des résultats erronés lors de l'ordonnancement des documents et ce quelques soit la valeur donnée au paramètre γ. Nous constatons d'ailleurs l'incohérence des résultats dans l'ensemble *D* lorsqu'on voit que le document B dont le titre et le link n'ayant aucun terme de la requête classé avant le document H dont le titre et les liens entrants sont porteurs d'informations pertinentes.

5 Conclusion

Dans cet article nous avons présenté notre contribution au problème de l'exploitation des liens dans les documents XML en proposant une propagation d'une fraction de score de chacun des éléments extraits du corpus vers ses voisins. Un modèle de recherche est utilisé pour obtenir une liste triée d'éléments, et en se basant sur cette liste, les scores des documents sont propagés selon les liens Xlink entrants vers les documents. En plus du score propagé, nous avons aussi évalué l'information portée par le link et celle portée par la balise titre des documents référencés en leur attribuant un score. Au final, le score de chaque document est calculé en fonction du score propagé, du score link et du score titre. L'exemple illustratif que nous avons présenté montre que l'approche répond correctement à l'objectif assigné et que la prise en compte conjointe des informations portées par le lien et le titre sont importantes pour avoir un bon ordonnancement des résultats. Les perspectives à envisager pour enrichir ce travail consistent en la mise en œuvre de cette approche et son expérimentation dans le cadre de la compagne d'évaluation d'INEX.

6 Bibliographie

1. S. Brin, L. Page.: The anatomy of a large-scale hypertextual web search engine. In Proceedings of the 7th international conference on World Wide Web (WWW), pages 107–117, Brisbane, Australia, 1998.
2. J. Kleinberg.: Authoritative sources in a hyperlinked environment. In Proc. 9th Annual ACM-SIAM Symposium Discrete Algorithms, pages 668–677, 1998.
3. T. Kolda, B. Bader.: The tophits model for higher-order web link analysis. In Workshop on Link Analysis, Counterterrorism and Security, 2006.
4. J. A. Tomlin.:A new paradigm for ranking pages on the world wide web. In: Proc. of the 12th International World Wide Web Conference, pages 350–355, 2003.
5. Z. Gy¨ongyi, H. Garcia-Molina, J. Pedersen.: Combating web spam with trustrank. In: Proc. of the 30th International Conference on Ver Large Databases, pages 576–587, 2004.
6. J. Li, Y. Zhao.: Pathrank :Web page retrieval with navigation path. In: ECIR '09 : Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, pages 350–361, Berlin, Heidelberg, 2009. Springer-Verlag.
7. G. Lin, S. Feng, B. Chavdar, S. Jayavel.: Xrank : Ranked search over xml documents. In SIGMOD'2003, San diego,CA, 2003.

8. H. Elghazel, K. Idrissi, A. Baskurt, C. B. Amar.: Approche textuelle pour la recherche d'image. In : 3rd International Conference SETIT'05, 2005.
9. J. Kamps, M. Koolen.: The importance of link evidence in wikipedia. In: Lecture Notes in Computer Science, pages 270–282, Heidelberg, 2008.
10. B. Kimelfeld, E. Kovacs, Y. Sagiv, D. Yahav.: Using language models and the hits algorithm for xml retrieval. In INEX 2006, pages 253–260, 2007.
11. J. Savoy, Y. Rasolofo.: Hyperliens et recherche d'information sur le web. In : 7eme journée internationales d'Analyse statistique des Données Textuelles, 2004.
12. P. Grosso, E. Maler, J. Marsh, N. Walsh.: Xml pointer language (xpointer). Technical report, World Wide Web Consortium (W3C), W3C Recommendation, 2003.
13. S. Deroze, E. Maler, D. Orchard.: Xml linking language (xlink). Technical Report 1.0, World Wide Web Consortium (W3C), W3C Recommendation, juin 2001.
14. Samia Berchiche-Fellag, Mohand Boughanem.: Traitement des requêtes CO (Content Only) sur un corpus de documents XML, In : Colloque sur l'Optimisation et les Systèmes d'Information 2010.

Measuring Semantic Proximity between Flexible Queries: A Distance-Based Approach

AGGOUNE Aicha [1], HADJALI A[2], MOUSSAOUI A[1]
[1] University of Sétif, Algeria
[2] IRISA/ENSSAT, University of Rennes 1
aggoune_ai@yahoo.fr

Abstract. We investigate the problem of evaluating the semantic proximity between queries involving fuzzy predicates. We use a particular distance, called the Hausdorff distance, to define this proximity. We show the usefulness of our approach for dealing with failing queries, i.e., queries that produce empty answers. The solution proposed leverages the proximity measure defined and a past query workload.

Keywords: Flexible queries, distance, proximity, empty answers

1 Introduction

Semantic distance has been used in the field of database since a long time Ichikawa [17] and Motro [3]. In these works, the notion of distance is used to determine the similarity between data values where each database domain is provided with a definition of distance between its values. In [10], a model of semantic distance is proposed in which a graph-based approach is used to quantify the distance between two data values organised as a direct graph. See also the work done in [12] for a distance-based approach to repairing inconsistent databases.

On the other hand, similarity between queries has also attracted the attention of many researchers [15][13][1]. For instance in [1], a similarity measure between the initial user query and some predefined successful queries is developed in the context of mediator system. That measure relies on a hierarchy of concepts, the more characteristics two concepts share, the closer they are. The aim of this work is to help a user to reformulate his/her query when it fails. Following the same objective, a recent approach to measure the similarity between fuzzy queries has been proposed in [16]. This work assumes that for each domain of values of an attribute (involved in the target database) a resemblance relation is available. Unfortunately, in practice, this strong assumption is not always fulfilled.

Starting from the assumption that proximity measures between fuzzy sets are generally derived from a distance function underlying the universe, we may use a particular distance to estimate the extent to which two (atomic or compound) queries are close or not, semantically speaking. That distance is the so-called Hausdorff distance. The Hausdorff distance is one popular way to extend the distance measure to subsets of a metric space. As pointed out in [5][4][9], the Hausdorff distance can be seen as a measure of how much two non-empty compact (closed and bounded) sets A and B in a metric space resemble each other with respect to their positions. Let us

note that this kind of distance has been generalized to fuzzy sets [5][7]. Two options are possible: one may look for a scalar or a fuzzy evaluation. Here, we only focus on the scalar evaluation. We show how to exploit this distance measure for estimating the proximity between two flexible queries.

To illustrate the practical interest of our proposal, we consider the problem of failing flexible queries, i.e., queries that return an empty set of answers, and we show how such queries can be approximately answered. The idea is to leverage a query workload (i.e., a collection of queries that have been executed on the database system in the past and have produced non-empty answers) and the measure of semantic proximity between some queries of the workload and the failing query at hand. Furthermore, for the approximate answers returned, a ranking method is proposed.

The remainder of the paper is organized as follows. Section 2 reviews some related work. In section 3, we introduce the Hausdorff distance and we review the main approaches proposed to compute this distance between sets both in crisp and fuzzy cases. Section 3 discusses a way to estimate a proximity measure between flexible queries. In section 4, we show how the problem of empty answers can be approached using the proximity measure defined. An illustrative example is provided in section 6. Last, we conclude and draw some working directions for the future.

2 Preliminary Notions

2.1 Flexible Queries

A *fuzzy set* [6] F on a universe of discourse U is characterized by a mapping μ_F from U to $[0, 1]$, also called the membership function of F. For all u in U, $\mu_F(u)$ is called the membership degree of u to F. The core and the support of F are respectively defined as $C(F) = \{u \in U, \mu_F(u) = 1\}$ and $S(F) = \{u \in U, \mu_F(u) > 0\}$. If F is defined on a continuous (numerical) domain, we use the trapezoidal membership function (*t.m.f.*) to represent F, i.e., $F = (A, B, a, b)$ where $C(F) = [A, B]$ and $S(F) = [A–a, B+b]$. In case of discrete domains, we write $F = \{\mu_F(u_1)/u_1, \mu_F(u_2)/u_2, \ldots, \mu_F(u_n)/u_n\}$ where u_i (for $i=1,n$) is an element of U. In the following, F_α denotes a cut of level α of the fuzzy set F, i.e., $\{u \in U, \mu_F(u) \geq \alpha\}$.

Flexible queries [8] are requests that express user preferences by means of gradual predicates modeled by fuzzy sets. The user does not specify crisp conditions, but fuzzy ones whose satisfaction may be regarded as a matter of *degree*. Then, the result of a query is no longer a flat set of elements but is a set of discriminated elements according to their global satisfaction w.r.t. the fuzzy constraints appearing in the query. An example of a flexible query is: "retrieve the employees which are *young* and *well-paid*", where *young* and *well-paid* are respectively represented by the following *t.m.f.* $(0, 25, 0, 15)$ and $(5, \infty, 2, 0)$.

2.2 The Hausdorff Distance

First recall that a *distance* (or a *metric*) d is a mapping from $U \times U$ to R^+, such that

M_1) $\forall u, \forall v, d(u, v) = d(v,u)$ …… *(Symmetry)*
M_2) $\forall u, \forall v, \forall w, d(u, w) \leq d(u,v) + d(v, w)$ ……*(Triangle inequality)*
M_3) $\forall u, \forall v, \text{if } u \neq v \text{ then } d(u, v) > 0$ ……*(Distinguishability of non-identicals)*

M_4) $\forall u, d(u, u) = 0$……. *(Indistinguishability of identicals)*

One of the most popular measures of distance is the Euclidean distance defined as $d(u, v) = |u - v|$.

In the following, we recall the principle of the *Hausdorff distance* and we review the main approaches that can be followed to compute such a distance.

2.2.1 Crisp Sets. Consider two subsets A and B of a space U (equipped with a metric). The most popular scalar extension of distance between A and B is the *Hausdorff distance* defined as [5][7][11]:

$$d_H(A, B) = max\{H(A, B), H(B, A)\}, \quad (1)$$

Where $H(A, B)$ stands for the *directed Hausdorff distance* from A to B. We have $H(A, B) = Sup_{u \in A}\ d(u, B)$ and $d(u, B) = Inf_{v \in B}\ d(u, v)$. The expression $d(u, v)$ stands for a standard distance (such as Euclidean distance). Formula (1) can be written in the following condensed form:

$$d_H(A, B) = max\{sup_{u \in A}\ inf_{v \in B}\ d(u, v), sup_{v \in B}\ inf_{u \in A}\ d(u, v)\}. \quad (1')$$

The idea that governs this distance is the following: for each element in A look for the closest element in B, then check for the element in A for which the distance to the closest element in B is maximal. The same is done exchanging B and A and the *longest* distance of the two component is kept. Intuitively, if the Hausdorff distance isδ, then every point of A must be within a distance δ of some point of B and vice versa.

Note that d_H is a metric and, in particular, the following statement holds: $d_H(A, B) = 0$ if and only if $A = B$. Usually, the following equalities are assumed to be true $d_H(A, \varnothing) = d_H(\varnothing, B) = +\infty$ and $d_H(\varnothing, \varnothing) = 0$.

Example 1. Let $A = [a_1, a_2]$ and $B = [b_1, b_2]$ be two regular intervals and let $d(u, v) = |u - v|$. Then, it easy to check that $d_H(A, B) = max(|a_1 - b_1|, |a_2 - b_2|)$.

2.2.2. Fuzzy Sets. The Hausdorff distance between fuzzy sets can be either fuzzy or scalar. Hereafter, we only focus on the scalar version. For the fuzzy evaluation, more details are available in [7][4].

Dubois and Prade's Approach

In [7], the Hausdorff distance was generalized to fuzzy sets in the following way. Let F and G be two fuzzy sets on U, for any $r \in R^+$ let $D_r(F)$ defined by

$$\mu_{D_r(F)}(u) = sup_{v \in U}\{\mu_F(v) / d(u, v) \leq r\}.$$

Here $D_r(F)$, the dilation of F by r, is the result of applying to all points of F a local *max* operation within a region of radius r. Now, the generalized Hausdorff distance writes:

$$d_H(F, G) = inf\{r \in R^+ / F \subseteq D_r(G) \wedge G \subseteq D_r(F)\} \quad (2)$$

One can easily that the dilation is only "horizontal". Then, we may never be able to cover the other set and hence $d_H(F, G)$ cannot be defined. Thus, two fuzzy sets must

have the same *supremum* for the Hausdorff distance between them to exist; this is a serious drawback of this definition.

Chaudhuri and Rosenfeld's Approach

Another definition of the Hausdorff distance between two fuzzy sets is proposed in [4]. This definition is more general and is valid in the case of two fuzzy sets with unequal maximum memberships. More details about this case are available in [4]. In the following, we consider only fuzzy sets with the same supremum.

Let F and G be two discrete fuzzy sets. Let $T = \{t_1, t_2, \ldots, tm\}$ the set of all the distinct membership values of F and G. The Hausdorff distance between F and G is defined by the following expression:

$$d_H^2(F, G) = \frac{\sum_{i=1}^{m} t_i d_H(F_{t_i}, G_{t_i})}{\sum_{i=1}^{m} t_i} \qquad (3)$$

Where F_{t_i} (resp. G_{t_i}) stands for the t_i-level cut of F (resp. G). $d_H^2(F, G)$ can be seen as a membership-weighted average of the crisp Hausdorff distances between the level sets of the two fuzzy sets.

Example 2. Let $U = \{1, 2, 3, 4, 5, 6, 7\}$ be a universe of discourse. Let also F and G be two fuzzy sets on U defined as follows: $F = \{0.7/1, 0.2/2, 0.6/4, 0.5/5, 1/6\}$ and $G = \{0.2/1, 0.6/4, 0.8/5, 1/7\}$. Let all the distinct membership values of F and G, pooled together, be $\{\alpha_1 = 0.2, \alpha_2 = 0.5, \alpha_3 = 0.6, \alpha_4 = 0.7, \alpha_5 = 0.8, \alpha_6 = 1\}$. Then, we have $T = \{0.2, 0.5, 0.6, 0.7, 0.8, 1\}$.

Table 1. The Hausdorff distance between the α-cuts of F and G

α_i	F_{α_i}	G_{α_i}	$H(F_{\alpha_i}, G_{\alpha_i})$	$H(G_{\alpha_i}, F_{\alpha_i})$	$d_H(F_{\alpha_i}, G_{\alpha_i})$
0.2	{1, 2, 4, 5, 6}	{1, 4, 5, 7}	1	1	1
0.5	{1, 4, 5, 6}	{4, 5, 7}	3	1	3
0.6	{1, 4, 6}	{4, 5, 7}	3	1	3
0.7	{1, 6}	{5, 7}	4	1	4
0.8	{6}	{5, 7}	1	1	1
1	{6}	{7}	1	1	1

To compute the distance $d_H^2(F, G)$, we have first to evaluate $d_H(F_{\alpha_i}, G_{\alpha_i})$ for $i = 1,6$.

Let us give the details about the calculus of $d_H(F_{\alpha_1}, G_{\alpha_1})$. By formula (1), we have:

$d_H(F_{\alpha_1}, G_{\alpha_1}) = \max\{H(F_{\alpha_1}, G_{\alpha_1}), H(G_{\alpha_1}, F_{\alpha_1})\}$ with $F_{\alpha_1} = \{u1, u2, u3, u4, u5\} = \{1, 2, 4, 5, 6\}$ and $G_{\alpha_1} = \{v1, v2, v3, v4\} = \{1, 4, 5, 7\}$. Then, we have

$H(F_{\alpha_1}, G_{\alpha_1}) = \sup_{u \in F_{\alpha_1}} \inf_{v \in G_{\alpha_1}} d(u, v)$, with $d(u, v) = |u - v|$

$= \sup \{\inf(|u1 - v1|, |u1 - v2|, |u1 - v3|, |u1 - v4|),$
$\inf(|u2 - v1|, |u2 - v2|, |u2 - v3|, |u2 - v4|),$
$\inf(|u3 - v1|, |u3 - v2|, |u3 - v3|, |u3 - v4|),$
$\inf(|u4 - v1|, |u4 - v2|, |u4 - v3|, |u4 - v4|),$

$inf(|u5 - v1|, |u5 - v2|, |u5 - v3|, |u5 - v|)\ \}$
$= sup\ (0, 1, 0, 0, 1) = 1.$
In the same way, we can compute $H(G_{\alpha_1}, F_{\alpha_1}) = sup_{v \in G_{\alpha_1}} inf_{u \in F_{\alpha_1}} d(u, v) = 1.$
Hence, $d_H(F_{\alpha_1}, G_{\alpha_1}) = 1$. We proceed in a similar way for the other $d_H(F_{\alpha_i}, G_{\alpha_i})$'s. Those results are reported in Table 1. Then,
$d_H^2(F, G) = (0.2 \cdot 1 + 0.5 \cdot 3 + 0.6 \cdot 3 + 0.7 \cdot 4 + 0.8 \cdot 1 + 1 \cdot 1) / 3.8 = 8.1/3.8 \cong 2.13$
In case of continuous fuzzy sets, formula (3) is modified in the following form [7]:
$d_H^2(F, G) =$

$$\int_0^1 t d_H(F_t, G_t)dt \Big/ \int_0^1 t dt = 2\int_0^1 t d_H(F_t, G_t)dt \qquad (4)$$

Example 3. Let now U represent the numeric universe of discourse of the variable "*age*" of a person. Let also F = "*about thirty*" and G = "*between_26_and_28*" two fuzzy sets on U defined by the following two *t.m.f.*: $F = (30, 30, 3, 3)$ and $G = (26, 28, 1, 1)$. Now, we evaluate the distance between F and G using formula (4). First, let us precise that F_α and G_α are regular intervals and can be expressed as follows: $F_\alpha = [3\alpha + 27, 33 - 3\alpha]$; $G_\alpha = [\alpha + 25, 29 - \alpha]$. Then, one can easily check that
$d_H^2(F, G) = 2\int_0^1 t\, max\left(\left|(t + 25) - (3t + 27)\right|, \left|(29 - t) - (33 - 3t)\right|\right)dt = 7/2$
It has been pointed out in [10] that expression (3) (resp. (4)) is a metric and reduces to the classical Hausdorff distance when sets are crisp.

Remark 1. It is worth noticing that for our purpose the main drawback stemmed from the unequal maximum memberships is excluded. Indeed, all fuzzy sets that are considered are normalized (hence, with the maximum membership value equal to 1).

3 Measuring proximity

Taking into account the strong intuitive connection between proximity and distance, and by using the Hausdorff distance measure, we may want to estimate to what extent two fuzzy queries are close, semantically speaking.

3.1 Single Predicate Queries (SPQ)

Let $Q = P$ and $Q' = P'$ be two SPQ where P and P' are gradual predicates represented by means of fuzzy sets (of course, P and P' are pertaining to the same attribute, say A). To evaluate to which extent Q and Q' are close, semantically speaking, we make use of the Hausdorff distance index between the fuzzy predicates involved in those two queries. Then, we write:
$Dist(Q, Q') = d_H^2(P, P'),$ (5)
With $Dist(Q, Q')$ stands for a distance measure between Q and Q'. It is well known that the distance measure produces values that have the reverse ordering of proximity measures. The smaller the index $Dist(Q, Q')$, the closer Q and Q'.

Let now *Prox(Q, Q')* denote a proximity measure between *Q* and *Q'*. *Prox(Q, Q')* based on the distance *Dist(Q, Q')* can be obtained in two steps:

i) Normalizing *Dist(Q, Q')* with a function f_{norm} that reduces the range to the interval [0, 1].

ii) Under this assumption, *Prox(Q, Q')* can be defined in the following way:

$$Prox(Q,Q') = 1 - f_{norm}(Dist(Q,Q')) \qquad (6)$$

The well known expressions of f_{norm} are [4]:
$f_{norm}(x) = min(1, x)$ and $f_{norm}(x) = x/(1+x)$.
Another approach to defining a proximity measure is to use a conversion function on the distance measure. For instance, *Prox(Q, Q')* can be defined by [11]:

$$Prox(Q,Q') = \left(1 + \left(\frac{Dist(Q,Q'))}{s}\right)^{t}\right)^{-1}. \qquad (7)$$

The positive constants *s* and *t* adjust the size of the proximity measure. The simplest conversion function can be obtained by setting *s = 1* and *t = 1*. Note that formula (7) (resp. (6)) can be directly applied in case of point and range queries.

Example 4. Consider a relational database, say D, of employees of a large company. Assume that D has been queried by three users about the "*salary*" of some employees of interest, see Table 2.

Table 2. Examples of users queries (assume that the maximum salary is *10 €*).

User	Query	Fuzzy predicates
#1	Q1=" retrieve employees who win about 2.8 €"	about 2.8=(2.8, 2.8, 0.6, 0.6)
#2	Q2=" retrieve employees who win approximately 3 €"	approximately 3=(2.8, 3.2, 0.4, 0.4)
#3	Q3=" retrieve employees whose salary is very high"	very high=(6, 10, 1, 0)

Assume now that one is interested in retrieving from D the employees that are "*well-paid*" in the company. Assume also that "*well-paid*" is modelled by the following t.m.f. (5, 10, 2, 0). In some particular situations (as it will be explained further), it might be desirable to be able to estimate the semantic proximity between this new query, say *Q*, and the previous processed user queries. Using formula (4), we can easily check that

Dist(Q, Q1) = 6, *Dist(Q, Q2) = 17/3,* *Dist(Q, Q3) = 8/3.*

By formula (7), we obtain

Prox(Q, Q1) = 0.14, Prox(Q, Q2) = 0.15, Prox(Q, Q3) = 0.27.

One can observe that *Prox(Q, Q1) < Prox(Q, Q2) < Prox(Q, Q3)*. Hence, *Q* is closer to *Q3* than *Q2* (resp. *Q1*). In case where *Q* results in null answers, it is practically better to provide the answers to Q_3 as response to *Q* than nothing.

3.2 Compound Queries

Let D be a relational database containing n attributes $A_1, A_2,\ldots, A_n$ with $D(A_i)$ being the domain of values pertaining to A_i. We make the assumption that $D(A_i)$ is closed and bounded (resp. finite) if A_i is a continuous (resp. discrete) attribute.

Let also Q be a flexible compound query of the form $P_1 \wedge\ldots\wedge P_k$ where the symbol '$\wedge$' stands for the connector *'and'* (which is interpreted by the '*min*' operator) and P_i ($i=1,k$) is a fuzzy predicate pertaining to the attribute A_i. Let also Q' be a flexible query of the form $Q' = P'_1 \wedge\ldots\wedge P'_s$ where P'_j ($j=1,s$) is a fuzzy predicate pertaining to the attribute A_j. To evaluate the semantic proximity between the two compound queries Q and Q', we distinguish three cases:

Case 1: Q and Q' cover the same attributes exactly.

Case 2: Q' covers all the attributes specified in Q.

Case 3: Q' does not cover all the attributes specified in Q.

Case 1: Q and Q' cover the same attributes exactly. Hence, $Q' = P'_1 \wedge\ldots\wedge P'_k$ with $(P_i, P'_i) \in D(A_i)$, for $i=1,k$, which means that both P_i and P'_i are predicates that constrain the same attribute A_i. In this case, $Prox(Q, Q')$ is evaluated in a three-step procedure:

i) for each pair (P_i, P'_i), $i=1,k$, compute $Dist(P_i, P'_i) = d_H^2(P_i, P'_i)$;
ii) for each pair (P_i, P'_i), $i=1,k$, compute $Prox(P_i, P'_i)$ using, for instance, formula (7);
iii) then, compute $Prox(Q, Q') = min_{i=1,k}\, Prox(P_i, P'_i)$.

This procedure is formalized in Algorithm 1.

Algorithm 1 *Prox(Q, Q')*

Input: $Q = P_1 \wedge\ldots\wedge P_k$ and $Q' = P'_1 \wedge\ldots\wedge P'_k$

```
1.  begin
2.    for i = 1 to k do
3.        compute Dist(Pi, P'i);
4.        compute Prox(Pi, P'i);
5.    end for
6.    compute Prox(Q, Q') = min i=1,k Prox(Pi, P'i);
7.    return Prox(Q, Q');
8.  end
```

Output: Proximity between Q and Q': $Prox(Q, Q')$

Case 2: Q' covers all the attributes specified in Q. Then, Q' can be transformed into the following form $Q' = P'_1 \wedge\ldots\wedge P'_k \wedge P'_{k+1} \wedge\ldots\wedge P'_s$ (thanks to a simple operation of variable changing) with $(P_i, P'_i) \in D(A_i)$, for $i=1,k$. Attributes A_j (for $j=k+1,s$) are not specified in Q.

To estimate the proximity between Q and Q ', we can apply two strategies:

(S1) Increased Q: The idea is to complete Q with constraints on the missing attributes (ie, j for j = k +1 to s). Since the user does not specify any constraints on these attributes, all values of their fields are allowed. Thus, Q can be written:
$Q = P_1 \wedge ... \wedge P_k \wedge D(A_{k+1}) \wedge \ldots \wedge D(A_s)$.

With this rewriting of Q, we find the same conditions as in case 1. *Prox (Q, Q ')* can be obtained by applying the algorithm 1.

(S2) weakening of Q': The idea here is to focus solely on the attributes specified by the user and to measure the closeness of their constraints specified in Q and those present in Q'. Superfluous attributes present in Q' can be deleted. Q' is rewritten as follows:

$$Q'_R = P'_1 \wedge ... \wedge P'_k .$$

It is easy to see that Q'_R is a variant of relaxed Q '. The following relation $\Sigma_{Q'} \subseteq \Sigma_{Q'R}$ is always true (ie, the responses of *Q'* are also of Q'_R).
This strategy will take any interest in the processing of requests to empty answer, since the objective is to provide approximate solutions and alternatives to these queries. Thus, if the measure *Prox (Q, Q'_R)* is high (or above a certain threshold), it is perfectly acceptable to offer the answers of Q' as approximate answers to the query Q if its evaluation does produced no response (i.e., $\Sigma_Q = \varnothing$). *Prox(Q, Q'_R)* can also be calculated using the algorithm 1.

The Strategy S2 can measure the proximity between the constraints of the attributes present in both queries *Q* and *Q'*. This proximity can be characterized by local an interest which in practice is shown above. As for the strategy S1, it calculates a global proximity between *Q* and *Q'*. All attributes with those not specified in *Q* are taken into account in this calculation. As part of the problem of interest, this property presents a significant disadvantage. Indeed, suppose that *Q = "Salary = around 3 € k"* is an empty query answer.
Let *Q '= "Age = 27 years* $\wedge$ *Salary = [2.8, 3.2]*. The fact reflect the closeness of the value of *age= 27* and *D (Age) (ie P (27, D (Age))*, With D (Age) = [0, 100], significantly weakening the global proximity (based on min operator) between Q and Q '. This could lead to the rejection of Q' and therefore his answers so that they might find interesting because they approximately satisfy the criterion on the attribute "Salary ".

Case 3: *Q'* does not cover all the attributes specified in *Q*. Then, *Q'* can be transformed into the following form $Q' = P'_1 \wedge ... \wedge P'_b \wedge P'_{k+1} \wedge \ldots \wedge P'_s$, with $b < k$ and $(P_i, P'_i) \in D(A_i)$, for $i = 1$ *to* b, and the predicats P_j (for $j = k+1$ *to* s) are not specified in Q. To calculate *Prox(Q, Q')* :

(i) Added to Q' the domaines for missing attributs specified in Q, Q' is wiritten so
$Q' = P'_1 \wedge ... \wedge P'_b \wedge D(A_{b+1}) \wedge \ldots \wedge D(A_k) \wedge P'_{k+1} \wedge \ldots \wedge P'_s$;
(ii) Applying one of two strategies described in case 2.

4 Approaching the Empty Answers Problem

In this section, we show how the proximity measure introduced between queries can provide a basis for approaching the problem of empty answers. Let us first introduce this problem in the flexible database querying context.

4.1 Problem Formulation

Let Q be a *flexible* query. In this case, Σ_Q contains the items of the database that *somewhat* satisfy the fuzzy requirements involved in Q. Formally, $\Sigma_Q = \{t \in D / \mu_Q(t) > 0\}$, where t stands for a database tuple.

Definition. We say that Q results in *empty answers* if $\Sigma_Q = \varnothing$. [2]

This means that no data in the database somewhat satisfies the fuzzy conditions involved in Q. Let us note that *query relaxation* is one of the basic approaches that were proposed to deal with this problem. In a flexible queries context, a proximity relation-based approach for relaxing queries has been suggested in [14][15]. Unfortunately, this approach sometimes leads to a problem with a high computational complexity. In the following, we propose an alternative approach to deal with this problem. The approach proposed leverages the previous queries that have been evaluated by the system and have produced non-empty answers.

4.2 Principle of the Approach

Assume that we have at our disposal a past query workload $W(D)$, i.e., a collection of queries that have been executed on our database system in the past and have produced non-empty answers. Let $Q = P_1 \wedge \ldots \wedge P_k$ be a flexible query with empty answers.

The intuition of how we intend to use the workload $W(D)$ in answering Q is the following. The workload may perhaps reveal that, some queries that have been processed in the past and produced a set of non-empty answers are *close* to Q, *semantically speaking*. Thus, it is more convenient to propose as a response to Q the answers of the query that is the closest one to Q than "*nothing*". To achieve this, we proceed in the following way:

i) We first form the subset, say $W_Q(D)$, of the workload $W(D)$ which contains queries that involve all the attributes specified in Q, i.e.,

$W_Q(D) = \{Q' / Q' \in W(D) \wedge |D(Q') \cap D(Q)| = k\}$,

where $D(Q)$ denotes the list of the domains of attributes specified in Q and $|E|$ denotes the cardinality of the set E. It is assumed that $W_Q(D) \neq \varnothing$.

ii) For each Q' in $W_Q(D)$ we estimate the proximity measure $Prox(Q, Q')$ as explained in section 3. Based on this measure, we rank-order the elements of $W_Q(D)$ from the higher index to the smaller index. This ordering allows for discriminating the elements of $W_Q(D)$ from the closest one to Q to the least close one.

Remark 2. It is worthy to note that when applying Algorithm 1 to compute *Prox(Q, Q')*, one can estimate $Dist(P_i, P'_i)$, see step 3 of the Algorithm, only in terms of the *directed Hausdorff distance* from P'_i to P_i, i.e. $H(P'_i, P_i)$. Then, $Dist(P_i, P'_i) = d_H^2(P_i, P'_i)$ is now computed as in formula (3) (resp. (4)) where $d_H(A, B) = H(B, A)$. The main argument in favor of this reasoning is the fact that it is the set of answers to Q' which could be returned as alternative answers to Q and not the reverse. Then, only the measure to which extent Q' is far apart of Q is the most relevant.

iii) To provide alternative approximate answers to Q, we choose the top element, say Q_{App}, of the ordered set $W_Q(D)$ and return its set of answers as a response to Q.

Remark 3. Let us note that Q_{App} could be to some sense considered as a relaxation to the failing query Q. One can use the distance indices, computed between the predicates involved in Q_{App} and Q, to estimate the extent to which Q_{App} is a relaxation to Q.

In Algorithm 2, we summarize this above procedure.

Algorithm 2 *Approximate answers to Q*

Input: a failing query $Q = P_1 \wedge \ldots \wedge P_k$
the set $W_Q(D)$

1. **begin**
2. rankOrder($W_Q(D)$);
3. $Q_{App} = First(W_Q(D))$;
4. **return** $\Sigma_{Q_{App}}$;
5. **end**

Output: Approximate answers to Q: $\Sigma_{Q_{App}}$

Procedure *rankOrder(var set E)*

1. **begin**
2. **let** *Aux* = *E*; $E = \emptyset$;
3. **while** $Aux \neq \emptyset$ **do**
4. **begin**
5. $Q' = First(Aux)$;
6. compute *Prox(Q, Q')*;
7. **if** $Prox(Q, Q') > 0$ **then** $E = E \cup \{Q'\}$;
8. $Aux = Aux - \{Q'\}$;
9. **end while**
10. rank-order *E* in decreasing sense; *(# w.r.t. the proximity measure #)*
11. **end**

5 An Illustrative Example

To illustrate our proposal, let us consider a database D representing the JOB, EXPERIENCE and SALARY of seven EMPLOYEES, as shown in Table 3. Assume that the two following queries Q_1 = *"EXP=Around_4 ∧ SALARY=Between_4_5"* and Q_2 = *"EXP=Approximately_10 ∧ SALARY=About_5"* have been evaluated against D. For instance, Q_1 means that one is interested to retrieve employees whose experience is *about 4 years* and whose salary is *between 4 and 5 k€*. The fuzzy predicate *Around_4* (resp. *approximately_10*) is a discrete fuzzy set defined by *{0.5/3, 1/4, 0.5/5}* (resp. *{0.5/9, 1/10, 0.5/11, 0.2/12}*), while *Betwee_4_5* (resp. *About_5*) is a continuous fuzzy set defined by *(4, 5, 0.5, 0.5)* (resp. *(5, 5, 1, 1)*).

Assume also that the set of answers of each executed query is stored in the database workload W(D), see Table 4.

Table 3. A database of employees

EMP#	JOB	EXP	SALARY
T1	Vice president	15	7
T2	Design engineer	12	5
T3	System engineer	3	4.5
T4	Design engineer	5	3.8
T5	Accountant	10	4.7
T6	Secretary	5	6
T6	Software engineer	4	5.5

Table 4. Database workload

Query	Set of answers $\sum_Q$
Q1	{0.5/t3, 0.16/t4}
Q2	{0.2/t2, 0.7/t5}

Now, let us consider a new query *Q = "EXP=Close_to_6 ∧ SALARY=Well-paid"* where Close_to_6 = {0.2/4, 0.5/5, 1/6, 0.5/7, 0.2/8} and Well-paid = (7, 10, 1, 0). One can easily see that *Q* returns an empty set of answers when it is evaluated against D. To approximately answer *Q*, we apply our approach given in Section 4.

- By algorithm 1 and taking into account remark 2, we obtain:

  $Prox(Q, Q_1) = min(0.34, 0.17) = 0.17$, $Prox(Q, Q_2) = min(0.2, 0.18) = 0.18$

- By algorithm 2, we obtain $Q_{App} = Q_2$

Then, the approximate answers to *Q* to provide to the user are $\{t_2, t_5\}$.

6 Conclusion

In this paper, we have proposed an approach for estimating the semantic proximity between queries in a flexible database querying setting. The key concept of this approach is the Hausdorff distance. The approach proposed can apply both for point and range (crisp) queries as well. On the other hand, we have also shown how this proximity measure constitutes a valuable tool to approximately answer failing queries.

In this work, only attributes with domains endowed with a metric have been considered (quantitative attributes). It would be extremely interesting to extend the approach to qualitative attributes (as *color* attribute). We acknowledge also that some experimental studies are needed to demonstrate the efficiency and effectiveness of the approach.

7 References

[1] A. Bidault, C. Froidevaux, B. Safar: Similarity Between Queries in a Mediator. Proc. of ECAI 2002, pp. 235-239 (2002)

[2] A. Motro, FLEX: Tolerant and Cooperative User Interface to Database. IEEE Transactions on Knowledge and Data Engineering on Office Information Systems, vol. 4, pp. 231-246 (1990)

[3] A. Motro: VAGUE: A user interface to relational databases that permits vague queries. ACM Transactions on Office Information Systems, vol. 6, pp. 187-214 (1988)

[4] B.B. Chaudhuri and A. Rosenfeld: A modified Hausdorff distance between fuzzy sets. Information Sciences, Vol. 118, pp. 159-171 (1999)

[5] B.B. Chaudhuri and A. Rosenfeld: On a metric distance between fuzzy sets. Pattern Recognition Letters, Vo. 17, pp. 1157-1160 (1996)

[6] D. Dubois, H. Prade: Fundamentals of Fuzzy Sets. The Handbooks of Fuzzy Sets Series (D. Dubois, H. Prade, Eds), Vol. 3, Kluwer Academic Publi., Netherlands (2000)

[7] D. Dubois and H. Prade: On distances between fuzzy points and their use for plausible reasoning. In Proc. Int. Conf. on Systems, Man and Cybernetics, pp. 300-303 (1983)

[8] H. Larsen, J. Kacprzyk, S. Zadrozny, T. Andreasen, and H. Christiansen: Flexible Query Answering Systems. Recent Advances, Physica Verlag (2001)

[9] J. Fan: Note on Hausdorff-like metrics for fuzzy sets. Pattern Recognition Letters, Vo. 19, pp. 793-796 (1998)

[10] J. F. Roddick, K. Hornsby, D. de Vries: A Unifying Semantic Distance Model for Determining the Similarity of Attribute Values. ACSC'03, pp. 111-118 (2003)

[11] M.L. Puri and D.A. Ralescu: differentials of fuzzy functions. Journal of Mathematical Analysis and Applications, Vol. 91, pp. 552-558 (1983)

[12] O. Arieli, M. Denecker, M. Bruynooghe: Distance semantics for database repair. Ann. Math. Artif. Intell. Vol. 50, pp. 389-415 (2007)

[13] O.R. Zaïane, A. Strilets: Finding Similar Queries to Satisfy Searches Based on Query Traces. OOIS Workshops'02, pp. 207-216 (2002)s

[14] P. Bosc, A. Hadjali, and O. Pivert, O: Relaxation paradigm in a flexible querying context. In Proc. Int. Conf. on Flexible Query Answering Systems, pp. 39-50 (2006)

[15] P. Bosc, A. Hadjali, and O. Pivert: Weakening of fuzzy relational queries: An absolute proximity relation-based approach. Mathware & Soft Comp. J., Vol. 14, pp. 35-55(2007)

[16] P. Bosc, C. Brando, A. Hadjali, H. Jaudoin, O. Pivert: Semantic proximity between queries and the empty answer problem. In IFSA World Congress, 20-24 July, 2009, Lisbon, Portugal.

[17] T. Ichikawa and M. Hirakawa: ARES: A relational database with the capability of performing flexible interpretation of queries. IEEE Transactions on Software Engineering, vol. 12, pp. 624 -634 (1986)

Optimisation I

The adaptive method with hybrid direction for solving linear programming problems with bounded variables

M.O. Bibi and M. Bentobache

L.A.M.O.S., Laboratory of Modelling and Optimization of Systems,
University of Bejaia, 06000, Algeria;
Department of Technology,
University of Laghouat, Algeria;
mohandbibi@yahoo.fr, mbentobache@yahoo.com

Abstract. In this paper, we will suggest a new search direction for the adaptive method developed by R. Gabasov and F.M. Kirillova [9] for solving linear programming problems with bounded variables. An algorithm called the adaptive method with hybrid direction (AMHD) is described. In order to compare the suggested algorithm with the classical primal simplex algorithm employing Dantzig's rule (PSA), we have realized a numerical implementation under the MATLAB programming language. In the implementation of the two algorithms, we have used the LU factorization of the basic matrix to solve the linear equations systems and the Sherman-Morrison-Woodbury formula to update the LU factors [7]. The experimental study involves the cpu time and the iterations number of the two algorithms applied to solve randomly generated test problems of different dimensions and density of 5%. It has been shown that when the problem dimension increases, the superiority of AMHD over PSA increases. Particularly, for problems with dimension 1000×1000, our algorithm is approximately 7 times faster than the primal simplex algorithm with Dantzig's rule.

Keywords: Adaptive method, Hybrid direction, Long step rule, Simplex method, Computational experiments.

1 Introduction

Linear programming is a mathematical discipline which deals with solving the problem of optimizing a linear function under a domain delimited by a set of linear equations and inequations. The first formulation of an economical problem as a linear programming problem is done by L.V. Kantorovich (1939) and the general formulation is given later by G.B. Dantzig in his work [5]. LP is considered by the operations research community as the most important technique in

operations research. Indeed, it is widely used in practice and most of the optimization techniques are based on LP ones. That is why many researchers have shown great interest in finding efficient methods to solve LP problems. Although some methods have existed even before 1947, they were restricted to solve some particular forms of the LP problem. Being inspired by the work of J.B. Fourier on linear inequalities, G.B. Dantzig (1947) develops the simplex method. The simplex method is known to be very efficient to solve LP problems which arise in practice. However, in 1972, Klee and Minty have found an example where the simplex method takes an exponential time to solve it [15]. In 1979, Khachian developed the first polynomial interior point algorithm to solve LP problems [14], but it's not efficient in practice. In 1984, Karmarkar presents, for the first time, a polynomial interior point algorithm competitive with the simplex method on large problems [16].

In addition to the simplex method and its variants in which the search is done only by extreme points and interior point methods in which the search is done only by interior points, there exist other methods that use both interior and extreme points. We can cite: support methods, Gabasov and Kirillova (1977) [9]; interior search methods within the simplex framework, Mitra et al. (1988) [19]; active set methods, Gill et al. (1973) [11,12], Gondzio (1996) [13], etc.

In [9], the authors have developed the support method which is a generalization of the simplex method. The principle of this method is to start by a support feasible solution which comprises a basis and a feasible solution and to go through interior or extreme points to achieve the optimal one. Later, they have developed the adaptive method to solve, particularly, linear optimal control problems and they have extended it later to solve general linear and convex quadratic problems. An experimental study on generated sparse LP problems has shown its efficiency [17].

In this work, we will suggest a new search direction, called hybrid direction, for the adaptive method. This direction takes extreme values in order to bring some solution components to their bounds and it takes for some other components the reduced gradient values. In order to test the efficiency of the new suggested direction, we carry out an experimental study on a set of randomly generated problems.

The paper is organized as follows: in Section 2, we give some definitions. In Section 3, we present the main theory of the adaptive method with hybrid direction. In Section 4, experimental results are presented. Finally, the Section 5 is devoted to the conclusion.

2 State of the problem and definitions

Consider the linear programming problem with bounded variables presented in the following standard form:

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

$$\text{s.t. } Ax = b, \tag{2}$$

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

where c and x are n-vectors; b an m-vector; A an $(m \times n)$-matrix with $rankA = m < n$; l and u are n-vectors. In the following sections, we will assume that $||l|| < \infty$ and $||u|| < \infty$. We define the following sets of indices:

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

So we can write and partition the vectors and the matrix A as follows:

$$x = x(J) = (x_j, j \in J), \; x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \; x_B = x(J_B) = (x_j, j \in J_B),$$

$$x_N = x(J_N) = (x_j, j \in J_N); \; c = c(J) = (c_j, j \in J), \; c = \begin{pmatrix} c_B \\ c_N \end{pmatrix},$$

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

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

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

where

$$a_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}, j = \overline{1, n}; \; A_i^T = (a_{i1}, a_{i2}, \ldots, a_{in}), i = \overline{1, m};$$

$$A = (A_B | A_N), \;\; A_B = A(I, J_B), \;\; A_N = A(I, J_N).$$

A vector x verifying the constraints (2)-(3) is called a *feasible solution* for the problem (1)-(3). A feasible solution x^0 is called *optimal* if $z(x^0) = c^T x^0 = \max c^T x$, where x is taken from the set of all feasible solutions of the problem (1)-(3).
A feasible solution x^ϵ is said to be ϵ-*optimal* or *suboptimal* if $z(x^0) - z(x^\epsilon) = c^T x^0 - c^T x^\epsilon \leq \epsilon$, where x^0 is an optimal solution for the problem (1)-(3) and ϵ is a positive number chosen beforehand. We consider the index subset $J_B \subset J$ such that $|J_B| = |I| = m$. Then the set J_B is called a *support* if $\det A_B = \det A(I, J_B) \neq 0$. The pair $\{x, J_B\}$ comprising a feasible solution x and a support J_B will be called a *support feasible solution* (SFS). An SFS is called *nondegenerate* if $l_j < x_j < u_j, \; j \in J_B$.

Remark 1. A support feasible solution (SFS) is a more general concept than the basic feasible solution (BFS). Indeed, the nonsupport components of an SFS are not restricted to their bounds. Therefore, an SFS may be an interior point, a boundary point or an extreme point, but a BFS is always an extreme point.

We define the m-vector of multipliers π and the reduced costs n-vector Δ as follows:

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

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

Theorem 1. *(The optimality criterion [9]). Let $\{x, J_B\}$ be an SFS for the problem (1)-(3). So the relations:*

$$\begin{cases} \Delta_j \geq 0 \; for \quad x_j = l_j, \\ \Delta_j \leq 0 \; for \quad x_j = u_j, \\ \Delta_j = 0 \; for \quad l_j < x_j < u_j, \;\; j \in J_N, \end{cases} \tag{4}$$

are sufficient and, in the case of nondegeneracy of the SFS $\{x, J_B\}$, also necessary for the optimality of the feasible solution x.

The quantity $\beta(x, J_B)$ defined by:

$$\beta(x, J_B) = \sum_{\Delta_j > 0, j \in J_N} \Delta_j (x_j - l_j) + \sum_{\Delta_j < 0, j \in J_N} \Delta_j (x_j - u_j), \tag{5}$$

is called the *suboptimality estimate*.

3 Adaptive method with hybrid direction (AMHD)

Contrarily to the direction used in the adaptive method, which takes only extreme or zero values, the hybrid direction takes extreme values for relatively big component values of the reduced cost vector in order to bring the corresponding SFS components to their bounds and it takes the reduced gradient values for the other components.

Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta \in [0, 1]$. We define the following sets of indices:
$J_N^{P+} = \{j \in J_N : 0 < \Delta_j \leq \eta(x_j - l_j)\}, \; J_N^{P-} = \{j \in J_N : \eta(x_j - u_j) \leq \Delta_j < 0\}$;
$J_N^{P0} = \{j \in J_N : \Delta_j = 0\}, J_N^+ = \{j \in J_N : \Delta_j > \eta(x_j - l_j)\}, \; J_N^- = \{j \in J_N : \Delta_j < \eta(x_j - u_j)\}$. Then we have $J_N^P = \{j \in J_N : \eta(x_j - u_j) \leq \Delta_j \leq \eta(x_j - l_j)\} = J_N^{P+} \cup J_N^{P-} \cup J_N^{P0}$ and $J_N = J_N^+ \cup J_N^- \cup J_N^P$.
The quantity $\gamma(\eta, x, J_B)$ defined by:

$$\begin{cases} \sum_{j \in J_N^+} \Delta_j (x_j - l_j) + \sum_{j \in J_N^-} \Delta_j (x_j - u_j) + \frac{1}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2, \text{ if } \eta > 0, \\ \beta(x, J_B), \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } \eta = 0, \end{cases} \tag{6}$$

is called the *optimality estimate*.

Remark 2. When $\eta = 0$, we get $J_N^{P+} = J_N^{P-} = \emptyset$, so $J_N^P = J_N^{P0}$, $J_N^+ = \{j \in J_N : \Delta_j > 0\}$, $J_N^- = \{j \in J_N : \Delta_j < 0\}$ and $\gamma(\eta, x, J_B) = \beta(x, J_B)$.

Lemma 1. *Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. The only optimal indices in J_N^P are such that $\Delta_j = 0$, i.e., all the indices with $\Delta_j \neq 0$ are nonoptimal.*

Proof. Let $j \in J_N^{P+}$ and assume that j is optimal. From the optimality criterion (4), we deduce that $x_j = l_j$ and $0 < \Delta_j \leq \eta(x_j - l_j) = 0$, contradiction with the fact that $\Delta_j > 0$.
Let $j \in J_N^{P-}$ and assume that j is optimal. From the optimality criterion (4), we deduce that $x_j = u_j$ and $\eta(x_j - u_j) = 0 \leq \Delta_j < 0$, contradiction with the fact that $\Delta_j < 0$.

Theorem 2. *(Sufficient and necessary condition for optimality). Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. So the condition $\gamma(\eta, x, J_B) = 0$ is sufficient and, in the case of nondegeneracy of the SFS $\{x, J_B\}$, also necessary for the optimality of the SFS $\{x, J_B\}$.*

Proof. **Sufficiency.** Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. If $\gamma(\eta, x, J_B) = 0$, then we have $\sum_{j \in J_N^+} \Delta_j(x_j - l_j) = 0$, $\sum_{j \in J_N^-} \Delta_j(x_j - u_j) = 0$ and $\frac{1}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 = 0$.
For all $j \in J_N^+$, we have $\Delta_j > 0$, so $\sum_{j \in J_N^+} \Delta_j(x_j - l_j) = 0 \Rightarrow \forall j \in J_N^+, x_j = l_j$.
For all $j \in J_N^-$, we have $\Delta_j < 0$, so $\sum_{j \in J_N^-} \Delta_j(x_j - u_j) = 0 \Rightarrow \forall j \in J_N^-, x_j = u_j$.
Furthermore, $\frac{1}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 = 0 \Rightarrow J_N^{P+} = J_N^{P-} = \emptyset \Rightarrow J_N^P = J_N^{P0}$ and following the optimality criterion (4), all the indices of J_N^+, J_N^- and J_N^P are optimal. Hence $\{x, J_B\}$ is optimal.
Necessity. We assume that $\{x, J_B\}$ is a nondegenerate optimal SFS. Hence, from Theorem 1, we deduce $x_j = l_j$ for $\Delta_j > 0$; $x_j = u_j$ for $\Delta_j < 0$ and $l_j \leq x_j \leq u_j$ for $\Delta_j = 0$, i.e., all the indices of J_N are optimal. Then we have $x_j = l_j$, if $j \in J_N^+$ and $x_j = u_j$, if $j \in J_N^-$. Hence $\sum_{j \in J_N^+} \Delta_j(x_j - l_j) = \sum_{j \in J_N^-} \Delta_j(x_j - u_j) = 0$. Furthermore, since all the indices of J_N are optimal, we deduce from Lemma 1, that $J_N^{P+} = J_N^{P-} = \emptyset \Rightarrow \frac{1}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 = 0$. Therefore $\gamma(\eta, x, J_B) = 0$.

3.1 An iteration of AMHD

Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta \in [0, 1]$. We compute $\gamma(\eta, x, J_B)$ with the relationship (6). If $\gamma(\eta, x, J_B) = 0$, then the SFS $\{x, J_B\}$ is optimal; else we improve it. We define the feasible direction d as follows:

$$\begin{cases} d_j = l_j - x_j, \text{ if } j \in J_N^+; \\ d_j = u_j - x_j, \text{ if } j \in J_N^-; \\ d_j = \frac{-\Delta_j}{\eta}, \text{ if } j \in J_N^P, \ \eta \neq 0; \\ d_j = 0, \text{ if } j \in J_N^P, \ \eta = 0; \\ d_B = -A_B^{-1} A_N d_N, \text{ where } d_B = (d_j, j \in J_B), d_N = (d_j, j \in J_N). \end{cases} \tag{7}$$

This direction, with respect to the standard direction of the adaptive method is called an hybrid direction.

In order to improve the objective function while remaining in the feasible region, we compute the step length θ^0 along the direction d as follows:

$$\theta_j = \begin{cases} (u_j - x_j)/d_j, \text{ if } d_j > 0, \\ (l_j - x_j)/d_j, \text{ if } d_j < 0, \\ \infty, \qquad\quad \text{if } d_j = 0, \end{cases} \tag{8}$$

and $\theta_{j_r} = \min\{\theta_j, j \in J_B\}$, $\theta^0 = \min\{\theta_{j_r}, 1\}$. So the new feasible solution is $\bar{x} = x + \theta^0 d$. Since $Ad = 0$, then the vector d is a feasible direction. Furthermore, d is an ascent direction. Indeed, the objective function increment is given by

$$\begin{aligned} \bar{z} - z = c^T\bar{x} - c^T x &= c_B^T(\bar{x}_B - x_B) + c_N^T(\bar{x}_N - x_N) \\ &= c_B^T[-A_B^{-1}A_N(\bar{x}_N - x_N)] + c_N^T(\bar{x}_N - x_N) \\ &= -\Delta_N^T(\bar{x}_N - x_N) \\ &= -\sum_{j\in J_N} \Delta_j(\bar{x}_j - x_j) \\ &= -\theta^0 \sum_{j\in J_N} \Delta_j d_j \\ &= \theta^0[-\sum_{j\in J_N^+} \Delta_j d_j - \sum_{j\in J_N^-} \Delta_j d_j - \sum_{j\in J_N^{P+}\cup J_N^{P-}} \Delta_j d_j] \\ &= \theta^0[\sum_{j\in J_N^+} \Delta_j(x_j - l_j) + \sum_{j\in J_N^-} \Delta_j(x_j - u_j) + \sum_{j\in J_N^{P+}\cup J_N^{P-}} \frac{\Delta_j^2}{\eta}] \\ &= \theta^0\gamma(\eta, x, J_B) \geq 0. \end{aligned}$$

If $\gamma(\eta, x, J_B) > 0$ and $\theta^0 > 0$, then $\bar{z} > z$, so we can strictly improve the feasible solution x.

For the new feasible solution $\bar{x}$, we define the following sets of indices:
$\bar{J}_N^{P+} = \{j \in J_N : 0 < \Delta_j \leq \eta(\bar{x}_j - l_j)\}$, $\bar{J}_N^{P-} = \{j \in J_N : \eta(\bar{x}_j - u_j) \leq \Delta_j < 0\}$, $\bar{J}_N^{P0} = J_N^{P0} = \{j \in J_N : \Delta_j = 0\}$, $\bar{J}_N^+ = \{j \in J_N : \Delta_j > \eta(\bar{x}_j - l_j)\}$, $\bar{J}_N^- = \{j \in J_N : \Delta_j < \eta(\bar{x}_j - u_j)\}$. Then we have $\bar{J}_N^P = \{j \in J_N : \eta(\bar{x}_j - u_j) \leq \Delta_j \leq \eta(\bar{x}_j - l_j)\} = \bar{J}_N^{P+} \cup \bar{J}_N^{P-} \cup \bar{J}_N^{P0}$, $J_N = \bar{J}_N^+ \cup \bar{J}_N^- \cup \bar{J}_N^P$.
Furthermore, we define the following quantities $\gamma(\eta, \bar{x}, J_B)$ and $\bar{\gamma}(\eta, \bar{x}, J_B)$:

$$\gamma(\eta, \bar{x}, J_B) = \sum_{j\in J_N^+} \Delta_j(\bar{x}_j - l_j) + \sum_{j\in J_N^-} \Delta_j(\bar{x}_j - u_j) + \frac{1}{\eta} \sum_{j\in J_N^{P+}\cup J_N^{P-}} \Delta_j^2,$$

$$\bar{\gamma}(\eta, \bar{x}, J_B) = \sum_{j\in \bar{J}_N^+} \Delta_j(\bar{x}_j - l_j) + \sum_{j\in \bar{J}_N^-} \Delta_j(\bar{x}_j - u_j) + \frac{1}{\eta} \sum_{j\in \bar{J}_N^{P+}\cup \bar{J}_N^{P-}} \Delta_j^2.$$

Lemma 2. *Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. Then we have:*
1) $J_N^+ \subset \bar{J}_N^+$, $J_N^- \subset \bar{J}_N^-$, $\bar{J}_N^{P+} \subset J_N^{P+}$ and $\bar{J}_N^{P-} \subset J_N^{P-}$.
2) $J_N^{P+} = \bar{J}_N^{P+} \cup (\bar{J}_N^+ \setminus J_N^+)$ and $J_N^{P-} = \bar{J}_N^{P-} \cup (\bar{J}_N^- \setminus J_N^-)$.

Lemma 3. *Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. Let $\bar{x} = x + \theta^0 d$ be the new feasible solution, where d and θ^0 are defined by the relations (7) and (8) respectively. Then we have:*

a) $\gamma(\eta, \bar{x}, J_B) = (1 - \theta^0)\gamma(\eta, x, J_B) + \frac{\theta^0}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2$.

b) $\gamma(\eta, x, J_B) \leq \beta(x, J_B)$, $\gamma(\eta, \bar{x}, J_B) \leq \gamma(\eta, x, J_B)$, $0 \leq \bar{\gamma}(\eta, \bar{x}, J_B) \leq \gamma(\eta, \bar{x}, J_B)$.

Theorem 3. *(Sufficient condition for the optimality of $\bar{x}$)*
Let $\{x, J_B\}$ be an SFS for the problem (1)-(3) and $\eta > 0$. The feasible solution $\bar{x}$ is optimal if $\gamma(\eta, \bar{x}, J_B) = 0$.

Proof. According to Lemma 3, we have

$$0 \leq \bar{\gamma}(\eta, \bar{x}, J_B) \leq \gamma(\eta, \bar{x}, J_B) = 0 \Rightarrow \bar{\gamma}(\eta, \bar{x}, J_B) = 0.$$

Following Theorem 2, the SFS $\{\bar{x}, J_B\}$ is optimal.

Corollary 1. *Let x be a nonoptimal feasible solution for the problem (1)-(3) and $\bar{x} = x + \theta^0 d$ the new feasible solution. We assume that $\theta^0 = 1$. Then the feasible solution $\bar{x}$, is optimal if $J_N^P \setminus J_N^{P0} = \emptyset$.*

Proof. We assume that $J_N^P \setminus J_N^{P0} = \emptyset$, i.e., $J_N^{P+} \cup J_N^{P-} = \emptyset$. When $\theta^0 = 1$, from part a) of Lemma 3, we deduce that

$$\gamma(\eta, \bar{x}, J_B) = (1 - \theta^0)\gamma(\eta, x, J_B) + \frac{\theta^0}{\eta} \sum_{j \in J_N^{P+} \cup J_N^{P-}} \Delta_j^2 = 0.$$

According to Theorem 3, the feasible solution $\bar{x}$ is optimal.

Remark 3. If $\eta = 0$, then $J_N^{P+} = J_N^{P-} = \emptyset \Rightarrow \gamma(\eta, \bar{x}, J_B) = (1 - \theta^0)\gamma(\eta, x, J_B)$. Since for $\eta = 0$, $\gamma(\eta, x, J_B) = \beta(x, J_B)$ and $\gamma(\eta, \bar{x}, J_B) = \beta(\bar{x}, J_B)$, so we find the classical updating formula of $\beta(x, J_B)$ [9]:

$$\beta(\bar{x}, J_B) = (1 - \theta^0)\beta(x, J_B).$$

3.2 Algorithm of the adaptive method with hybrid direction and long step rule

Let $\{x, J_B\}$ be an initial SFS for the problem (1)-(3) and η, μ two nonnegative real parameters such that $\eta \in [0, 1]$ and $\mu \geq 1$. The scheme of the adaptive method with hybrid direction is described in the following steps:

(1) Compute $\pi^T = c_B^T A_B^{-1}$, $\Delta_N^T = \pi^T A_N - c_N^T$, $\Delta_B = 0$;
(2) find the sets J_N^+, J_N^-, J_N^P and J_N^{P0};
(3) compute $\gamma(\eta, x, J_B)$ with the formula (6);
(4) if $\gamma(\eta, x, J_B) = 0$, then the algorithm stops with $\{x, J_B\}$, an optimal SFS;
(5) compute the search direction, d, using the relations (7);

(6) compute $\theta_{j_r} = \min\limits_{j \in J_B} \theta_j$, where θ_j is determined by the formula (8);
(7) compute $\theta^0 = \min\{\theta_{j_r}, 1\}$; $\bar{x} = x + \theta^0 d$; $\bar{z} = z + \theta^0 \gamma(\eta, x, J_B)$;
(8) if $\theta^0 = 1$, then
  (8.1) if $J_N^P \setminus J_N^{P0} = \emptyset$, then according to Corollary 1, $\bar{x}$ is optimal. Stop.
  (8.2) we put $\eta = 0$, $x = \bar{x}$, $z = \bar{z}$ and go to step (2).
(9) compute the dual direction t with the components:

$$\begin{cases} t_{j_r} = -sign(d_{j_r}); \\ t_j = 0,\ j \neq j_r, j \in J_B; \\ t_N^T = t_B^T A_B^{-1} A_N, \text{ where } t_B = (t_j, j \in J_B), t_N = (t_j, j \in J_N); \end{cases}$$

(10) compute the dual step length and the new support with the long step rule:
  (10.1) compute σ_j, $j \in J_N$:

$$\sigma_j = \begin{cases} \frac{-\Delta_j}{t_j}, \text{ if } \Delta_j t_j < 0; \\ 0, \quad \text{if } \Delta_j = 0,\ t_j < 0,\ j \in J_N; \\ \infty, \quad \text{in other cases.} \end{cases}$$

  (10.2) Find the entering index j_q:
  - arrange the indices $\{j \in J_N : \sigma_j \neq \infty\}$ in increasing values σ_j:

$$\sigma_{j_1} \leq \sigma_{j_2} \leq \cdots \leq \sigma_{j_p};\ j_k \in J_N,\ \sigma_{j_k} \neq \infty,\ k = \overline{1,p};$$

  - if $p = 1$, then $j_q = j_1$; go to step (10.3);
  - for every $j_k, k = \overline{1,p}$, we compute the jump of the rate of the dual objective function

$$\Delta\alpha_{j_k} = |t_{j_k}|(u_{j_k} - l_{j_k});$$

  - compute the initial rate of change of the dual objective function

$$\alpha_0 = -(1 - \theta^0)|d_{j_r}|;$$

  - compute α_{j_k}, $k = \overline{0,p}$, where

$$\begin{cases} \alpha_{j_0} = \alpha_0, \\ \alpha_{j_k} = \alpha_0 + \sum_{s=1}^{k} \Delta\alpha_{j_s} = \alpha_{j_{k-1}} + \Delta\alpha_{j_k}, \quad k = \overline{1,p}; \end{cases}$$

  - we choose j_q such that $\alpha_{j_{q-1}} < 0$ and $\alpha_{j_q} \geq 0$;
  (10.3) we put $\sigma^0 = \sigma_{j_q}$, $\bar{\Delta} = \Delta + \sigma^0 t$, $\bar{J}_B = (J_B \setminus \{j_r\}) \cup \{j_q\}$;
(11) we put $x = \bar{x}$, $J_B = \bar{J}_B$, $z = \bar{z}$, $\Delta = \bar{\Delta}$;
(12) we put $\eta = \eta/\mu$ and go to the step (2).

Remark 4. If degeneracy is encountered, i.e., after a certain number of iterations the step length is equal to zero, then we put $\eta = 0$ and go to step (2). This allows the algorithm to progress toward the optimal solution.

Remark 5. We distinguish two variants of the AMHD algorithm: the first variant consists of putting beforehand $\mu = 1$, so the η parameter is reduced only when we encounter degeneracy or entering to step (8.2). The second variant consists of setting $\mu > 1$, hence the η parameter is reduced in the end of each iteration (step (12)) by the μ factor. The experimental study that we have carried out has shown that there is no significant difference in the performances of the two variants. This is why in our numerical experiments, we compare only the first variant with the simplex method.

Remark 6. The main difference between the adaptive method and the adaptive method with hybrid direction consists in the computing of the search direction to improve the primal solution x, and their stopping criteria. Note that, when we put $\eta = 0$ in the adaptive method with hybrid direction, we switch to the iterations of the adaptive method.

4 Experimental results

In order to perform a numerical comparison between the Primal Simplex Algorithm (PSA) and the Adaptive Method with Hybrid Direction (AMHD), we have programmed them under the MATLAB programming language version 7.4.0 (R2007a).

In each iteration of PSA, we solve, using LU factorization, two systems $A_B^T \pi = c_B$ and $A_B d_B = a_{j_0} sign(\Delta_{j_0})$, where j_0 is the entering index found using Dantzig's rule. And in each iteration of AMHD, we solve the two systems $A_B d_B = -A_N d_N$ and $A_B^T v = t_B$, but the system $A_B^T \pi = c_B$ is solved only in the beginning of the algorithm and the reduced cost vector is updated in each iteration using the formula $\bar{\Delta} = \Delta + \sigma^0 t$, where σ^0 is the dual step length computed by the long step rule (step (10) of AMHD).

The updating of the L and U factors is done, in each iteration, using the Sherman-Morrison-Woodbury formula [7].

We have generated a set of sparse random LP problems of the form

$$\{\max c^T x,\ \text{s.t. } Ax + x^e = b,\ l \leq x \leq u,\ l^e \leq x^e \leq u^e\},$$

with different dimensions and density equal to 5%, $A \in \Re^{m \times p}$, $x, c, l, u \in \Re^p$, $b \in \Re^m$, $c_j \in [-100, 100]$, $a_{ij} \in [-500, 500]$, $l_j \in [-50, 200]$, $u = l + r_1^{(p)}$, $b = A\frac{(l+u)}{2} + r_2^{(m)}$, $l^e = b - Al - r_3^{(m)}$, $u^e = b - Al + r_4^{(m)}$, where $r_j^{(k)}, j = \overline{1,4}$ are k-vectors of random numbers drawn from the uniform distribution on the interval $[1, 100]$. Totally, we have generated 150 small and medium LP problems. We have considered three sets of problems: LPs with A of size $n \times n$, $n \in \{200, 400, 600, 800, 1000\}$; LPs with A of size $n \times 2n$, $n \in \{100, 200, 300, 400, 500\}$ and LPs with A of size $2n \times n$, $n \in \{100, 200, 300, 400, 500\}$. For each class of LP problems, for example LPs of dimension 200×200, we generate ten problems.

We have solved the randomly generated set of test problems with the two considered algorithms, PSA and AMHD, on a personal computer with Intel (R)

Core (TM) 2 Quad CPU Q6600 2.4 GHz, 3.24 GB of RAM, working under the Windows XP SP2 operating system.

We have initialized PSA by the first BFS $x = (l; b - Al)$ and AMHD by the SFS $\{x, J_B\}$, where $x = (l; b - Al)$ and $J_B = \{p + 1, p + 2, \ldots, p + m\}$. Furthermore, we have put for AMHD $\eta = 1$ and $\mu = 1$.

The numerical results are reported in Table 1, where *cpu* and *niters* represent respectively the mean cpu time and the average number of iterations of the ten problems generated for each class. In the sixth and the seventh column of Table 1, we give the cpu time ratios (cpu time of PSA)/(cpu time of AMHD) and iteration ratios (number of iterations of PSA)/(number of iterations of AMHD) for the corresponding dimensions. At the end of each set of LPs, we give the mean values of the above ratios.

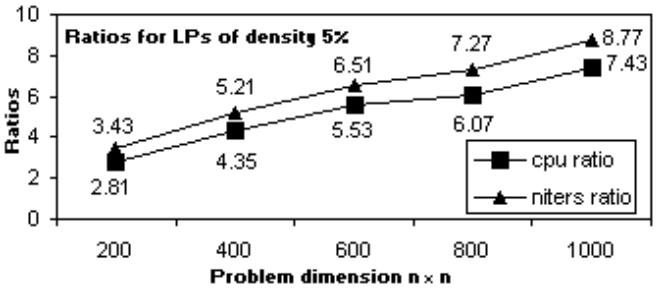
We plot the ratios of PSA over AMHD for problems with dimension $n \times n$. The graph is shown in Figure 1. From the graph shown in Figure 1, we see

Table 1. Results for LPs of density 5%

Algorithm→	PSA		AMHD		PSA/AMHD	
Dimension↓	cpu	niters	cpu	niters	cpu	niters
200 × 200	2.36	1434.70	0.84	417.70	2.81	3.43
400 × 400	27.37	6224.30	6.29	1193.90	4.35	5.21
600 × 600	124.15	13270.50	22.43	2038.80	5.53	6.51
800 × 800	373.47	22643.30	61.51	3114.30	6.07	7.27
1000 × 1000	959.42	36324.10	129.09	4140.50	7.43	8.77
Mean ratio					5.24	6.24
100 × 200	0.68	608.50	0.46	288.20	1.48	2.11
200 × 400	4.15	1592.40	3.20	809.30	1.30	1.97
300 × 600	15.74	3058.10	10.78	1386.70	1.46	2.21
400 × 800	46.63	5127.00	29.82	2168.00	1.56	2.36
500 × 1000	103.59	7338.10	65.07	2981.40	1.59	2.46
Mean ratio					1.48	2.22
200 × 100	0.25	223.00	0.15	129.10	1.67	1.73
400 × 200	3.80	1624.10	1.08	429.90	3.52	3.78
600 × 300	16.53	3858.80	3.73	830.90	4.43	4.64
800 × 400	43.70	7182.70	8.88	1270.90	4.92	5.65
1000 × 500	104.73	10821.40	18.66	1790.40	5.61	6.04
Mean ratio					4.03	4.37
Total mean ratio					3.58	4.28

that when the problem dimension increases, the superiority of AMHD over PSA increases. Particularly, for problems with dimension 1000×1000, our algorithm is 7.43 times faster than PSA in terms of cpu time and solves the problems 8.77 times faster than PSA in terms of number of iterations. However, we need to compare the two algorithms performances on real benchmarks LP test problems to obtain more refined conclusions.

Fig. 1. Ratios of PSA over AMHD for LPs of dimension $n \times n$



5 Conclusion

In this work, we have suggested a new search direction for the adaptive method; sufficient and necessary conditions are derived to characterize the optimality of the current solution and an algorithm called the adaptive method with hybrid direction and long step rule is described. In order to compare the suggested algorithm (AMHD) with the primal simplex algorithm employing Dantzig's rule (PSA), we have implemented the two algorithms under the MATLAB environnement. In the implementation, we have used the LU factorization of the basic matrix to solve the linear equations systems and the Sherman-Morrison-Woodbury formula to update the LU factors. Furthermore, we have compared the two algorithms on a large set of randomly generated LP problems of different dimensions and density of 5%. The obtained numerical results are quite encouraging because the proposed algorithm outperforms the primal simplex algorithm using Dantzig's rule on almost all the randomly generated tests. Indeed, our algorithm is very efficient in relation to the primal simplex algorithm, particularly, on the test problems of higher dimension. In future works, we will apply some crash procedure like that proposed in [18] in order to initialize AMHD with a good initial support. Furthermore, we will implement some modern sparse algebra techniques to update the LU factors [8] and test the performances of our algorithm on practical LP test problems [20].

References

1. R.H. Bartels and G.H. Golub, *The simplex method of linear programming using LU decomposition*, Communications of the ACM 12 (5), 1969, pp. 266—268.
2. M. Bentobache, *A new method for solving linear programming problems in canonical form and with bounded variables*, Master thesis, University of Bejaia, Algeria, 2005 (in French).
3. M. Bentobache and M.O. Bibi, *Two-phase support method for solving linear programming problems with nonnegative variables: Numerical experiments*, in Proceedings of COSI'08, University of Tizi Ouzou, June 2008, pp. 314–325 (in French).

4. M. Bentobache and M.O. Bibi, *Two-phase support method for solving linear programming problems with bounded variables: Numerical experiments*, in Proceedings of COSI'09, University of Annaba, May 2009, pp. 109–120 (in French).
5. G.B. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, in R.C. Koopmans (ed.), Activity Analysis of Production and Allocation, Wiley, New York, 1951, pp. 339–347.
6. G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.
7. M.C. Ferris, O.L. Mangasarian and S.J. Wright, *Linear Programming with MATLAB*, MPS-SIAM Series on Optimization, 2007.
8. J.J.H. Forrest and J.A. Tomlin, *Updating triangular factors of the basis to maintain sparsity in the product form simplex method*, Mathematical Programming 2, 1972, pp. 263—278.
9. R. Gabasov and F.M. Kirillova, *Methods of linear programming*, Vol. 1, 2 and 3, Edition of the Minsk University, 1977, 1978 and 1980 (in Russian).
10. R. Gabasov, F.M. Kirillova and S.V. Prischepova, *Optimal Feedback Control*, Springer-Verlag, London, 1995.
11. P.E. Gill and W. Murray, A Numerically Stable Form of the Simplex Algorithm, Linear Algebra and its Applications 7, 1973, pp. 99–138.
12. P.E. Gill, W. Murray and M.H. Wright, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison-Wesley Publishing Company, Redwood City, CA. 94065, 1991.
13. J. Gondzio, *Another simplex type method for large scale linear programming*, Control and Cybernetics 25 (4), 1996, pp. 739—760.
14. L.G. Khachiyan, *A polynomial algorithm for linear programming*, Soviet Mathematics Doklady 20, 1979, pp. 191–194.
15. V. Klee and G.J. Minty, *How Good is the Simplex Algorithm?*, in O. Shisha (Ed.), Inequalities III, Academic Press, New York, 1972, pp. 159–175.
16. N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica 4, 1984, pp. 373–395.
17. E. Kostina and S.V. Prishchepova, *A numerical experiment with respect to the solution of large sparse problems of linear programming by the adaptive method*, Izv. Akad. Nauk BSSR, Ser. Fiz.-Mat. Nauk 6, 1990, pp. 3–5 (in Russian).
18. I. Maros and G. Mitra, *Strategies for creating advanced bases for large-scale linear programming problems*, Informs Journal on Computing 10 (2), 1998, pp. 248—260.
19. G. Mitra, M. Tamiz and J. Yadegar, *Experimental investigation of an interior search method within a simplex framework*, Communications of the ACM 31, 1988, pp. 1474–1482.
20. *Netlib test problems*; available at http://www.netlib.org/lp/data.
21. K. Paparrizos, N. Samaras and G. Stephanides, *An efficient simplex type algorithm for sparse and dense linear programs*, European Journal of Operational Research 148, 2003, pp. 323—334.

Mono-Criteria and Bi-Criteria Optimal Control Problem for Moving an Electric Vehicle

A. Merakeb[1], F. Messine[2] and M . Aidène[1]

[1] Département de Mathematiques, Faculté des Sciences
Université Mouloud Mammeri, Tizi-Ouzou, Algeria
[2] ENSEEIHT-IRIT, Université PRES Toulouse, France
merakeb_kader@yahoo.fr, Frederic.Messine@n7.fr, aidene@mail.ummto.dz

Abstract. In simulation, the electrical vehicle energy management problem can be expressed as an optimal control problem. In this work, we discuss on a new formulation and about the way to approximate the global solution of an optimal control problem of Bang-Bang type via a discretization technique associated with a Branch and Bound algorithm. The problem that we focus on is to compute the minimal energy consumption of an electrical car achievable on a given driving cycle. In last section, we consider a second criteria that is the maximization of the distance browsed leading to a bi-criteria optimal control problem. We extend our Branch&Bound algorithm to construct the Pareto front.
keywords : Discretization techniques, Optimal Control, Bang-Bang control, Branch and Bound, Multiple criteria optimization, Pareto front, Goal Programming.

1 Introduction

Electrical vehicle uses an electrical energy source for its displacement that can be reversible. This work focuses on the energy management. The objective is to find the control strategies and compute the minimal energy consumption achievable by the electrical vehicle on a given driving cycle. The main approaches have been studied, including the Pontriagin maximum principle (PMP) and direct methods. Indirect methods, based on the PMP are famous for their speed and accuracy. However, their implementation using shooting methods may face some difficulties in practice when the structure of control is bang-bang type. Indeed, these methods transform the original problem by solving a system of nonlinear equations. In this case, the system is not regular, and its numerical solution is particularly sensitive to the choice of initial point. Note also that the presence of a constraint on the state only increases its complexity. Direct methods, in turn, traditionally involve total or partial discretization of the problem, and then use various approaches (SQP and interior point techniques for example) to solve the problem arising. Nevertheless, they are relatively imprecise and can lead to problems of large sizes depending on the used step of discretization. Also, these methods are less suitable for certain special cases, including problems with a bang-bang structure yielding a large number of switching operations. We discuss

about the way to solve efficiently the problem of the minimization of the energy which is consummated by an electrical car during an imposed displacement, see [1] for an overview on this type of problems. To solve this problem, we reformulate it following the construction of current regulator technique to obtain a global optimization problem that can be solved using Branch and Bound algorithm. Using this technique, the benefit obtained is the reduction of the computation requirements and the algorithm can be embedded within a real time predictive control framework.

2 Model Overview

Fig.1 concerns the standard traction link between the different components of the constitution of the vehicle in question. The modeling of this transmission chain consists of two parts: electrical part in association with battery, converter and motor; and mechanical part in association with transmission and vehicle. Each part is described by a differential equation (one for the current inside the motor and one for speed).

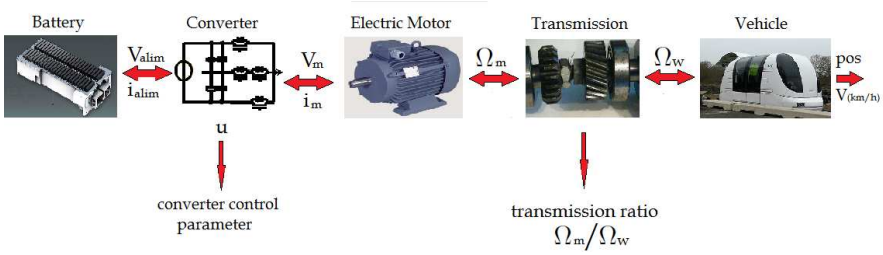


Fig. 1. Standard traction link

The energy confined in the battery is regulated in the converter using the control parameter u, and the current delivered to the motor is submitted to the differential equation (1)

$$\frac{di_m(t)}{dt} = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega_m(t)}{L_m} \tag{1}$$

The movement of the motor is transmitted to the vehicle via the transmission provided with a coefficient ratio. The differential equation in this part is given in the speed rotor parameter by (2)

$$\frac{d\Omega_m(t)}{dt} = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega_m(t) r}{K_r}\right)^2\right)\right) \tag{2}$$

To know the position of the vehicle, we can infer it from the third differential equation (3).

$$\frac{dpos(t)}{dt} = \frac{\Omega_m(t) \times r}{K_r} \tag{3}$$

$V(t) = \frac{3.6 \times r}{K_r} \Omega_m(t)$ gives linear celerity of the car in km/h.
The performance index is given by the energy formula (4)

$$E(t_f, i_m, u) = \int_0^{t_f} (u(t) i_m(t) V_{alim} + R_{bat} u^2(t) i_m^2(t)) dt \tag{4}$$

where E represents the electrical energy consummated during the displacement on the cycle $[0, t_f]$. The quadratic term reflects the losses due to the internal resistance of the battery. The system allows to recover the kinetic energy under deceleration to recharge the battery.

The problem that we are interested with can be formulated as an optimal control problem (5):

$$\begin{cases} \min_{i_m(t), \Omega(t), pos(t), u(t)} \quad E(t_f, i_m, u) \\ s.t. \\ \begin{cases} \dot{i_m}(t) = \frac{u(t) V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\ \dot{\Omega}(t) = \frac{1}{J} \left(K_m i_m(t) - \frac{r}{K_r} \left(MgK_f + \frac{1}{2} \rho S C_x \left(\frac{\Omega(t) r}{K_r} \right)^2 \right) \right) \\ \dot{pos}(t) = \frac{\Omega(t) r}{K_r} \end{cases} \\ \\ |i_m(t)| \leq 150 \\ u(t) \in \{-1, \ +1\} \\ \\ (i_m(0), \ \Omega(0), \ pos(0)) = (i_m^0, \ \Omega^0, \ pos^0) \in R^3 \\ (i_m(t_f), \ \Omega(t_f), \ pos(t_f)) \in \mathcal{T} \subseteq R^3 \end{cases} \tag{5}$$

The state variables are: (i) i_m the current inside the motor; (ii) Ω the angular speed; (iii) pos is the position of the car. The control u is in $\{-1, 1\}$ (a Bang-Bang structure). In this problem, we have a constraint on a state variable to limit the current inside the motor in order to discard the possibility to destroy it. The other terms are fixed parameters and represent some physical things: $K_r = 10$, the coefficient of reduction; $\rho = 1.293 kg/m^3$, the air density; $C_x = 0.4$, the aerodynamic coefficient; $S = 2m^2$, the area in the front of the car; $r = 0.33m$, the radius of the wheel; $K_f = 0.03$, the constant representing the friction of the wheels on the road; $K_m = 0.27$, the coefficient of the motor torque; $R_m = 0.03\Omega$, the inductor resistance; $L_m = 0.05$, inductance of the rotor; $M = 250kg$, the mass; $g = 9.81$, the gravity constant; $J = M \times r^2 / K_r^2$; $V_{alim} = 150$, the battery voltage; $R_{bat} = 0.05\Omega$, the resistance of the battery. This problem is subject to the boundary conditions. The initial conditions are given by the starting point $(i_m^0, \ \Omega^0, \ pos^0)$ at the starting time $t_0 = 0$, but the target set $\mathcal{T}$ at the final time t_f is free and depends on the instances of the problem; it could be a point of R^3

but one or two variables could not be fixed: for example just the final position equal to $100m$ is required (see the numerical section).

For the moment, the fact that we have a constraint on the state associated with the fact that it is a Bang-Bang control involves a lot of difficulties when using the Pontriagin method which does not permit to obtain solutions (even local ones).

With direct methods, the procedure leads to problems of large sizes depending on the used step of discretization. In our case, if we discretize all the cycle of time by fixing the value of the control, it is necessary to have very small steps else the value of the current inside the motor will change too roughly.

Dynamic programming using Hamilton- Jacobi Bellman equation, is a technique which compares the optimal solution with all the other solutions. This global comparison, therefore, leads to optimality conditions which are sufficient. The only disadvantage of DP (which often rules out its use), is that it can easily give rise to enormous computational requirements, which is the case with our problem [2].

Thus, in this paper we propose another original methodology to solve this problem yielding to some discretized problems which are solved using an exact Branch and Bound algorithm. This new method provide exact results for the discretized formulations which correspond to approximations of the global solutions of Problem (5).

3 Approximation of Problem (5)

First we remark that the energy formula is only depending on the current and the control. Therefore, it is just required to search the trajectory of the current that minimizes the consumption of the energy. If we discretize all the interval of time $[0, t_f]$ by fixing the value of the control u, it is necessary to have very small steps about 10^{-3} for at least to be able to control the current through the motor. That will generate a very huge mixed integer non-linear global optimization problem which is, for the moment, impossible to solve using direct methods of optimal control.

An idea, which directly comes from the numerical simulation of the behavior of the car, is to impose during some short sample of time the value of the current inside the electrical motor of the vehicle. This is possible using the control parameter $u(t)$. Thus, if we impose a reference current $iref$, if $i_m(t) > iref + \frac{\Delta}{2}$ then $u(t) := -1$ and if $i_m(t) < iref - \frac{\Delta}{2}$ then $u(t) := 1$ (see fig.2).

The control switches between the two values of u when the current inside the motor exceeds the value of iref with respect to the tolerance Δ.

The exceeding boundaries of the Δ band over a step time is due to the fact that the current is close to the borders of the Δ band at the end of the previous step. The maximum overflow is 0.3 amps for 10^{-4} time discretization stepsize.

This technique is just a way to construct a regulator of current which is a first step before making a speed regulator for an electrical car. Hence, using this, the following differential system of equations can be solved:

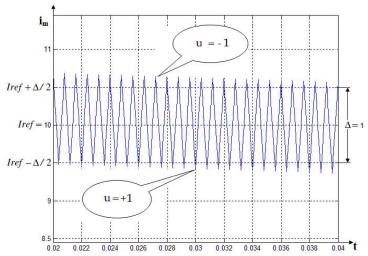


Fig. 2. Management principle by reference current

$$VS_{t_0,iref}(t) := \begin{cases} \dot{E}(t) = u(t)i_m(t)V_{alim} + R_{bat}u^2(t)i_m^2(t) \\ \dot{i_m}(t) = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\ \dot{\Omega}(t) = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega(t)r}{K_r}\right)^2\right)\right) \\ \dot{pos}(t) = \frac{\Omega(t)r}{K_r} \\ u(t) := \begin{cases} -1 \text{ if } i_m(t) > iref + \frac{\Delta}{2} \\ +1 \text{ if } i_m(t) < iref - \frac{\Delta}{2} \\ u(t) \text{ else.} \end{cases} \\ (E(t_0), i_m(t_0), \Omega(t_0), pos(t_0)) = (E^{t_0}, i_m^{t_0}, \Omega^{t_0}, pos^{t_0}) \in R^4 \\ u(t_0) := 1; \end{cases} \tag{6}$$

where t_0 is the initial time which is not necessary equal to 0.

This system of differentiable equations can be efficiently solved using a classical differentiable integrator such as for example *Euler, RK2, RK4* with a step of time less than 10^{-3}. The function $VS_{(t_0,iref}(t)$ will compute in theory all the values for $E(t), i_m(t), \Omega(t), pos(t)$, for all $t \in [t_0, t_f]$ but in practice only values for a discretized time $t_i \in [t_0, t_f]$ is available. Here, we are interested by the final values of the state variables, hence we define a function:

$$VSF(iref, t_0, t_f) := (E(t_f), i_m(t_f), \Omega(t_f), pos(t_f)) \in R^4.$$

All the computations are performed using the function $VS_{(t_0,iref}(t)$ which solves the system of differential equations (6) under the initial conditions (E^{t_0}, $i_m^{t_0}$, Ω^{t_0}, pos^{t_0}).

4 The global optimization problem

The main idea of this work is to subdivide the cycle of time $[0, t_f]$ into P subintervals. In each sample of time $[t_{k-1}, t_k]$ with $k \in \{1, \cdots, P\}$ ($t_k = k \times \frac{t_f}{P}$),

we apply a reference current $iref_k$ which takes values in $[-150, 150]$ in order to directly satisfy the constraint on the state variable of Problem (5).

Thus, we focus on the resolution of the following global optimization problem:

$$\begin{cases} \min_{iref \in [-150,150]^P} \sum_{k=1}^{P} E_k \\ u.c. \\ \qquad\qquad (E_k, i_k, \Omega_k, pos_k) := VSF(iref_k, t_{k-1}, t_k) \\ \qquad\qquad (E_0, i_0, \Omega_0, pos_0) = (E^0, i_m^0, \Omega^0, pos^0) \in R^4 \\ \qquad\qquad (i_P, \Omega_P, pos_P) \in \mathcal{T} \subseteq R^3 \end{cases} \tag{7}$$

Problem (7) is a good approximation of the initial problem (5) which generates just a few number of variables P.

5 Dedicated Branch and Bound Algorithm

For the moment, we are not able to solve exactly the global optimization problem (7), thus we need to discretize also the possible values for the reference current: $iref \in \{-150, -150 + s, -150 + 2 \times s, \cdots, 150\}^P$; we will take integer values for s which divide exactly $[-150, 150]$. Therefore, the set of solution becomes finite and could be enumerated. Nevertheless, if we want to have a good approximation for the resolution of the global optimization problem (7) we have to discretize into small samples and the finite set of possible points becomes rapidly too huge to be entirely enumerated in a reasonable CPU-time. The idea is then to use a Branch and Bound algorithm in order to not explore all the finite set of solutions.

5.1 Computing bounds technique

For using such an algorithm, we have to elaborate a technique to compute bounds for the four main parameters: $E_k, i_k, \Omega_k, pos_k$ over a box $IREF \subseteq \{-150, -150+s, -150+2\times s, \cdots, 150\}^P$ and for given t_0 and t_f. In order to be more efficient, in a previous sample, we compute 4 matrices: $M_E, M_i, M_\Omega, M_{pos}$ where the columns corresponds to values when $iref$ is fixed with $i_m^{t_0} = iref$ and the rows provides values for the entities when a speed Ω^{t_0} is given (we discretize also the possible values of the speed).

	$iref = -150 + (j-1)s$
	$\vdots$
$\Omega^{t_{k-1}} = (i-1)pasV$	$\cdots \quad m_\Theta(i,j)$

$pasV$ is a discretization step of the speed values.
Θ represents one of these symbols $E,\ \Omega,\ pos$.

$e_E = (1,0,0,0),\ \ e_\Omega = (0,0,1,0),\ \ e_{pos} = (0,0,0,1)$

$$m_\Theta(i,j) =< VSF(iref, t_{k-1}, t_k),\ e_\Theta >$$

taken when computing the function $VS_{t_{k-1},iref}(t)$ over a sample time $[t_{k-1},\ t_k]$ and under the initial conditions

$$(E^{t_{k-1}},\ i_m^{t_{k-1}},\ \Omega^{t_{k-1}},\ pos^{t_{k-1}}) = (0,\ iref,\ \Omega^{t_{k-1}},\ 0)$$

For example $m_E(i,j)$ represents the value of the energy which is consummated during a sample of time $[t_{k-1}, t_k]$ when $iref$ corresponds to the jth components of the set $\{-150, -150+s, -150+2\times s, \cdots, 150\}$ with $i_m^{t_0} = iref$ and the ith row corresponds to the discretized value of the speed, the other initial values are taken equal to 0: i.e., $E^{t_{k-1}} = pos^{t_{k-1}} = 0$.

When a box $IREF$ is considered, we can compute bounds for E, i, Ω and pos by computing the integer sets I and J of the indices corresponding to the possible values of the speed at the previous sample and the possible values of $iref$. Then, we have to compute the bounds which correspond to the minimal and maximal values of $m_E(i,j), m_i(i,j), m_\Omega(i,j), m_{pos}(i,j)$ with $(i,j) \in I \times J$. To obtain the final value for E and pos, we have to sum all the lower and upper bounds.

The rest of the Branch and Bound algorithm that we develop is simple and uses the following classical principle: (i) subdivision into two (distinct) parts of the enumerate set $IREF$ (which represents the possible values for $iref$); (ii) the upper bound is updated by taking the middle of the box $IREF$ if the constraints are satisfied and if its value is better than the previous one (we start with $+\infty$);(iii) we branch following the heuristic of lowest lower bound of the energy.

5.2 Heuristics alternative

The Branch and Bound algorithm uses the data pre-processing when computing matrices. These data are exploited in order to reduce the computing time bounds. Indeed, the exact method described above to compute bounds is expensive for CPU-time. So, interest was paid for two heuristics $H1$ and $H2$.

For $H1$, it is taken as lower bounds, the values of each sub-matrices induced corresponding to the first row and first column $m_E(i_1,j_1);\ m_\Omega(i_1,j_1);\ m_{pos}(i_1,j_1)$. As upper bounds, the values corresponding to the last rows and last columns $m_E(i_n,j_m);\ m_\Omega(i_n,j_m);\ m_{pos}(i_n,j_m)$.

For $H2$, we keep the same bounds as the heuristic $H1$ for the position. For the energy and the speed, as lower bounds, we compute the minimum value on the first row. As upper bounds, the maximum value on the last row.

Lower bounds $\min_{j\in J} m_E(i_1,j);\ \min_{j\in J} m_\Omega(i_1,j);\ m_{pos}(i_1,j_1)$.

Upper bounds: $\max_{j\in J} m_E(i_n,j);\ \max_{j\in J} m_\Omega(i_n,j);\ m_{pos}(i_n,j_m)$.

Time computing matrices depends on the sample of time $t_k - t_{k-1} = \frac{t_f}{P}$, on the $pasV$ (step of discretization of the speed), on the integer value of s

(which sudivide the intervalle $[-150, 150]$ of reference current), on the $RK4$ integrator step of time and the value of Δ. If we fix the $RK4$ integrator step to 10^{-4}, $pasV = 0.1$ and $\Delta = 1$, the preprocessing time to compute matrices is proportional with $(t_k - t_{k-1})$ and inversely with s.

6 Numerical Experiments

To illustrate our method, we simulated it for a displacement of 100 meters, and a cycle $t_f = 10$ seconds: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = R \times R \times \{100\}$.

In our simulation, we compare the three heuristics for the parameters $P = 5$ and $s = 10$, we obtain (for the discretized problem) the exact solution $iref = (150, 90, 30, -30, -110)$ corresponding to the minimal value $Emin(10) = 23272J$. Moreover, we have $pos(10) = 100.24m$. This computation is performed for 10^{-4} RK4 integrator step and $pasV = 0.1$. We add $167s$ of pre-processing to compute matrices simulated on MatLab 9 on a standard PC Laptop with 4GB of RAM.

	$CPU - time\ (s)$	$E_{min}\ (J)$	$iref$	$iterations$
ME	13.15	23272	(150, 90, 30, − 30, − 110)	31531
$H1$	1.68	23272	(150, 90, 30, − 30, − 110)	15102
$H2$	3.97	23272	(150, 90, 30, − 30, − 110)	16023

Note that if we compute this solution directly using $RK4$ numerical integrator (without using matrices) we obtain $\overline{E}min = 23313J$ and $\overline{pos} = 99.36m$ and the error computation is about 0.17% for the energy and 0.87% for the position.

Although the three heuristics give the same solution for the instance shown, the faster heuristic $H1$ remains less effective in some cases. however, $H2$ is more robust and gives in most cases, satisfactory results with a time saving of 70% compared to the exact method ME. This solution is represented in fig.3.

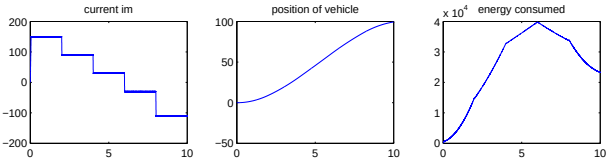


Fig. 3.

For $P = 5$ and $s = 2.5$, we obtain the solution $iref^* = (150.0,\ 82.5,\ 35.0,\ -25.0,\ -105.0)$ corresponding to the minimal value $Emin^*(10) = 23102J$. Moreover, we have $pos^*(10) = 100.04m$. The CPU-time computation is about $1356s$ (add $659s$ of pre-processing) corresponding to 309669 iterations of the Branch

and Bound algorithm. This long CPU-time strongly depends on the parameter s and also P which is understandable for a Branch and Bound code (the complexity of such an algorithm depends on $(\frac{300}{s}+1)^P$ number of solutions).

Therefore an idea to obtain much more precise solutions, is simply to run the Branch and Bound code iteratively by defining more precise zones around the previous exact solutions and by increasing parameter P and decreasing s.

With the solution obtained above (fig.3) $iref = (150, 90, 30, -30, -110)$ over a sampling period $[0, 2]$, $(P = 5; s = 10)$, this one is equivalent to $iref = (150, 150, 90, 90, 30, 30, -30, -30, -110, -110)$ over the sampling period $[0, 1]$, $(P = 10; s = 10)$. Spreading it on a maximum range of 40 amps, we generate a box $IREF = [130, 150] \times [130, 150] \times [70, 110] \times [70, 110] \times [10, 50] \times [10, 50] \times [-50, -10] \times [-50, -10] \times [-130, -90] \times [-130, -90]$.

Each solution found is returned for a new iteration. The following table gives the refined solutions after 3 iterations.

$Instance$	$iref$	E_{min} (J)	$CPU\text{-}time$ (s)	$max.range$ $(amps)$	$Iterations$
$P = 5$, $s = 10$	(150, 90, 30, −30, −110)	23272	3.97	300	16023
$P = 10$, $s = 10$	(150, 140, 90, 90, 30, 30, −20, −20, −100, −130)	22860	920	40	127460
$P = 10$, $s = 5$	(150, 140, 95, 85, 30, 30, −15, −30, −100, −130)	22852	2900	20	227396
$P = 10$, $s = 1$	(150, 139, 97, 85, 30, 30, −16, −31, −101, −131)	22691	805	4	124854

With the improved solutions, we obtain a gain of 2.50% on the performance index for $4625s$ more. The curves (fig.4) are from the latest refined solution.

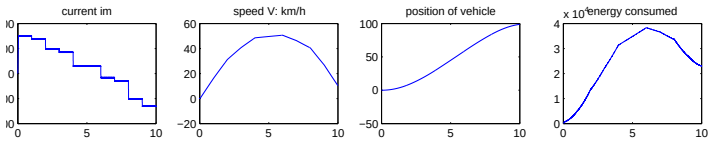


Fig. 4.

We remark that the current i_m remains trapped around $iref$ with respect to the tolerance Δ. The values of u switches many times between -1 and $+1$; this is due to the fact that the current in the motor increases too quickly (around of $3A$ every $10^{-3}s$). Moreover, the curve of the energy decreases at the end of the cycle because this corresponds to the recovery phase. Note that the final speed is not equal to zero because the final time is too short. So, the Branch and Bound code can take this parameter into account. to compare the solutions, we simulate the same previous instance by adding a constraint on the final speed -ie- a displacement of 100 meters, and a cycle $t_f = 10$ seconds with null final speed: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = R \times \{0\} \times \{100\}$. We obtain the following table which compute refined solutions. The minimum energy consumption for this case is higher than the case where we not consider the constraint on the speed.

$Instance$	$iref$	E_{min} (J)	$CPU\text{-}time$ (s)	$max.range$ $(amps)$	$Iterations$
$P = 5$, $s = 10$	$(150,\ 110,\ 40, -70, -150)$	26517	14	300	37437
$P = 10$, $s = 10$	$(150,\ 150,\ 130,\ 90,\ 30,$ $20, -50, -60, -150, -150)$	25645	543	40	97429
$P = 10$, $s = 5$	$(150,\ 145,\ 130,\ 90,\ 35,$ $15, -45, -55, -150, -150)$	25362	472	20	91304
$P = 10$, $s = 1$	$(150,\ 143,\ 129,\ 92,\ 37,$ $14, -45, -55, -148, -150)$	25259	63	4	37764

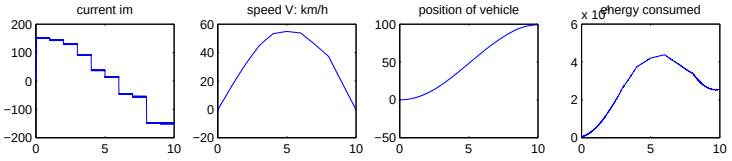


Fig. 5.

the Branch and Bound code can take also the speed parameter into account. We simulate the same previous instance by adding a constraint on the speed state variable who indicates the speed limits authorized for the displacement, -ie- a displacement of 90 meters, and a cycle $t_f = 10$ seconds with null final speed: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = R \times \{0\} \times \{100\}$. We specify that for such parameters, it is not possible to do the displacement of $100m$, with a speed limited to $50km/h$, in one time limited to $10s$. We obtain the following table which compute refined solutions.

$Instance$	$iref$	E_{min} (J)	$CPU\text{-}time$ (s)	$max.range$ $(amps)$	$Iterations$
$P = 5$, $s = 10$	$(150,\ 80,\ 20, -30, -150)$	21284	0.65	300	2911
$P = 10$, $s = 10$	$(150,\ 150,\ 80,\ 70,\ 30,$ $0, -20, -30, -130, -150)$	20218	100	40	42633
$P = 10$, $s = 5$	$(150,\ 145,\ 85,\ 70,\ 25,$ $10, -20, -40, -125, -150)$	20061	145	20	52770
$P = 10$, $s = 1$	$(149,\ 145,\ 84,\ 69,\ 27,$ $12, -19, -39, -127, -150)$	20054	44	4	34442

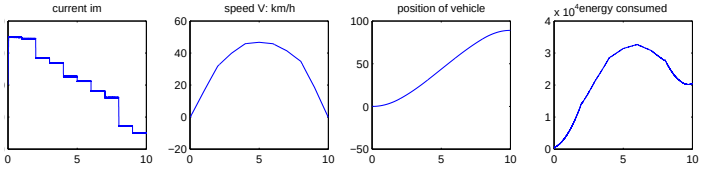


Fig. 6.

7 Application to a Bi-Criteria optimization problem

Let consider the following bi-criteria optimization problem

$$\begin{cases} \min_{i_m(t),\Omega(t),u(t)} \quad (E(t_f, i_m, u), \ -pos(t_f, \Omega)) \\ s.c. \\ \begin{cases} \dot{i_m}(t) = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\ \dot{\Omega}(t) = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega(t) r}{K_r}\right)^2\right)\right) \end{cases} \\ |i_m(t)| \leq 150 \\ \Omega(t) \leq \frac{Vl \times K_r}{3.6 \times r} \\ u(t) \in \{-1, \ +1\} \\ \\ (i_m(0), \ \Omega(0)) = (i_m^0, \ \Omega^0) \in R^2 \\ (i_m(t_f), \ \Omega(t_f)) \in \mathcal{T} \subseteq R^2 \end{cases} \tag{8}$$

Vl is the limit speed authorized for the circulation given in km/h.
$pos(t_f, \Omega)$ is the position of the vehicle at time t_f that is led by the equation (9)

$$pos(t_f, \Omega) = \int_0^{t_f} \frac{\Omega(t) \times r}{K_r} dt. \tag{9}$$

The methodology proposed to solve this problem leans on our Branch and Bound algorithm that provides exact solutions for the discretized problem which correspond to approximate solutions for the global optimization problem (10)

$$\begin{cases} \min_{iref \in \{-150, -150+s, -150+2s, \cdots, 150\}^P} (\sum_{k=1}^{P} E_k, \ -\sum_{k=1}^{P} pos_k) \\ s.c. \\ (E_k, i_k, \Omega_k, pos_k) := VSF(iref_k, t_{k-1}, t_k) \\ \Omega_k \leq \frac{Vl \times K_r}{3.6 \times r} \\ (E_0, i_0, \Omega_0, pos_0) = (E^0, i_m^0, \Omega^0, pos^0) \in R^4 \\ (i_P, \Omega_P) \in \mathcal{T} \subseteq R^2 \end{cases} \tag{10}$$

The two considered criteria (minimize the energy consumed and maximize the distance performed on a fixed cycle) are of conflicting type. By optimizing only one criteria without taking into account the other, we get the ideal point $(E^*_{min}, \ pos^*_{max}) = (31780J, \ 115.38m)$. Theses results are obtained for a cycle of time $t_f = 10s$, $Vl = 50km/h$ with initial conditions $(i_m(0), \Omega(0)) = (0, 0)$ and final conditions $(i_m(t_f), V(t_f)) \in \mathcal{T} = R \times \{50km/h\}$. The parameters of simulation are $P = 5$ and $s = 10$.

For $E^*_{min} = 31780J$, the performed distance is $pos_* = 26.75m$. For $pos^*_{max} = 115.38m$, the energy consumed is $E_* = 47846J$.

To construct the Pareto front (fig.7), we discretize the values of position between pos_* and pos^*_{max}.

In multiple criteria optimization problem, the research of the best compromise used in Goal Programming method implies to minimize the value

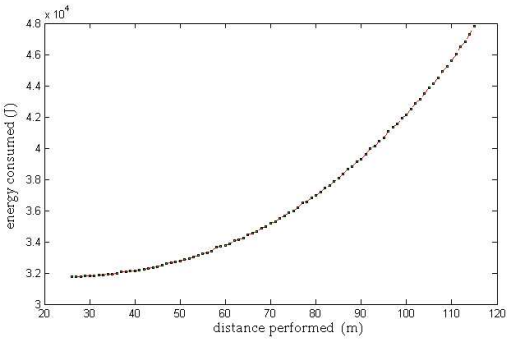


Fig. 7. Pareto front

$\|(E(t_f, i_m, u),\ pos(t_f, \Omega)) - (E^*_{min},\ pos^*_{max})\|_p$, where $\|.\|_p$ is the norm of L^p. It is useful in this case to put the two criteria in the same size order. The obtained solution is reached for $(E = 37615\ J,\ 83\ m)$ with L^2 norm ; $(E = 36544\ J,\ 78\ m)$ with L^1 norm and $(E = 37892\ J,\ 84\ m)$ with L^∞ norm.

8 Conclusion

The energy management problem of an electrical vehicle has been written as an optimal control problem with bang bang control, which is, in general, difficult to solve using PMP and direct methods.

In this paper, we show an original way based on discretization and a Branch and Bound method to solve a hard global optimization problem which is an approximation of an optimal control problem. An algorithm was derived and can be exploited for multicriteria problems. It remains to be improve it by investigating topics such as management of routes with slopes and regulation using the reference speed.

Also, in a future work, we want to improve the efficiency of our Branch and Bound algorithm. Furthermore, we are interested by the resolution of Problem (7) directly by computing bounds.

References

1. A. Sciarreta, L. Guzzella, *Control of Hybrid Electric Vehicles - A Survey of Optimal Energy-Management Strategies*, IEEE Control Systems Magazine, Vol. 27, N. 2, pp. 60–70, 2007.
2. C. Musardo, G. Rizzoni, Y. Guezennec, B. Staccia, *A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*, European Journal of Control, N. 11 (4–5), pp. 509–524, 2005.

3. J. Bernard, S. Delprat, T.M. Guerra, F. Buechi, *Fuel Cell Hybrid Vehicles: Global Optimization based on Optimal Control Theory*, International Review of Electrical Engineering, 1, 2006.
4. R. Trigui, F. Harel, B. Jeanneret, F. Badin, S. Dérou, *Optimisation globale de la commande d'un moteur synchrone à rotor bobiné. Effet sur la consommation simulée de véhicules électriques et hybrides.* Colloque National Génie Électrique Vie et Qualité:GEVIQ'2000. Marseille, 21-22 mars 2000.
5. L. Idoumghar, D. Fodorean et A. Miraoui, *Simulated Annealing Algorithm for multi-objective optimization: Application to Electric Motor Design.* Proceeding of the 29th IASTED International Conference on Modelling, Identification and Control, pp. 190-196, February 15-17, 2010.
6. M.S. Couceiro, C.M. Figueiredo, C. Lebres, N.M. Fonseca Ferreira, and J.A. Tenreiro Machado, *Electric Vehicle Drive System with Adaptive PID Control.* Modelling, Identification, and Control - 2010.

Combinaison de la méthode du gradient et de la méthode de discrétisation en programmation semi-infinie convexe

Ouanes Mohand[1,2], Le Thi Hoai An[1] and Tran Duc Quynh[1]

[1] Laboratoire de l'Informatique Théorique et Appliquée
UFR MIM, Université Paul Verlaine - Metz, Ile du Saulcy, 57045 Metz, France.
lethi@univ-metz.fr
[2] LAROMAD, Département de Mathématiques.
Faculté des Sciences, Université Mouloud Mammeri de Tizi-Ouzou, Algérie.
ouanes_mohand@yahoo.fr

Résumé. Nous proposons une nouvelle méthode pour résoudre les problèmes semi-infinis convexes par l'utilisation d'une combinaison de deux méthodes. L'une est la méthode de descente du gradient et l'autre est la méthode de discrétisation. A chaque itération, la valeur de la fonction objectif décroit strictement et de plus on a une solution admissible, donc si on decide d'arrêter l'algorithme aprés un nombre fini d'itérations, nous disposons d'une solution admissible et pour certains problèmes l'admissibilité est aussi importante que l'optimalité. Des techniques d'optimisation globale sont utilisées pour maintenir l'admissibilité. La convergence de notre algorithme est démontrée. Un exemple numérique est traité pour dérouler notre algorithme.
Mots clés: Optimisation semi-infinie convexe, Optimisation globale, Branch and Bound, Bornes inférieure et supérieure, Méthode du gradient.

1 Introduction

On considère le problème suivant:

$$(SIP)\begin{cases}\min f(x)\\ g(x,s)\leq 0, \forall s\in S\subset R\\ x\in R^n\end{cases}$$

avec f et g de classe C^2 et convexes par rapport à x, S est un compact de R. On suppose que g est lipschitzienne par rapport à x(i.e $|g(x,s)-g(y,s)|\leq L||x-y||, \forall x,y\in R^n, \forall s\in S$). Il existe plusieures méthodes pour résoudre le problème (SIP) exemple: la méthode de discrétisation, la méthode des coupes, la méthode du lagrangien etc...(voir [1], [5], [6], [8], [11]). L'inconvenient pour ces méthodes est que si on décide d'arrêter l'alogorithme après un nombre fini d'itérations la solution approchée n'est pas admissible puisqu'on a des approximations externes, pour certains problèmes l'admissibilité est aussi importante que l'optimalité. Avec notre méthode, on gènère deux suites de points, l'une admissible et l'autre non admissibles donc on a soit une solution optimale ou

bien une solution approchée admissible. Notre méthode consiste en une combinaison de la méthode du gradient et de la méthode de discrétisation en partant d'un point strictement admissible qu'on peut trouver par la méthode des centres des hyperrectangles. A chaque itération la valeur de la fonction objectif décroit strictement. L'article est organisé comme suit: A la section 2, on présente une méthode de construction d'une fonction borne supérieure, à la section 3, l'algorithme et sa convergence sont présentés. un exemple numérique pour dérouler notre algorithme est présenté à la section 4.

2 Fonction borne supérieure

Nous allons maintenant expliquer comment calculer la fonction borne supérieure d'une fonction de classe C^2 sur un intervalle $[a, b]$ qu'on va utiliser dans la méthode Branch and Bound pour le calcul des w_k en résolvant le problème d'optimisation globale $max_{s\in S} g(x^k, s)$.

Pour $m \geq 2$, Soient $\{v_1, v_2, ..., v_m\}$ des fonctions linèaires par morceaux (voir [2]):

$$v_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{si } x_{i-1} \leq s \leq x_i \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{si } x_i \leq x \leq x_{i+1} \\ 0 & \text{sinon.} \end{cases}$$

On a

$$\sum_{i=1}^{i=m} v_i(x) = 1, \forall x \in [a, b] \text{ et } v_i(x_j) = 0 \text{ si } i \neq j, \ \ 1, \text{ sinon.}$$

Soit $L_h f$ la fonction d'interpolation de f aux points $x_1, x_2, ..., x_m$:

$\mathrm{L}_h f(x) = \sum_{i=1}^{i=m} f(x_i) v_i(x)$.

Le résultat suivant de [2] donne une borne supérieure et une borne inférieure de f sur l'intervalle $[a, b]$, $(h = b - a)$.

Théorème 1,[2]
Pour tout $x \in [a, b]$, on a $|L_h f(x) - f(x)| \leq \frac{1}{8} K h^2$, i.e.,

$$L_h f(x) - \frac{1}{8} K h^2 \leq f(x) \leq L_h f(x) + \frac{1}{8} K h^2.$$

Dans [16] La fonction borne infèrieure de f est proposée

$$Lf(x) := L_h f(x) - \frac{1}{2} K(x - a)(b - x) \leq f(x), \ \forall x \in [a, b] .$$

On a montré (voir [16]) que cette borne inférieure est meilleure que celle donnée dans [2]:

$$Lf(x) \geq L_h f(x) - \frac{1}{8}Kh^2.$$

On introduit maintenant une fonction borne supérieure quadratique de f.

Théorème 2
Pour tout $x \in [a, b]$, on a

$$L_h f(x) + \frac{1}{8}Kh^2 \geq Uf(x) := L_h f(x) + \frac{1}{2}K(x-a)(b-x) \geq f(x). \qquad (1)$$

Preuve
Soit $E(x)$ la fonction definie sur $[a, b]$ par

E(x) =$L_h f(x) + \frac{1}{8}Kh^2 - Uf(x)$

E est convexe sur $[a, b]$, et sa dérivée est égale à zéro au point $x^* = \frac{1}{2}(a + b)$. Alors, pour n'importe quel $x \in [a, b]$ on a

E(x)$\geq \min\{E(x) : x \in [a, b]\} = E(x^*) = 0.$

donc, la première inégalité en (1) est vérifiée. Considerons maintenant la fonction ϕ definie sur S par

$\phi(x) := Uf(x) - f(x) = L_h(x) + \frac{1}{2}K(x-a)(b-x) - f(x).$

On a $\phi''(x) = -K - f''(x) \leq 0$ pour tout $x \in S$. alors ϕ est une fonction concave, et pour tout $x \in [a, b]$ on a

$\phi(x) \geq \min\{\phi(x) : x \in [a, b]\} = \phi(a) = \phi(b) = 0.$

La deuxième inégalité dans (1) est donc prouvée.

3 Principe de la méthode

On commence notre méthode par la résolution du problème discrétisé incluant les deux extrémités de S, et on vérifie l'admissibilité, alors on a deux cas
1er cas: La solution est admissible donc elle est optimale puisque le domaine du problème discrétisé contient le domaine du problème (SIP).
2ème cas: La solution n'est pas admissible alors la solution du problème (SIP) se trouve sur la frontière de son domaine (le gradient de f ne s'annule pas sur le domaine de (SIP).
Ensuite on va chercher un point strictement admissible en utilisant la méthode suivante:
Supposons que nous sommes à l'itération k et qu'on n'a pas encore trouvé de

point strictement admissible alors on résout les $2n$ problèmes convexes $(i = 1, .., n)$ suivants :

$$\left(P_{H_k}^{i,\min}\right) \begin{cases} \min x_i \\ g(x, s_j) \leq 0,\ j = 1, .., m \\ g(x,\ s_{x_{H_t}}) \leq 0, t = 0, ..., k-1 \end{cases}$$

$$\left(P_{H_k}^{i,\max}\right) \begin{cases} \max x_i \\ g(x, s_j) \leq 0,\ j = 1, .., m \\ g(x,\ s_{x_{H_t}}) \leq 0, t = 0, ..., k-1 \end{cases}$$

on obtient l'hyperrectangle H_k,on calcule son centre C_{H_k}
ensuite on résout le problème suivant:

$$\max_{s \in S} g(C_{H_k}, s)$$

soit $s_{C_{H_k}}$ sa solution

i) Si $g(C_{H_k}, s_{C_{H_k}}) < 0$ stop C_{H_k} est strictement admissible .

ii) Si $g(C_{H_k},\ s_{C_{H_k}}) > 0$ alors C_{H_k} n'et pas admissible, on ajoute le point $s_{C_{H_k}}$ à la discretisation et on répète la procédure

Comme les hyperrectangles sont emboités et qu'ils contiennent tous le domaine du problème semi-infini, alors ceci va nous permettre de trouver un point strictement admissible.

Soit $\overline{x}^1$ le point strictement admissible, on calcule w_1 en résolvant le problème d'optimisation globale

$max_s g(\overline{x}^1, s) \leq w_1 \leq max_s g(\overline{x}^1, s) + 1$

A l'itération k, on calcule

$max_s g(\overline{x}^k, s) \leq w_k \leq max_s g(\overline{x}^k, s) + \frac{1}{k}$

On pose

$x^{k+1} = x^k + \alpha_k d_k$

Si $\frac{-w_k}{L} \geq \frac{1}{k}$ alors poser

$\alpha_k = \frac{1}{k}$ et $d_k = -\frac{\nabla f(x^k)}{||\nabla f(x^k)||}$

Sinon poser

$\alpha_k = \frac{-w_k}{L}$ et $d_k = \frac{x^{mk} - x^k}{||x^{mk} - x^k||}$

avec x^{mk} la solution du problème discrétisé suivant:

$$(P^{mk}) \begin{cases} \min f(x) \\ g(x, s_i) \leq 0, i = 1, ..., k \\ x \in R^n \end{cases}$$

Théorème 3

Soit x^F un point quelconque de la frontère du domaine du problème (SIP), alors $||x^F - \overline{x}^k|| \geq \frac{-w_k}{L}$

Preuve

x^F est un point de la frontière alors il existe $s^F \in S$ tel que $g(x^F, s^F) = 0$ on a

$|g(\overline{x}^k, s^F) - g(x^F, s^F)| \leq L||\overline{x}^k - x^F||,$

donc

$L||\overline{x}^k - x^F|| \geq (g(x^F, s^F) - g(\overline{x}^k, s^F) \geq -w_k$

ce qui implique

$||x^F - \overline{x}^k|| \geq \frac{-w_k}{L}$

w_k est tel que $\max_s g(\overline{x}^k, s) \leq w_k < 0$

Ainsi les points $\overline{x}^k, k = 0, 1, 2, 3.....$ trouvés par notre algorithme sont admissibles pour le problème (SIP).

4 Algorithme et convergence

On décrit maintenant notre algorithme

4.1 Algorithme

Initialisation: Soit $\varepsilon > 0, S = [s_0, s_1]$, calculer L

Etape 1: Résoudre le problème discrétisé

$$(P_0) \begin{cases} \min f(x) \\ g(x, s_i) \leq 0, i = 0, 1 \\ x \in R^n \end{cases}$$

pour avoir x^{m1}, sinon infeasibilité. Si x^{m1} est admissible stop il est optimal, sinon la solution optimale de (SIP) se trouve à la frontère de son domaine.

Etape 2: Trouver un point strictement admissible $\overline{x}^1$

Etape 3: (Iteration k=1,2,3,.....)

3i) Calculer w_k tel que

$max_s g(\overline{x}^k, s) \leq w_k \leq max_{s \in S} g(\overline{x}^k, s) + \frac{1}{k}$

3ii) Si $-\frac{w_k}{L} \geq \frac{1}{k}$

$\overline{x}^{k+1} = \overline{x}^{(k)} + \frac{\nabla f(\overline{x}^k)}{k||\nabla f(\overline{x}^k)||}$

Si $||\overline{x}^{k+1} - \overline{x}^k|| \leq \epsilon$

stop $\overline{x}^{k+1}$ est une solution $\epsilon-$optimale solution

sinon poser $k := k + 1$ et aller en 3i)

3iii) Si$-\frac{w_k}{L} < \frac{1}{k}$

$\overline{x}^{k+1} = \overline{x}^{(k)} + \frac{-w_k(x^{mk} - \overline{x}^k)}{L||x^{dk} - \overline{x}^k||}$

Si $f(\overline{x}^{k+1}) - f(x^{mk}) \leq \epsilon$

stop $\overline{x}^{k+1}$ est une solution $\epsilon-$optimale

sinon poser $k := k + 1$ et et aller en 3i)

4.2 Convergence

Pour la convergence on a deux cas

1er cas: $\frac{-w_k}{L} \geq \frac{1}{k}$

Théorème 4
$\overline{x}^{k+1} = \overline{x}^{(k)} + \frac{\nabla f(\overline{x}^{(k)})}{k||\nabla f(\overline{x}^k)||}$

alors la suite $\overline{x}^k$ converge vers la solution optimale du problème (SIP)

Preuve

Soit x^* une solution optimale du problème (SIP) ,

$\overline{x}$ la limite de la suite $\overline{x}^k$ (i.e. on a une suite bornée donc elle admet une sous suite convergente vers $\overline{x}$ et on montre facilement que toute la suite converge vers $\overline{x}$)

$d_k = \frac{\nabla f(\overline{x}^k)}{||\nabla f(\overline{x}^k)||}$,

On a

$|\overline{x}^{k+1} - x^*||^2 = ||\overline{x}^k - \alpha_k d_k - x^*||^2 = ||\overline{x}^k - x^*||^2 - 2\alpha_k d_k(\overline{x}^k - x^*) + (\alpha_k)^2||d_k||^2 \leq ||\overline{x}^k - x^*||^2 - 2\alpha_k(f(\overline{x}^k) - f(x^*)) + (\alpha_k)^2$(inégalité de convexité)

On applique cette inégalité recurressivement, on obtient

$||\overline{x}^{k+1} - x^*||^2 \leq ||x^0 - x^*||^2 - 2\sum_{i=0}^{k} \alpha_i(f(\overline{x}^i) - f(x^*)) + \sum_{i=0}^{k}(\alpha_i)^2$

Alors

$2\sum_{i=0}^{k} \alpha_i(f(\overline{x}^k) - f(x^*)) \leq ||x^0 - x^*||^2 + \sum_{i=0}^{k}(\alpha_i)^2 - ||\overline{x}^{k+1} - x^*||^2$

On a

$2min_i(f(\overline{x}^i) - f(x^*))\sum_{i=0}^{k} \alpha_i \leq 2\sum_{i=0}^{k} \alpha_i(f(\overline{x}^i) - f(x^*)) \leq ||x^0 - x^*||^2 + \sum_{i=0}^{k}(\alpha_i)^2 - ||\overline{x}^k - x^*||^2 \leq ||x^0 - x^*||^2 + \sum_{i=0}^{k}(\alpha_i)^2$

ce qui implique en passant à la limite dans les deux membres

$0 \leq (f(\overline{x}) - f(x^*)) \leq \frac{||x^0 - x^*||^2 + \sum_{i=0}^{\infty}(\alpha_i)^2}{2\sum_{i=0}^{\infty} \alpha_i} = 0$

$\Rightarrow$f($\overline{x}$) = $f(x^*)$

Donc $\overline{x}$ est une solution optimale.

2ème cas: $\frac{-w_k}{L} < \frac{1}{k}$

$\overline{x}^{k+1} = \overline{x}^{(k)} + \frac{-w_k(x^{mk} - x^k)}{L||x^{dk} - x^k||}$

On utilise la solution du problème discrétisé x^{mk} qui converge vers la solution optimale du problème (SIP) (voir [11]).

5 Exemple numérique

Example ([1])

$$\begin{cases} \min x_2^2 - 4x_2 \\ x_1\cos(s) + x_2\sin(s) - 1 \leq 0, \forall s \in [0, \pi] \\ x_1, x_2 \in R \end{cases}$$

On résout le problème discrétisé en utilisant les extrémités de l'intervalle $0, \pi$. La solution trouvée est $(x_1, 2), -1 \leq x_1 \leq 1$. On vérifie son admissibilité, on trouve qu'elle n'est pas admissible, alors la solution du problème se trouve sur

la frontière du domaine admissible.

On calcule un point strictement admissible, on trouve $\overline{x}^1 = (0,0)$, on calcule $w_1 = -1$, f(0,0)=0

On calcule la constante de Lipschitz (pour la contrainte), on trouve $L = 1$, on calcule aussi $\nabla f(0,0) = (0,-4)$, et $||\nabla f(0,0)|| = 4$

On a $\overline{x}^2 = \overline{x}^1 + \frac{w_1 \nabla f(0,0)}{L||\nabla f(0,0)||} = (0,1)$
$f(\overline{x}^2) = f(0,1) = -3$
on résout le problème $max_s g(\overline{x}^2, s) = 0$ ce qui implique $w_2 = 0, s^2 = \frac{\pi}{2}$ (i.e. on a le 2ème cas),
On résout le problème discrétisé avec les points $0, \pi, \frac{\pi}{2}($ i.e. pour avoir une nouvelle direction de descente)
On trouve la solution $x^{m1} = (x_1, 1); -1 \leq x_1 \leq 1$
$f(x^{m1}) = -3$
$f(\overline{x}^2) - f(x^{m1}) = 0$
donc $\overline{x}^2 = (0,1)$ est une solution optimale qui est la même que celle trouvée dans [1].

6 Conclusion

Nous avons proposé une nouvelle méthode pour résoudre les problèmes semi-infinis convexes en combinant la méthode du gradient et la méthode de discrétisation. Des techniques d'optimisation globale sont utilisées pour maintenir l'admissibilité. La convergence de notre méthode est démontrée ainsi qu'un exemple illustratif trouvé dans la litterature est traité.

References

1. B. Bhattacharjee,W.H.Green,Jr and P.Barton (2005), *Interval Methods for Semi-infinite Programs*, Computational Optimization and applications, 30:1-13
2. De Boor, C.: A practical Guide to Splines Applied Mathematical Sciences. Springer Verlag (1978)
3. Bradley and M.Bell (1989), *Global convergence of semi-infinite optimization method*, Applied maths and optimization, Springer Verlag N.Y
4. P.G. Ciarlet (1980), *Elements finis*, Dunod,
5. Conn and Gould (1979), *An exact penalty function for semi-infinite problems*, University of Bonn, W. Germany
6. Coope and Watson (1985), *A projected lagragian algorithm for semi-infinite progamming*, Mathematical Programming, North Holland
7. Gribbik (1979), *A central cutting plane algorithm for semi-infinite programming problems*, Univ. Pitsburg, U.S.A
8. Guo-Xin Liu (2007), *A homotopy interior point method for semi-infinite programming problems*, Journal of Global optimization, 37:637-646

9. Gustavson and Glashoff (1983), *Linear optimization and approximation*, Royal institut of technology, Stockholm, Swenden
10. Gustavson and Kortaneck (1984), *Linear optimization and applications*, Royal institut of technology, Stockholm, Swenden
11. Hettich and Kortaneck (1993), *Semi-infinite programming : Theory, methods an applications*, SIAM revue, 35.3:380-429
12. H. Hu (1996), *A globally convergente method for semi-infinite linear programming*, Journal of Global optimization, 8.2:189-199
13. H. Hu (1998), *A one phase algorithm for linear semi-infinte programming*, Mathematical Programming, North Holland
14. O.L. Kostyukova (2000), *An algorithm for constructing solutions for a family of linear semi-infinite problems* ,Variationnal calculus, optimal control and applications, 1-34
15. Le Thi Hoai An and Pham Dinh Tao (1998), *A branch and bound method via D.C. optimization algorithm and ellipsoiadal technique for box constrained nonconvex quadratic problems*, Journal of Global optimization, 13.2:171-206
16. Le Thi Hoai An and M. Ouanes (2004), *A tighter lower bound by convex quadratic function for univariate global optimization* ,Edited by Le Thi Hoai An and Pham Dinh Tao in Hermes Science Publishing :223-231
17. Le Thi Hoai An and M. Ouanes (2006), *Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint*, RAIRO Recherche Opérationnelle, 40:285-302
18. M.K. Luandjula, M. Belhouas and M. Ouanes (1994), *Semi-infinite programming: A collocation approach*, International center for theoritical physics, Triest, Italy
19. M.K. Luandjula and M. Ouanes (2001), *A cutting plane method for semi-infinite optimization* , African journal of science and technology, 1-10
20. G.Still (2001), *Discretization in semi-infinite programming: The rate of the convergence*, Optimization, 53-69

E-learning et travail collaboratif

Un modèle de Garbage Collection pour un éditeur collaboratif en temps réel dans les réseaux mobiles et P2P

Mechaoui Moulay Driss[1], Imine Abdessamad[2], Bendella Fatima[3]

[1]Université de Mostaganem Algérie
moulaydrissnet@yahoo.fr
[2]Inria Lorraine et université de Nancy 2 France
imine@loria.fr
[3]Université des sciences et de la technologie d'Oran USTO Algérie
bendella_fatima@yahoo.fr

Résumé. La progression de la technologie mobile ces dernières années a ouvert de nouvelles issues pour le développement d'applications mobiles dédiées aux dispositifs mobiles tels que iPhone, Andoid et Windows Phone. La transformation opérationnelle (TO) est l'une des meilleures techniques qui supporte la collaboration en utilisant des dispositifs mobiles, elle est utilisée par les éditeurs collaboratifs en temps réel pour permettre à plusieurs utilisateurs de modifier le même document partagé simultanément. Toutefois, le journal (log) d'un éditeur collaborative basé sur l'approche TO peut se développer d'une manière vertigineuse au fur et à mesure que des opérations arrivent au site et que de nouveaux utilisateurs rejoignent le groupe de collaboration, cela finira par épuiser la capacité de stockage sur le site et en particulier si le site est un dispositif mobile. Dans cet article, nous proposons une nouvelle conception d'un modèle de Garbage Collection (ramasse-miettes) distribué pour les environnements mobiles et paire à paire (P2P), ce modèle gère de façon optimale les ressources des dispositifs mobiles et améliore les performances de l'application de l'éditeur collaborative par l'optimisation de la taille du journal.

Mots-clé : Garbage Collection, transformation opérationnelle, dispositif mobile.

1 Introduction

La réconciliation des données divergentes et l'un des problèmes cruciaux des environnements mobiles et distribués [1]. Il existe plusieurs systèmes qui permettent la réconciliation des données divergentes tels que les synchronisateurs de fichiers, les outils pour les PDA (assistant de données personnelles), les outils de gestion de configuration avec les outils de fusion, les algorithmes de réplication optimiste dans les bases de données, algorithmes de groupware en CSCW et les algorithmes des systèmes répartis [2].
Un éditeur collaborative en temps réel est une classe de systèmes distribués basée sur l'interaction de plusieurs utilisateurs tentant de modifier simultanément un objet partagé (texte, image, graphique), tels que des articles scientifiques, pages wiki, agenda distribué, et le code source d'un programme. Un tel système est un moyen

d'établir des collaborations afin de parvenir à une tâche commune. Chaque site dispose d'une copie de l'objet partagé qui peut le modifier à volonté, et les changements sont propagés pour être exécuté sur d'autres copies. Pour intégrer les changements effectués dans les autres sites, les éditeurs collaboratifs utilisent une approche synchrone des transformées opérationnelles (TO) pour la sérialisation des transactions concurrentes.

Motivation. Comme il est nécessaire pour les éditeurs collaboratifs basés sur les transformations opérationnelles utilisant des journaux de mémoriser la trace des opérations reçues pour assurer la convergence des données partagées. La motivation pour tenir un journal sur chaque site, c'est qu'une opération distante doit intégrer l'effet de toutes les opérations concurrentes à être exécutées sur le site récepteur.

Par conséquent, la taille du journal devient très grande lorsque le nombre d'utilisateurs du groupe de collaboration et les opérations échangées augmentent, et aussi lors de l'exécution d'une collaboration intense sur une longue duré de temps. Dans cette situation, le temps d'intégration et de génération des opérations augmentent et par conséquent diminuent les performances des applications collaboratives.

Toutefois, si la collaboration est mobile; de nombreuses opérations peuvent s'accumuler, défiant la capacité des algorithmes actuels basés sur TO qui sont principalement conçus pour des groupes en temps réel et la question posée ici est comment d'intégrer une opération distante à temps ? Même si le périphérique mobile a assez d'espace mémoire, le temps d'intégration d'une opération distante reste toujours coûteux, cela est dû à la modeste capacité de calcul des dispositifs mobiles.

Néanmoins, les opérations reçues par un site collaboratif ne sont pas toutes obligatoirement conservées dans le journal, car une opération peut être supprimée en toute sécurité à partir du journal d'un site si (i) elle est déjà reçue dans tous les autres sites et (ii) toutes les opérations qui en dépendent sont reçues par tous les sites.

Par conséquent, à un moment donné nous avons besoin de supprimer le contenu de tous les journaux dans chaque site, pour recommencer une autre session de la collaboration avec des journaux vides et un même état dans tous les sites, et cela sans provocation de divergence d'état.

Contributions. Dans cet article, nous proposons un système de Garbage Collection pour les éditeurs collaboratifs en temps réel basé sur l'approche de la transformée opérationnelle (TO) en mode distribué.

Notre système permet de (i) nettoyer tous les journaux du groupe de collaboration et (ii) la conservation de la cohérence et la convergence des données partagées et (iii) capable de fonctionner sur des dispositifs mobiles avec le respect de leurs conditions particulières et (iv) effectuer le Garbage Collection sans bloquer l'intégration des opérations à distance et / ou la génération des opérations locales et (v) de répondre aux exigences de Garbage Collection mentionnées dans [3].

Dans ce travail, nous nous sommes basé sur des travaux précédents [3,4,5] sur les éditeurs de collaboration, nous visons à étendre le modèle de collaboration pour qu'il soit supporté sur les dispositifs mobiles et d'améliorer la performance des modèles éditeurs de collaboration existants.

Nous démontrons aussi une bonne mesure de notre modèle de Garbage Collection par une étude expérimentale que nous allons la discuter plus tard.

1.1 Approche de transformation opérationnelle (TO).

Une TO est une technique optimiste qui a été proposée pour résoudre le problème de divergence [6]. Le modèle TO considère N sites, chaque site dispose d'une copie des objets partagés, quand un objet est modifié sur un site, l'opération est exécutée immédiatement et se propage vers d'autres sites pour être exécutée à nouveau. Chaque site sauvegarde toutes les opérations exécutées dans un tampon aussi appelé un journal (Log). Chaque opération est traitée en quatre étapes: (i) la génération sur le site local, (ii) la diffusion vers d'autres sites, (iii) la réception par d'autres sites, (iv) l'exécution sur d'autres sites. Le contexte d'exécution d'une opération *o* reçu peut être différent de son contexte de génération. L'objet partagé est une suite finie d'éléments de tout type de données, il est représenté par une structure linéaire de données, un élément peut être considéré comme un caractère, un paragraphe, une page, un nœud XML, etc. Cette structure linéaire de données peut être facilement étendue à une série de documents multimédia [7]. Il est supposé que l'objet partagé ne peut être modifiée que par les opérations primitive suivantes: (i) Ins (p; e) insère l'élément e à la position p; (ii) Del (p) supprime l'élément à la position p.

1.2 Modèle de Coordination

Dans le modèle utilisé, nous définissons la requête *q* comme un quadruplet (*c*, *r*, *a*, *o*) où *c* est l'identifiant du site collaborateur (ou l'utilisateur) émetteur de la requête et *r* est son numéro de série. Il est à noter que la concaténation de *c* et de *r* est définie comme l'identifiant de *q*. La composante *a* est l'identifiant de la requête de précédence, et enfin *o* est l'opération à être exécutée sur l'état partagé.
Étant donné un journal (log) *L*, *L* [i] désigne la i ^{ième} requête de *L*; | *L* | est la longueur de *L*, *L* [i, j] est le sous-journal (sub-log) de *L* allant de son i^{ème} à la j^{ème} requêtes avec $0 < i \leq j \leq n-1$ tel que $n = | L |$.
Dans cet article, nous présentons un nouveau système de Garbage Collection développé au-dessus d'un modèle d'éditeur collaboratif présenté dans [5].
Nous nous sommes intéressés à ce modèle d'éditeur collaboratif car il offre un nouveau cadre de coordination pour les éditeurs collaboratifs en temps réel qui se caractérisent par les aspects suivants: (i) L'utilisation d'un système de réplication optimiste fournissant un accès simultané à des documents partagés. (ii) la conservation de lien de causalité grâce à une nouvelle technique simple appelée dépendance sémantique.
En outre, ce modèle supporte les groupes dynamiques dans le sens où les utilisateurs peuvent quitter ou joindre les groupes à tout moment. Quand un nouvel utilisateur veut joindre un groupe de collaboration, il demande l'état courant du document et le journal (log) actuel de l'utilisateur le plus proche afin de commencer la collaboration avec les membres de ce groupe.
Un état stable dans un éditeur collaboratif en temps réel est atteint lorsque toutes les opérations générées ont été effectuées sur tous les sites. Pour cela les critères suivants doivent être assurés [5]:
Définition 1. (Critères de cohérence) Un éditeur collaboratif en temps réel est cohérent si est seulement si, il satisfait les propriétés suivantes:

1) la préservation de la dépendance: si o_1 dépend de o_2 alors o_1 est exécuté avant o_2 sur tous les sites.
2) la convergence: lorsque tous les sites ont effectué le même ensemble d'opérations, les copies des documents partagés sont identiques.
A l'état stable, journaux (logs) des sites ne sont pas nécessairement identiques parce que les opérations simultanées peuvent être exécutées dans un ordre différent. Néanmoins, ces journaux doivent être équivalents dans le sens où elles doivent aboutir au même état final.
Définition 2. (Journal Equivalent) Deux journaux sont équivalents si et seulement si ils produisent le même état lorsqu'il est appliqué à un état donné *St*.
Pour éviter le problème TP2 puzzle [4,5], on définit une classe de journaux qui nous permettent de construire des chemins de transformation menant à une convergence des données.
Définition 3. Un journal L est canonique si et seulement si L est la concaténation de deux sous-journaux L_i (ensemble d'opérations ins insertion) et L_d (ensemble de opérations supprimer del) de telle sorte que L_i ne contient pas de requêtes de suppression et L_d ne contient pas de requêtes d'insertion.
Les journaux L_i et L_d peuvent également contenir des requêtes de modification. Dans un journal canonique, on impose un ordre sur les requêtes d'insertion et de suppression: une requête d'insertion doit toujours être avant les requêtes de suppression. A noter que les journaux vides et les journaux contenant uniquement des requêtes d'insertions et / ou suppressions sont également canoniques.

2 Notre modèle de Garbage Collection

2.1 Arborescence du journal

Pour supprimer un journal par une procédure de Garbage Collection, il faut savoir que toutes les opérations du journal ont été intégrées dans tous les sites. Pour vérifier cette condition, nous comparons la totalité du journal de l'initiateur de session de Garbage Collection les journaux des autres sites. Cependant, comparer un journal est une tâche coûteuse surtout si nous manipulons un journal de grande taille. Pour surmonter ce problème, nous proposons la solution suivante. Chaque site maintien son propre journal qui peut être vu comme un arbre de dépendance en utilisant les relations de dépendance sémantique. (voir figure 1).
Les feuilles de cet arbre représentent un résumé de ce qu'un site a reçu comme opérations. Soit L l'ensemble des feuilles maintenues par chaque site s. Par exemple, dans l'exemple de la figure1, $L = \{o_3; o_4\}$.
Deux sites ayant le même ensemble de feuilles signifient qu'ils ont exécuté les mêmes opérations et par conséquence, ils ont un journal équivalent. Cet ensemble de feuilles est mis à jour à chaque fois qu'une opération est générée localement ou une opération distante est reçue en remplaçant l'ancienne feuille par la nouvelle.
La racine de l'arbre de dépendance notée R représente l'identifiant de l'opération racine générée par le site qui a effectué la procédure de Garbage Collection.

L'opération de racine n'a pas d'effet sur l'état des données partagées et respectent les critères canoniques de journal (voir la définition 3), elle sert plutôt comme un point de rupture (breakpoint) indiquant l'initiation de Garbage Collection.

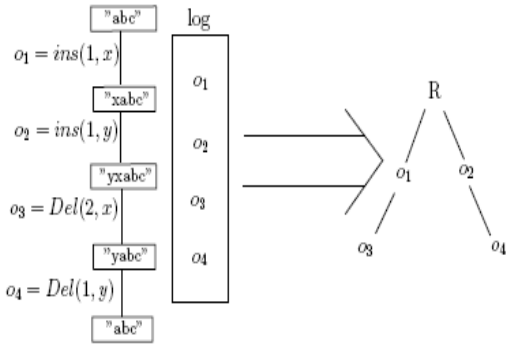


Figure 1. L'arbre de dépendance construite à partir du journal

La suppression d'un élément dépend de l'opération qui a inséré cet élément. o_3 dépend sémantiquement de o_1 et o_4 dépend de o_2 (*pour plus de détails, le lecteur peut se référer à* [5]). Supposons que o_1 et o_2 dépendent de l'opération la racine *R*, le journal est considéré comme l'ensemble des feuilles $\{o_3, o_4\}$ (figure 1). Toutes les opérations concurrentes dépendent systématiquement de l'opération de racine *R*.
Nous supposons d'abord que chaque site commence par un journal contenant seulement une opération de racine générée par le premier utilisateur qui démarre le travail collaboratif. Il est évident que la structure arborescente permet d'extraire l'ensemble des feuilles et facilite ainsi la comparaison entre les journaux des différents d'utilisateurs. Par conséquent, nous réduisons la taille des opérations analysées pour déterminer si la procédure de Garbage Collection est autorisée ou non, puisqu'il suffit juste de comparer l'ensemble des feuilles plutôt que l'ensemble de journaux. Donc l'arborescence permet la comparaison des feuilles est utilisée aussi pour vérifier si tous les utilisateurs ont le même contexte ou pas.

2.2 Algorithme de Garbage Collection

Nous avons conçu un nouveau système de Garbage Collection placé comme une couche supérieure de l'algorithme de contrôle de concurrence [5] désigné pour gérer toutes les interactions simultanées se produisent dans l'éditeur collaborative en temps réel. Cet algorithme [5] repose sur (i) la réplication des documents partagés afin de fournir l'accès aux données sans contraintes, et (ii) un modèle de cohérence basée sur la dépendance causale. Dans notre approche, un éditeur collaborative se compose d'un groupe de N sites (où N est variable dans le temps) commencent une session de collaboration à partir d'un même état initial *St*. Chaque site stocke toutes les requêtes exécutées dans un journal canonique *L*. Notre algorithme de contrôle d'accès concurrent avec le modèle de Garbage Collection est donné dans l'algorithme 1.

Soit *W* l'ensemble des opérations attendues par un site et *D* l'ensemble des sites en ligne du groupe. L'idée de notre solution est de créer un sous-journal (sub-log) dans le même journal canonique d'application lors du lancement de la session de Garbage Collection, ce sous-journal contient l'ensemble d'opérations exécutées après le début de Garbage Collection, et respecte les critères canoniques d'un journal.
Pour appliquer le Garbage Collection, les sites de groupe de collaboration échangent des messages de Garbage Collection afin de décider si le nettoyage de journal est possible ou pas. En fait, il n'est pas toujours possible de supprimer des opérations à partir d'un journal. Par exemple, deux sites ayant des ensembles différents de feuilles sont incapables de nettoyer leurs journaux. On peut déduire qu'il existe des opérations qui ne sont pas encore reçues par tous les sites, pour le faire, les sites collaborateurs échangent des messages afin de décider le lancement de la procédure Garbage Collection :

- Initiation de Garbage Collection (GCI): est un message envoyé par le site initiateur de la procédure de Garbage Collection , le GCI contient l'identifiant du site ainsi que l'ensemble des feuilles de *L*.
- Acquittement (ACK): quand un site reçoit le message GCI, il calcule la différence entre son ensemble de feuilles *L* et celle reçue dans le GCI, puis envoie le résultat à l'initiateur.
- Ordre de Garbage Collection (GCO): ce message contient l'opération de racine de nouveau arbre vide qui remplacera l'ancienne racine. Ainsi, quand il est reçu par les sites, il conduit à la suppression du journal.

Un site peut avoir deux états en fonction de l'état de la collaboration: active et passive. Il est passif quand une procédure de Garbage Collection est en cours d'exécution et dès que la procédure est terminée, il se tourne vers l'état actif. En résumé, le Garbage Collection (GC) effectuece qui suit:
1) Le site initiateur de GC génère une nouvelle opération racine qui sera la nouvelle racine *R′* du journal après la suppression de la racine actuelle et envoie un message GCI pour le reste du groupe. Le GCI message contient l'identifiant (ID) de la nouvelle racine et l'ensemble des feuilles du journal de l'initiateur. Simultanément, l'initiateur continue l'intégration des opérations distantes en utilisant la procédure INTEGRATE_REMOTE_GC. Pour la génération des opérations locales, chaque opération concurrente exécutée localement après le début de la session GC dépendra de la nouvelle racine *R′*. Puisque la génération locale des opérations est assurée par la procédure GENERATE_LOCAL_GC et le contexte de la dépendance est calculé uniquement sur les opérations générées après le début du GC, alors toute opération exécutée après le début du GC dépendra de la nouvelle racine *R'* ou d'une autre opération exécutée aussi après le début du GC. Il est à noter que les opérations locales générées lors de la session GC ne seront pas envoyées jusqu'à ce que tous les accusés de réception (ACK) soient reçus par le site initiateur de GC.
2) A la réception de message GCI, chaque site vérifie si les opérations contenues dans ce message sont déjà exécutées ou pas. Pour faire cette vérification, on calcule la différence entre l'ensemble des feuilles reçues par le message GCI et les feuilles locales du site, le résultat est envoyé dans un message ACK.

```
1: MAIN
2: JOIN
3: INITIALIZATION
4: while not aborted do
5:   if there is an input message m then
6:    GENERATE_MESSAGE(m)
7:  else
8:   RECEIVE MESSAGE
9:  end if
10: end while

11: INITIALIZATION:
12: state ← active
13: R ← " "
14: W ← ø
15: L ← ø
16: s ← Identification of local user
17: initiator ← false
18: FirstInsPos ← 0
19: InsPos ← 0
20: LastDelPos ← 0
21: OfflineOp ← ø

22: GENERATE_MESSAGE(m):
23: if m is an operation then
24:     if state = active then
25:     INTEGRATE_LOCAL_OPERATION
26:     else
27:     INTEGRATE_LOCAL_GC
28:     end if
29: else
30:     if m is a GCI then
31:     LAUNCH_GC()
32:     end if
33: end if

34: RECEIVE MESSAGE:
35: if m is an operation then
36:     if m ∈ W then
37:       W ←W - m
38:    end if
39:    if state = active then
40:    INTEGRATE_REMOTE_OPERATION
41:     else
42:    INTEGRATE_REMOTE_GC
43:     end if
44: else
45: if m is a log request then
46:     if state=active then
47:     SEND LOG
48:     end if
49: end if
50: else
51:     RECEIVE_GCI_MESSAGE (m)
52: end if
```

Algorithme 1. Algorithme de contrôle de la concurrence avec le modèle Garbage Collection

```
1. INTEGRATE_LOCAL_GC :
2. L ← Do(o, L)
3: q ← (c, r, null, o)
4: q' ← COMPUTEBF_GC(q,L)
5: L ← Up_LocalOp(q,L)
6: broadcast q' to other users
```

Algorithme 2. Génération des opérations locales pendant GC session

```
1:INTEGRATE_REMOTE_GC:
2: if there is q in Q that is causally-ready then
3: Delete q from Q
4: q'← ComputeFF(q,L)
5: apply o on current state
6: L← Up_RemoteOp (q',L)
7: end if
```

Algorithme 3. Intégration des opérations distante pendant GC session

```
1: RECEIVE_GC_MESSAGE (m):
2: if m = GCI(s';L'_s) then
3: if s < s_0 and initiator = true then
4: initiator ← false {Abort garbage collection initiation}
5: end if
6: state ← passive
7: L_r ← L \ L_s
8: send ACK(L_r)
9: else
10: if m = ACK(L'; s') and initiator = true then
11: W ← W ∪ L'
12: D ← D \ {s'}
13: end if
14: else
15: if m = GCO(s';R') then
16: L ← ReOrder_Log (L)
17: clean log
17: integrate the new root R'
18: R ← R'
19: state ← active
20: end if
21: end if
```

Algorithme 4. Réception et traitement des messages de GC

Le résultat de la différence représente les feuilles exécutées par le site et qui ne sont pas encore vu par le site initiateur. Chaque site continue la génération locale des opérations en utilisant GENERATE_LOCAL_GC et l'identifiant de la nouvelle racine *R′* générée par le site initiateur de la session de GC reçu dans le message GCI en tant que nouvelle racine. La procédure INTEGRATE_REMOTE_GC est appelée pour l'intégration des opérations distantes.

3) Chaque fois que l'initiateur reçoit un message ACK, il stocke la différence localement dans sa propre liste des opérations attendu *W*. Cette liste est mise à jour chaque fois que l'initiateur reçoit l'une des opérations attendues par l'extraction de cette dernière de l'ensemble. En d'autres termes, un message ACK est causalement prêt quand toutes les feuilles qu'il contient sont exécutées localement.

4) Tous les sites continus de la génération locale des opérations en session GC en utilisant GENERATE_LOCAL_GC jusqu'à ce que l'initiateur reçoit tous les ACKs et que toutes les opérations attendues sont exécutés localement. A noter que les ACKs attendus seulement sur les sites connectés qui ont été découverts initialement par l'initiateur et poursuivent la collaboration. Les nouveaux sites ou les sites déconnectés sont ignorés. De cette façon, nous assurons que l'initiateur ne va pas attendre les ACKs indéfiniment.

5) Lorsque tous les ACKs attendus sont reçues et $W = \emptyset$; le site initiateur de GC ré-ordonne son journal en utilisant la fonction ReOrder_Log (voir l'algorithme 5) et supprime le sous-journal qui contient les opérations exécutées avant d'entamer la session GC, et envoie un message d'ordre de Garbage Collection (GCO) avec l'opération de racine R' aux autres sites du groupe et envoie également les opérations générées localement lors de la session du GC.

6) Lors de la réception du message GCO., chaque site supprime le sous-journal qui contient les opérations exécutées avant d'entamer la session du GC, et intègre l'opération de racine *R′* reçu dans le message GCO et envoie les opérations générées localement lors de la session du GC aux autres sites et intègre les opérations distantes qui dépendent de la nouvelle opération racine *R′*.

2.2.1 Génération des requêtes locales

La génération d'une requête locale en session GC est assurée par la procédure décrite dans l'algorithme 2. L'idée est de générer des opérations locales uniquement sur le sous-journal contenant les opérations locales vues (exécutées) après le Garbage Collection (GC) et non pas sur la totalité du journal. Pour cela, il faut placer les opérations locales générées après le GC entre les opérations d'insertion (ins) et les opérations de suppression (del) vues avant l'initiation de GC en utilisant la fonction Up_LocalOperation , ceci pour faciliter la génération de ces opérations locales, et de respecter les critères de journal canonique (voir la définition 3) et permet aussi d'appliquer le GC sans bloquer l'utilisateur.

L'algorithme 2 utilise la fonction Do(*o*, *L*) (voir [5]) pour effectuer l'effet de *o* sur l'état courant de document *St*, et utilise ComputeBF_GC(o) pour calculer les relations de dépendance sur le sous-journal contenant les opérations qui sont générées après l'initiation de GC , les variables *FirstInsPos* et *LastDelPos* délimitent l'intervalle des sous-journaux.

La fonction ComputeBF_GC (o) a les mêmes fonctionnalités de ComputeBF(o) [5], la seule différence c'est qu'elle déduit les relations de dépendance entre les opérations générées localement sur le sous-journal seulement.
Après la définition de la dépendance, *o* est stocké dans le journal local selon deux cas:

- Si *o* est une opération d'insertion (***ins"***), elle est placée après les opérations d'insertion exécutées avant le GC, et avant les opérations de suppressions exécutées avant GC (L = {ins, ins, ***ins"***, del, del}).
- Si *o* est une opération de suppression (***del"***), elle est placée avant les opérations de suppression exécutées avant le GC. (L = {ins, ins, ***del"***, del, del}).

L'ordre de placement est assuré par la fonction Up_LocalOperation. Par exemple, supposons *L* un journal qui contient un ensemble d'opérations exécutées avant l'initiation de GC *L* = {ins, ins, ins, del, del} et soit ***ins"***, ***del"*** des opérations exécutées après le début de la session de GC. Après la génération de ces opérations, nous obtenons le journal suivant : *L* = {ins, ins, ***ins"***, ***del"***, del, del}.

2.2.2 Intégration des requêtes distantes

Lorsqu'une opération distante *o* est reçue après l'initiation de GC. Si *o* est causalement prête (si sa dépendance a déjà été intégrée sur le site récepteur) elle est intégrée en utilisant la procédure INTEGRATE_REMOTE_GC.
Cette procédure fait appel à la fonction COMPUTEFF() [5] afin de calculer la forme transformée q_0 a être exécuter sur l'état actuel *St*. En suite la fonction Up_RemoteOperation() est appelé pour placer l'opération transformée à la bonne place sur le sous-journal créé après l'initiation de GC.
Il y a deux cas pour placer l'opération dans le journal:

- Si *o* est une insertion (***ins***), elle est placée avant les opérations d'insertions exécutées après GC. (*L* = {ins, ins, ***ins***, ins", del", del, del) et le variables *InsPos*, *LastDelPos* sont incrémentées d'une position.
- Si *o* est une opération de suppression (***del***), elle est placée à la fin du journal. (*L* = {ins, ins, ins", del", del, del, ***del*** })

Nous notons que les opérations ins", del" représentent les opérations générées après l'initiation de la session de GC.

2.2.3 La construction du sous-journal

Le sous-journal contient toutes les opérations générées localement après l'initiation de GC dans l'ancien journal. Quand une session de GC est lancée, chaque site reçoit le message GCO, commence à calculer les relations de dépendance uniquement sur le sous-journal délimitée par l'intervalle *FirstInsPos* et *LastDelPos* positions respectivement dans le même journal.
La variable *FirstInsPos* mise à jour lors qu'une opération distante est intégrée et la variable *LastDelPos* est mise à jour à chaque génération ou l'intégration d'une l'opération.
Initialement la variable *FirstInsPos* égale à la position de la dernière opération d'insertion dans le journal, plus une position.
Lorsque l'utilisateur génère une opération d'insertion, on calcule ces relations de dépendance par la fonction ComputeBF_GC et on place cette opération dans la

position *InsPos* en utilisant la fonction Up_LocalOperation et on incrémente *InsPos* et *LastDelPos* par une position. ComputeBF_GC a les même fonctionnalités que ComputeBF [5], mais seulement sur l'intervalle délimité par les positions *FirstInsPos* et *LastDelPos* qui constituent le sous-journal.
Dans le cas d'une opération distante, l'opération est intégrée sur la totalité du journal et elle est placée juste avant *FirstInsPos* d'une position à l'aide de la fonction Up_RemoteOperation et les variables *FirstInsPos*, *InsPos*, *LastDelPos* sont incrémentées d'une position.

2.2.4 Réorganisation de sous-journal

Lorsque l'initiateur de la GC reçoit tous les messages ACK, l'initiateur peut commencer la suppression des opérations obsolètes dans le journal (opérations exécutées avant GC). Mais avant cela, le journal est réorganisée pour séparer le sous-journal de l'ancien journal, en d'autres termes, placer le sous-journal généré lors de la session du GC dans le bas du journal en utilisant la fonction ReOrder_Log (voir l'algorithme 5), pour tenir un journal équivalent, la fonction ReOrder_Operations (voir l'algorithme 6) est appelée pour effectuer des opérations de permutation successives.

```
1: ReOrder_Operations(q,i): L'
2: L' ← [L; q]
3: j ← LastDelPos -1
4: while (j < i)
5:   <L'[j], L'[j+1] > ← PERM (L'[j+1], L'[j])
6:   j ← j + 1
7: end while
8: LastDelPos ← LastDelPos -1
9: return L'
```

Algorithme 5. Réorganiser la position d'une opération

```
1: ReOrder_Log(L): L'
2: i ← L-1
3: while FistInsPos < LastDelPos
4:   ReOrder _Operation(L[i],i)
5:   i ← i - 1
6: end while
7: return L'
```

Algorithme 6. Réorganisation de structure du journal

Le résultat sera deux sous-journaux, le premier contient les opérations exécutées avant la session du GC et le deuxième contient les opérations exécutées après la session du GC. Ces sous-journaux sont équivalents à l'ancien journal.
Par exemple, soit L un journal, *ins*, *del* des opérations exécutées avant le GC et ***ins",del"*** des opérations exécutées après le GC alors *L = {ins, ins, **ins", del"**, del, del}* et après l'exécution de la fonction ReOrder_Log, le journal devient *L = {ins, ins, del, del, **ins", del"**}*

Après la réorganisation du journal, l'initiateur ne supprime que le sous-journal qui contient les opérations effectuées avant le début de GC (L = {ins, ins, ***ins"***, ***del"***, del, del} et après la suppression le journal devient L = {***ins"***, ***del"***}). L'initiateur de GC génére aussi un GCO message, et envoie le GCO pour les autre sites du groupe.

3 Mise en œuvre et Performances

Nous avons implémenté notre solution sur l'émulateur Android 1.5[1], il fonctionne sur Windows XP SP2 avec un processeur dual core 2,8 et 2Go de RAM.

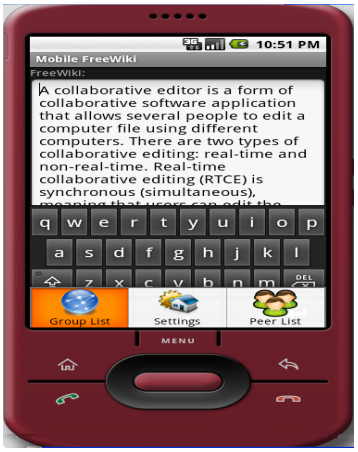


Figure 2. Capture d'écran de notre prototype

Pour notre évaluation de performances, nous considérons les temps suivants utilisés pour calculer le temps de réponse de notre modèle:

- Tg est le temps nécessaire pour générer une opération locale;
- Ti est le temps d'intégration une opération distante;
- Tc est le temps nécessaire pour communiquer une opération à un site à travers le réseau;
- Tr est le temps de réponse, et il représente la somme des Tg, Ti et Tc (Tr = Tg + Ti + Tc).

En général, il est établi que les éditeurs collaboratifs basés sur l'approche OT doivent fournir un temps de réponse tr <100ms [8], la collaboration est meilleur si le temps de réponse est inférieur a 100ms. En fait, l'utilisateur est capable de voir les différentes mises à jour effectuées sur les documents partagés instantanément.
Pour étudier la performance de notre prototype réalisé pour les téléphones mobiles, nous avons fait des essais expérimentaux sur le comportement de notre modèle développé sur Android1.5. L'expérience consiste à calculer le temps de réponse des opérations dans le pire des cas. Le pire des cas se produit lorsque le journal ne contient que des opérations de suppression (voir [5]). Puis nous avons mesuré le temps requis pour générer une insertion et nous l'avons intégré sur un site distant.
La figure 3 montre le temps de réponse des différentes valeurs de la taille du journal. Ces mesures reflètent les temps de génération Tg, et l'intégration Ti des opérations et leur temps de réponse Tr. Le temps d'exécution relève de 100 ms pour tous |H| <2000.

[1] http://developer.android.com/index.html

Nous concluons qu'au-delà de 2000 opérations le journal doit être nettoyé par le mécanisme de Garbage Collection. Ce résultat démontre l'utilité de Garbage Collection dans l'environnement mobile pour maintenir la performance d'exécution des applications. En atteignant des tailles données, tous les utilisateurs commenceront à nouveau la collaboration avec des journaux vides qui rendent la collaboration de plus en plus efficace.

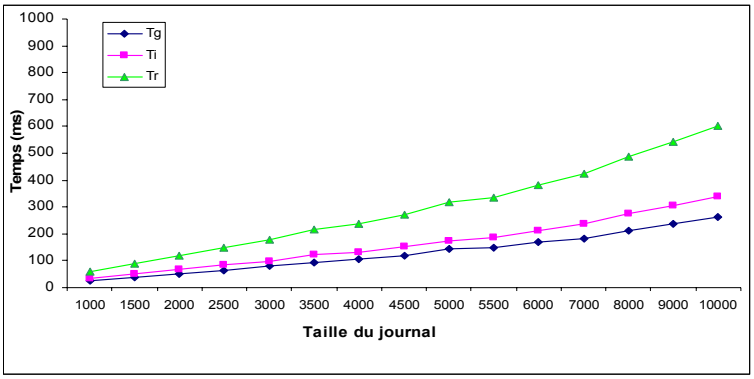


Figure 3. Temps de réponse sur Android 1.5.

L'expérience suivante est réalisée pour mesurer le temps nécessaire par le groupe afin d'effectuer le Garbage Collection. L'expérience des mesures cette fois pour différentes valeurs de l'ensemble des feuilles appartenant à l'initiateur. Selon les résultats présentés dans la figure 4.

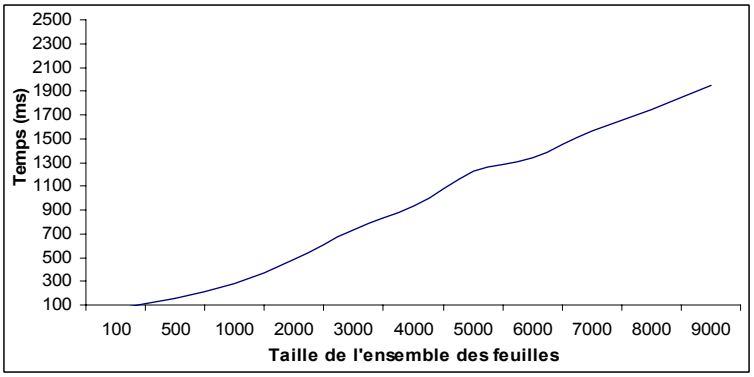


Figure 4. Variation de temps de Garbage collection avec taille de l'ensemble de feuilles.

Nous déduisons que le temps nécessaire pour la réalisation d'une session de Garbage Collection pour une taille d'un ensemble feuille égale à 2000 opérations est inférieur à 500 ms. Il faut de noter que la valeur 2000 ne représente pas la taille du journal, mais plutôt l'ensemble de la taille des feuilles ce qui signifie que le journal pourrait contenir plus que ce nombre d'opérations.

4 Travaux connexes

Pour appliquer le Garbage Collection sur un éditeur collaborative distribué nous supposons avoir une vue globale sur l'état du système distribué. Chandy et Lamport [9] proposent un algorithme pour déterminer les états globaux des systèmes distribués par l'enregistrement d'une logique snapshot (ou de causalité) du système.
Dans le domaine des bases de données, le travail de [10] est pertinent. Il propose une technique de réduction de taille permettant la suppression des différentes mises à jour qui ne sont plus nécessaires pour le système, aussi [11] propose d'appliquer un Garbage Collection de journal de base de données en utilisant un horodatage de coupure pour déterminer un état global du système. Cependant, l'utilisation de l'horodatage, est inappropriée dans un contexte dynamique. La solution proposée dans [12], est un protocole qui permet aux sites dans un système de bases de données répliquées de supprimer les anciennes mises à jour pour maintenir la cohérence mutuelle. Cette solution ne pouvait pas satisfaire les exigences des éditeurs collaboratifs en temps réel car il est difficile d'assurer un ordre global de manière décentralisée avec des groupes dynamiques. Autres travaux tels que [13] ont été proposés pour mettre l'accent sur le Garbage Collection dans les bases de données orientées objet, mais ils intéressent plutôt da gestion de mémoire pour supprimer les objets non utilisés et non dans le nettoyage des journaux.
A notre connaissance, [6] est le seul travail qui a proposé une technique de Garbage collection afin de réduire la taille du journal pour les éditeurs collaboratifs distribués.
Dans [3] nous avons proposé un système de Garbage Collection pour les éditeurs collaboratifs en temps réel basés sur l'approche TO et bâtis sur des réseaux paire à paire (P2P) et mobiles. La limite de cette proposition est le blocage de la collaboration lors du lancement d'une procédure de Garbage Collection, en d'autres termes, le groupe reste bloqué jusqu'à la fin du processus de Garbage Collection. Nous notons que notre solution est une extension du schéma proposé par [3] pour surmonter ses limites.

5 Conclusion

Dans ce travail nous avons proposé un système de Garbage Collection distribuée pour l'éditeur de collaboration en temps réel basée sur l'approche TO et construit sur des réseaux P2P et mobiles. L'objectif principal de ce modèle est d'améliorer les performances de l'application d'éditeur collaboratif en éliminant les anciennes opérations à partir du journal une fois qu'il a été déterminé par un accord commun d'une procédure de Garbage Collection. Notre système de Garbage Collection permet l'intégration des éditeurs de collaboration dans les dispositifs mobiles, conserve la convergence des données, permet d'optimiser la taille du journal d'application, et offre donc de meilleures performances.
Notre modèle de Garbage Collection est applicable dans l'éditeur de collaboration en temps réel sur les dispositifs mobiles ainsi que sur les ordinateurs ordinaire (PC). Vu au nombre limité de pages, on n'a pas pu détaillée toutes les fonctionnalités de notre solution.

Dans nos prochains travaux, nous prévoyons d'améliorer notre système de Garbage Collection par l'élaboration d'un modèle de la sécurité qui répond aux besoins de travail collaboratif et qui respecte les caractéristiques des dispositifs mobiles.

Références

[1] Clarence A. Ellis and Simon J. Gibbs, "Concurrency Control in Groupware Systems". SIGMOD Conference 18, pp. 399–407 1989

[2] P. Molli, G. Oster, H. Skaf-Molli, and A. Imine," Using the transformational approach to build a safe and generic data synchronizer". Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, pp. 212–220. ACM Press, Florida, USA (2003).

[3] Mechaoui M.D, Cherif A., Imine A. and Bendella F.,"Log Garbage Collector-based Real Time Collaborative Editor for Mobile Devices". In 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (IEEE CollaborateCom 2010), Chicago, USA (2010).

[4] Imine A., "Conception Formelle d'Algorithmes de Réplication Optimiste. Vers l'Edition Collaborative dans les Réseaux Pair-à-Pair". Phd thesis, University of Henri Poincaré, Nancy, France (2006).

[5] Imine A., "Coordination Model for Real-Time Collaborative Editors". COORDINATION, pp. 225-246 (2009).

[6] Sun C., "Achieving Convergence, Causality-preservation, and Intention-preservation in Real-time Cooperative Editing Systems". ACM Transactions on Computer-Human Interaction 5, pp. 63–108 (1998).

[7] Sun C., Xia S., Sun D., Chen D., Shen H., and Cai W., " Transparent adaptation of single-user applications for multi-user real-time collaboration". ACM Trans. Comput.-Hum. Interact., 13(4):pp. 531–582 (2006).

[8] Du Li and Rui Li, "An Operational Transformation Algorithm and Performance Evaluation ", Computer Supported Cooperative Work, pp. 469-508 (2008).

[9] Chandy K. M, and Lamport L.," Distributed snapshots: Determining Global States of Distributed System",vol 3, No.1. ACM Transactions on Computer Systems, pp. 63-75 (1985).

[10] Samarati P. and Ammann P. and Jajodia S.,"Maintaining replicated authorizations in distributed database systems". Data & Knowledge Engineering journal 18, pp. 55–84 (1996).

[11] Sunil K. Sarin, Charles W. Kaufman, and Janet E. Somers," Using History Information To Process Delayed Database Updates",VLDB '86 Proceedings of the 12th International Conference on Very Large Data Bases 1986. pp. 71-78. San Francisco, CA, USA (1986).

[12] Sunil Sarin and Nancy Lynch and A. Lynch, "Discard Obsolete Information In A Replicated Database System". IEEE Transaction on Software Engineering, Vol. SE-13, No. 1, pp. 39-47 (1987).

[13] Roy P. and Seshadri S. and Silberschatz A. and Sudarshan S. and Ashwin S., "Garbage collection in object oriented databases using transactional cyclic reference counting", In VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, pp. 366–375 (1997).

[14] Matthias Ressel and Doris Nitsche-Ruhland and Rul Gunzenhauser," An Integrating, Transformation-Oriented Approach to Concurrency Control and Undo in Group Editors". ACM CSCW'96", Boston, USA. pp. 288-297 (1996).

[15] Brad Lushman and Gordon V. Cormack ,"Proof of correctness of Ressel's adOPTed algorithm", Information Processing Letters 86, pp. 303–310, Elseiver B.V (2003).

[16] Shao B., Li D. and Gu N., "A Sequence Transformation Algorithme for Supporting Cooperative Work on Mobile Devices", Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW 2010, Savannah, Georgia, USA pp.159-168 (2010).

Modélisation d'une situation d'évaluation de l'apprenant avec UML: CAS d'application pour l'apprentissage des langages de programmation

Boussaha Karima ,Departement d'informatique, université larbi ben m'hidi om elbouaghi ,Laboratoire d'automatique et d'informatique de Guelma ,Algerie
Karima_et_rania @yahoo.fr

Abstract. Cet article témoigne de l'évolution de nos travaux de recherche entre 2007 et 2009. Notre travail sur la modélisation de TéléTP en programmation s'est concrétisé par une conception d'un environnement de TéléTP (E-TéléTP@AALP)[17]. Un retour d'usage sur cette première conception nous a permis de proposer une nouvelle conception améliorée notamment par l'intégration d'un mécanisme d'évaluation de l'apprenant au cours de son processus d'apprentissage.

Keywords: Auto-évaluation, didactique de la programmation, environnement informatique, programmation sur exemple, scénario d'évaluation,EIAH

1 Introduction

L'enseignement à distance par Internet, appelé EIAH (Environnement Informatique pour l'Apprentissage Humain), constitue une avancée pédagogique importante. L'EIAH utilise le web (structure hypertexte, capacités multimédias, etc.) comme support de diffusion des connaissances et d'interaction entre les différents acteurs (enseignants, apprenants, etc.). Plusieurs plateformes d'EIAH ont été développées et plusieurs sont disponibles sur le web en libre accès[11]. Ces plateformes sont des environnements qui permettent à un enseignant de créer et de gérer très facilement un cours sur Internet, en lui laissant le libre choix de la méthode pédagogique, et sans nécessiter de compétences informatiques particulières. Elles offrent aussi des outils de communication (forums, chat), des instruments d'évaluation (exercices, sondages, travaux), et la possibilité de déposer des ressources pédagogiques (fichiers PDF, séquences vidéo, etc.). Le travail présenté dans cet article, s'intéresse à l'énoncé d'une problématique que nous jugeons Importante à prendre en considération, lors de l'élaboration d'un environnement d'apprentissage des travaux pratiques en programmation c'est l'évaluation de l'apprenant. C'est important pour lui, qu'il puisse distinguer ses forces et ses faiblesses durant tout son parcours pédagogique.
Nous pensons que l'évaluation, dans sa fonction formative, est au coeur de l'apprentissage vu sa fonction régulatrice qui est primordiale[7].

Le travail proposé s'inscrit dans le domaine de recherche des environnements informatiques pour l'apprentissage humain (EIAH). Et concerne plus particulièrement l'évaluation de l'apprenant dans les environnements d'apprentissage des travaux pratiques à distance (TéLéTP) en programmation.
L'évaluation dans les EIAH reste le parent pauvre, car très souvent absente ou obsolète dans les formalismes proposés. Or l'évaluation ,qui est une part importante de l'activité pédagogique , mérite de disposer elle aussi des méthodes ,des techniques et des outils qui font d'une manière générale évaluer le contexte D'EIAH.

2 principales Difficultés dans l'apprentissage de la programmation

La programmation est une discipline longtemps utilisée de manière naïve, sans formalisme particulier. Cette discipline est souvent source de problème pour l'enseignant ainsi que pour l'apprenant.

2.1 Pour l'enseignant

Parce qu'il doit trouver les méthodes adéquates pour faire assimiler des concepts assez abstraits à des étudiants qui ne sont qu'à leur phase d'initiation. Des études confirment que le taux d'échec ou d'abandon aux cours d'initiation à la programmation en premier cycle universitaire varient de 25 à 80% de part le monde.

2.2 Pour les apprenants

Nous avons demandé aux apprenants qui ont l'expérience d'utiliser la plateforme E-TéléTP@AALP de signaler les obstacles et les difficultés rencontrés pendant la réalisation de leurs TP [18]. Dans ce sens un questionnaire est disponible. On a remarqué que la majorité des apprenants concernés par l'enquête estiment que la tâche la plus difficile dans la programmation est la correction des erreurs. La plus part des apprenants collabore dans cette phase. Malheureusement, généralement les apprenants bloquent sur les mêmes types d'erreurs. Ce qui nous amène à la question suivante " quelles méthodes pédagogiques et avec quels outils d'évaluation peut on améliorer l'apprentissage de la programmation?

3 situation initiale

E-TéLéTP@AALP se veut un environnement pour l'enseignement et l'apprentissage à distance des travaux pratiques en programmation en informatique. Réalisée en 2009, elle possède une architecture de trois niveaux (téléformation, télé programmation, interface)[19].

La plateforme est développée en ASP/ MYSQL et fonctionnée sur n'importe quel environnement.

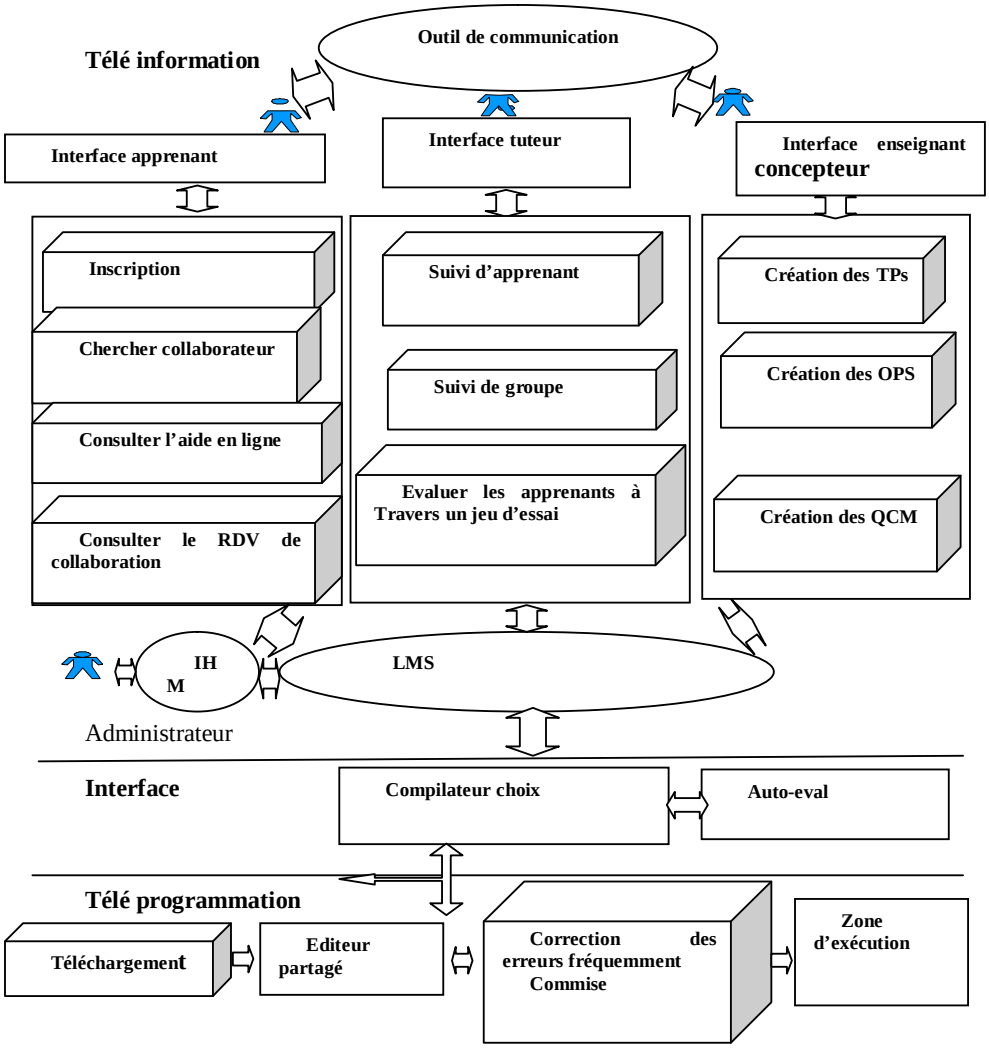


Fig. 1. architecture de la plateforme E-TéLéTP@AALP[18].

4 Conclusion tirée de cette situation

Dans un processus d'apprentissage des travaux pratiques à distance, il est important pour un apprenant d'évaluer ses forces et ses faiblesses afin de pouvoir corriger les

erreurs de compilation et réaliser son Tp facilement., l'évaluation est un dispositif clé. En effet, cette évaluation dans le contexte d'un EIAH doit :
– être motivante pour l'apprenant,
– encourager une activité d'apprentissage soutenue,
– contribuer à la progression de l'apprenant,
– être faible en coût humain et facilement maintenable.
Donc la réutilisation et le retour d'usage de cette conception de la plateforme E-TéléTP@AALP ont orienté nos recherches pour améliorer les services de cette plateforme avec l'intégration d'un nouveau module d'évaluation.

5 Nouveaux objectifs

L'évaluation permet à l'apprenant de connaître le niveau d'acquisition des connaissances, méthodes et sa capacité à résoudre un problème donné tout seul sans l'aide de l'environnement. C'est une opération qui s'avère compliquée dans le cas de la programmation puisqu'un problème peut être résolu de différentes manières, et en utilisant des méthodes différentes[7].

L'objectif de ce travail est de concevoir un prototype d'évaluation des apprenants et d'identifier jusqu'à où il est possible d'intégrer ce module dans notre plateforme E-TéLéTP@AALP destinée à l'apprentissage des travaux pratiques sur les langages de programmation en informatique. Pour cela nous avons proposé une architecture de ce module d'évaluation, ainsi que l'architecture d'un générateur permettant de gérer à partir d'un programme réalisé par l'apprenant un diagramme de classe pour pouvoir le comparer avec ceux enregistrés dans la base.

6 l'évaluation dans les EIAH

L'évaluation est tributaire des activités proposées par l'EIAH. En effet, l'évaluation menée dans le cadre d'une activité collective n'est pas toujours valide dans une activité individuelle.

6.1 L'évaluation dans des activités individuelles

6.1.1.Evaluation de production : le bilan de compétences

Le test du TOEFL et le logiciel Pépite sont certainement les plus connus.

6.1.2. Evaluation de production : l'auto-évaluation

Le meilleur exemple de ce type d'évaluation c'est le logiciel GenEval.

6.1.3. Evaluation de la démarche : l'assistance à l'évaluation

S'il est possible d'évaluer relativement finement une production, il n'en est pas de même pour une démarche. Aussi, que ce soit dans des activités individuelles ou collectives, des solutions d'assistance à l'évaluation sont proposées.

6.2 L'évaluation dans des activités collaboratives

6.2.1. Evaluation de production : l'auto-évaluation collaborative

Une des premières pratiques d'évaluation des activités collaboratives en EIAH est l'auto-évaluation par pair. Cette évaluation est menée par un groupe d'apprenants. et porte sur une production réalisée par un autre groupe.

6.2.2. Evaluation de la démarche: l'évaluation de la participation

L'évaluation de la participation se retrouve quasi essentiellement dans le cadre d'activités usant d'un forum,Tel est le cas du logiciel DIAS.

6.2.3. L'évaluation de la démarche : l'assistance à l'évaluation

Alors que l'assistance à l'évaluation trouvait du sens dans une activité individuelle, elle devient nécessaire dans le cadre de l'activité collective qui démultiplie la complexité pour l'enseignant d'en évaluer le déroulement.

7 Notre solution proposée

7.1 Travaux similaires

Dans ce contexte, nous citons Allogène qui est un système d'apprentissage par L'exemple. L'association du déroulement graphique de l'algorithme (à travers les Variables) est intéressante. MELBA (Metaphor-based Environment to Learn the Basics of Algorithmics) propose des métaphores pour expliciter le concept de

variable, de type, de paramètre et de référence.

Une autre approche est proposée par Duchâteau [12]. Cette méthode d'apprentissage des bases de la programmation était révolutionnaire : son principe de base est de fournir à un débutant une représentation simple et imagée de la manière dont cette boîte noire, que représente l'exécutant ordinateur, fonctionne et qu'il puisse s'appuyer sur cette «vision » mentale pour concevoir ses programmes. La Méthode CAR n'apporte fondamentalement rien de neuf par rapport à « Images pour programmer ». Les concepts, images et analogies sont généralement conservés, mais l'adoption d'un Support interactif permet toutefois l'apport d'éléments supplémentaires tels que les Animations, les auto-tests, etc.

Tous les travaux que nous citons s'intéressent à l'apprentissage mais l'évaluation reste le parent pauvre. L'approche que nous proposons répond aux deux objectifs : l'apprentissage et l'évaluation à la fois.

7.2 Solution

7.2.1 Selon la phase d'apprentissage de l'apprenant

On demande à l'apprenant de deviner une solution particulière, celle proposée par l'environnement et cela de différentes manières **:**

L'environnement propose à l'apprenant de renforcer son savoir faire à l'aide d'entraînement au moyen des TPs de type différents :

a) à aide : on fournit des indices sous forme d'aide pour orienter l'apprenant vers une solution donnée.
b) A trou : le programme solution est fournit mais avec des vides à remplir par l'apprenant.
c) A séquence: le programme solution est fournit mais dans une fausse séquence, l'apprenant doit séquencer les instructions du programme.

Il faut remarquer que dans ce cas l'apprenant est orienté vers une solution particulière. Donc notre environnement applique un type particulier d'apprentissage c'est *l'apprentissage par exemple*

7.2.2 l'apprenant pourra écrire son propre programme et le tester

L'autoévaluation permet à l'apprenant de connaître le niveau d'acquisition des connaissances. C'est une opération qui s'avère compliquée dans le cas de la programmation puisqu'un problème peut être résolu de différentes manières.

Une première solution est d'intégrer une sorte de ***générateur de digramme de classe*** dans l'environnement. Le rôle de ce dernier est de fournir pour chaque programme-solution une seule représentation graphique (pour chaque solution on associe une représentation graphique "diagramme de classe").

8 Scénario d'évaluation

Le scénario de base que nous avons adopté est représenté dans la figure (Fig.2).
L'apprenant lit les énoncés pour chaque Tp, étape dont on ne peut se passer, s'entraîne volontairement avec des exemples. L'objectif des exemples est d 'aider l'apprenant à construire ses propres programmes, et de prendre assez de recul pour compléter une autre solution qui n'est pas forcément celle à la quelle il a pensé au départ. Cela lui permet de construire sa propre base de connaissances, d'atouts et d'astuces pour répondre à un problème quelconque.
Dans une deuxième étape il doit passer à écrire son propre programme. Ce dernier doit être représenté par un diagramme de classe à l'aide de générateur de diagramme intégré dans l'environnement. Cette nouvelle représentation graphique doit être comparée avec les représentations stockées dans la base de diagramme de classe. La comparaison se fait selon les algorithmes d'appariement des graphes sachant que le diagramme de classe représentant chaque programme solution est considéré comme un graphe.

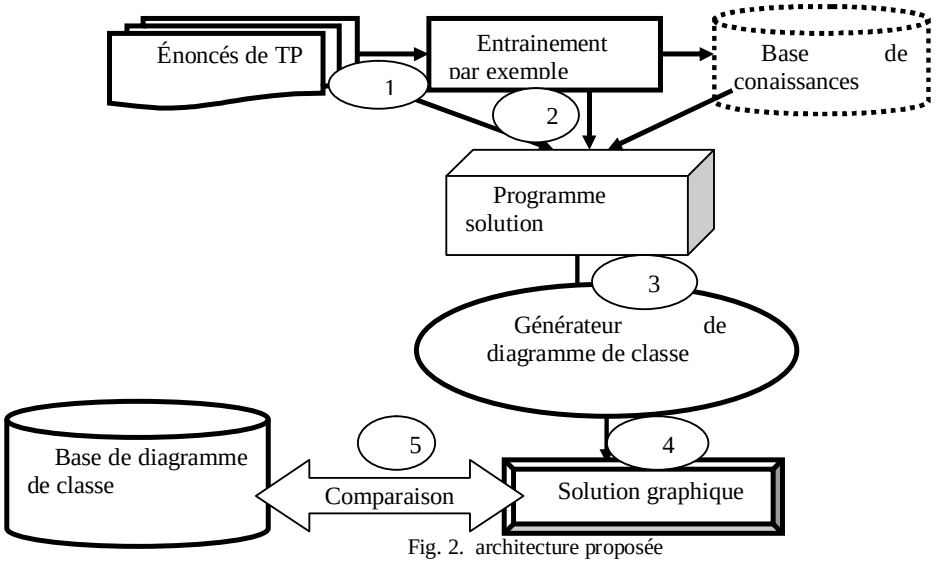


Fig. 2. architecture proposée

Le principe d'appariement de graphes est de trouver une correspondance entre les sommets d'un graphe et les sommets d'un autre graphe, tout en préservant les liens entre ces sommets.

9 Types d'appariement

Il existe plusieurs catégories de problèmes d'appariement, nous définissons ci-après les quatre problèmes les plus connus [6] :

- – l'isomorphisme de graphes, qui permet de vérifier que deux graphes sont structurellement identiques.
- l'isomorphisme de sous-graphes, qui permet de vérifier qu'un graphe est « inclus » dans un autre.
- le plus grand sous-graphe commun, qui permet d'identifier la plus grande partie commune à deux graphes [6];
- – la distance d'édition de graphes (ou Graph Edit Distance, GED), qui permet d'identifier les opérations à effectuer pour transformer un graphe en un autre à un moindre coût [6]

Les deux premières catégories correspondent à un appariement « exact », où la méthode d'appariement requiert une correspondance stricte (isomorphisme de graphes) ou une inclusion stricte (isomorphisme de sous-graphes) entre les sommets et les arcs des deux graphes.

Les deux dernières catégories correspondent à une méthode d'appariement « inexact », où la correspondance entre les deux graphes revient à chercher le « meilleur » appariement possible, même si leurs structures sont différentes, moyennant quelques transformations.

Nous nous intéressons aux mesures qui vont nous permettre de « quantifier » la similarité ou la dissimilarité entre les structures de deux graphes, c'est-à-dire les mesures de similarité qui permettent « d'évaluer » la ressemblance entre deux graphes dans le cas des appariements inexacts. Ce qui correspond aux techniques de comparaison de graphes à tolérance d'erreurs telles que la recherche du plus grand sous-graphe commun ou la distance d'édition de graphes.

9.1 Plus grand sous-graphe commun

Le principe est le suivant : g3 est un plus grand sous-graphe commun à deux graphes g1 et g2 (noté mcs(g1, g2) pour maximum common subgraph), si g3 est un sous-graphe commun à g1 et g2 ayant un nombre de sommets maximum, c'est-à-dire qu'il n'existe aucun autre sous-graphe commun à g1 et g2 avec un ordre (ou nombre de sommets) strictement plus grand que celui de g3.

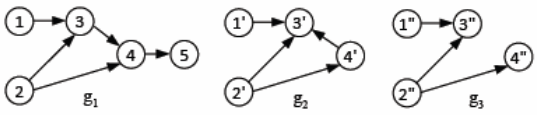


Fig. 3. Exemple : le plus grand sous graphe commun

Dans[2].Bunke propose de définir une mesure de similarité entre deux graphes basée sur la méthode du plus grand sous-graphe commun par :

$$d(g_1, g_2) = 1 - \frac{|mcs(g_1,g_2)|}{max(|g_1|,|g_2|)}$$

Fig. 4. équation de mesure de similarité

où mcs(g1, g2) est le nombre de sommets du plus grand sous-graphe commun à g1 et g2, g1 (respectivement g2) est le nombre de sommets du graphe g1 (respectivement g2)

9.2 Distance d'édition de graphes

Une alternative à la mesure de similarité construite à partir du « plus grand sous-graphe commun » est la distance d'édition de graphe. Cette méthode est très largement utilisée pour déterminer des appariements de graphes tolérants aux erreurs. Dans[5], Bunke a proposé plusieurs algorithmes de mesure de similarité basés sur la distance d'édition de graphes ou « Graph Edit Distance (GED)».
Le principe de ces algorithmes est le suivant : étant donné deux graphes g1 et g2, l'idée est d'appliquer une séquence de transformations (ou d'opérations d'édition) sur g1 (insertion, substitution et suppression de sommets ou d'arcs) pour finalement transformer g1 en g2. Ce processus tend à déformer le modèle du graphe initial jusqu'à ce qu'il s'identifie avec le modèle de l'autre graphe.

Chaque transformation ayant un coût prédéfini, le coût d'une séquence de transformations est égal à la somme pondérée des coûts de chaque transformation. La séquence de Transformation qui obtient un coût minimal représente la distance d'édition entre les deux graphes.

$$d(g_1, g_2) = \min_{t \in E(g_1,g_2)} \{ \sum_{i=1}^{k=card(t)} c(e_i) \}$$

Fig.5. Distance d'édition de graphe

« ei » est appelé une opération d'édition. « t = (e1, ...ek) » est une séquence d'opérations d'édition transformant g1 en g2.

10 Conclusion

Après avoir discuté de l'état de l'art de l'évaluation dans les EIAH, nous avons présenté dans cet article les choix opérés pour le traitement d'un exemple typique

d'évaluation de l'apprenant dans les EIAH des travaux pratiques en programmation et les réflexions que ce traitement nous inspire pour nos futurs travaux. Pour démontrer la faisabilité de cette proposition, nous voulons implémenter les différentes parties de générateur de diagramme de classe UML pour chaque programme. Cette implémentation est actuellement en cours de réalisation avec les étudiants dans un projet de fin d'étude en informatique.
Ce travail va donner lieu à une expérimentation dans le cadre d'un projet de collaboration entre les étudiants de deux universités algériennes.

References

1. Auxepaules, L. (2009). Analyse des diagrammes de l'apprenant dans un EIAH de la modélisation orientée objet. Thèse de doctorat, Université du Maine. pages 226.

2. Bunke, H. (2000). Graph matching : Theoretical foundations, algorithms, and applications. In Proc. Vision Interface 2000, Montreal, pages 82–88.
3. Conte, D., Foggia, P., Sansone, C. et Vento, M. (2004). Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence, 18(3):265–298.
4. Delorme, F. (2005). Evaluation et modélisations automatiques des connaissances des apprenants à l'aide de cartes conceptuelles. Thèse de doctorat, INSA de Rouen. pages 182.
5. Riesen, K. et Bunke, H. (2008). Approximate graph edit distance computation by means of bipartite graph matching. Elsevier, 27(7):950–959.
6. Sorlin, S. (2006). Mesurer la similarité de graphes. Thèse de doctorat, Université Claude Bernard Lyon I. pages 154.
7. Tanana, M., Delestre, N., Pécuchet, J.-P. et Bennouna, M. (2008a). Plate-forme web pour la gestion et l'evaluation de travaux pratiques en electronique numérique. In Colloque International Organisation Numérique des Universités (CIONU 2008).
8. Tanana, M., Delestre, N., Pécuchet, J.-P. et Bennouna, M. (2008b). Évaluation du savoirfaire en électronique numérique à l'aide d'un algorithme de classification. In Colloque International TICE 2008, pages 44–51.
9. Tanana, M., Delestre, N., Pécuchet, J.-P. et Bennouna, M. (2009). Génération d'exemples pour l'évaluation de l'apprenant en électronique numérique à l'aide d'un algorithme de classification. In Conférence EIAH 2009, pages 345–352.
10. Tanana, M., Pécuchet, J.-P. et Guégot, F. (2006). Un outil pour l'évaluation automatique des apprenants. In Colloque International TICE 2006, pages 2.
11. Tchounikine, P. (2009). Précis de recherche en ingénierie des eiah. (http ://membresliglab. imag.fr/tchounikine/Precis.html, dernière viste : juillet 2009).
12. Duchâteau, C., Images pour programmer. Vol. 1. 2000.
13. Dyke, G., Lund, K., and Girardot, J.-J. 2009. Tatiana : an environment to support the CSCL analysis process. CSCL 2009. Rhodes, Greece, 58–67.
14. Fenwick, J. B., Norris, C., Barry, F. E., Rountree, J., Spicer, C. J., and Cheek, S. D. 2009. Another look at the behaviors of novice programmers. SIGCSE '09. ACM, New York, NY, 296–300.
15. Rodrigo, M. T., Baker, R. S., Jadud, M. C., Amarra, A. M., Dy, T., et al. 2009. Affective and behavioral predictors of novice programmer achievement. ITiCSE '09. Paris, France. 156-160.

16. Guibert, N., Guittet, L., & Girard, P. (2005). Initiation à la Programmation « par l'exemple » : concepts, environnement, et étude d'utilité. Acte de colloque EIAH'05, Montpellier : 25-27 Mai, 461-466

17. Boussaha ,K, Bensebaa .T. E-TéléTPC@AAlP: environnement pour l'apprentissage collaboratif des langages de programmation. Colloque sur l'Optimisation et les Systèmes d'Information COSI'2009, 25-27 Mai 2009, Annaba, Algérie

18. Boussaha ,K, Bensebaa ,T.Une nouvelle plateforme pour L'apprentissage de travaux pratiques à distance (TéléTPs). ICAI09 - International Conference On Applied Informatics,15-17 novembre 2009, centre universitaire El Bachir EL IBrahimi de Bordj bou arréridj.

19. Boussaha ,K, Bensebaa ,T. Environnements informatiques favorisant les travaux pratiques à distance: réalités et perspectives.2 émé JSS'08, 24 avril 2008,université de guelma

20. Boussaha ,K, Bensebaa ,T.Design of an environment of Remote practice works between realities and prospects international conference of novel digital technology 31 July republic check 2009.

21. Boussaha ,K, Bensebaa ,T.Conception d'un environnement de travaux pratiques collaboratifs à distance: application à l'apprentissage des langages de programmation,JCI'08,journée des jeunes chercheurs en informatique 20 mai 2008 , université de guelma

The use of Web resources for creating educational and adaptive content to learners in an e-Learning platform

Mohammed Chaoui[1,2], Mohamed Tayeb Laskri[1,3]

[1] GRIA/LRI, Department of Computer Science, Badji Mokhtar University,
BP 12, Sidi Amar Annaba 23000, Algeria
[2] chaoui.mohamed@yahoo.fr
[3] laskri@univ-annaba.org

Abstract. The evolution of Web technologies has been a popularity of online learning systems. The adaptation of content in an e-Learning platform presents today a domain making the appearance of many recent projects. This paper presents a study which attempts to explore a new Web architecture to create adaptive educational content to learners in an open source LMS (Open Elms LMS). And with a novelty in creation of courses made by our work; is the use of Web resources and not courses created previously by drafters; a direct adaptation of Web content to the profiles of learners has emerged. This architecture will reduce time and effort carried out by the drafters of such e-Learning platform for the creation of courses, and Web content adaptation will greatly enrich the quality of online training courses.

Keywords: Web Content, Adaptation of Web Content, Platform for Education, e-Learning, Educational Content, LMS, Learners Profiles.

1 Introduction

Online training content are stored in a 'BD' database. It fed or changed by the editors, and not the pages themselves. This is called to websites and dynamic content: stored information is then presented in some form by "LMS" Learning Management System [1].

The presentation of the contents stored in base is defined in priori by a model, a template, often called "style sheet". She is responsible for the layout: how to extract information from the BD, which displays information, where and under what conditions [2].

With this separation of content-presentation, content can be adapted to all type of media from the same source. With the XML description language “Language Semantic Description of Documents” in particular [3], [4], one document can be adapted in various formats.

1.1 Problematic

The web presents a very broad area of information requiring a good search and precise filtering to extract the most relevant information. We are facing a very large mass of information available on the WEB, and editors spend an indefinite time to create courses and more specifically, having a content database that will be adapted to the learners profiles. The base consists with the use of the publishers provided by the LMS.

Our first step was the creation of CADEL-WEB "Automatic Construction of an On-Line Learning domain with the Web", "Construction Automatique d'un domaine d'enseignement en ligne à partir du Web" [5], which introduced a system of automatic construction of courses via the Web. Then, we have proposed a new method of searching the Web for Web content customization to a community of Internet users especially educational [6].

In the continuation of our work, we presented throughout this paper a new Web architecture for search and automatic filtering of Web resources, and eventually an adaptation of the latter to profiles of learners.

1.2 Context

E-Learning is a means of education that integrates personal motivation, communication, efficiency and technology [7]. Learn remotely, in any place and in any time to allow free, fast and the most important training custom. This is provided by the Internet and multimedia technologies that have a positive influence on the efficient use of online learning environments [8].

Although the removal of the limitations of time and space Web [6] which provides a means of finding the most relevant resources to create an educational support in an e-Learning platform to allow a good management of Web resources.

1.3 Motivation

To create a practical learning environment for e-users, and to a broad audience (different objectives, knowledge levels, funds or learning abilities), it is necessarily that the designers of e-learning systems thinking on adaptive learning environments and flexible with this potential need, so they must improve the performance to the learners.

Recent works dealing with the problem of adaptation are a very powerful difficulty, because a profile such a learner can change a lot of time in period for learning.

And before the learners needs to cultivate, to deepen more on such a field or theme of learning, we are obliged to produce a system that uses the Web as a documentary medium, and provides techniques to custom navigation for learners.

1.4 Objectives

Through our experiences on the Web domain, we found a solution [6] very effective for personalized search of Web content; this solution helps a user to find its educational needs with a free guidance, proposing multiple choices according to his mentioned query. However, we are obliged to use the principle of CADEL-WEB system with this solution to a creation of the automatic and custom profiles of learners.

We quote our secondary objectives to deal with the ultimate goal:

- The creation of a Web architecture playing two roles: research of Web resources and adaptation of these latest to profiles of learners.
- The creation of domain ontology for the presentation of the contents of courses or training on-line followed.
- Integration of a method of searching and filtering in the same architecture.
- Integration of a method for adaptation of content in the same architecture.
- The implementation of the various steps listed above on a single system, then the integration of the latter in the Open Elms LMS Version 6.x (open source) [9].

1.5 Structure of paper

The rest of the paper is presented as follows:

Part two presents a background to the concepts related to our project. These definitions are necessary in order to properly understand the following of our proposed approach.

The third part provides examples of related works, to follow the recent approaches used by researchers in the field, so these work analysis, gives us a point very hard to take place our proposal by reducing a few weak points of completed systems.

Fourth part explains our proposed approach, citing the resolved issues and the followed principle. Subsequently we detail our system architecture. Then, we give the main objective.

The fifth part describes preliminary results after the implementation of most components of the system and our solution for adaptation Web content.

At the end part “the sixth”, we accomplish by a conclusion and we include a few perspectives and our future work.

2 Background

We will give and define some concepts and theories necessary for the understanding of our work:

2.1 Learning

Learning is defined as a process where the knowledge is created by the transformation of experience [10]-[11]. The most common perceptions about learning include that it is a quantitative increase in the acquisition of knowledge or information, or make a good memory, or store information that can be replicated, the acquisition of gestures, skills and methods that can be stored and used according to the needs of learners. In a broader sense, it is to interpret and understand the reality in a different manner [12], [13].

2.2 E-Learning and Education

E-Learning is useful for education, businesses and all types of learners. It is affordable, saves time, and produces measurable results. E-Learning can be defined as learning by electronic means: the acquisition of knowledge and skills by using electronic technologies such as computers and tutorials on the Internet and local and extended networks of another definition of e-learning is education via the Internet, network, or a stand-alone computer. E-Learning is essentially the transfer network of skills and knowledge. E-learning refers to using electronic applications and learning processes [14].

2.3 Adaptive Hypermedia Systems (AHS)

There are the systems that use usage patterns and the concept to provide customized version information for the end-user [15].

2.4 Adaptive Hypermedia Systems for Education (AHSE)

They are the ones that create a unique learning experience for each student based on learner-base knowledge, goals, learning style and so forth [15]-[16].

2.5 Adaptation of learning content

There are two general approaches for adaptation of content for learning [17], [18], [19]. The first approach seeks to adapt learning content with special needs, and the second focuses on providing the most appropriate order of learning content to the needs of learners.

The first is called a content adaptation and the last is called adaptation at the level of links. None of these approaches was preferred over the other in the literature.

3 Related Work

Some researchers are in making extensions for learning content standards to improve the quality of the learning process. These researchers argue that current standards do not support an adaptive system so that they must be changed, for example there is works of (Lu & Hsieh and two teams of Rey-Lopez [21], and Sampson [22]).

There is another necessity as the metadata for learning content standards is somehow insufficient for some applications, other researchers make the emergence of new work trying to replace these standards with ontologies on the Semantic Web, as a basis of (Chi [23], Yang [24], Junuz [25] and teams of: Jovanovic [26], Lee [27], Verbert [28], Shih [29], Wang [30] and Zitko [31]).

Our objective aims to claim that the current adaptive systems are not stable, given differentiation in the profiles of learners and the difficulty instant before the drafters of course to make the latest update according to the needs of learners, thus types trainings based on the informational side and non-semantic, so currently there is no education system online which is adapting in the market.

We want from our system the integration of our approach into an LMS open source, in order to receive the benefits of the adaptation of content really in online teaching platforms.

4 The proposed approach

The amount of learning material on the Internet has grown rapidly in recent decades. Accordingly, the information consumers are confronted with the challenge of choosing the right things. In e-Learning systems such an approach has led to confusion for learners. Inevitably, the adaptive learning has gained much attention in this area (the two teams of Wang [32], and Yang [33]).

The adaptive learning can be defined as the process of generation of a unique learning experience for each learner based on personality of learner, interest and performance in order to achieve objectives such as improving university learners, satisfaction of learners from the learning process and so forth (Monova-Zheleva [34], and Rosmalen team [35]).

The proposed approach is based on the quoted points previously, where learners and even editors of course satisfaction, forms a very important point in our objectives. It is in this context that we pulled our problem.

In a commercial platform of online teaching, course management system offers a tool for creating courses, where manager which can be author or administrator, can upload courses already created. Other systems offer interfaces, where the editors create blocks of information that can be later creating courses according to profiles of learners.

In both cases we have the following weaknesses:

- The difficulty of creating courses in both cases (it should be a background before starting the creation by searching in the web or books. As a result: an indeterminate time spent by editors) ;

- The non-freedom of learners especially in the first case, because the authors cannot know the need, the level, the knowledge of learners to create a well suited courses ;
- The weakness of the training program, such as a lack of enrichment of courses by the recent news ;

To reduce the limitations suggested by the course authoring tools, we chose to develop a database of segments via the Web, and then adjust them using an open source LMS to have an adaptive content for learners. We can explain the approach as follows:

Entry of system: the Web.

→ Internet access
→ Web Resources
→ Research + Extraction + Filtering
→ Create database of segments supported on domain ontology
→ Adaptation of segments to learners from the base and supported on pedagogical ontology and patterns principal

Result: Educational Adaptive Web Content.

We will produce the architecture of the proposed approach, as in Fig. 1, to facilitate understanding and implementation of our system.

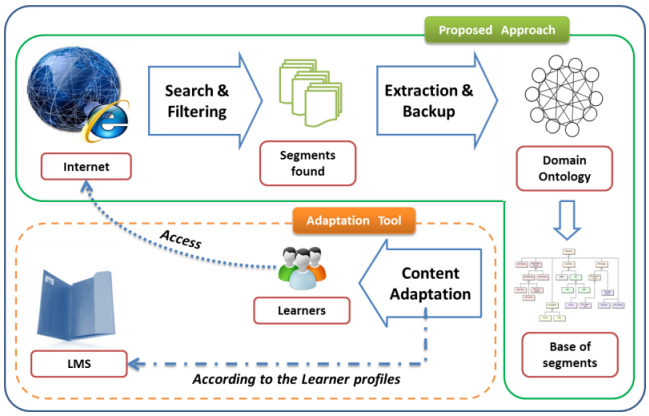


Fig. 1. Proposed architecture

Our project aims to reduce the time spent by editors to create the basis of the course in a first place, so the time spent by developers to adapt the content to learners in another place.

We can summarize the approach as follows:

1. The web presents documentary support to the creation of content courses;
2. Search tool based patterns, and more semantics extracted for ontology;
3. A filtering tool enabling extraction of different segments and stored results in the database of patterns;
4. And finally, a tool for content adaptation which will be detailed in the next part.

5 Preliminary results and solution for Adaptation

We chose a specific domain (medical domain). Our ontology is created by the tool Protégé 2000 version 3.4, and validated by experts in the field. Fig.2 presents a graphical extract of created ontology through the graphic model named Jambalaya.

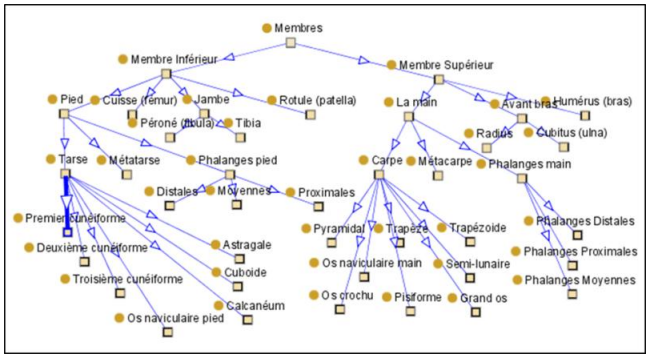


Fig. 2. A graphical extract of the created ontology

Each concept of ontology has a set of keywords and a set of semantic relationships. Keywords are required in the calculation of the degree of relevancy of each segment found in the Web, in order to choose the most relevant among all extracted segments and subsequently save these latest in a database of segments. For Web access, we use the Google API of search. This API allowed us to have a Web result like that found in the use of the search engine (title, link, textual description), as in Fig. 3.

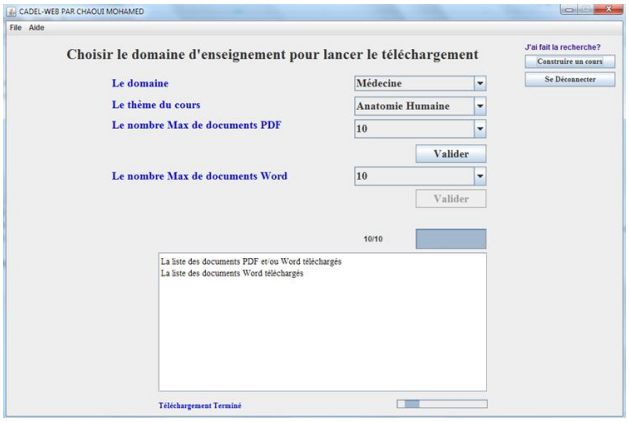


Fig. 3. Research of Web resources via our system

Fig. 3 shows an example of using our tool to search for example the documents in PDF and DOC type, then the downloading tool that will save the Web resources found in a physical medium, to the later reused.

After finding the necessary information, we turn to step of filtering and retrieval. We tested our method of filtering (1) [5] with the abstract of this paper and its keywords, and we found a degree of relevance "0.0862069" as in Fig.4.

Extraction consists of finding the highest degrees of relevance to assign each part extracted in the concept associated in the database of segments.

$$DR = (F_c + \sum_{k=0}^{n} (F_k * W_k) / N. \quad (1)$$

Where:

DR is the Degree of Relevance

F_c : present the frequency of a concept in a segment.

c : The concept.

k : The keyword.

$\sum_{k=0}^{n} (F_k * W_k)$: The sum of the frequencies of all keywords (k=0…n) of a concept in the same segment multiplied of the weight of correspondent keys.

n : The maximum number of keywords.

W_k : The weight of a keyword k.

N : The total number of words in the same segment.

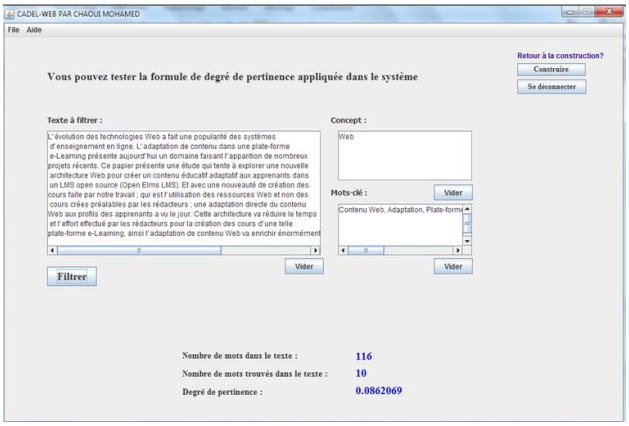


Fig. 4. A test of degree of relevance with the abstract of this paper

In the end, we can integrate our base in Open Elms LMS; which is open source; to build adaptive courses to learners. Adding on a tool responsible for adaptation of content in our system, this tool is the sequel to the use of our ontology, because in the latter, we have a hierarchy of a course. A course is a set of concepts, if we know the level of a learner and its goals; we can associate a course to each learner. The course

will be created in an automatic manner via our tool for adaptation of content and from different bricks / segments stored in the database of segments.

Profiles of learners are also presented in other pedagogical ontology, to give a direct adaptation via the current state of the learner by making a test or examination. According to the test, we can know their conditions according to the pedagogical ontology; the latter will initiate a process of construction of course using segments stored in the database.

More generally, Web resources will be adapted directly to learners during their navigation in the system. We conducted our research based patterns method [6]. Through this work, we will provide a basis for patterns, it is necessary for Web content adaptation to profiles of learners. We give part of performed adaptation by our system, as in Fig. 5.

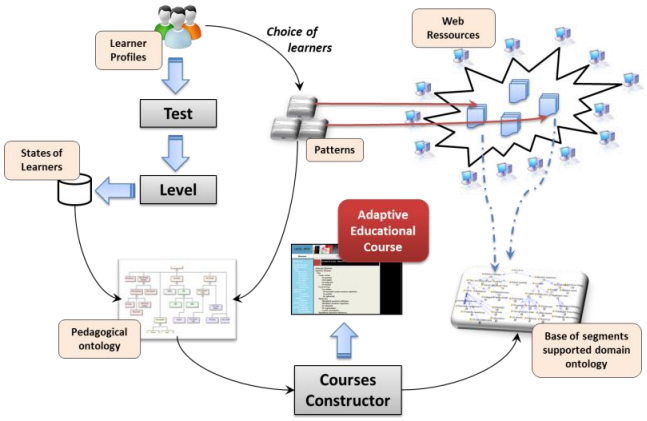


Fig. 5. Architecture of our Adaptation

Our adaptation is based on the following points:

1. The States of learners;
2. The choice of patterns in research action from the Web;
3. The backup of extracted segments into database;
4. A courses constructor tool based on level, the choice of the patterns and segments database, this tool will create educational and adaptive courses to learners.

From this principle, we can benefit from the reuse of Web resources stored in the base of the segments, in order to reduce the time for search and filtering.

In the strengths points of our system, we can see the use of patterns, learners have the opportunity to choose freely the details of the courses, this is provided by the pedagogical ontology improved and enriched by the authors.

The instant update of learners states in the associated database, allows good adaptation of segments. And in the other hand, learners can use a quite powerful tool who gives researched custom from the Web [6].

6 Conclusion & perspectives

After considering all consultation, creating tools and management of information, we have reached a conclusion that gives us the opportunity to have a new Web architecture to extract the most relevant Web resources, and then classified them in database of segments via domain ontology. The base is integrated into an Open Elms LMS with adaptation to the profiles of learners.

Through this study developed, we succeed in building Web architecture for adaptive resources to learners in an e-Learning. The architecture offers research and filtering of Web resources, after that, creating areas of learning with the possibility of adaptation of content for the platform to the profiles of learners.

The world in the last years saw very rich side resources available on the Web; our method is to reduce this informational space in an adaptive educational space, personalized and mostly reusable for the entire community of learners.

In our future work, we plan to have an experiment made by a very large number of learners (students, teachers, researchers …etc.), and in various fields (not only in the field chosen by our approach). Thus, we want to make a fusion between resources found from the Web, to improve the quality of the segments stored in the database, and the quality of courses construction.

References

1. Blanchard, E., Razaki, R., Frasson, C.: Cross-cultural adaptation of elearning contents: a methodology. In: G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005, pp. 1895--1902. Chesapeake, VA: AACE (2005)
2. JRUBESCUP, T.: Learning Content Management Systems. In: Revista Informatica Economică, No. 4 (48), pp. 91--94. INFOREC Association (2008)
3. KANOVSKY, I., OR-BACH, R., Yezreel, E.: E-Learning – Using XML technologies to meet the special characteristics of higher education. In: Journal of Systemics, Cybernetics and informatics, Vol. 2, No. 1, pp. 32--36. IIIC (2004)
4. Quang, P., Tien, T. : Rapport du stage de fin d'étude: Sujet : Projet KSCS / Webographe : Modèle de stockage, de diffusion, d'échange entre documentalistes dans un réseau de pairs. Report, ENST, Paris, France (2006)
5. Chaoui, M., Laskri, M.T.: Automatic Construction of an On-Line Learning Domain. In: IEEE ICMWI 2010, International Conference on Machine and Web Intelligence, pp. 418--422. Algiers, Algeria (2010)
6. Chaoui, M., Laskri, M.T.: New method of finding information on the Web in unstructured information resources for educational use by learners. In: International Journal of Research and Reviews in Computer Science (IJRRCS), Vol. 2, No 1, pp. 33--39. Science Academy Publisher United Kingdom (2011)
7. Phobun, P., Vicheanpanyaa, J.: Adaptive intelligent tutoring systems for e-learning systems. In: WCES-2010, World Conference on Educational Science, Istanbul, Turkey. Procedia-Social and Behavioral Sciences, Vol. 2, No. 2, pp. 4064--4069. Elsevier Ltd (2010)
8. Beldagli, B., Adiguzela, T.: Illustrating an ideal adaptive e-learning: A conceptual framework. In: WCES-2010, World Conference on Educational Science, Istanbul, Turkey. Procedia-Social and Behavioral Sciences, Vol. 2, No. 2, pp. 5755--5761. Elsevier Ltd (2010)

9. Open Source e-Learning Management System for business, http://www.openelms.org/downloads/
10. Arthurs, J.: A Juggling Act in the Classroom: Managing Different Learning Styles. In: Teaching and Learning in Nursing, Vol. 2, No. 1, pp. 2--7. Elsevier Science Inc. (2007)
11. Kolb.: Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall, Upper Saddle River (1984)
12. Ramsden, P.: Learning to Teach in Higher Education. London: Routledge (1992)
13. Smith, M.K.: Learning Theory. The Encyclopedia of Informal Education, Online: (01.01.2011) : www.infed.org/biblio/b-learn.htm, (1999)
14. MIHALCA, R., UȚĂ, A., ANDREESCU, A., ÎNTORSUREAN, I.: Knowledge Management in E-Learning Systems. In: Revista Informatica Economică, No. 2 (46), pp. 60--65. INFOREC Association (2008)
15. Ruiz, M., Diaz, M., Soler, F., Perez, J.: Adaptation in current e-learning systems. In: Computer Standards & Interfaces, Vol. 30, No. 1-2, pp. 62--70. Elsevier Science Inc. (2008)
16. Damjanovic, V., Kravcik, M. Devedzic, V.: An approach to the realization of personalized adaptation by using eQ agent system. In: UM'2005 Workshop on Personalized Adaptation on the Semantic Web (PerSWeb'05) (in conjunction with the User Modeling 2005 International Conference), Edinburg, Scotland, UK (2005)
17. Olfman, L., Mandviwalla, M.: Conceptual versus procedural software training for graphical user interfaces: A longitudinal field experiment. In Management Information Systems Quarterly, Vol. 18, No. 4, pp. 405--426. MISK (1994)
18. Papanikolaou, Mabbott, A., Bull, S., Grigoriadou, M.: Designing learner-controlled educational interactions based on learning/cognitive style and learner behavior. In: Interacting with Computers, Vol. 18, No. 3, pp. 356--384. Elsevier Science Inc. (2006)
19. Samuelis, L.: Notes on the components for intelligent tutoring systems. In: Acta Polytechnica Hungarica, Vol. 4, No. 2, pp. 77--85. (2007).
20. Lu, E., Hsieh, C.: A relation metadata extension for SCORM content aggregation model. In: Computer Standards & Interfaces, Vol. 31, No. 5, pp. 1028--1035. Elsevier Science Inc. (2008)
21. Rey-Lopez, M., Diaz-Redondo, R., Fernandez-Vilas, A., Pazos-Arias, J., Garcia-Duque, J., Gil-Solla, A., Ramos-Cabrera, M.: An extension to the ADL SCORM standard to support adaptivity: The T-learning case study. In: Computer Standards and Interfaces, Vol. 31, No. 2, pp. 309--318. Elsevier Science Inc. (2009)
22. Sampson, D., Karagiannidis, C., Cardinali, F.: An architecture for Web-based e-learning promoting re-usable adaptive educational e-content. In Educational Technology & Society, Vol. 5, No. 4, pp. 27--37. ERIC (2002)
23. Chi, Y.: Ontology-based curriculum content sequencing system with semantic rules. In Expert Systems with Applications. Vol. 36, No. 4, pp. 7838--7847. Elsevier Science Inc. (2009)
24. Yang, S.: Context aware ubiquitous learning environments for peer-to-peer collaborative learning. In Educational Technology & Society, Vol. 9, No. 1, pp. 188--201. ERIC (2006)
25. Junuz, E.: Preparation of the Learning Content for Semantic E-Learning Environment. In: World Conference on Educational Sciences, Nicosia, North Cyprus. Procedia - Social and Behavioral Sciences, Vol. 1, No. A, pp. 824--828. Elsevier Ltd (2009)
26. Jovanovic, J., Gasevic, D., Knight, C., Richards, G.: Ontologies for effective use of context in e-learning settings. In: Educational Technology & Society, Vol. 10, No. 3, pp. 47--59. ERIC (2007)
27. Lee, M., Tsai, K., Wang, T.: A practical ontology query expansion algorithm for semantic-aware learning objects retrieval. In: Computers & Education, Vol. 50, No. 4, pp. 1240--1257. Elsevier Ltd. (2008)

28. Verbert, K., Gasevic, D., Jovanovic, J., Duval, E.: Ontology-based learning content repurposing. In: Proceedings of 14th international conference on World Wide Web, pp. 1140--1141. ACM Press (2005)
29. Wang, H.-C., Hsu, C.-W.: Teaching-material design center: An ontologybased system for customizing reusable e-materials. In: Computers & Education, Vol. 46, No. 4, pp. 458--470. Elsevier Ltd. (2006)
30. Shih, W., Yang, C., Tseng, S.: Ontology-based content organization and retrieval for SCORM-compliant teaching materials in data grids. In: Future Generation Computer Systems, Vol. 25, No. 6, pp. 687--694. Elsevier B.V. (2009)
31. Zitko, B., Stankov, S., Rosic, M., Grubisic, A.: Dynamic test generation over ontology-based knowledge representation in authoring shell. In: Expert Systems with Application, Vol. 36, No. 4, pp. 8185--8196. Elsevier Ltd. (2009)
32. Wang, T., Wang, K., Huang, Y.: Using a style-based ant colony system for adaptive learning. In: Expert Systems with Applications, Vol. 34, No. 4, pp. 2449--2464. Elsevier Ltd. (2008)
33. Yang, Y., Wu, C.: An attribute-based ant colony system for adaptive learning object recommendation. In: Expert Systems with Applications, Vol. 36, No.2, pp. 3034--3047. Elsevier Ltd. (2009)
34. Monova-Zheleva, M.: Adaptive learning in Web-based educational environments. In: Cybernetics and Information Technologies, Vol. 5, No. 1, pp. 44--55. IIIC (2005)
35. Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., Koper, K.: Authoring a full life cycle model in standards-based adaptive e-learning. In: Educational Technology & Society, Vol. 9, No. 1, pp. 72--83. ERIC (2006)

Optimisation II

Numerical solution for optimal control of the Fisher equation by decomposition method

N. A. Messaoudi[1], S. Manseur[2]
Department of mathematics, Lamda Ro laboratory, Faculty of Sciences.
The University of Blida, B. P 270 Soumaa Blida, Algeria.
[1]*namessaoudi@yahoo.fr* [2]*smanseur@yahoo.fr*

Abstract

The purpose of this paper is to use a direct scheme for solving the optimal control problem governed by a parabolic nonlinear partial differential equation (PDE). The dynamics of the problem is discretized in time and the control is approximated by the piecewise constant functions. The decomposition method is applied to solve the PDE in small time intervals. The optimal control problem is reduced to a constrained optimization problem and can be solved by optimisation methods. The Fisher equation is given to illustrate the scheme efficiency.

Keywords: Optimal control, parabolic nonlinear partial differential equation, Fisher equation, Adomian decomposition method.

1 Introduction

The modelling of a phenomenon be it physical, mechanical, chemical, economic, biological, ...etc, leads to a mathematical model in form of partial differential equations (PDEs). Control theory's main purpose is to study

the possibility of acting upon the system, so as either to determine the best possible desired working way or stabilise the system by making it insensitive to some disturbances. The action variable being *control*. In pharmacokinetics, such an optimal control problem consist to determine the optimal therapy.

The optimal control problem can be solved using two numerical methods [13] : direct methods and indirect methods. Direct methods consist of discretising both state variable and control, and approximating the optimal control problem as an optimisation problem. On the other hand, indirect methods solve the boundary value problem obtained by applying the maximum principle using the shooting method.

The problem of optimal control systems governed by partial differential equations (PDE) has been extensively studied in the literature (see [4, 8, 9, 10, 11] for example).

The optimal control problem of nonlinear PDE has been solved by Mampassi and al. [11], once the problem is transformed into an optimisation problem, a fast Alienor method- based algorithm [6] is proposed to determine the global solution(s). In [15] the authors studied the fully discrete mixed finite element method for quadratic convex optimal control governed by semilinear parabolic equations. The discretization space of the state variable was done using usual mixed finite elements, whereas the time discretization was based on difference methods. The state and co-state are approximated by the lowest order Raviart-Thomas mixed finite element spaces while the control was approximated by piecewise constant elements. They have derived a priori error estimates both for the coupled state and the control approximation, but there were only few published results on this topic for nonlinear optimal control problems.

However, the purpose of this work is to use a direct method to compute the optimal control of a system governed by a nonlinear parabolic equation. We choose the controls in a finite dimensional space (piecewise constant functions), and we use the Adomian decomposition method (ADM) for solving the PDE.

The ADM ([1], [5]) has been used for solving the linear and nonlinear systems (differential, partial, algebraic, integral, ...). The solution is an analytical function given in series form explicitly dependent of the parameters of the system. This method is based on the decomposition of the nonlinear part of the system, using special polynomials called Adomian polynomials. These polynomials are calculated by recursive formulas ([1], [5]). The resolution of the system is done on small time intervals, where the control is assumed to be constant and bounded. The optimal control problem is reduced to a minimisation problem with a single variable : the control.

This article is organised as follows : the second section deals with stating the problem. Section 3 presents a numerical method for solving optimal control problem governed by a parabolic nonlinear PDE. An application of the Fisher equation is presented in section 4. The numerical experiments are investigated in section 5.

2 Problem statement

Consider the nonlinear parabolic equation [5]:

$$\frac{\partial u}{\partial t} = D\,\frac{\partial^2 u}{\partial x^2} + f\,(u,q) \ \text{ in } \Omega \ \times \ ]0,T[\tag{1}$$

$$u(x,0) = v(x) \quad , x \in \partial\Omega \tag{2}$$

$$u(x,t) = 0 \qquad , x \in \partial\Omega,\ t > 0 \tag{3}$$

where q is a control parameter.

In (1), Ω denotes a bounded region in $\mathbb{R}^n (n = 1, 2)$ with a boundary $\partial\Omega, T > 0$ is the terminal time, $u = u(x, t)$ is a function describing the state of the system, D is a constant coefficient, f is a nonlinear function. v is non-null positive function on Ω. We shall assume that the problem (1-2-3) is well posed, in other words there exists a unique non-null positive solution bounded on the interval $[0, T]$.

This equation occurs frequently in chemistry, when there is a substance that diffuses into a liquid or gas. In biology, the same equation can be used to describe the motion of particles ranging from molecules to bacteria as well as the transport of a substance into and out of a cell. The latter provides a simple model for gene selection/migration, and arisde as a model for the spread of a gene through a geographically distributed population (see [5], [8]).

As in biological processes, the aim of the optimal control problem is to find the control "$q(t)$",solution of :

$$\underset{q \in Q}{Min} \int_w \int_0^T g(u(z, s, q))ds\ dz \tag{4}$$

g is assumed a continuous differentiable function real and positive. The parameter T is known, w is a subdomain of Ω. Some constraints are imposed on control and state. More precisely, q is taken in the closed convex subset of $L^2(0, T)$ defined as :

$$Q = \{q \in L^2(0, T) : a \leqslant q(t) \leqslant b\} \tag{5}$$

where a and b are given constants so that $a < b$.

And the state u satifies the equation (2) and (3).

3 The numerical method

In our work, a direct numerical method [13] is used to solve the optimal control problem. It allows to transform the optimal control problem (1-2-3) - (4) into a nonlinear constrained optimisation problem in finite dimension. We present a numerical scheme in two stages. The first step consists of reshaping the optimal control problem as a constrained optimization problem. In the second step, the problem is solved by an appropriate method.

3.1 Transformation of the problem

The decomposition method can be used to solve linear and nonlinear functional equations of various kinds (differential, partial differential, integral, algebraic, ...) ([1], [2], [5], [12], [14]). Here, we propose to use the ADM to solve the nonlinear partial differential equation (PDE). To improve the method convergence, we use Adomian solution on time subintervals of uniform size Δt. Given a parameter N ($N > 0$), we set :

$$\Delta t = \frac{T}{N} \tag{6}$$

and we denote by $t_0 = 0, t_k = k.\Delta t$ for $k = 1, ..., N$ the time grid points.

We find out the Adomian solution on each sub-interval $[t_k, t_{k+1}]$.

The direct method consists of choosing the control in a finite dimensional space, namely the piecewise constant controls.

Assuming that $Q \subset \mathbb{R}^N([0,T])$, we can approximate $q(t)$ for $t \in [t_k, t_{k+1}]$ to a constant written as :

$$q(t) = q(t_k) = q^k, t \in [t_k, t_{k+1}] \ , \ k = 0, ..., N-1 \tag{7}$$

We assume that $q(t)$ is an admissible control satisfing the following constraint :

$$a \leqslant q(t) \leqslant b \tag{8}$$

with $a, b \in \mathbb{R}^+$and $a < b$.

Then, each q^k satisfies the constraint (8) :

$$a \leqslant q^k \leqslant b, \quad \text{for} \quad k = 0, ..., N-1 \tag{9}$$

First, by integrating equation (1) on the intervals $[t_k, t]$, we obtain :

$$u(x,t) = u(x,t_k) + \int_{t_k}^{t} D\frac{\partial^2 u(x,s)}{\partial x^2} ds + \int_{t_k}^{t} f_k(u(x,s), q^k) \ ds \tag{10}$$

where $f_k(u(s), q^k)$: is an approximation of $f(u(x,s), q(s))$ on the interval $[t_k, t_{k+1}]$ by substitute $q(t)$ by (7).The formula (10) is called a canonical form and used to obtaine the following series :

$$u^k(x,t,q^k) = \sum_{i=0}^{\infty} \Phi_i^k(x,t), \quad t \in [t_k, t_{k+1}] \tag{11}$$

u^k is Adomian solution on $[t_k, t_{k+1}]$. The Φ_i^k are the terms of Adomian solution wich is given later (see formula (14)).

The nonlinear term is decomposed as follows :

$$f_k(u, q^k) = \sum_{i=0}^{\infty} A_i^k \tag{12}$$

where the functions A_i^k are the so called Adomian polynomials ([1],[2]). For each i, A_i^k depends only on $\Phi_0^k, ..., \Phi_i^k$.

Practical formulas for computing the Adomian polynomials are given by K. Abbaoui and Y. Cherruault ([1], [2]) :

$$\begin{aligned} A_0^k &= f_k(\Phi_0^k) \\ n!A_n^k &= \frac{d^n}{d\lambda^n}\left[f_k\left(\sum_{i=0}^{n}\lambda^i.\Phi_i^k]\right)\right]_{\lambda=0}, \quad n=0.1,2,... \end{aligned} \tag{13}$$

λ is a parameter introduced for convenience.

The terms of Adomian solution are given by the following expression ([1], [5]) :

$$\begin{aligned} \Phi_0^k(x,t) &= u(x,t_k) \\ \Phi_i^k(x,t) &= \int_{t_k}^{t} D\ L_{xx}[\Phi_{i-1}^k(x,s)]\ ds + \int_{t_k}^{t} A_{i-1}^k ds \quad ,i=1,2,... \end{aligned} \tag{14}$$

where $L_{xx} = \frac{\partial^2}{\partial x^2}$.

It can be easily seen that this process leads to the solution of the state system. So, we use an approximation of the Adomian series obtained from the truncated series of p terms given as follows:

$$u^k(x,t,q^k) = \sum_{i=0}^{p-1} \nu_i^k(x,D,q^k)\ \frac{(t-t_k)^i}{i!} \quad \text{for } t\in[t_k,t_{k+1}] \tag{15}$$

where the term $\nu_i^k(x,D,q^k)$ of the Adomian series explicitly depends on x, D and q^k.

On each interval $[t_k,t_{k+1}]$ we solve the equation (1). The initial condition in t_k comes from the resolution of the previous equation on $[t_{k-1},t_k]$.

By using (15) in the objective function (4), we get the following approximation :

$$J \approx \sum_{k=0}^{N-1} G_k(q^k) \tag{16}$$

by setting :

$$G_k(q^k) = \int_w^{t_{k+1}} \int_{t_k} g(u^k(z,t,q^k))ds\ dz \quad , \text{ with } N.\Delta t = T.$$

We notice the canonical form (10) does not take into account the boundary condition (3). Obviously this equation may be regarded as a constraint of the optimisation problem.

This condition may be written using the Adomian solution (15), wich leads to the following equation :

$$\sum_{i=0}^{p-1} \nu_i^k(x, D, q^k).(t - t_k)^i = 0 \quad , \ x \in \partial\Omega, \ t \in [t_k, t_{k+1}] \tag{17}$$

wich implicates that :

$$\nu_i^k(., q^k) \equiv 0 \quad \text{on} \ \ \partial\Omega. \tag{18}$$

By integrating (18) on $\partial\Omega$, we obtain the following condition :

$$\int_{\partial\Omega} \left(\nu_i^k(., q^k)\right)^2 = 0, \ i = 0, ..p-1; \quad k = 0, ...N-1 \tag{19}$$

Which is a necessary and sufficient condition for solving (18) almost everywhere.

Finally, the optimal control problem is approximated by the following discretised problem:

$$\underset{q^k \in \mathbb{R}}{Min} \sum_{k=0}^{N-1} G_k(q^k) \tag{20}$$

subject to :

$$\begin{cases} \Gamma_i^k(q^k) = 0 \qquad , i = 0, ..p-1; \quad k = 0, ...N-1 \\ a \leqslant q^k \leqslant b, \qquad\qquad\qquad k = 0, ..., N-1 \\ \text{where } \Gamma_i^k(q^k) = \displaystyle\int_{\partial\Omega} \left(\nu_i^k(., q^k)\right)^2 \end{cases} \tag{21}$$

This is a constrained optimisation problem with N unknown variables and $(N + pN)$ constrained equations . We should note that the convergence of the solution (15) is obviously satisfied since the ADM is developed over small size subintervals (see for example Edwards and al [7]). The problem (20-21) can be solved by using the Lagrangian method.

4 Application to the Fisher equation

To illustrate the numerical method developed in the previous section, we consider the unforced Fisher equation in one dimension [12, 14] :

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + qu(1-u) \ \ \text{in} \ \ ]0, l[\times \ \ ]0, T[\tag{22}$$

$$u(x, 0) = v(x), \qquad\qquad \text{for} \qquad x \in]0, l[\tag{23}$$

where $q(t)$ is an unknown control function.

This equation provides a simple model for gene selection/migration with $u = u(x,t)$ denoting the frequency of an advantageous gene and q measuring the intensity of selection (see, e.g., [8]).

It is well-known that the system state admits a single positive solution. We want to regulate the frequency of an advantageous gene u at $x_* = l$ during $[0,T]$, that is, to make it close to a desired frequency u_d. So that, u_d is a positive constant. The problem consists of finding $q(t)$, solution of :

$$\underset{q}{Min} \int_0^T (u(x_*,s) - u_d)^2 \, ds \tag{24}$$

with $0 \leq q(t) \leq 1$. Subject to (22-23).

The optimal control $q^*(t)$ satisfies the following condition :

$$0 \leq q^*(t) \leq 1 \tag{25}$$

- Using the numerical scheme described in the previous section, the $q(t)$ is taken as a constant function. i.e. $q = q^k$ on $[t_k, t_{k+1}]$ wich leads to :

$$\begin{array}{rcll} \dfrac{\partial u}{\partial t} & = & D\dfrac{\partial^2 u}{\partial x^2} + q^k \; u \; - q^k \; u^2 \;\; , & 0 < x < l, \quad k = 0, ...N-1 \\ u(x,0) & = & v(x), & 0 < x < l \end{array} \tag{26}$$

- The nonlinear term is decomposed as follows :

$$f(u) = u^2 = \sum_{i=0}^{\infty} A_i^k \tag{27}$$

where A_i^k are the Adomian polynomials given by (13).

- Applying the Adomian method to solve (26) yields :

$$\left\{ \begin{array}{l} u_0(x,t) = u_0 = u(x,t_k) \\ u_i(x,t) = \displaystyle\int_{t_k}^{t} D\frac{\partial^2 u_{i-1}(x,s)}{\partial x^2} ds + \int_{t_k}^{t} q^k \; u_{i-1}(x,s) ds \; - \; \int_{t_k}^{t} q^k \; A_{i-1}^k(u_0, ..., u_{i-1}) ds \\ \text{for } t \in [t_k, t_{k+1}] \, , i = 1, 2, ... \end{array} \right. \tag{28}$$

where $u(x, t_k = k.\Delta t) = u^k(x, t, q^{k-1})$ and u^k $(k = 1, ...N)$ is Adomian solution given as follows :

$$u^k(x, t, q^k) = \sum_{i=0}^{p-1} u_i(x, t) = \sum_{i=0}^{p-1} \nu_i^k(x, D, q^k).\frac{(t - t_k)^i}{i!} \tag{29}$$

- Thus the approximated objective function is as follows :

$$\underset{0 \leqslant q^k \leqslant 1}{Min} \int_0^T \left(u^k(x_*, s, q^k) - u_d\right)^2 ds \tag{30}$$

On each interval $[t_k, t_{k+1}]$, we have a minimization problem of one-variable function q^k (see [6]). So, we find out q^k on a compact domain, wich proves the existence of an optimal control. The uniqueness of solution results from the particular proprieties of the objective function.

5 Numerical experiments

- We decomposed the time interval $[0, 2]$ into differents sub-intervals using a uniform partition $\Delta t = 0.1, 0.25, 0.35$. We consider the problem (22)-(23) with the initial condition :

$$u(x, 0) = x.$$

We set $D = 0.765$ and $u_d = 0.5$ at point $x_* = 1$.The approximated solution is computed with the ADM.

All numerical experiments are done on Pentium IV using Maple programming software. Simulation results are given in figures 1, 2, 3 and 4.

Discussion :

We have solved the optimal control problem of the Fisher equation by direct method. The numerical solution convergence of the optimal control problem is evaluated in figure 1. The computing time increases as the sub-intervals value decreases. We notice that J value is minimal when Δt takes on 0.1 as value. In figures 2 and 3 we have plotted the optimal control and the corresponding state at point $x_* = 1$. The optimal state solution obtained by ADM is given in figure 4. We can see in figures 3 and 4 that the optimal state has reached the desired value.

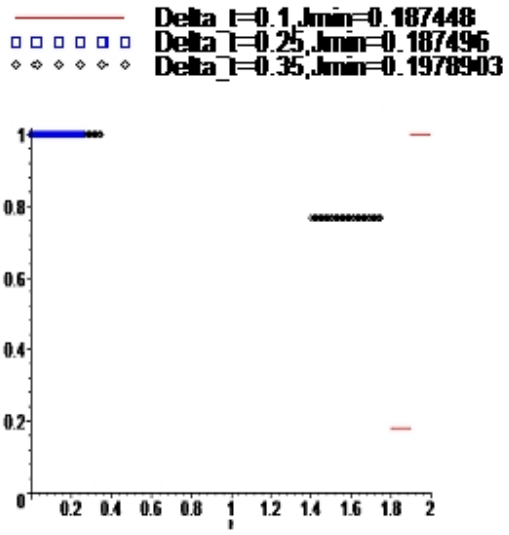


Fig.1. Optimal control for different values of the step time Δt

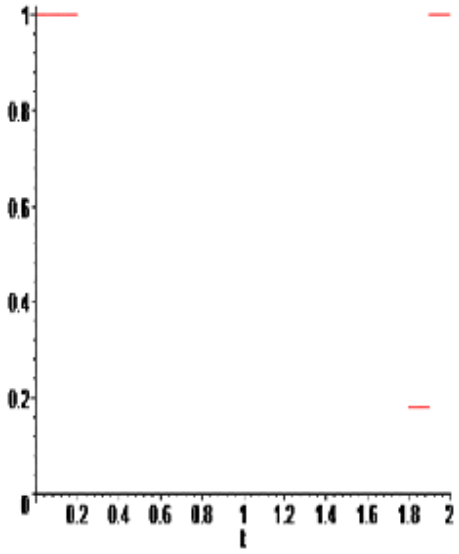


Fig. 2. Optimal control $q^*(t)$

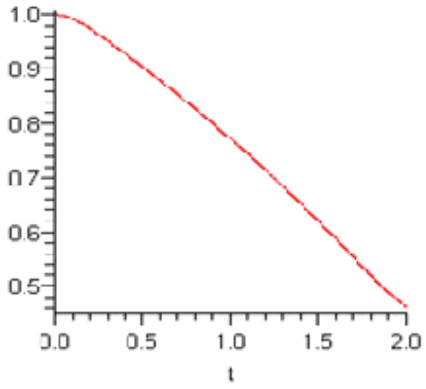


Fig.3. Optimal state solution $u(1, t, q^*)$

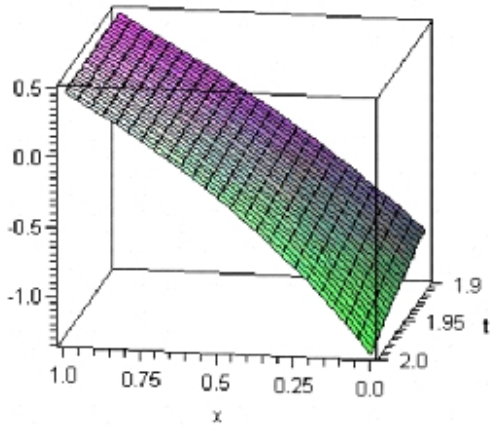


Fig. 4. Optimal state solution obtained by ADM

6 Conclusion

The ADM is an efficient tool for solving nonlinear functional equations (differential, partial differential, ...). The resolution of a parabolic nonlinear partial differential equation allows to express the solution in an explicit function of unknown parameters.

The main idea developed in this paper consists of approximating the optimal control problem of PDE to a classical optimisation problem using the Adomian decomposition

method where the control is assumed to be constant and bounded.

The finite difference and finite elements methods give numerical solutions that depend implicitly on parameters : the controls (see [3]), the thing that shows the main advantage of our method.

An application to an optimal control problem is realised for the Fisher equation and has leads to interesting results, as the optimal state reached the desired value. Our future works will be focused on solving the constrained optimization problem.

References

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

[2] Abbaoui, K., Cherruault,Y., and N'Dour, M.N. (1995), *The decomposition method applied to differential systems.* Kybernetes, 24 (8) (1995), 32-40.

[3] Bergmann, M, *Optimisation aérodynamique par réduction du modèle POD et contrôle optimal. Application au sillage laminaire d'un cylindre circulaire.* Thèse de doctorat de l'institut polytéchnique de Lorraine, Nancy, France, 2004.

[4] Bounaim, A, *Méthode de décomposition de domaine : Application à la résolution de problèmes de contrôle optimal.* thèse de l'université de Grenoble 1, France, 1999.

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

[6] Cherruault, Y. Mora,G, *Optimisation globale : théorie des courbes α-denses* . Edition Economica, 2005.

[7] Edwards, J,T. Roberts ,J,A. Ford, N,J, *A comparaison of Adomian decomposition method and Runge-Kutta methods for approximate solution of some predator prey model equation.* Numerical Analysis Report n°309, October 1997.

[8] Gunzburger, M. Yang, S,D. Zhu,W, *Analysis and discretization of an optimal control problem for the forced Fisher equation.* Discrete and continuous dynamical systems Series B. 8 (3) (2007), 569–587.

[9] Joshi, H, R, *Optimal control of the convective velocity coefficient in parabolic problem.* Elsevier, Non linear Analysis, 63 (2005), 1383-1390.

[10] Lions, J.L, *Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles.* Edition Dunod , Paris, 1968.

[11] Mampassi, B. Cherruault,Y. Konfe, B. Benneouala,T, *New challenges for computing optimal control of distributed parameter systems.* Kybernetes, 34 (7/8) (2005).

[12] Ngarhasta,N. Some, B. Abbaoui,K. Cherruault, Y, *New numerical study of Adomian method applied to a diffusion model .* Kybernetes , 31 (1) (2002), 61-75.

[13] Trélat,E, *Contrôle optimal : théorie et applications.* Edition Vuibert, 2005.

[14] Wazwaz,A, M. Gorguis,A, *An analytic study of Fisher s equation by using Adomian decomposition method.* Applied Mathematics and Computation, 154 (2004), 609-620.

[15] Yanping ,C. Zuliang, L, *Error estimates of fully discrete mixed finite element methods for semilinear quadratic parabolic optimal control problem.*Computer Methods in Applied Mechanics and Engineering, 199 (23-24) (April 2010), 1415-1423.

Une Synthèse sur le Problème de Transport à Quatre Indices avec Capacités

Aaid Djamel [1], Noui Amel [2], Hoài Lê Thi An[2], Zidna Ahmed[2],

[1] Université de Constantine, Laboratoire LMAHIS Skikda.
[2] Université de Sétif, Laboratoire LMFN Sétif.
[2] Université Paul-Verlain -Metz Laboratoire LITA Metz.
[2] Université Paul-Verlain -Metz Laboratoire LITA Metz.

Résume. Dans ce papier, nous nous intéressons à l'étude théorique d'un problème de transport à quatre indices avec capacités. Ce modèle non traité convenablement auparavant, et lié à des problèmes pratiques, entre autre les problèmes de localisation, nous donnons une présentation générale du problème à savoir : Définitions, propriétés du problème, conditions d'existence d'une solution, conditions d'optimalité. Nous exploiterons les caractéristiques spécifiques du problème à fin d'innover un algorithme de résolution.

Mots clés: Programmation Mathématiques**,** Programmation linéaire, Problème de transport**,** Problème pratique.

1 Introduction

La programmation linéaire est l'une des plus importantes techniques d'optimisation utilisées en recherche opérationnelle pour résoudre les problèmes des divers domaines (militaire, économie, gestion …), parmi ceux qui sont importants on trouve le problème de transport formulé pour la première fois par F. Hitchcock en 1941.

En 1949, Kantorovitch et Savourine ont donné une première méthode pour résoudre un tel problème dit des potentiels, indépendamment de la méthode du simplexe appliquée par G. Dantzig à la résolution de ce problème en 1951.

Plusieurs recherches concernant l'étude et la résolution d'un problème de transport à trois indices, ont été publiées au cours des années soixante.

Dans le but de généraliser le problème de transport, P. X. Ninh propose en 1979 une méthode pour résoudre un problème de transport à indices multiples sans capacités tout à fait différente de celle introduite par B. C. Verkhobski au début des années soixante-dix qui utilise la réduction d'indices.

L'étude d'un problème de transport à capacités à indices supérieur à deux est un prolongement naturel dans les recherches concernant ce problème, à ce propos nous nous somme intéressé par l'étude des caractéristiques particulières des problèmes de transport à quatre indices avec compacités.

2 Position du problème

Étant données m origines $A_1, \dots, A_m$ de disponibilités $\alpha_1, \dots, \alpha_m$, n destinations $B_1, \dots, B_n$ de demandes $\beta_1, \dots, \beta_n$. P moyens de transport choisis convenablement $S_1, \dots, S_P$ de charges réservées $\gamma_1, \dots, \gamma_p$ et q qualités d'une marchandise prises en même unités $H_1, \dots, H_q$ de quantités $\delta_1, \dots, \delta_q$.Désignant par : d_{ijkl} les capacités des chemins de transport et par c_{ijkl} le coût unitaire de transport d'une quantité x_{ijkl} de la marchandise H_l transportée de l'origine A_i vers la destination B_j à l'aide du moyen de transport S_k.

2.1 Formulation du Problème

Le problème de transport à quatre indices avec capacités qu'on note (TC4) est formulé comme suit :

$$Minimiser\ Z = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} c_{ijkl}x_{ijkl}$$

Sous les contraintes :

$$\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \alpha_i \quad pour\ i = 1, \dots, m,$$

$$\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \beta_j \quad pour\ j = 1, \dots, n,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} x_{ijkl} = \gamma_k \quad pour\ k = 1, \dots, p,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijkl} = \delta_l \quad pour\ l = 1, \dots, q,$$

$$0 \le x_{ijkl} \le d_{ijkl} \quad pour\ tout\ (i,j,k,l).$$

$$\forall (i,j,k,l)\ :\ \alpha_i > 0, \quad \beta_j > 0, \quad \gamma_k > 0, \quad \delta_l > 0, \quad d_{ijkl} > 0, \quad c_{ijkl} \ge 0$$

Cette formulation est équivalente au programme linéaire suivant :

$$(PLVB)\ \begin{cases} min\ Z = c^t x \\ Ax = b \\ 0 \le x \le d \end{cases},$$

avec:

- $x = (x_{1111}, \dots, x_{mnpq})^t \in \Re^N$,
- $c = (c_{1111}, \dots, c_{mnpq})^t \in \Re^N$,
- $d = (d_{1111}, \dots, d_{mnpq})^t \in \Re^N$,
- $b = (\alpha_1, \dots., \alpha_m, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_p, \delta_1, .., \delta_q)^t \in \Re^M$,
- $A\ est\ une\ M \times N\ matrice, M = m + n + p + q$

$$et\ \ N = mnpq.$$

3 Propriétés

3.1 La matrice des contraintes.

Dans cette représentation $x = (x_{1111}, \dots, x_{mnpq})$, on associe à chaque $(i, j, k, l) \in \{1, \dots, m\} \times \{1, \dots, n\} \times \{1, \dots, p\} \times \{1, \dots, q\}$ un vecteur $P_{ijkl} \in \Re^M$. Seulement quatre composantes du vecteur P_{ijkl} sont non nulles, elles sont situées dans les lignes $i, m + j, m + n + k\ et\ m + n + p + l$, et ayant 1 comme valeur commune, les autres éléments de A sont tous nuls.
Notons que $rang(A = M - 3)$.

3.2 Tableau de Transport

Il est utile de présenter les données du problème grâce à un tableau qu'on appelle tableau de transport.

d_{1111}	d_{1211}	.	d_{mnpq}	
c_{1111}	c_{1211}	.	c_{mnpq}	
x_{1111}	x_{1211}	.	x_{mnpq}	
1	1	.	0	α_1
. . .	. . .	.	. . .	. . .
0	0	.	1	α_m
1	0	.	0	β_1
0	1	.	0	β_2
. . .	. . .	..	. . .	. . .
0	0	..	1	β_n
1	1	..	0	γ_1
. . .	. . .	..	. . .	. . .
0	0	..	1	γ_p
1	1	..	0	δ_1
. .	. .	..	. .	. .
0	0	..	1	δ_l

3.3 Cycle

On appelle cycle, tout ensemble μ contenant un nombre $\omega > 1$ de cases (i, j, k, l) dont les vecteurs P_{ijkl} correspondants sont liés, mais toute sous famille de $\omega - 1$ éléments de cette famille de vecteurs est libre. Les vecteurs P_{ijkl} correspondant à un cycle μ vérifient la relation $\sum_{(i,j,k,l)\in\mu} \alpha_{ijkl} P_{ijkl} = 0,$

Les nombres $\alpha_{ijkl} \neq 0$ sont appelés coefficients du cycle.

- Une solution réalisable x du problème (TC4) est dite de base si les colonnes de la matrice A_x obtenue de A en gardant seulement les colonnes correspondant aux variables $0 < x_{ijkl} < d_{ijkl}$ sont linéairement indépendantes.

- Une solution réalisable de base est dite non dégénérée si

$$rang(A_x) = rang(A).$$

- Soit x une solution réalisable de base, le 4-uplet (i, j, k, l) tel que $0 < x_{ijkl} < d_{ijkl}$ est appelé case intéressante. Dans le cas contraire, (i, j, k, l) est dite non intéressante.

4 Conditions de Réalisabilité et d'optimalité

4.1 Conditions de réalisabilité

Théorème 1

1) Une condition nécessaire pour que le problème (TC4) possède une solution réalisable est que la condition

$$\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j = \sum_{k=1}^{p} \gamma_k = \sum_{l=1}^{q} \delta_l = H \ldots\ldots (A.1)$$

et les conditions suivantes …..(A.2)

$$\begin{cases} \alpha_i \leq \sum_{j=1}^{n} \sum_{k=1}^{p} \sum_{l=1}^{q} d_{ijkl} \quad pour\ tout \quad i = 1, \ldots, m \\ \beta_j \leq \sum_{i=1}^{m} \sum_{k=1}^{p} \sum_{l=1}^{q} d_{ijkl} \quad pour\ tout \quad j = 1, \ldots, n \\ \gamma_k \leq \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{q} d_{ijkl} \quad pour\ tout \quad k = 1, \ldots, p \\ \delta_l \leq \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} d_{ijkl} \quad pour\ tout \quad l = 1, \ldots, q \end{cases}$$

soient vérifiées.

2) Une condition suffisante pour que le problème (TC4) possède une solution réalisable est que la condition (A.1) et la condition suivante

$$\frac{\alpha_i \beta_j \gamma_k \delta_l}{H^3} \leq d_{ijkl} \quad \text{pour tout } (i, j, k, l)$$

soient vérifiées.

Preuve

1) On vérifie facilement que si $x = (x_{ijkjl})$ est une solution réalisable pour le problème (TC4), alors les conditions (A.1) et (A.2) sont satisfaites.

2) Soit $x = (x_{ijkl})$ un vecteur de $\Re^N$ tel que

$x_{ijkl} = \frac{\alpha_i \beta_j \gamma_k \delta_l}{H^3}$, pour tout (i, j, k, l).

On vérifie facilement que x est une solution réalisable pour le problème(TC4).

Si l'ensemble des solutions réalisables du problème est non vide alors il est un polyèdre convexe borné, et comme la fonction objectif est continue, alors le problème admet au moins une solution optimale.

4.2 Conditions d'optimalité

Théorème 2

Supposons que le problème (TC4) est réalisable, alors une solution réalisable x de ce problème est optimale si et seulement s'il existe un vecteur

$(u_1, \dots, u_m, v_1, \dots, v_n, w_1, \dots, w_k, t_1 \dots, t_l) \in \mathbb{R}^N$

tel que : $u_i + v_j + w_k + t_l \leq c_{ijkl}$ Si $x_{ijkl} = 0$

$u_i + v_j + w_k + t_l = c_{ijkl}$ Si $0 < x_{ijkl} < d_{ijkl}$

et

$u_i + v_j + w_k + t_l \geq c_{ijkl}$ Si $x_{ijkl} = d_{ijkl}$.

Preuve

Considérons la formulation suivante du problème (TC4) :

$\min_x [c^t x \, ; Ax = b, -x \geq -d]$,

Alors, son dual qu'on note (DTC4) est formulé comme suit :

$$\max_{Z \geq 0} [b^t y - d^t z : \; A^t y - z \leq c, y \in \mathbb{R}^M]$$

Soit (y, z) une solution optimale du problème (DTC4), alors une solution réalisable $x = x_{ijkl}$ du problème (TC4) est optimale si et seulement si les deux conditions de complémentarité suivantes sont vérifiées

$(A^t y - z - c)_{ijkl} x_{ijkl} = 0$,…… *(A.3)*

et

$(d - x)_{ijkl} z_{ijkl} = 0$ …… *(A.4)*

1. Si $x_{ijkl} = 0$, alors $x_{ijkl} < d_{ijkl}$ et (A.4) impliquent $z_{ijkl} = 0$. Donc $(A^t y)_{ijkl} \leq c_{ijkl}$.

2. Si $0 < xijkl < d_{ijkl}$ alors (A.4) implique $z_{ijkl} = 0$ et (A.3) implique $(A^t y - z - c)_{ijkl} = 0$. Donc $(A^t y)_{ijkl} = c_{ijkl}$.

3. Si $x_{ijkl} = d_{ijkl}$ alors $x_{ijkl} > 0$ et (A.3) impliquent
$(A^t y - z - c)_{ijkl} = 0$, Ce qui implique
$(A^t y - c)_{ijkl} = z_{ijkl}$. Donc $(A^t y)_{ijkl} \geq c_{ijkl}$. ■

5 Adaptation d'une méthode de point intérieurs à la résolution du problème

Pour résoudre le problème de transport à quatre indices avec capacités on utilise une variante de la méthode de Karmarkar dite Ye-Lustig, nous devons considérer la formulation équivalente **(PLVB)** dont la matrice des contraintes n'est pas pleins rang.

Afin de remonter cette difficulté on utilise l'élimination de Gauss avec permutation de lignes ou de colonnes, à la fin du processus on remet les colonnes dons l'ordre initial et on résout le problème résultant.

Algorithme 1:Algorithme de Ye-Lustig

Phase 1

Initialisation $x^0 = a, \lambda^0 = 1, \tilde{x} = (x^0, \lambda^0)$ et $k = 0$.

1. Si $\|Ax^0 - b\| \leq \varepsilon$. Stop, x^k est une solution initiale du $(PLVB)$ qu'on note x^0 ultérieurement.

Sinon aller à l'étape (2).

2. Si $\lambda^k \leq \varepsilon$ Stop; x^k est une solution initiale du $(PLVB)$.

Sinon aller à l'étape 3.

3. Prenons:

$D_k = diag(\tilde{x}^k), B = [A, b - Aa]$ et $r = \frac{1}{\sqrt{(n+1)(n+2)}}$

Calculer

$p_k = [I - B_k^t (B_k B_k^t)^{-1} B_k](D_k \tilde{c}, -\tilde{c}^t \tilde{x}^k)^t$

Tel que : $B_k = [BD_k, -b]$

Calculer

$y^{k+1} = \frac{e_{n+2}}{n+2} - \alpha^k r \frac{p_k}{\|p_k\|}, \tilde{x}^{k+1} = (y_{n+2}^{k+1})^{-1} D_k y^{k+1}[n+1]$.

4. Faire $k = k + 1$ et retourner à l'étape 2.

Fin.

Phase 2 : (amélioration d'une solution strictement réalisable)

Initialisation : $x^0 > 0, k = 0$

Tan que

$$\frac{\|d^k\|}{c^t x^0} > \varepsilon$$

Faire

1. Construire $D_k = diag(x^k), A_k = [AD_k, -b], B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$
2. Calcul de la projection

$$p_k = [I - B_k^t (B_k B_k^t)^{-1} B_k](D_k c, -c^t x^k)^t$$

3. Normaliser la projection p_k : $d^k = \frac{p_k}{\|p_k\|}$
4. Calculer l'itéré suivant $y^k = \frac{e_{n+1}}{n+1} - \alpha^k r d^k,\ x^{k+1} = \frac{D_k y^k[n]}{y_{n+1}^k}$

avec $r = \frac{1}{\sqrt{(n+1)}}$

5. $k = k + 1$

Fin tant que, Fin algorithme

6 Conclusion

Le problème de transport énoncé précédemment se modélise comme un programme linéaire à variable bornées. L'algorithme de Ye-Lustig permet donc d'en calculer une solution optimale.

En tenant compte des particularités du problème (TC4), on propose prochainement un algorithme qui attaque directement le problème tel qu'il est sans reformulation et sans augmentation dans la taille du problème en question.

Références

1. Cenk Calıs_kan, A specialized network simplex algorithm for the constrained maximum flow problem, European Journal of Operational Research 210 (2011) 137–147

2. D.AAID, Étude numérique comparative entre des méthodes de résolution d'un problème de transport à quatre indices avec capacités mémoire de magister Soutenu publiquement en 14/02/2010 à l'université de Constantine.

3. M. S. Bazaraa, J. J. Javis and H. D. Sherali, Linear programming and network flows, John Wiley & Sons, 1990.

4. Rafael A. Melo a, Laurence A. Wolsey, Optimizing production and transportation in a commit-to-delivery business mode, European Journal of Operational Research 203 (2010) 614–61

Méthode de Résolution d'un Problème de Contrôle Optimal avec une Application Financière

Mohand-Ouamer Bibi et Mourad Azi

Laboratoire LAMOS, Université de Béjaia, 06000 Béjaia, Algérie.

Résumé Dans cet article, nous développons une méthode adaptée de résolution d'un problème de contrôle optimal d'un système dynamique linéaire sous forme de Bolza, avec des contraintes inégalités et une commande vectorielle. Cette méthode est élaborée en se basant sur la méthode développée par R.Gabasov et F.M.Kirillova. Sa particularité réside dans le fait qu'elle évite toute transformation préliminaire du problème, et elle possède un critère de suboptimalité qui permet d'arrêter l'algorithme avec une précision désirée.
Les problèmes de contrôle optimal sous forme de Bolza ont des applications importantes en économie financière. Pour cela, nous appliquons notre méthode sur un modèle de financement optimal d'entreprise, afin d'analyser et d'interpréter l'évolution de ce modèle.

Mots clés :Méthode adaptée, Commande vectorielle, Problème de Bolza, Financement optimal.

1 Introduction

Un système de contrôle est un système dynamique dépendant d'un paramètre appelé contrôle (commande), avec lequel on peut amener le système d'un état initial donné à un certain état final, en optimisant éventuellement certains critères. Après l'apparition du principe du maximum de Pontriaguine [11] dans les années 50, plusieurs théoriciens se sont intéressés à la résolution des problèmes de contrôle optimal, afin d'automatiser les systèmes de gestion, de s'affranchir des tâches pénibles, de prédire et de contrôler les événements futurs, et enfin d'optimiser certains critères. Le principe du maximum a révolutionné la théorie moderne du contrôle optimal et il a ouvert un vaste champ de recherche dans cette discipline.

Dans cet article, nous développons une nouvelle méthode de résolution d'un problème de contrôle optimal sous forme de Bolza avec une commande multivariable et des contraintes sous forme d'inégalités, et ce, en se basant sur la méthode développée par R. Gabassov et F.M.Kirillova dans [1], et sur les travaux de M.O.Bibi [4,6]. Après avoir construit le support et l'accroissement de la fonctionnelle, nous donnons les critères d'optimalité et de suboptimalité, qui permettent d'élaborer un algorithme de résolution du problème.

Le modèle de contrôle optimal considéré ici a des applications importantes en économie financière. C'est pour cela que nous exposons à la fin de ce travail l'application numérique de l'algorithme élaboré, implémenté sous Matlab, pour un modèle de financement optimal d'entreprise.

2 Position du problème et définitions

Le problème de contrôle optimal étudié dans ce travail est le suivant :

$$J(u) = c_1' x(t^*) + \int_0^{t^*} c_2'(t)u(t)dt \qquad \longrightarrow \max, \tag{1}$$

$$\dot{x} = Ax + Bu + r, \qquad x(0) = x_0, \tag{2}$$

$$g_* \leq Hx(t^*) \leq g^*, \quad d^- \leq u(t) \leq d^+, \quad t \in T = [0, t^*], \tag{3}$$

où $x(t) \in \mathbb{R}^n$ est un vecteur qui représente l'état du système à l'instant t et x_0 la position initiale du système ; $u(t) \in \mathbb{R}^r$ est la commande agissant sur le système à l'instant t (signal d'entrée), avec $d^- = (d_1^-, d_2^-, \cdots, d_r^-)$, $d^+ = (d_1^+, d_2^+, \cdots, d_r^+)$; A est une matrice carrée d'ordre n, caractérisant le système (pour plus de simplicité on la suppose constante, c'est-à-dire ne dépendant pas de la variable t), de la même manière $B = B(K, J)$ et $H = H(I, K)$ sont des matrices réelles d'ordre $n \times r$ et $m \times n$ respectivement ; $g_* = g_*(I)$ et $g^* = g^*(I)$ sont des m-vecteurs, où $K = \{1, ..., n\}$, $I = \{1, ..., m\}$, $J = \{1, ..., r\}$ sont des ensembles d'indices ; c_1 et $c_2(t)$ sont deux vecteurs de coûts, de dimension n et r respectivement.

La solution du système dynamique (2) est donnée par la formule de Cauchy :

$$x(t) = F(t)\left[x_0 + \int_0^t F^{-1}(\tau)(Bu(\tau) + r(\tau))d\tau\right], \quad t \in T, \tag{4}$$

où $F(t) = \exp(At)$, est une matrice carrée d'ordre n, solution du système différentiel homogène :

$$\dot{F}(t) = AF(t), F(0) = I_n, \tag{5}$$

I_n étant une matrice identité d'ordre n.
En remplaçant cette solution dans le problème (1)-(3), celui-ci devient un problème de la seule variable $u(t)$ suivant :

$$\begin{cases} J(u) = c_1' F(t^*)x_0 + \int_0^{t^*} c'(t)u(t)dt + \int_0^{t^*} c_3'(t)r(t)dt \longrightarrow \max, \\ \overline{g}_* \leq \int_0^{t^*} \varphi(t)u(t)dt \leq \overline{g}^*, \\ d^- \leq u(t) \leq d^+, \quad t \in T = [0, t^*], \end{cases} \tag{6}$$

avec $c'(t) = c_1' F(t^*)F^{-1}(t)B + c_2'(t)$, $c_3'(t) = c_1' F(t^*)F^{-1}(t)$, $\varphi(t) = HF(t^*)F^{-1}(t)B$, $\overline{g}_* = g_* - HF(t^*)x_0 - \int_0^{t^*} HF(t^*)F^{-1}(t)r(t)dt$, $\overline{g}^* = g^* - HF(t^*)x_0 - \int_0^{t^*} HF(t^*)F^{-1}(t)r(t)dt$.

Choisissons dans l'ensemble I un sous ensemble $I_s \subset I$, avec $|I_s| = p \leq m$. Sur l'intervalle T, choisissons un ensemble de moments isolés $T_s = \{t_k, k \in K_s\}$, $K_s = \{1, \ldots, k_s\}$, $k_s \leq p$. A chaque moment $t_k \in T_s$ faisons correspondre un ensemble d'indices $J_k \subset J$, $\sum_{k \in K_s} |J_k| = p$.
Posons $J_s = \{J_k, k \in K_s\}$ et $Q_s = \{I_s, J_s, T_s\}$.
Construisons la matrice :

$$\varphi_s = \varphi(Q_s) = (\varphi_{ij}(t_k),\ i \in I_s,\quad j \in J_k,\ k \in K_s). \tag{7}$$

Définition 1. La commande constante par morceaux $u(t), t \in T$, est dite admissible si elle satisfait les contraintes (2), (3).

Définition 2. La commande admissible $u^0(t), t \in T$, est dite optimale si

$$J(u^0) = \max J(u). \tag{8}$$

La trajectoire correspondante $x^0(t)$ est dite trajectoire optimale. En outre, on appelle commande suboptimale (ou ϵ-optimale) toute commande admissible $u^\varepsilon(t), t \in T$, satisfaisant l'inégalité :

$$J(u^0) - J(u^\varepsilon) \leq \varepsilon, \tag{9}$$

où $\varepsilon \geq 0$ est un nombre donné et u^0 est une commande optimale.

Définition 3. L'ensemble $Q_s = \{I_s, J_s, T_s\}$ est appelé support du problème (1)-(3) si la matrice φ_s est inversible.

Définition 4. La paire $\{u, Q_s\}$ formée de la commande admissible u et du support Q_s est appelée commande de support.

Définition 5. La commande de support $\{u, Q_s\}$ est dite non dégénérée si :

1. Pour tout moment t_k de T_s et pour tout indice $i \in I_k$, $k \in K_s$, l'une des deux conditions est vérifiée :
   – dans le voisinage de t_k , la composante $u_i(t)$, $t \in T$, est non critique :
   $$d_i^- < u_i(t) < d_i^+,\quad t \in [t_k - \delta,\ t_k + \delta],\quad \delta > 0,$$
   – t_k est un point de discontinuité de la fonction $u_i(t)$, $t \in T$;
2. En outre, la contrainte suivante est vérifiée :
$$g_*(I_H) < H(I_H, K)x(t^*) < g^*(I_H),\quad I_H = I \backslash I_s. \tag{10}$$

3 Formule de l'accroissement de la fonctionnelle

Soit $\{u, Q_s\}$ une commande de support du problème (1)-(3). Considérons une autre commande admissible $\overline{u}(t) = u(t) + \Delta u(t)$ et sa trajectoire correspondante
$\overline{x}(t) = x(t) + \Delta x(t)$, $t \in T$.
L'accroissement de la fonctionnelle s'écrit alors :

$$\begin{aligned}
\Delta J(u) &= J(\overline{u}) - J(u) \\
&= c_1' F(t^*)x_0 + \int_0^{t^*} (c'(t)\overline{u}(t) + c_3'(t)r(t))dt - c_1' F(t^*)x_0 - \int_0^{t^*} (c'(t)u(t) + c_3'(t)r(t))dt \\
&= \int_0^{t^*} c'(t)\overline{u}(t)dt + \int_0^{t^*} c_3'(t)r(t)dt - \int_0^{t^*} c'(t)u(t)dt - \int_0^{t^*} c_3'(t)r(t)dt \\
&= \int_0^{t^*} c'(t)\overline{u}(t)dt - \int_0^{t^*} c'(t)u(t)dt = \int_0^{t^*} c'(t)(\overline{u}(t) - u(t))dt \int_0^{t^*} c'(t)\Delta u(t)dt.
\end{aligned}$$

Définissons le vecteur :

$$c_s = (c_j(t_k), j \in J_k, k \in K_s),$$

où $c_j(t)$ est le $j^{ème}$ élément du vecteur

$$c'(t) = (c_1(t), \ldots, c_r(t)),\ t \in T.$$

Construisons le vecteur des potentiels :

$$y'(I_s) = c'_s \varphi_s^{-1}, \qquad y(I_H) = 0, \tag{11}$$

et la co-commande $E'(t) = (E_1(t), \ldots, E_r(t)),\ t \in T$:

$$\begin{aligned} E'(t) &= y'\varphi(t) - c'(t) \\ &= y'HF(t^*)F^{-1}(t)B - (c'_1F(t^*)F^{-1}(t)B + c'_2(t)) \\ &= (y'H - c'_1)F(t^*)F^{-1}(t)B - c'_2(t). \end{aligned} \tag{12}$$

En introduisant la fonction $\psi(t)$ défini comme suit :

$$\psi'(t) = -(H'y - c_1)'F(t^*)F^{-1}(t),\ t \in T,$$

qui est la solution du système conjugué :

$$\dot{\psi} = -A'\psi, \quad \psi(t^*) = c_1 - H'y, \tag{13}$$

alors, la co-commande peut s'écrire sous la forme :

$$E'(t) = -\psi'(t)B - c'_2(t),\ t \in T. \tag{14}$$

En vertu des définitions (11) et (14), l'accroissement de la fonctionnelle prend la forme suivante :

$$\begin{aligned} \Delta J(u) &= J(\overline{u}) - J(u) \\ &= \int_0^{t^*} c'(t)\Delta u(t)dt \\ &= \int_0^{t^*} [y'\varphi(t) - E'(t)]\Delta u(t)dt \\ &= y' \int_0^{t^*} \varphi(t)\Delta u(t)dt - \int_0^{t^*} E'(t)\Delta u(t)dt \\ &= y'H\Delta x(t^*) - \int_0^{t^*} E'(t)\Delta u(t)dt. \end{aligned} \tag{15}$$

En posant $\upsilon = H\Delta x(t^*)$, l'accroissement de la fonctionnelle devient :

$$\Delta J(u) = \sum_{i \in I_a} y_i \upsilon_i - \int_0^{t^*} E'(t)\Delta u(t)dt. \tag{16}$$

Par conséquent, il est clair que le maximum de l'accroissement de la fonctionnelle $\Delta J(u)$ sous les contraintes :

$$\begin{cases} g_{i*} - H(i,k)x(t^*) \le \upsilon_i \le g_i^* - H(i,k)x(t^*), & i \in I_s, \\ d^- - u(t) \le \Delta u(t) \le d^+ - u(t), & t \in T, \end{cases} \tag{17}$$

est égal à :

$$\beta(u,Q_s) = \sum_{j=1}^{r}\left[\int_{T^+(j)} E_j(t)(u_j(t)-d_j^-)dt + \int_{T^-(j)} E_j(t)(u_j(t)-d_j^+)dt\right] + \sum_{y_i<0, i\in I_s} y_i\upsilon_i^- + \sum_{y_i>0, i\in I_s} y_i\upsilon_i^+, \tag{18}$$

avec $T^+(j)=\{t\in T, E_j(t)>0\}$, $T^-(j)=\{t\in T, E_j(t)<0\}$, $j\in J$,
et $\upsilon^-(I)=(\upsilon_i^-, i\in I)=g_*-Hx(t^*)$, $\upsilon^+(I)=(\upsilon_i^+, i\in I)=g^*-Hx(t^*)$.
Le nombre $\beta(u,Q_s)$ est appelé estimation de suboptimalité. Ainsi, nous obtenons une majoration de l'accroissement de la fonctionnelle :

$$\Delta J(u) = J(\overline{u}) - J(u) \le \beta(u,Q_s). \tag{19}$$

4 Critère d'optimalité et de suboptimalité

Nous avons les théorèmes suivants :

Théorème 1. ***(Critère d'optimalité).***
Soit (u,Q_s) une commande de support du problème (1)-(3). *Les relations suivantes :*

$$\begin{cases} E_j(t)\ge 0, \ si \ \ u_j(t)=d_j^-, \\ E_j(t)\le 0, \ si \ \ u_j(t)=d_j^+, \\ E_j(t)=0, \ si \ \ d_j^-<u_j(t)<d_j^+, \qquad t\in T, \ \ j\in J; \\ y_i\ge 0, \quad si \ \ H(i,K)x(t^*)=g_i^*, \\ y_i\le 0, \quad si \ \ H(i,K)x(t^*)=g_{*i}, \\ y_i=0, \quad si \ \ g_{*i}<H(i,K)x(t^*)<g_i^*, \ i\in I_s, \end{cases} \tag{20}$$

sont suffisantes et dans le cas de la non dégénérescence, elles sont aussi nécessaires pour l'optimalité de la commande de support $\{u,Q_s\}$.

Théorème 2. ***(Critère suboptimalité).***
Soit $\varepsilon>0$. Pour que la commande admissible $u(t)$, $t\in T$, soit ε-optimale, il est nécessaire et suffisant qu'il existe un support Q_s de telle sorte que le long des trajectoires $x(t)$,$\psi(t)$, $t\in T$, on ait les conditions suivantes vérifiées :

$$H(t,x(t),\psi(t),u(t)) = \max_{d^-\le v\le d^+} H(t,x(t),\psi(t),v) - \varepsilon(t), \quad t\in T, \tag{21a}$$

$$y'Hx(t^*) = \max_{g_*\le Z\le g^*} y'Z - \varepsilon_1, \tag{21b}$$

avec $\int_0^{t^*}\varepsilon(t)dt+\varepsilon_1+\le\varepsilon$; ε_1 et $\varepsilon(t)$ sont deux quantités positives.

Remarque 1. La démonstration des théorèmes 1 et 2 peut se faire de la même manière que dans [1].

5 Algorithme de la méthode

Soient $\varepsilon>0$ et $\{u,Q_s\}$ une commande de support initiale. Le but de l'algorithme est de construire une commande suboptimale u^ε ou carrément optimale u^0, en faisant des itérations qui consistent à faire le passage de $\{u,Q_s\}$ à $\{\overline{u},\overline{Q}_s\}$ tel que $J(\overline{u})\ge(u)$. Pour cela, l'algorithme se décompose en trois procédures :

- changement de commande $u \longrightarrow \overline{u}$;
- changement de support $Q_s \longrightarrow \overline{Q}_s$;
- procédure finale.

5.1 Changement de commande

Soient $\epsilon \geq 0$ donné et une commande de support $\{u, Q_s\}$ vérifiant $\beta(u, Q_s) > \epsilon$. Construisons une autre commande admissible $\overline{u}(t) = u(t) + \theta \Delta u(t)$, $t \in T$, de telle façon à avoir $J(\overline{u}) \geq J(u)$, où $\Delta u(t)$ est la direction du changement de la commande, et $\theta \geq 0$ est le pas maximal admissible le long de cette direction. Pour cela, choisissons les nombres $\alpha > 0$, $h > 0$ (paramètres de l'algorithme) et construisons les ensembles :

$$T_\alpha = \{t \in T : \ \eta(t) \leq \alpha\}, \qquad T_* = T \backslash T_\alpha, \qquad avec\ \eta(t) = \min_{j \in J} |E_j(t)|, \ t \in T.$$

Subdivisons l'ensemble T_α en intervalles $[\tau_k, \tau^k]$, $k = \overline{1, N}$, $\tau_k < \tau^k \leq \tau_{k+1}$, $T_\alpha = \bigcup_{k=1}^{N} [\tau_k, \tau^k]$, de telle façon que nous ayons $\tau^k - \tau_k \leq h$; $T_s \subset \{\tau_k, k = \overline{1, N}\}$; $u_j(t) = u_{jk} = const$, $t \in [\tau_k, \tau^k]$, $k = \overline{1, N}$, $j \in J$.
Calculons les quantités suivantes :

$$\beta_{jk} = -\int_{\tau_k}^{\tau^k} E_j(t)dt, \qquad q_{jk} = \int_{\tau_k}^{\tau^k} \varphi_j(t)dt, \quad k = \overline{1, N}, \quad j \in J; \tag{22}$$

$$\beta_{N+1} = -\sum_{j=1}^{r} \int_{T_*} E_j(t) \Delta u_j(t)dt + \sum_{i \in I_s} y_i \overline{v}_i; \tag{23}$$

$$q_{i(N+1)} = \sum_{j=1}^{r} \int_{T_*} \varphi_{ij}(t) \Delta u_j(t)dt - \overline{v}_i, i \in I_s, \tag{24}$$

$$q_{i(N+1)} = \sum_{j=1}^{r} \int_{T_*} \varphi_{ij}(t) \Delta u_j(t)dt, \ i \in I_H; \tag{25}$$

avec

$$\overline{v}_i = \begin{cases} g_i^* - H(i, K)x(t^*), \ si \ \ y_i < 0, \\ g_{*i} - H(i, K)x(t^*), \ si \ \ y_i > 0, \end{cases} \tag{26}$$

et

$$\Delta u_j(t) = \begin{cases} d_j^+ - u_j(t), \ si \ \ E_j(t) < -\alpha, \\ d_j^- - u_j(t), \ si \ \ E_j(t) > \alpha, \quad t \in T_*, \quad j = \overline{1, r}. \end{cases} \tag{27}$$

Posons :

$$\begin{cases} l_{jk} = \theta \Delta u_j(t), t \in [\tau_k, \tau^k], j = \overline{1, r}, k = \overline{1, N}, \\ l_{N+1} = \theta, \qquad avec \ \ 0 \leq \theta \leq 1; \end{cases} \tag{28}$$

$f_*(I_H) = g_*(I_H) - H(I_H, K)x(t^*), f^*(I_H) = g^*(I_H) - H(I_H, K)x(t^*), f_*(I_s) = f^*(I_s) = 0;$

$$l = (l_{11}, \ldots, l_{1N}, \ldots, l_{r1}, \ldots, l_{rN}, l_{N+1})'; \tag{29}$$

$$\beta = (\beta_{11}, \ldots, \beta_{1N}, \ldots, \beta_{r1}, \ldots, \beta_{rN}, \beta_{N+1})'. \tag{30}$$

Les vecteurs l, β, et q_{jk} ont pour dimensions respectives $(Nr + 1)$ et m. En utilisant ces quantités, le problème (16)-(17) sera équivalent au problème de support suivant :

$$\beta' l \qquad \longrightarrow \max, \tag{31a}$$

$$f_* \leq \sum_{j=1}^{r} \sum_{k=1}^{N} q_{jk} l_{jk} + q_{N+1} l_{N+1} \leq f^*, \tag{31b}$$

$$d_j^- - u_{jk} \leq l_{jk} \leq d_j^+ - u_{jk}, \quad j = \overline{1, r}, \quad k = \overline{1, N}, \quad 0 \leq l_{N+1} \leq 1. \tag{31c}$$

Résolvons le problème (31) par la méthode adaptée, présentée dans [9], de la manière suivante :
introduisons $S_B \subset S = \{1, \dots, N+1\}$, $|S_B| \le p$, et à chaque indice $k \in S_B$ faisons correspondre un ensemble $J_k \subset J = \{1, \dots, r\}$, $\sum_{k \in S_B} |J_k| = p$.
Posons $J_B = \{J_k, k \in S_B\}$, $T_B = \{k, k \in S_B\}$ et $Q_B = \{I_s, J_B, T_B\}$ et introduisons la matrice $P_B = P(Q_B) = (q_{ijk}, \quad i \in I_s, \quad j \in J_k, \quad k \in S_B)$, où q_{jk} est un m-vecteur et $\det P_B \neq 0$.

On prend comme plan initial $l^0_{jk} = 0$, à qui nous associons le support Q_B. Après un certain nombre d'itérations, on obtient la solution ε-optimale $\{l^\varepsilon, \widetilde{Q}_B\}$.
Si l'indice supplémentaire $(N+1) \in \widetilde{T}_B$, alors par la méthode duale, on l'exclut du support et on construit un nouveau support $\overline{Q}_B$.
Si l'indice $(N+1)$ n'est pas dans T_B, alors on pose $\overline{Q}_B = \widetilde{Q}_B$.
Ainsi, construisons la nouvelle commande de support $\{\overline{u}, \widetilde{Q}_s\}$, avec :

$$\overline{u}_j(t) = \begin{cases} u_j(t) + l^\epsilon_{jk}, & t \in [\tau_k, \tau^k], \quad k = \overline{1, N}, \\ u_j(t) + l^\epsilon_{N+1} \Delta u_j(t), & j = \overline{1, r}, \quad t \in T_*, \end{cases} \tag{32}$$

et le support $\widetilde{Q}_s = \{\widetilde{I}_s, \widetilde{J}_s, \widetilde{T}_s\}$ du problème (1)-(3) est construit de la manière suivante :

$$\widetilde{I}_s = \overline{I}_s, \qquad \widetilde{J}_s = \{\overline{J}_k, \quad k \in \overline{S}_B\}, \quad \widetilde{T}_s = \{\tau_k, \quad k \in \overline{S}_B\}. \tag{33}$$

En utilisant ces ensembles, on construit la matrice :

$$\widetilde{\varphi}_s = (\varphi_{ij}(t_k), \quad i \in I_s, \quad j \in \widetilde{J}_k, \quad k \in \widetilde{K}_s).$$

La nouvelle commande ainsi construite vérifie l'inégalité $J(\overline{u}) \ge J(u)$.

Calculons alors la nouvelle valeur de suboptimalité $\beta(\overline{u}, \widetilde{Q}_s)$. A partir de cette valeur on distingue trois cas :

- si $\beta(\overline{u}, \widetilde{Q}_s) = 0$, alors $\overline{u}$ est une commande optimale pour le problème (1)-(3) ;
- si $\beta(\overline{u}, \widetilde{Q}_s) \le \varepsilon$, alors $\overline{u}$ est une commande ϵ-optimale ;
- sinon, nous passons soit à une nouvelle itération en démarrant avec une commande d'appui $\{\overline{u}, \widetilde{Q}_s\}$ et les paramètres $\overline{\alpha} < \alpha$, $\overline{h} < h$, soit à la procédure de changement de support.

5.2 Changement de support

Soit $\{\overline{u}, \widetilde{Q}_s\}$ la commande de support obtenue après résolution du problème (31). Calculons par les formules (11)-(14) la co-commande $\widetilde{E}'(t) = -\widetilde{\psi}'(t)B - c_2'(t), \ t \in T$, correspondant à $\{\overline{u}, \widetilde{Q}_s\}$. Par la suite, construisons la quasi-commande $w(t), \ t \in T$:

$$w_j(t) = \begin{cases} d_j^-, & si \quad \widetilde{E}_j(t) > 0, \\ d_j^+, & si \quad \widetilde{E}_j(t) < 0, \\ \in [d_j^-, d_j^+], & si \quad \widetilde{E}_j(t) = 0, \ j = \overline{1, r}, \quad t \in T, \end{cases} \tag{34}$$

et sa quasi-trajectoire correspondante $\chi(t), \quad t \in T$, vérifiant l'équation :

$$\dot{\chi} = A\chi + Bw + r(t), \quad \chi(0) = x_0. \tag{35}$$

Construisons les vecteurs

$$\gamma(\widetilde{J}_s, \widetilde{T}_s) = \widetilde{\varphi}_s^{-1} \left(g_*^*(\widetilde{I}_s) - H(\widetilde{I}_s, K)\chi(t^*) \right), \tag{36}$$

$$\gamma^*(\widetilde{I}_H) = (\gamma_i^*, i \in \widetilde{I}_H = I \backslash \widetilde{I}_s), \quad \gamma_*(\widetilde{I}_H) = (\gamma_{*i}, \quad i \in \widetilde{I}_H),$$

où

$$g_{*i}^* = \begin{cases} g_{*i}, \; si \;\; \widetilde{y}_i < 0, \\ g_i^*, \; si \;\; \widetilde{y}_i > 0, \end{cases} \tag{37}$$

et

$$\gamma_i^* = \sum_{j \in \widetilde{J}_k, k \in \widetilde{K}_s} \varphi_{ij}(t_k)\gamma(j, t_k) + H(i, K)\chi(t^*) - g_i^*,$$

$$\gamma_{*i} = \sum_{j \in \widetilde{J}_k, k \in \widetilde{K}_s} \varphi_{ij}(t_k)\gamma(j, t_k) + H(i, K)\chi(t^*) - g_{*i}.$$

En introduisant un paramètre μ suffisamment petit deux cas peuvent se présenter :

1. Si les relations suivantes :

$$||\gamma(\widetilde{J}s, \widetilde{T}_s)|| \leq \mu, \quad \gamma^*(\widetilde{I}_H) \geq 0, \quad \gamma_*(\widetilde{I}_H) \leq 0, \tag{38}$$

   sont vérifiées, alors on passe à la procédure finale avec le support $\overline{Q}_s = \widetilde{Q}_s$.
2. Sinon on va changer le support ($\widetilde{Q}_s \longrightarrow \overline{Q}_s$) par une itération de la méthode duale, et on refait une nouvelle itération avec $\{\overline{u}, \overline{Q}_s\}$, $\overline{\alpha} < \alpha$ et $\overline{h} < h$.

5.3 Procédure finale

Admettons, que les relations (38) sont vérifiées pour la quasi-commande $w(t), \; t \in T$ et la quasi-trajectoire $\chi(t), \quad t \in T$, construite par le support $\overline{Q}_s$.
La procédure finale consiste à déterminer le support optimal $Q_s^* = \{I_s^*, J_s^*, T_s^*\}$ de telle manière à avoir $g_* \leq H\chi(t^*) \leq g^*$.
Ainsi, le support Q_s^* est déterminé en résolvant le système d'équations suivant :

$$\sum_{j \in \overline{J}_k} \sum_{k \in \overline{K}_s} (d_j^+ - d_j^-) sign\dot{E}_j(t_k) \int_{t_k}^{V_k(T_s^*)} \varphi_{ij}(t)dt - g_{*i}^* + H(i, K)\chi(t^*) = 0, \;\; i \in I_s^*, \tag{39}$$

où $V_k(T_s^*), k \in K_s^*$, est déterminé par les relations :

$$E_j(V_k(T_s^*), T_s^*) = 0, \quad V_k(\overline{T}_s) = t_k, \quad j \in \overline{J}_k, \quad k \in \overline{K}_s; \quad E(t, T_s^*) = c_s^{*'} \varphi_s^{*-1}\varphi(t) - c(t).$$

Supposons Q_s^l la $l^{ème}$ approximation, Q_s^0 l' approximation initiale , avec $I_s^0 = \overline{I}_s$, $J_s^0 = \overline{J}_s$, $T_s^0 = \overline{T}_s$. Supposons que la $l^{ème}$ approximation est connue, alors la $(l+1)^{ème}$ approximation sera construite de la manière suivante :

$$T_s^{l+1} = T_s^l + \left(\frac{1}{d_j^+ - d_j^-} sign\dot{E}_j(t_k)\gamma(j, t_k^l), \quad j \in J_k^l, \quad k \in K_s^l \right). \tag{40}$$

En outre, la $(l+1)^{ème}$ approximation sera construite d'une manière à satisfaire les relations (38). Ainsi, si à chaque approximation, les conditions (38) ne sont pas vérifies, nous changeons le support comme suit :

- Si $\exists \; i_* \in I_s^l$, avec $y_{i_*}^{l+1} = 0$, $\gamma_{*_i}^{l+1} \geq 0$, $\gamma_i^{*l+1} \leq 0$, $i \in I_H^k$, posons
  $Q_s^{l+1} = \{I_s^{l+1}, \;\; J_s^{l+1}, \;\; T_s^{l+1}\}$, avec $I_s^{l+1} = I_s^l \backslash i_*, \;\; J_s^{l+1} = J_s^l \backslash j_0, \;\; T_s^{l+1} = T_s^l \backslash t_{s_0}$.
  Calculons $\gamma^*(I_H^{l+1})$, $\gamma_*(I_H^{l+1})$, $\gamma^*(J_s^{l+1}, T_s^{l+1})$.
  Si les conditions (38) sont vérifiées, nous posons $Q_s^0 = Q_s^{l+1}$. Sinon, changeons le support jusqu'à ce que les conditions (38) soient satisfaites.

- Si $\exists\, i_* \in I_s^l$, $y_{i_*}^{l+1} = 0$; $\exists\, i_0 \in I_H^k$, $\gamma_{*i_0}^{l+1} > 0$, $\gamma_{i_0}^{*l+1} < 0$, nous changeons le support de la manière suivante :

$$I_s^{l+1} = (I_s^l \backslash i_*) \cup i_0, \quad J_s^{l+1} = J_s^l, \quad T_s^{l+1} = T_s^l.$$

- Si $\forall\ \ i \in I_s^l$, $y_i^{l+1} \neq 0$, et $\exists\ \ i_0 \in I_H$: $\exists\, i_0 \in I_H^k$, $\gamma_{*i_0}^{l+1} > 0$, $\gamma_{i_0}^{*l+1} < 0$, posons

$$I_s^{l+1} = I_s^l \cup i_0, \quad J_s^{l+1} = J_s^l \cup j_1, \quad T_s^{l+1} = T_s^l \cup t_s.$$

Faisons une nouvelle itération jusqu'à ce que les approximations successives ne diffèrent pas.

Soit $Q_s^* = \{I_s^*, J_s^*, T_s^*\}$ la solution du système (39), alors la quasi-commande $w^*(t), \quad t \in T$ calculée par (34) et le support Q_s^* est une commande optimale pour le problème (1)-(3), et Q_s^* est le support optimal.

6 Application en économie financière

Dans cette section, on présente les résultats numériques d'un modèle de financement de firme. Nous supposons une entreprise qui utilise le capital propre (dividendes non distribués) comme source de financement, et son objectif est de maximiser la somme des dividendes distribués aux actionnaires $V(D, t^*)$, sous la contrainte que son niveau de production peut satisfaire les engagements envers ses clients à une date donné t^*. Nous supposons aussi que la fonction de production de cette entreprise est celle de Leontieff à un seul facteur de production k. Le chiffre d'affaires de l'entreprise peut s'écrit alors $S = \nu k(t)$, où ν est la productivité nominale du capital. En effet, ce modèle s'écrit comme suit :

$$\begin{gathered} V(D, t^*) = \int_0^{t^*} e^{-\rho t} D(t) dt \to \max, \\ \begin{cases} \dot{k} = \nu k - (\delta + wl)k - D, \\ k(0) = k_0, \\ k(t^*) \geq k_f, \\ 0 \leq D(t) \leq D_{\max}, t \in T, \end{cases} \end{gathered} \tag{41}$$

où $k(t)$ est le capital de production à l'instant t, la commande $D(t)$ représente la quantité de dividende distribués à l'instant t, ν la productivité nominale, $lk(t)$ la quantité du travail à l'instant t, w le taux de salaire de la main-d'œuvre, ρ le taux d'actualisation et δ est le taux de dépréciation du capital.

Afin d'interpréter l'évolution de ce modèle, on a implémenté l'algorithme précédent sur machine en se servant des avantages du langage de programmation mathématique du Matlab7.

Pour tester le programme, les valeurs numériques suivantes ont été utilisées : $t^* = 10$; $k_0 = 20$; $k_f = 70$; $D_{\max} = 5$; $\rho = 0.05$; $\delta = 0.01$; $wl = 0.1$; $\nu = 0.3$.

Avec une précision de $\varepsilon = 10^{-4}$, la commande ε-optimale est la suivante :

$$D(t) = \begin{cases} 0 & pour \quad t \in [0, \quad 3.5229], \\ 3.3145 & pour \quad t \in [3.5229, \quad 3.5329], \\ 5 & pour \quad t \in [3.5329, \quad 10], \end{cases} \tag{42}$$

La trajectoire optimale $k(t)$ et l'évolution de la fonctionnelle associée à la commande ε-optimale dans le temps sont représentées sur les figures (1) et (2) :

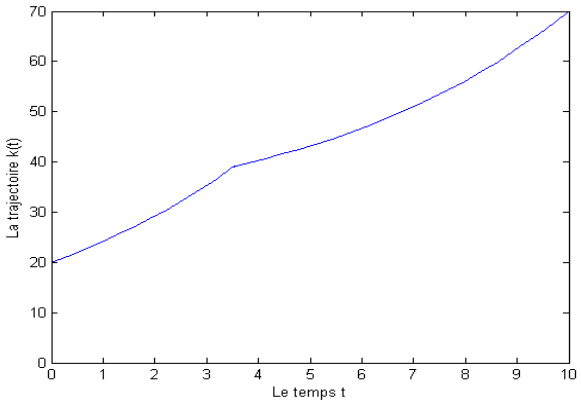


Figure 1. La trajectoire optimale.

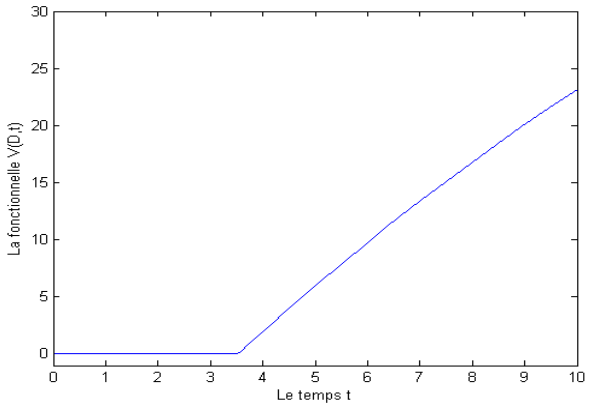


Figure 2. L'évolution de la fonctionnelle.

Ainsi, la valeur de la fonctionnelle est $V(D, t^*) = 23.1826$.

A partir des résultats obtenus, la décision optimale pour les actionnaires est de réinvestir leurs dividendes jusqu'à une date donnée pour profiter du taux de rendement que l'entreprise offre, puis de recevoir les dividendes à leurs valeurs maximales autorisées afin de maximiser les dividendes distribués.
Pour l'analyse de sensibilité du changement des bornes de la commande sur la valeur de la fonctionnelle et sur la commande optimale, prenons l'ensemble des valeurs de $D_{\max}$ suivant $M_v = \{2, 3, 7, 10\}$.
Les résultats pour les différentes valeurs de $D_{\max}$ sont donnés dans le tableau suivant :

D_{max}	La commande ε-optimale	$k(t*)$	V(D,t*)
2	$D(t) = 2$ sur $[0, \ 10]$.	73.8664	15.7388
3	$D(t) = 0$ sur $[0, \ 1.492]$; $D(t) = 3$ sur $[1.492, \ 10]$.	70	19.2948
7	$D(t) = 0$ sur $[0, \ 4.711]$; $D(t) = 4.0516$ sur $[4.711, \ 4.712]$; $D(t) = 7$ sur $[4.712, \ 10]$.	70	25.6838
10	$D(t) = 0$ sur $[0, \ 5.8178]$; $D(t) = 2.9357$ sur $[5.8178, \ 5.8278]$; $D(t) = 10$ sur $[5.8278, \ 10]$.	70	28.1607

Table 1. Résultats des scénarios d'exécution

L'évolution de la fonctionnelle dans le temps pour chaque valeur de D_{max} est représentée sur la figure suivante :

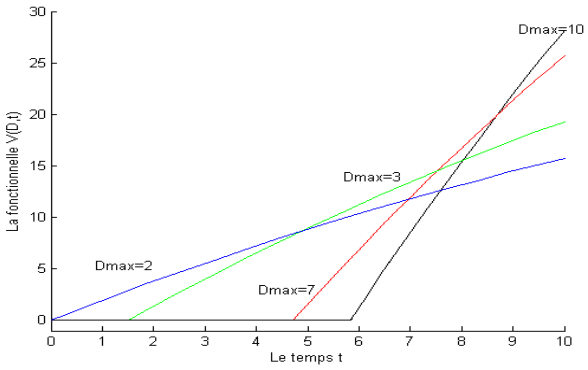


Figure 3. L'évolution des fonctionnelles.

Ces résultats montrent que plus la valeur de D_{max} est grande, plus l'instant de commutation de la commande est plus important ; en effet, les actionnaires vont profiter du taux de rendement que l'entreprise offre.
A partir des résultats obtenus, nous remarquons qu'un accroissement positif de D_{max} conduit à une valeur plus importante des dividendes distribués jusqu'à l'instant t^*. Ainsi, la fonctionnelle $V(D, t^*)$ est une fonction croissante de D_{max}.

7 Conclusion

Dans ce travail, nous nous sommes intéressés en premier lieu au problème de contrôle optimal d'un système dynamique linéaire. En s'inspirant de la méthode adaptée, développée par R.Gabassov et F.M.Kirillova, nous avons mis au point une généralisation de cette méthode au cas de contrôle optimal multivariable, avec un critère de Bolza, et des contraintes inégalités. Cette méthode se base sur trois procédures essentielles :
i) changer la commande u par $\overline{u}$ d'une manière à diminuer la mesure de non optimalité de la commande ; ii) changer le support Q_s par $\overline{Q_s}$ de telle sorte que la mesure de non optimalité de support sera diminuée ; iii) procédure finale qui consiste à rendre la quasi-commande w à la fois réalisable et optimale.

En second lieu, nous avons présenté les résultats du modèle de financement de firme avec la méthode adaptée. Ces résultats montrent que la décision optimale pour l'entreprise est de réinvestir les dividendes jusqu'à une date donnée, puis de les distribuer à leurs valeurs maximales autorisées jusqu'à l'instant t^*. De plus, nous avons remarqué que la fonctionnelle $V(D, t^*)$ est une fonction croissante de D_{max} (valeur maximale des dividendes autorisés à chaque instant).

Références

1. Gabasov R. and Kirillova F. M. Constructive Theory of Extremal problems. -University Press, Minsk, 1984.
2. Sethi S. P. and Thompson G. L. Optimal Control Theory : Applications to Management Sciences and Economics. -Springer, New York, 2006.
3. Dmitruk M. N. and Gabasov R. The Optimal Policy of Dividends, Investments, and Capital Distribution for the Dynamic Model of a Company. -Automation and Remote Control, 2001, 62(8) :1349-1365.
4. Bibi M. O. Optimization of a linear dynamic system with double terminal constraint on the trajectories. -Optimisation, 1994, 30 :359 - 366.
5. Bibi M. O. Support Method for Solving a Linear-Quadratic Problem with Polyhedral Constraints on Control. -Optimisation, 1996, 37 :139 - 147.
6. Bibi M. O. Méthode de Résolution d'un Problème Linéaire-Quadratique de Commande Optimale Multivariable. -Actes du Premier Colloque Maghrébin sur les Modèles Numériques de l'Ingénieur, 1996, 37 :97-102.
7. Bibi M. O. Optimal control of a quadratic problem with a piecewise linear entry.-Actes de la $5^{\text{ième}}$ Conférence Internationale en Recherche Operationnelle (CIRO'10), université de Marrakech, Maroc, 2010 :1 - 4.
8. Gabasov R. and Kirillova F. M. Constructive Methods of Optimization. P.II : Control Problems. -University Press, Minsk, 1984.
9. Gabasov R. and Kirillova F. M. Méthodes de Programmation Linéaire. P.III : -BGU, Minsk, 1980.
10. Krouse C. G. and Lee W. Y. Optimal equity financing of the corporation. -The Journal of Financial and Quantitative Analysis, 1973, 8(3) :539 - 563.
11. Pontryaguine L. S., Boltyanskii V. G., Gamkrelidze R. V. and Mishchenko E. F. The Mathematical Theory of Optimal Processes. -John Wiley and Sons, New Jersey, 1962.

Une approche basée sur la réduction de l'espace d'état pour la résolution du problème de tournées de véhicules

H. AIT HADDADENE[(1)], A. LAMAMRI[(2)], A. NAGIH[(3)]

[(1)]Laboratoire LAID3, USTHB, Algérie. [(2)] USTHB, Algérie.

[(3)]Laboratoire LITA, Université Paul Verlaine – Metz, France.

[(1)]aithaddadenehacene@yahoo.fr, [(2)]prslamamri_inch@yahoo.fr, [(3)]anass.nagih@univ-metz.fr

Abstract: We are interested in problems from combinatorial optimization, more precisely, the vehicle routing problem with resource constraints. Given the large size of the problems encountered in practice, these models are solved by an approach based on column generation that can handle implicitly all feasible solutions and a master problem determining the best solution. We propose in this paper an approach to improve the acceleration of the method of column generation for solving the problem of construction vehicle routing, it is projected in each arc, the resources a vector of size smaller by using a Lagrangean relaxation algorithm to determine the coefficients of the projection arc combined with an algorithm for re-optimization, then generates a sub-set of complementary solutions to the master problem. The preliminary experiments of our technique gave good results on instances of random vehicle routing.

Keywords: Mathematical programming, path problems in graphs, decomposition methods, column generation.

Résumé :

Nous nous sommes intéressés aux problèmes issus de l'optimisation combinatoire, plus précisément, le problème de tournées de véhicules avec contraintes de ressources. Etant donné la grande taille des problèmes rencontrés en pratique, ces modèles sont résolus par une approche basée sur la génération de colonnes qui permet de gérer implicitement l'ensemble des solutions réalisables et un problème maître déterminant la meilleure solution. Nous proposons dans cet article une approche permettant d'améliorer l'accélération de la méthode de la génération de colonnes pour la résolution de problème de construction des tournées de véhicules, il s'agit de projeter, en chaque arc, les ressources sur un vecteur de dimension inférieure en utilisant un algorithme de relaxation lagrangienne pour déterminer les coefficients de la projection par arc combinée à un algorithme de ré-optimisation, puis génère un sous-ensemble des solutions complémentaires vers le problème maître. Les expérimentations préliminaires de notre technique ont donné de bons résultats sur des instances aléatoires de tournées de véhicules.

Mots clés : Programmation mathématique, Problèmes de cheminements dans les graphes, Méthodes de décompositions, Génération de colonne.

1. Introduction :

Le problème de tournées de véhicules avec contraintes de ressources (PTVCR) est donné par un ensemble de clients N et d'un ensemble de véhicules $\mathcal{K}$ disponibles dans un dépôt. Ce problème consiste à trouver un ensemble de tournées de coût minimal, partant et retournant à un même dépôt, où chaque client est visité par un seul véhicule pour satisfaire une certaine demande. Chaque client doit être servi au cours d'une fenêtre de temps donnée. Un véhicule

arrivant à l'avance chez un client, attend jusqu'à la date de début de service sans surcoût, les fenêtres de temps dans ce cas sont dites 'dures'. Certains modèles pénalisent l'attente avec un surcoût, ces modèles sont dits 'soft', mais une grande partie de la recherche est dédiée aux fenêtres de temps 'dures'. Un véhicule arrivant en retard chez un client n'est pas admis pour effectuer son service. Une tournée réalisable est une suite de visites (effectuées par un même véhicule) respectant les fenêtres de temps, et qui commence et se termine au même dépôt.

Le PTVCR est défini par un graphe orienté $G^k = (X^k, A^k)$ où X = $N \cup \{s, t\}$, $N = \{1, \ldots, n\}$ est l'ensemble des sommets représentant les clients, s et t les sommets source et destination respectivement, représentant le dépôt, et A l'ensemble des arcs qui interconnectent les clients ainsi que le dépôt. Un arc (i, j) $\in A$ désigne la possibilité d'enchaîner le service des clients *i* et *j*. Afin d'écrire la formulation de ce problème, nous introduisons les notations suivantes :

– c_{ij} est le coût de l'arc (i, j) $\in A$.
– t_{ij} la durée de l'arc (i, j) $\in A$.
– [ai, bi] la fenêtre de temps durant laquelle le service du client $i \in N$ doit commencer.
– di la demande du client i $\in N$.
– Q la capacité de chaque véhicule.

Une affectation de clients aux véhicules est dite réalisable si :
– la demande cumulée des clients visités par un véhicule ne dépasse pas sa capacité.
– les contraintes temporelles sont respectées par chaque véhicule.
– chaque client est visité par un seul véhicule.
– chaque véhicule qui part du dépôt retourne au dépôt après avoir effectué sa tournée.
Le problème consiste à trouver une affectation réalisable des tournées aux véhicules, de coût minimum.

2. Modélisation mathématique

Le problème de tournées de véhicules avec contraintes de ressources (PTVCR) peut se modéliser, si la fonction de coût est linéaire, par la Programmation Linéaire en variables mixtes. Nous avons un problème de flot réalisable à coût minimal sur l'ensemble des sous-réseaux, avec variables binaires de flot et variables continues de ressources :

$$
(\text{PTVCR}) \equiv \begin{cases}
\min \sum_{k=1}^{K} \sum_{(i,j)\in A} c_{ij}\, x_{ij}^k & (1) \\
\sum_{k=1}^{K} \sum_{j:(i,j)\in A} x_{ij}^k = 1 \;\; pour \;\; i \in N = \{1, \ldots, n\} & (2) \\
\sum_{(i,j)\in A} d_i x_{ij}^k \leq \mathrm{Q} \quad pour \;\; k \in \mathcal{K} & (2') \\
\sum_{i:(o^k,i)\in A} x_{o^k i}^k = 1 \;\; pour \;\; k \in \mathcal{K} & (3) \\
\sum_{i:(i,d^k)\in A} x_{i d^k}^k = 1 \;\; pour \;\; k \in \mathcal{K} & (4) \\
\sum_{i:(i,j)\in A} x_{ij}^k = \sum_{l:(j,l)\in A} x_{jl}^k \;\; pour \;\; j \in N & (5) \\
x_{ij}^k \left(T_i^{k,q} + t_i^{k,q} - T_j^{k,q}\right) \leq 0 \;\; pour (i,j) \in A\,, k \in \mathcal{K}\,, q \in \mathcal{Q} & (6) \\
a_i^{k,q} \leq T_i^{k,q} \leq b_i^{k,q} \;\; pour \;\; i \in N, k \in \mathcal{K}\,, q \in \mathcal{Q} & (7) \\
x_{ij}^k \in \{0,1\}\,, T_i^{k,q} \geq 0 \;\; pour \;\; (i,j) \in A\,, k \in \mathcal{K}\,, q \in \mathcal{Q} & (8)
\end{cases}
$$

Les variables binaires x_{ij}^{k} indiquent si la tournée emprunte l'arc $(i,j) \in A$, tandis que les variables $T_i^{k,q}$ indiquent la consommation cumulée de chaque ressource q à chaque nœud i.

L'objectif (1) minimise le coût total des tournées. Les contraintes (2) expriment la couverture de chaque client par une tournée et les contraintes $(2')$ traduisent la limitation de la capacité des véhicules. Les contraintes $(3-5)$ définissent une structure de chemin dans un réseau G : passage d'un flux d'une unité $(3\ ou\ 4)$ et conservation du flux aux sommets (5). Les contraintes $(6-7)$ sont les contraintes de ressources associées à chaque tournée et les contraintes (5) permet d'obtenir la consommation cumulée de ressource q au nœud j, puisqu'on a :

$$T_j^{k,q} = \max\left(a_i^{k,q}, T_i^{k,q} + t_i^{k,q}\right) \qquad (9)$$

Les contraintes (7) sont des contraintes de bornes aux noeuds du réseau (fenêtres de temps par exemple). Remarquons que les contraintes $(3-7)$ sont des contraintes locales valables pour le seul réseau G. Seules les contraintes de partitionnement (2) sont des contraintes globales liant les K sous-réseaux. La relaxation de ces contraintes liantes et la décomposition du problème initial par sous-réseau sera donc une option de résolution intéressante. Notons enfin que les contraintes de ressources $(6-7)$ rendent le problème (PTVCR) NP-difficile. Même le problème de réalisabilité associé est NP-complet [5].

3. Approches de résolution

3.1 Principes de décomposition

On distingue deux types de contraintes dans le système $(2)-(7)$:
(i) les contraintes de partitionnement (2), dites liantes ou globales, qui lient l'ensemble des véhicules $k = 1, \ldots, K$,
(ii) les contraintes $(3)-(7)$ propres à chaque véhicule $k \in \{1, \ldots, K\}$ et définissant un itinéraire légal.
La matrice associée aux contraintes $(3)-(7)$ étant bloc-diagonale, et l'objectif (1) étant séparable (car linéaire), la résolution de la relaxation continue de ce modèle peut être basée sur la décomposition de Dantzig-Wolfe. Dans ce type de décomposition, les contraintes $(3)-(7)$ définissent K sous-problèmes indépendants et les contraintes globales (2) sont conservées dans le problème maître. Dans un schéma de type génération de colonnes, il s'agit de résoudre alternativement le problème maître et les K sous-problèmes. Pour obtenir une solution entière, ce schéma peut être appliqué au niveau de chaque nœud de l'arbre de recherche. La difficulté majeure réside dans la résolution des sous-problèmes dont les espaces des états peuvent augmenter de façon exponentielle avec le nombre de ressources Q, rendant incontournable l'utilisation d'heuristiques. D'autre part, la convergence du schéma de génération de colonnes étant sensible à la qualité des solutions fournies par la résolution de ces sous-problèmes, la résolution effective d'instances réelles issues de l'industrie nécessite de trouver un bon compromis entre la qualité des solutions et le temps de résolution des sous-problèmes. Dans ce qui suit, nous détaillons le principe général de la génération de colonnes pour le problème$(PTVCR)$.

3.2 Génération de colonnes, problème maître et sous-problème

Dans cette approche, le problème maître est reformulé par un Problème de Partitionnement (PP) (*Set Partitioning*) :

$$(PP) \equiv \begin{cases} \min \sum_{r \in \mathcal{R}} c_r x_r & (10) \\ \sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \; pour \;\; i \in N = \{1, \ldots\ldots, n\} & (11) \\ x_r \in \{0,1\} \quad pour \quad r \in \mathcal{R} & (12) \end{cases}$$

Où $\mathcal{R}$ désigne l'ensemble des tournées réalisables satisfaisant les contraintes de ressources et d'enchaînement entre clients, c_r représente le coût de la tournée $r \in \mathcal{R}$, $a_{ir} = 1$ si et seulement si le client i est visité par la tournée r, et la variable binaire x_r indique le choix ou non de la tournée r dans la solution.

On note $(\overline{PP})$ la relaxation continue du problème (PP) où les contraintes d'intégrité (12) sont remplacées par $x_r \geq 0$ pour $r \in \mathcal{R}$. Le nombre total de tournées admissibles $|\mathcal{R}|$ étant généralement une fonction exponentielle du nombre $n = |N|$ de clients à couvrir, l'énumération complète de $\mathcal{R}$ est à proscrire. Pour autant, il est possible de trouver en un temps raisonnable une solution optimale de $(\overline{PP})$ en ne générant qu'un sous-ensemble restreint de tournées (i.e., de colonnes de la matrice de contraintes).

De façon générale, on résout à chaque itération t le problème maître restreint $(\overline{PP}^t)$:

$$(\overline{PP}^t) \equiv \begin{cases} \min \sum_{r \in \mathcal{R}^t} c_r x_r & (13) \\ \sum_{r \in \mathcal{R}^t} a_{ir} x_r = 1 \; pour \;\; i \in N = \{1, \ldots\ldots, n\} & (14) \\ x_r \geq 0 \;\; pour \quad r \in \mathcal{R}^{\,t} & (15) \end{cases}$$

où, si δ^{t-1} désigne le vecteur de multiplicateurs associé aux n clients dans la résolution de $(\overline{PP}^{t-1})$, la tournée r^{t-1} de plus faible coût réduit négatif est définie par

$$r^{t-1} = arg \min_{r \in \mathcal{R}} \left(c_r - \sum_{i=1}^{n} \delta_i^{t-1} a_{ir} \right) \qquad (16)$$

Le terme de *génération de colonnes* provient de l'ajout de la colonne a_{r^t} à la matrice des contraintes du problème maître, à chaque itération t. Ce processus de résolution itérative du problème maître $(13-15)$ et du sous-problème (16) est stoppé dés que toutes les tournées sont de coût réduit positif dans la résolution du sous problème, signe que l'optimum continu est atteint.

Une variante de cette méthode, permettant d'accélérer le processus en pratique [3], consiste à ajouter à chaque itération un sous-ensemble de tournées complémentaires de coût réduit négatif au lieu de la seule meilleure tournée du sous-problème (16)(voir [4]). La taille maximale souhaitée de ce sous-ensemble de colonnes entrantes pourra être paramétrée de manière à évoluer au cours de l'algorithme. La complexité globale de la méthode est fortement dépendante de la complexité du sous-problème, que les contraintes de ressources rendent NP-difficile. Il est souvent possible cependant de le résoudre en un temps raisonnable grâce à une énumération *implicite* de $\mathcal{R}$, en exploitant la structure de graphe du sous-problème et en appliquant des variantes d'algorithmes de plus court chemin.

3.3 Résolution du sous-problème pour la génération de colonnes

Notant que dans le cas de plusieurs sous réseaux $k = 1, \dots, K$, la résolution du sous problème(16)étant décomposable par sous réseaux, on omettra l'indice k et le graphe du sous problème sera noté $G = (\{o\} \cup N \cup \{d\}, A)$.

Pour résoudre ce problème, Desrochers et Soumis [1] proposent un algorithme de programmation dynamique du type pulling.

Définition 1 A chaque chemin de l'origine o au nœud j, on associe une étiquette $E(\mathrm{C_j}, \mathrm{T_j}) = E(\mathrm{C_j}, \mathrm{T_j^1}, \dots, \mathrm{T_j^Q})$ représentant l'état de ses ressources et son coût.

Définition 2 Soient $E(\mathrm{C_j}, \mathrm{T_j})$ et $E'(\mathrm{C'_j}, \mathrm{T'_j})$deux étiquettes associées à deux chemins réalisables Pet P'de o à j. On dit que $E(\mathrm{C_j}, \mathrm{T_j})$ domine $E'(\mathrm{C'_j}, \mathrm{T'_j})$, (resp. Pdomine P'), et on note $E(\mathrm{C_j}, \mathrm{T_j}) \leq E'(\mathrm{C'_j}, \mathrm{T'_j})$, (resp. $P \leq P'$) si et seulement si $\mathrm{C_j} \leq \mathrm{C'_j}$ et $T_j^q \leq T'^q_j$, $\forall q \in Q$.

L'algorithme de programmation dynamique (APD) procède en trois grandes étapes. En chaque nœud $j \in V$, il effectue les opérations suivantes :
1. Prolongation des chemins (génération des étiquettes),
2. Filtrage (test de réalisabilité),
3. Dominance (élimination des étiquettes non efficaces).

Pour un nœud j donne, des étiquettes sont créées en prolongeant celles présentes aux nœuds i, tels que $(i, j) \in A$. Ainsi, une nouvelle étiquette $E(\mathrm{C_j}, \mathrm{T_j})$ est donnée par

$$\mathrm{C}_j = \mathrm{C}_i + c_{ij}$$
$$T_j^q = \max\{a_j^q, T_i^q + t_{ij}^q\}, q \in Q$$

En considérant que tous les prédécesseurs du nœud $j \in N$ sont déjà traités, la dominance au nœud j peut être interprétée comme la détermination des Pareto optimaux du problème multicritère a $|Q| + 1$ fonctions :

$$\begin{cases} \min_{i;(i,j)\in A} \left(C_i + c_{ij}\ ; \max\{a_j^q, (T_i^q + t_{ij}^q)\}, q \in Q\right) \\ T_i^q + t_{ij}^q \leq b_j^q \quad , \qquad q \in Q \end{cases}$$

La relation de dominance $\leq$ étant une relation d'ordre partiel, le nombre d'étiquettes efficaces à traiter augmente de façon exponentielle en fonction du nombre de ressources, ce qui rend la procédure de prolongation très ardue.

Dans un récent travail Nagih et Soumis [2] proposent une méthode d'agrégation des ressources pour les PCC-CR par projection, en chaque nœud en utilisant simultanément un algorithme de programmation dynamique et une relaxation lagrangienne.

Comme le nombre de coefficients à ajuster sera plus important pour l'approche de projection par arcs, trouver les multiplicateurs optimaux u_{ij}^* nécessiterai plusieurs itérations successives de APD-L [2], cette méthode peut s'avérer coûteuse. Afin d'obtenir rapidement de bonnes solutions heuristique (réalisables), notre approche appliqué une seule fois APD-L puis appliquer APD-LND [2], en utilisant les multiplicateurs u_{ij} trouves afin de produire des colonnes réalisables et de coût marginal négatif. Plus précisément, on choisit d'abord une suite de pas (p_k) telle que la série $(\sum p_k)$ est divergente et $\lim_{k\to\infty} p_k = 0$, autrement dit les conditions qui assurent la convergence de l'algorithme du sous-gradient. On applique en premier lieu APD-L en utilisant les multiplicateurs u_{ij}^{k-1}de l'itération précédente, on trouve les

sous-gradients correspondants à l'arc (i, j) ensuite on calcule les nouveaux multiplicateurs de Lagrange u_{ij}^k. Cette heuristique est certainement basée sur le fait que lorsque k est grand, le vecteur C_k des coûts réduits sur les arcs du réseau ne change pas beaucoup d'une itération à une autre de l'algorithme de génération de colonnes. Ainsi pour k grand, on peut espérer voir u_{ij}^k converger vers une valeur optimale.

Les étapes principales de notre approche sont résumées ci-dessous :

Problème maitre.

Sous problème
- − calculer les multiplicateurs de Lagrange u_{ij}^k (Projection par arc).
- − calculer la solution $maxL(u_{ij}^k)$, en utilise APD − L
- − calculer les solutions réalisables $\Phi(u_{ij}^k)$, en utilise APD − LND

Généré la solution de meilleur coût négative.

---

4. Expérimentations

Cette section présente l'évaluation préliminaire de notre approche pour le Problème de construction des tournées de véhicules avec une seule ressource. Jusqu'à maintenant, notre méthode n'a été testée que sur des problèmes de petite taille. Considérant les trois problèmes suivants :

Problèmes	**Nombre de clients**	**Arcs**	**Le nombre total de tournées**
P1	4	15	16
P2	10	32	40
P3	20	70	102

Les résultats de plusieurs tests sont présentés dans le tableau 4.1. Chaque colonne donne des informations sur la valeur optimale (la meilleure valeur) du problème trouvé par « ALG.P.NS », « ALG.P.s.CC », « ALG.P.arc.M » [2], et notre méthode « N.M », le nombre d'opération pour la méthode de génération de colonne.

problèmes	**ALG.NS**		**ALG.P.s.CC**		**ALG.P.arc.M**		**N.M**	
	V	N.O	V	N.O	V	N.O	V	N.O
P1	7	2	7	1	6*	2	6*	1
P2	15.5	10	15.5	6	15.5*	10	15.5*	4
P3	24	10	24	6	23*	10	23*	3

Tableau 4.1. Quelques résultats.

Dans ce tableau, on considère scellement deux critères, la meilleure valeur du problème trouvé par cette technique et le nombre d'opération globale.

La comparaison entre les quatre techniques ont permis de constater que « N.M » a fourni de bons résultats. Ceux-ci sont meilleurs quand certaines conditions sont réunies : l'initialisation de l'algorithme, le choix des multiplicateurs de Lagrange et le pas de déplacement.

5. Méthode de séparation

La méthode de génération de colonnes est utilisée pour résoudre le problème relaxé au nœud U. Elle hybride l'algorithme de simplexe (méthode existant dans la librairie ILOG) avec une méthode nommée Pricing. Si la solution obtenue est fractionnaire alors une méthode de séparation est appliquée au problème P^u. Elle consiste à subdiviser l'ensemble des solutions entières Su en deux sous ensembles disjoints, ceci a pour conséquence d'éliminer la réalisabilité de la solution fractionnaire pour les deux nouveaux problèmes qui sont des fils de P^u.

6. Conclusion

Dans ce papier, nous avons proposé un algorithme pour le problème de tournées de véhicules avec Contraintes de Ressources (PTV-CR) qui est une extension du (PTV) standard pour prendre en compte un aspect plus pratique du problème, nous avons principalement développé les approches de génération de colonnes et de décomposition en problème maître et sous-problème. La difficulté de la résolution du sous-problème étant directement liée au nombre de ressources, nous avons particulièrement étudié les techniques de réduction de l'espace des ressources, cette notion de réduction étant un élément-clé de l'efficacité de la résolution globale du problème.

Références

[1] M. Desrochers, F. Soumis (1988a), A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows, INFOR 26, 191–212.

[2] A. Nagih, F. Soumis (2005) « Nodal aggregation of resource constraints in a shortest path problem », European Journal of Operational Research.

[3] N. Touati, L. Létocart, and A. Nagih. (2008) Solutions diversification in a column generation scheme. En soumission à Discrete Optimization.

[4] N. Touati, L. Létocart, and A. Nagih. (2008) Reoptimization in a column generation scheme. En soumission à Computers and Operations Research.

[5] Vangelis Paschos (2005), Livre, optimisation combinatoire 3: applications, Hermès Science. ch 10.

Ontologies et leurs applications

Gestion distribuée de compétences

Badrina Gasmi, Nacer Boudjlida, Hassina Nacer Talantikite

Département d'informatique, Université de Béjaia, Algérie
LORIA, Université de Henri Poincaré, Nancy1, France
Département d'informatique, Université de Béjaia, Algérie
{b_goussem@yahoo.fr, Boudjlida@loria.fr, sino_nacer@yahoo.fr}

Résumé. La gestion de compétences est un problème crucial dans plusieurs types d'applications distribuées comme le e-business. Ce problème concerne la capture, la mise en forme et l'exploitation de connaissances sur l'expertise ou les compétences d'un « «objet » (comme un partenaire industriel, une personne, voire un composant logiciel. Le travail décrit dans ce papier rentre dans ce cadre. Il traite de la publication des compétences (capacités ou savoir-faire) d'un « objet ». Ces compétences sont organisées et structurées afin d'être exploitées pour la recherche d'objets pouvant satisfaire un objectif ou répondant à un besoin. L'approche que nous proposons est fondée sur une représentation des connaissances à base des graphes conceptuels, et sur un modèle d'architecture à base des médiateurs.

Mots clés: Gestion de compétence, Représentation de connaissances, Graphes conceptuels, Médiateur.

1 Introduction

La gestion de compétences est la façon avec laquelle les organisations gèrent les compétences de la corporation, des groups et des individus. Son premier objectif est de définir et de maintenir continuellement les compétences selon les objectifs de la corporation [1]. La compétence est une notion multidisciplinaire, elle peut désigner : les fonctions d'un système ou d'un composant logiciel, l'expertise d'un partenaire industriel, les compétences acquises par un humain dans un domaine d'expertise donné, etc. Dans ce papier, nous visons à proposer une approche générale qui peut être instanciée dans différents domaines. Pour cela, nous définissons la compétence comme étant la capacité ou service d'une entité physique ou non physique (un objet, un composant logiciel, un partenaire industriel potentiel, un web service, etc.)

La gestion de compétences est un processus complexe, nous le traitons dans ce travail en trois étapes : (1) l'identification de compétences, (2) leur l'organisation, et (3) leur exploitation.

L'identification de compétences consiste à mettre les compétences sous une représentation formelle pour que celles-ci puissent être exploitées par les machines. Une fois représentées, les compétences sont organisées et structurées afin d'être exploitées lors de la recherche d'entités dotées de certaines compétences requises afin de satisfaire un objectif donné (par exemple : rechercher un composant dans le

contexte de la programmation par composants, rechercher un partenaire industriel d'une expertise donnée, rechercher un employé dont l'expertise satisfait le profil d'un poste donné, etc.).

La gestion de compétences étant la gestion de connaissances sur les compétences, donc, elle peut tirer avantages des langages de représentation de connaissances pour supporter le processus de la gestion de compétences. Notre travail est fondé sur une représentation basée sur les graphes en utilisant le formalisme des graphes conceptuels [9]. Par techniques basées sur les graphes nous entendons les techniques dans lesquelles non seulement la représentation est faite à base des graphes, mais aussi les raisonnements sont faits à base des opérations sur les graphes. Du point de vu architecture système, nous avons opté pour un modèle basé sur des médiateurs [2] distribués et coopératifs.

Le reste de ce travail est structuré comme suit. Dans la section 2, nous exposons quelques motivations et domaines d'applications possibles de notre travail, ainsi que l'architecture du système visé. Dans la section 3, nous introduisons brièvement les éléments du formalisme des graphes conceptuels. Dans la section 4 nous présentons l'utilisation que nous faisons du formalisme des graphes conceptuels pour la gestion de compétences, et finalement, dans la section 5 nous concluons notre travail.

2 Motivations et architecture

La gestion et la recherche de services ou de compétences que peut rendre une « entité » trouve son application dans différents domaines, dont la programmation par composants, la découverte dynamique de web services [3], les affaires électronique (e-business), la gestion des ressources humaines ou encore la constitution et l'exploitation de mémoires d'entreprises [4, 20].

Ainsi, dans le premier domaine, un composant logiciel est décrit par les services (ou fonctions) qu'il peut prendre en charge. Ces services sont la plupart du temps décrits essentiellement par leurs interfaces (ou signatures des services) spécifiant les types des paramètres en entrée et en sortie du service. Il est clair que la recherche d'un composant ne peut pas se satisfaire d'une description basée exclusivement sur la syntaxe : un niveau de description sémantique est nécessaire. En outre, une connaissance des relations éventuelles entre services peut aussi contribuer à trouver « le meilleur » service satisfaisant une demande. De même, dans le domaine de e-business, nous voyons l'application de nos travaux dans le cadre de la constitution de consortium ou lors de la recherche de partenaires ou de sous-contractants, etc. Par exemple, pour la réalisation ou la conduite d'un projet, les services visés par nos travaux aideraient un maitre d'œuvre, voire un maitre d'ouvrage, à trouver des partenaires ayant les compétences requises pour satisfaire les exigences du projet. Dans le dernier domaine, et considérant le e-recrutement, l'application de notre travail peut être nécessaire durant la recherche des employés satisfaisant le profit d'un poste donné.

Des techniques basées sur la représentation de connaissances ont été déjà utilisées dans le contexte de la gestion de compétences. Les logiques de description (DL) ont été largement utilisées dans ce contexte afin de fournir une représentation sémantique des compétences acquises par des entités d'un domaine d'application donné, et afin de

les organiser et de les découvrir [6,7,8]. DL en tant que langage de représentation de compétences a l'avantage de pouvoir utiliser un seul mécanisme qui est la classification, à la fois pour construire une représentation et pour l'exploiter.

Contrairement aux travaux existants utilisant des langages logiques, nous proposons dans ce travail une nouvelle approche basée sur les graphes en utilisant le formalisme des graphes conceptuels [9]. Ce formalisme est caractérisé par le fait que non seulement la représentation du domaine est faite par des graphes mais les raisonnements sont également faits par des opérations sur les graphes. De ce fait, nous pouvons utiliser directement les opérations sur les graphes sans avoir besoin à les transformer en formules logiques afin d'effectuer des raisonnements logiques. Par conséquent, notre approche peut bénéficier des algorithmes existants développés pour la manipulation des graphes parce que les graphes conceptuels sont des graphes étiquetés.

D'autres raisons motivent le choix du formalisme des graphes conceptuels pour supporter le processus de la gestion de compétences : (1) le formalisme des graphes conceptuels est un modèle flexible, il nous permet de crier des nouvelles assertions d'une manière libre, parce qu'il offre des structures variées. (2) En terme de facilité d'utilisation, un graphe édité à l'écran par le biais d'une interface conviviale n'a pas, auprès d'utilisateurs non informaticiens, le côté rebutant que peuvent avoir les formules logiques. (3) Le modèle des graphes conceptuel distingue clairement les connaissances ontologiques (support) des connaissances factuelles (les graphes conceptuels), alors le modèle peut être utilisé pour fournir une représentation sémantique pour les entités d'un domaine ainsi que leurs compétences. (4) Le formalisme des graphes conceptuels supporte des relations n^{aire}, n étant quelconque, les types de ces relations sont bien structurés selon multiples hiérarchies, cela aide à considérer des relations de différents nombres d'argument sans avoir besoin à incorporer d'autres mécanismes, contrairement aux logiques de descriptions par exemple qui ne traitent que des relations binaires.

Par ailleurs, étant donnés les objectifs visés, il est nécessaire d'offrir à une entité des possibilités de se décrire, c'est-à-dire d'expliciter (on dira de publier ou d'exporter) ses compétences pour que celles-ci puissent être explorés par d'autres « entités ». Ces besoins nous ont naturellement orientés vers un modèle d'architecture de médiateurs (ou traders) [2] très similaire à la notion d'une agence de découverte dans une architecture des web services [10]. Dans cette architecture, une entité appelée exportateur publie ses compétences auprès d'un ou de plusieurs médiateurs (arc (a) sur la figure 1). Des entités appelées importateurs adressent au médiateur des demandes de recherche d'exportateurs dotés de certaines compétences, à charge pour lui d'essayer de trouver les exportateurs satisfaisant la demande (arcs (b) et (c) sur la figure 1). Pour cela, le médiateur se fonde sur les descriptions des compétences qui lui ont été adressées ainsi que sur les relations entre elles, des relations qu'il aura lui-même établies. Si la requête peut être satisfaite par quelques exportateurs connus par le médiateur, les références de ces exportateurs sont ensuite envoyées à l'importateur (arc (c) sur la figure 1).

Le but ultime de ce travail est la conception et le developpement des services pour l'export, l'import et la médiation. Les paragraphes qui suivent détaillent l'approche proposée en utlisant le formalisme des graphes conceptuels [9] en commençant par une introduction à ces derniers, à savoir les graphes conceptuels.

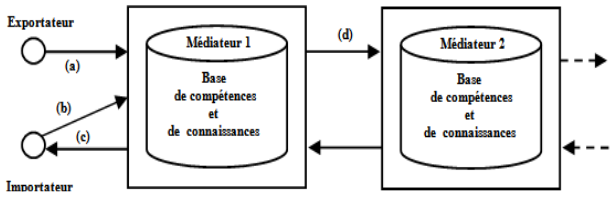


Fig. 1. Architecture à base des médiateurs

3 Le formalisme des graphes conceptuels

Les graphes conceptuels sont un formalisme introduit pour la première fois par SOWA [9]. Ils sont présentés comme un modèle générale d'écriture de réseaux sémantiques pour la représentation de connaissances. Ils sont conçus pour représenter la sémantique du langage naturelle ; ils sont émergés par la suite pour devenir un system complet dans le sens de la logique. Ce formalisme de représentation de connaissances permet la description des objets par des « concepts » et des relations entre objets par des « relations » appelées aussi des « relations conceptuelles ». Ces « concepts » et « relations » sont supposés être instanciés à partir d'un composant appelé « support ».

Le reste de cette section est organisé comme suit : section 3.1 définie le formalisme des graphes conceptuels et explique ces composants, quand à la section 3.2 introduit les opérations et les notions de base qui sont nécessaires à la compréhension du reste de ce papier.

3.1 Définition des graphes conceptuels

Un graphe conceptuel[1] est défini comme un graphe fini, orienté, biparti et non nécessairement connexe[2].

Deux niveaux de connaissances sont considérés dans les graphes conceptuels : (i) le niveau ontologique; il est spécifié par le « support » qui introduit le vocabulaire du domaine étudié ; nous parlons ici de la connaissance ontologique dans le sens ou ces connaissances sont organisées dans des hiérarchies de types. (ii) Le niveau assertionnel qui contient des connaissances factuelles encodées sous forme des graphes conceptuels.

Le support des graphes conceptuels (voir l'exemple de la figure 2) est composé de :

1. Une hiérarchie de types de concepts organisée autour d'une relation de spécialisation/généralisation. Cette hiérarchie contient deux types de concepts particuliers qui sont : le concept le plus général ("T_c" dans la figure 2), il représente tous les concepts, et le type de concept le plus spécifique ("$\perp_c$" dans la figure 2) qui représente « aucun concept ».

[1] Nous parlons ici des graphes conceptuels simples.
[2] Historiquement, les graphes conceptuels ont été connectés [9].

2. Un ensemble de types de relations organisés autour de différentes hiérarchies, chacune d'entre elles organise des types de relations ayant la même arité ; des relations de différentes arités sont incomparables.
3. Un ensemble de marqueurs ou référents individuels représentant les entités du domaine étudié (I dans la figure 2).
4. Une relation de conformité ("τ" dans la figure 2) qui associe chaque marqueur au type de concept dont il relève.
5. Les signatures des relations qui représentent tous les graphes exprimant les contraintes liées à chaque relation. Une signature définie le nombre d'arguments d'une relation ainsi que leurs types. Les signatures des relations constituent des graphes élémentaires avec lesquelles on peut construire des graphes plus complexes.

Un graphe conceptuel est composé de : (1) deux types de nœuds (concepts et relations) instanciés à partir du support et (2) des arcs orientés reliant ces nœuds et qui dénotent l'existence et la direction des relations. Une relation peut avoir plus d'un seul arc, dans ce cas les arcs sont numérotés. Un concept est une représentation d'un objet dans le monde de discours. Il est composé de deux parties : (1) un référent (général ou individuel) identifiant l'objet représenté (un référent générique est noté '*' et il peut être implicite dans la notation graphique ou linéaire des graphes conceptuels) et (2) un type qui classifie l'objet représenté. Une relation représente une association entre un ou plusieurs concepts, elle est composée d'un nom qui identifie le type de la relation, et d'arcs reliant la relation aux concepts attachés.

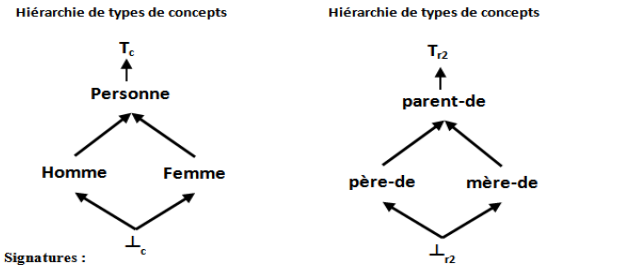


Fig. 2. Le support des graphes conceptuels

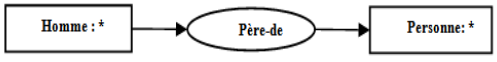


Fig. 3. Exemple de graphes conceptuels.

Les graphes conceptuels peuvent être représentés sous différents notations concrètes comme la notation graphique, la notation textuelle et la notation CGIF [18].

Les graphes conceptuels étant un system de logique, ils peuvent être traduits à une autre forme de logique comme la logique des prédicats par un opérateur Φ. L'exemple de la figure 3 peut être écrit de la manière suivante :

∃ h ∃ p: Personne(p) ∧ Homme(h) ∧ père-de(h,p).

3.2 Les opérations et les notions de base sur les graphes conceptuels

- *La projection :* La projection [11] est une opération de base sur les graphes conceptuels. Une projection d'un graphe 'H' sur un graphe 'G' existe veut dire que toute l'information exprimée par 'H' est exprimée aussi par 'G'. La projection est adéquate [9] et complète [12] par rapport à la sémantique des logiques associées. Formellement, la projection est définie comme une application 'Π' des nœuds d'un graphe 'H' sur les nœuds d'un graphe 'G' tel que: (i) pour chaque concept "c" de "H", Π(c) (nommé aussi l'image de "c") est soit identique soit une spécialisation de "c", (ii) pour chaque relation "r" de "H", Π(r) (ou l'image de "r") est soit identique soit une spécialisation de "r", (iii) si le i^{eme} arc de "r" est relié à un concept "c" dans "H", alors le i^{eme} arc de Π(r) doit être relié à Π(c) dans "G".
- *La normalisation*: La normalisation [16] est une opération qui met un graphe conceptuel sous sa forme normale. Un graphe sous forme normale respecte une structure ou les référents sont uniques; il est obtenu en fusionnant les concepts ayant le même marqueur individuel. La forme normale évite les ambigüités logiques et sémantiques dans les graphes conceptuels, sans la forme normale, deux graphes peuvent avoir la même interprétation logique sans avoir le même graphe qui les représente.
- *La somme disjointe:* La somme disjointe de deux graphes est une opération permettant de mettre deux copies des graphes originaux en juxtaposition pour avoir un seul graphe.
- *Headed graphs:* Ce sont des graphes conceptuels ayant un certain nœud choisi comme une tête sémantique**.**
- *Les règles des graphes conceptuels:* Les règles des graphes conceptuels [13] sont proposées comme une extension aux graphes conceptuels simples pour représenter des connaissances de type : "SI A ALORS B", tel que "A" et "B" sont des graphes conceptuels simples. Le mécanisme d'application des règles sur un graphe conceptuel représentant un effet par exemple, est basé sur l'opération de projection. Ce mécanisme est prouvé être adéquat et complet par rapport à la déduction par les formules logiques associées [14].

Après cette brève introduction aux graphes conceptuels, revenons à l'utilisation que nous faisons de ce formalisme pour la gestion de compétences.

4 La gestion de compétences

Dans cette section, nous détaillons notre approche pour la gestion de compétences selon ces trois étapes: la représentation de compétences (section 4.1), leur organisation (section 4.2) et leur utilisation (section 4.3).

4.1 La représentation de compétences

La représentation de compétences est basée sur le formalisme des graphes conceptuels. Les compétences sont représentées par des « relations », et les entités sont représentées par des « concepts ». Par exemple, dire qu'un programmeur "p" a les compétences pour programmer en Java, peut être représenté comme suit :

[Programmeur: p]->(programmer)->[Langage-prog: Java].

Cependant, le formalisme des graphes conceptuels simples ne permet pas de représenter d'une manière adéquate les entités d'un domaine ainsi que leurs compétences; en effet, dans le formalisme des graphes conceptuels simples, le sens d'un type (concept ou relation) est donné juste par rapport à sa position dans les hiérarchies de types, le seul mécanisme les définissant est la relation de spécialisation/généralisation. Cette représentation est pauvre et manque beaucoup d'expressivité pour représenter des informations génériques sur les types ainsi que quelques propriétés sur les relations tel que (la transitivité, la symétrie, etc.).

Pour répondre à ces objectifs, nous proposons d'utiliser les règles de graphes conceptuels tel que décrit dans les sections qui suivent.

Définition des types de concepts. "La définition d'un type de concept" est définie dans ce travail comme étant *"les conditions nécessaires ou nécessaires et suffisantes qui doivent être vérifiées pour appartenir à ce type de concept "*. Ces conditions sont formalisées par des règles de graphes.

Exemple 1 . Un type de concept "Mère" peut être défini comme suit:

[Mère: *x] => [Femme: ?x]->(mère-de)->[Personne]

[Femme: *x]->(mère-de)->[Personne]=>[Mère: ?x]

Exemple 2. Le type de concept "Mère-de-filles" peut être aussi défini comme suit:

[Mère-de-filles: *x]=>[Mère: ?x]

[Mère-de-filles:*]->(mère-de)->[Personne:*y]=>[Femme: ?y].

Même si ce dernier exemple n'exprime que les conditions nécessaires pour appartenir au type de concept "Mère-de-filles", cela permet de déduire quelques informations. Par exemple, ayant le graphe:

"[Mère-de-filles: x]->(mère-de)->[Personne: y]", nous pouvons déduire le graphe: "**[Femme: y]"** (i.e. « y » est un concept de type "Femme").

Définition de type de relation. De la même manière nous considérons "la définition d'un type de relation" comme étant "*les conditions nécessaires ou nécessaires et suffisantes pour appartenir à ce type de relation* ".

Exemple 3. La relation "grand-mère-de" peut être définie par les deux règles suivantes:

[Femme: *x]->(grand-mère-de)->[Personne: *y] ==>
[Femme: ?x]->(mère-de)->[Personne: *]->(parent-de)->[Personne: ?y].

[Femme: *x]->(mère-de)->[Personne: *]->(parent-de)->[Personne: *y]
==>[Femme: ?x]->(grand-mère-de)->[Personne: ?y]

Meta-connaissances sur les relations. Les propriétés des relations peuvent être aussi formalisées en utilisant les règles de graphes conceptuels.

Exemple 4. La propriété de transitivité de la relation "sœur-de" peut être formalisée en utilisant les règles de graphes comme suit :

[Femme: *x]-> (sœur-de)->[Femme: *]->(sœur-de) -> [Personne: *y]
=>[Femme: ?x]->(sœur-de)->[Femme: ?y].

Suite à cette représentation par les règles de graphes, le « support » dans notre approche sera composé des composants usuels du « support » dans les graphes conceptuels simples, à lesquels on rajoute une base de règles notée 'RB' et composée de toutes les règles utilisées pour définir les types ainsi que celles utilisées pour définir les propriétés sur les relations.

Le « support » étant défini, nous pouvons maintenant construire des graphes conceptuels représentant des situations particulières. Dans ce travail, ces graphes représentent les entités du domaine d'études accompagnées de leurs compétences acquises; ces graphes sont appelés ici "la base de compétences" et elle est notée (CB). La construction ainsi que l'organisation d'une base de compétences sont présentées dans la section suivante.

4.2 L'organisation de Compétences

Rappelons que d'un point de vu architecture system, nous avons opté pour une architecture à base des médiateurs (voir la section 2). Donc, la base 'CB' est construite et mise à jour à chaque nouvelle publication.

Pour chaque compétence publiée (représentée par un graphe G) nous appliquons les étapes du processus suivant:

1. Premièrement, le graphe "G" est ajouté à la base 'CB'. Pour cela, nous utilisons l'opération de la somme disjointe de "G" et "CB".
2. Deuxièmement, "CB" est normalisé. Dans le cas de la recherche de compétences, nous voulons éviter la redondance afin de minimiser la recherche mais aussi pour homogénéiser les connaissances. C'est pour cette raison que nous utilisons la normalisation.

3. Finalement, la troisième étape consiste à appliquer les règles présentes dans “RB”. Cette étape est très importante ; elle permet d’ajouter à la base “CB” toutes les connaissances publiées implicitement.

Exemple 5. Considérons le type de concept “OOPL” (langage de programmation orienté objet) qui est une spécialisation du type de concept “langage-prog” (Langage de programmation).

Nous définissons le type “OOPL” par les deux règles suivantes:

R1: [Langage-prog:*x]->(supporte)->[classes] ==> [OOPL:?x]
R2: [OOPL:*x]==> [Langage-prog:?x]->(supporte)->[classes]

Supposons qu’un programmeur appelé “p” publie sa capacité de programmer en langage JAVA; cette capacité est formalisée comme suit:

[Programmeur: p]->(programmer)->[OOPL: Java]

Alors, en appliquant la règle R2, nous obtenons le graphe suivant représentant la nouvelle base de compétences ‘BC’:

[Programmeur: p]->(programmer)->[OOPL: Java]-(supporte)->[classes]

4.3 L’utilisation de compétences

Dans ce papier nous nous intéressons à l’exploitation des compétences lors de la recherche d’entités ayant certaines compétences requises. La section 1 explique la représentation de la requête, tandis que la section 2 présente sa satisfaction.

La représentation de la requête. La requête est représentée par un « headed graph »; la tête du graph représente le type des entités recherchées. Nous introduisons dans ce travail un marqueur spécial qui est ‘ ?’ équivalent au marqueur générique ‘*’ pour indiquer la tête du graphe.

Exemple 6 . Une requête de recherche de programmeurs ayant les compétences pour programmer en C peut être représentée comme suit:

[Programmeur: ?]-->(programmer)->[Language-prog: C]

La requête peut être complexe en ayant des conditions sur les compétences recherchées. Dans ce cas, les compétences recherchées sont représentées par des relations directement attachées à la tête de la requête, tandis que le reste du graphe représente les conditions sur les compétences recherchées.

Exemple 7. Une requête de recherche de programmeurs ayant les compétences requises pour programmer en un langage qui supporte les classes, peut être représentée de la manière suivante:

[Programmeur: ?]->(programmer)->[Langage-prog: *] ->(supporte)->[Classes]

Satisfaction de la requête. La satisfaction d'une requête (représentée par un graphe R) est accomplie en suivant les étapes suivantes:

1. Premièrement, "R" est normalisé afin de minimiser sa taille et par conséquent minimiser la recherche.
2. Supprimer de "R" toutes les composantes connexes non connectées à la tête, parce que ces composantes sont indépendantes des compétences recherchées.
3. La troisième étape consiste à rechercher des réponses à cette requête. Pour cela, nous utilisons l'opération de projection de 'R' sur 'BC'.
4. Si au moins une projection est trouvée alors la requête est satisfaite. Dans ce cas, les réponses sont les projections du nœud tête.

Exemple 8 . Considérons le graphe "BC" de l'exemple 5. La satisfaction de la requête de l'exemple 7 est accomplie en utilisant l'opération de projection comme illustré dans la figure 4.

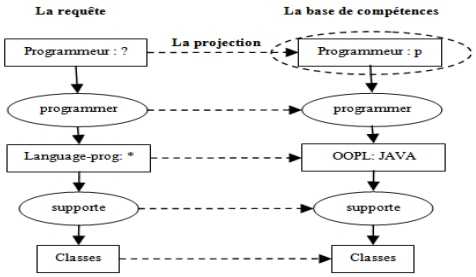


Fig. 4. Exemple de satisfaction d'une requête

Le concept encerclé dans la figure 4 représente l'image de la tête de la requête. Alors la réponse est: [Programmer: p].

5 Implémentation

Pour la validation expérimentale de nos idées, nous avons implémenté un prototype de médiation qui répond aux objectifs visés par ce papier. Le système est implémenté en utilisant la bibliothèque CoGITaNT [3] [17], version 5.2.0. Le prototype proposé a une architecture de médiation basée sur l'architecture client serveur de CoGITaNT. La communication entre les clients (importateurs ou exportateurs) et le médiateur est basée sur le protocole TCP.

[3] Bibliothèque C++ développée par LIRMM CNRS, France.

Il est à noter que l'implémentation actuelle de notre prototype ne permet pas la génération des graphes conceptuels automatiquement à partir des descriptions des sources. Cette tache est importante mais elle nécessite l'enrichissement de notre prototype par un system de traitement linguistique. Alors pour tester notre prototype, nous construisons les graphes conceptuels ainsi que les autres composants en utilisant les logiciels d'édition graphiques des graphes conceptuels tels que Cogui [19].

6 Conclusion

Dans ce papier, nous avons traité le problème de la gestion de compétences. Notre objectif était de trouver une méthode pour représenter formellement les compétences acquises par des entités d'un domaine particulier, de les organiser et de les utiliser lors de la recherche d'entités ayant certaines compétences requises.

Notre approche est fondée sur le formalisme des graphes conceptuels pour fournir une description sémantique d'un domaine d'application. Pour la représentation de compétences nous avons utilisé les règles de graphes pour exprimer les informations génériques sur les types.

Du point de vue organisation, les compétences sont organisées sous forme d'un graphe conceptuel. L'application des règles au moment de la publication facilite la recherche vu que les informations implicites seront présentes avant leur utilisation grâce à l'application de ces règles.

Finalement, et en point de vu utilisation de compétences, nous avons utilisé l'opération de projection pour rechercher des entités ayant un certain nombre de compétences requises. Cependant, les réponses considérées sont seulement de type oui/non. En effet, dans certaines applications, les réponses souhaitées sont de types coopératives; dans le sens où, si aucune entité ne satisfait à elle seule la requête; on essayera de trouver quel ensemble d'entités réunis la satisfaisant. Dans ce cas, la satisfaction conduit à plusieurs cas de figures [6]: (i) il existe des exportateurs satisfaisant complètement la requête; (ii) il existe des exportateurs satisfaisant partiellement la requête, mais en combinant les compétences de différents exportateurs la requête sera satisfaite; (iii) il n'existe ni un seul exportateur ni un ensemble d'exportateurs satisfaisant la requête. Dans ce dernier cas, le médiateur peut initier un processus de coopération avec les autres médiateurs pour essayer de satisfaire la requête (flèche (d) sur la figure 1). Ces différents cas sont traités dans [15] mais en utilisant le concept de complément dans DL. Une nouvelle version de notre travail est en cours de construction, elle prend en compte la fédération de médiateurs.

7 Références

1. G. Berio and M. Harzallah, "Knowledge management for competence management", j-jukm.J., Vol. 0, no 1, pp. 21—28, 2005.

2. G. Wiederhold, “Mediators in the architecture of future information systems”, IEEE Computer, Vol 25, no 3, pp. 38-49, 1992.
3. H. T Nacer, D. Aissani and N. Boudjlida, “ Semantic annotations for web services discovery and composition ”, Comput. Stand. nterfaces.J., Vol. 31, no 6, pp. 1108—1117, 2009.
4. N.Ikujiro and T. Hirotaka. “The knowledge creating company; how japanese companies creat the dynamics of innovation”, Oxford University Press, 1995.
5. DL-org. Description logics. Available: http//dl.kr.org/. Last consultation: December 2009.
6. N. Boudjlida. “A mediator-based architecture for capability management”, In Proc of the 6th IASTED International Conference, Software Engineering and Applications, SEA'2002. pp 45--50. MIT, Cambridge, USA, November 2002.
7. C.Dong, “Gestion et découverte de compétences dans des environnements hétérogènes”, Ph.D thesis, Henri Poincaré-Nancy1 Univ, France, 2008.
8. A.Borgida and P.Devanhu. “Adding more”DL” to IDL: Towards more knowledgeable component interoperability”, In 21rst Int. Conf. on Software Engineering, ICSE’99, pages 378—387, Los Angeles, CA, May 1999. ACM Press.
9. J. Sowa, “Conceptual structures: Information Processing in Mind and Machine”, Addison-Wesley, 1984.
10. UDDI. Universal Description, Discovery and Integration. Available: http://www.uddi.org/. Last consultation: September 2009.
11. M. Chein and M.L Mugnier. “Conceptual graphs: fundamental notions“,In Revue d'Intelligence Artificielle. J., Vol. 6, no 4, pp. 365--406, 1992.
12. M.L. Mugnier and M. Chein. “Représenter des connaissances et raisonner avec des graphes“. In Revue d'Intelligence Artificielle, Vol 7, no 1, pp 7-56, 1996.
13. E.Salavat, “Raisonner avec des opérations de graphes : graphes conceptuels et règles d'inférence“, PhD thesis, Montpellier II Univ, 1997.
14. E. Salvat, M. Mugnier, ”Sound and complete forward and backward chainings of graph rules”.in Proc of the 4th International Conference on Conceptual Structures, pp 248-262, 1996.
15. N.Boudjlida and C. Dong. “Complement concept and capability discovery “. In J. Grundspenkis and M. Kirikova, editors, Proc of the EMOI-INTEROP'04 (Enterprise Modelling and Ontologies for Interoperability), in connection with the 16th Conference on Advanced Information Systems Engineering, CAiSE'2004, Vol 4, pp 337--342, Riga, Latvia, Jun 2004.
16. M.L.Mugnier,“Contributions algorithmiques pour les graphes d’heritage et les graphes conceptuels”, Ph.D thesis, Montpellier II Univ, France, 1993.
17. CoGITaNT. “Conceptual Graphs Integrated Tools allowing Nested Typed graphs“. Available: http://cogitant.sourceforge.net. Last consultation: December 2009.
18. CGIF.“Conceptual Graphs Interchange Format“.Available: http://www.webkb.org/doc/CGs.html. Last consultation: January 2009.
19. Cogui.“Conceptual graphs graphical user interface“ .Available: http://www.lirmm.fr/cogui. Last consultation: January 2009.
20 M. Zacklad et M. Grundstein (eds), Ingénierie et capitalisation des connaissances. Hermès, 2001.

Interrogation d'une ontologie hybride en langage naturel : application au domaine médical

Chouaib Bouhalika, Zizette Boufaida

Laboratoire LIRE, Université Mentouri de Constantine, Algérie
bouhalika.chouaib@yahoo.fr, zboufaida@gmail.com

Résumé. L'interrogation d'une ontologie nécessite l'utilisation d'un langage de requêtes formel et difficile à maitriser, ce qui pose d'importantes difficultés pour les utilisateurs non-experts. Dans ce travail, nous proposons une approche qui vise à simplifier et à améliorer la recherche sémantique d'une manière générale et celle ayant trait au domaine médical. Ceci est rendu possible à travers la conversion des requêtes en langage naturel libre vers le langage de requête nRQL (new Racerpro Query Language), ce qui facilite la formulation des requêtes et améliore sensiblement l'exploitation de l'ontologie. A cet effet, nous combinons plusieurs outils et standards du web sémantique et de traitement du langage naturel, de façon à ce que le système développé soit plus robuste, pour la résolution des ambigüités linguistiques et des problèmes liés à la complexité et à la richesse d'expression des requêtes imposées par le domaine médical.

Mot clés : Recherche Sémantique, Ontologie Médicale, Interface en Langage Naturel, Traitement Automatique du Langage Naturel, Langage nRQL,

1 Introduction

La croissance très importante des informations disponibles sur Internet nécessite des outils de recherche de plus en plus performants permettant de discerner efficacement les informations intéressantes parmi des centaines voire des milliers de documents. Seulement, la qualité des résultats fournis par les moteurs de recherche traditionnels n'est pas toujours pertinente. Ceci est dû principalement aux ambiguïtés linguistiques et aux concepts abstraits qui ne sont pas bien traités, ainsi qu'à la complexité des langages d'interrogation. Donc aider un utilisateur à trouver l'information qu'il cherche dans ce contexte devient une tâche de plus en plus difficile. Le recours au langage naturel et aux différentes techniques et outils du traitement automatique de la langue naturelle (TALN) est une solution très intéressante. Par ailleurs, la prise en compte de la sémantique par le biais des ontologies est une solution pour la réduction des conflits sémantiques. Toutefois, afin d'acquérir les connaissances ontologiques, les utilisateurs doivent se familiariser avec le vocabulaire des langages des ontologies, les langages de requêtes**,** et la structure de l'ontologie utilisée. En outre, interroger une ontologie médicale impose d'importantes difficultés liées à la richesse et à la complexité du vocabulaire du domaine médical. Les termes utilisés sont souvent flous, imprécis et complexes.

C'est dans le but de combler ce fossé entre un web sémantique assez complexe et formel d'une part et le langage naturel simple et compréhensible par tous les utilisateurs d'autre part, et pour résoudre et pallier aux ambiguïtés par l'utilisation d'un vocabulaire médical, que nous proposons l'architecture d'un système de conversion des requêtes exprimées en langage naturel vers le langage de requête nRQL. Ce système permet à l'utilisateur de formuler ses requêtes en langage naturel libre sans se soucier de la structure ni de la syntaxe des requêtes nRQL adressées au système pour l'interrogation de l'ontologie médicale et sans avoir des connaissances préalables sur la structure de cette dernière, en apportant ainsi souplesse et facilité d'utilisation. C'est en s'appuyant sur les techniques du TALN et des ontologies et sur différentes mesures de similarités, ainsi que sur l'exploitation de Wordnet et UMLS, que le système est en mesure de régler toutes les conflits linguistiques et sémantiques qui peuvent survenir.

En plus la recherche ne se limite pas à trouver des ressources référencées par des mots clés, mais tente d'identifier la sémantique des mots de la requête et de représenter cette dernière formellement en logique de description afin de générer une requête nRQL plus pertinente.

Le reste du papier est organisé comme suit. La section 2 décrit les travaux existants. Nous présentons ensuite dans la section 3 l'architecture générale du système proposé et nous décrivons ses différents composants en détaillant le fonctionnement et le rôle de chacun. Dans la section 4, nous illustrons une étude de cas traité par ce système. Enfin, dans la section 5, nous terminons notre article par une conclusion et des perspectives.

2 Travaux existants

Les interfaces en langage naturel aux bases de connaissances, pouvant aider les utilisateurs ordinaires à exprimer leurs besoins d'information en langage naturel ont été étudiées pendant des décennies**.** Ces dernières années, les chercheurs se sont penchés sur l'étude des interfaces en langage naturel pour l'accès aux ontologies, car celles-ci délivrent les utilisateurs de la nécessité de connaître la structure de la base de données et offrent une interaction beaucoup plus commode et plus flexible.

- **ONLI [1]** (Ontology Natural Language Interaction) est un système d'interrogation d'ontologies en langage naturel, basé sur les restrictions sémantiques. Il suppose que l'utilisateur est familiarisé avec le domaine de l'ontologie. Ce systeme prend comme entrée une requête en langage naturel libre et la traduit en nRQL. Il génère ensuite les réponses pertinentes. Le système a été évalué sur l'ontologie de FungalWeb. La correspondance sémantique dans ce système est basée sur les prédicats des triplets linguistiques. Le système ignore les triplets vides, ce qui nuit à la pertinence des résultats attendus.

- **Querix [2]** est une interface en langage naturel indépendante du domaine et qui traduit les requêtes en langage naturel vers le langage SPARQL. En comparaison avec une interface en langage naturel fondée sur la logique, Querix n'essaie pas de résoudre l'ambiguïté de la langue naturelle, mais demande à l'utilisateur dans une fenêtre de dialogue de contribuer a la résolution de l'ambiguïté.

- **Aqualog [3]** est l'un des systèmes de Questions/Réponses (Q/R) les plus mûrs qui prend en entrée des requêtes exprimées en langage naturel et une ontologie et retourne des réponses tirées de la sémantique des données compatibles dans l'ontologie. Il combine plusieurs techniques puissantes pour donner un sens aux requêtes en langage naturel et les classe sous forme d'une représentation intermédiaire de triplets linguistiques et les reformule pour qu'ils soient compatibles avec les données de l'ontologie. Il est également soutenu par un mécanisme d'apprentissage, afin que sa performance s'améliore au fil du temps, en réponse au vocabulaire utilisé par les utilisateurs. Cependant, il exige que la requête soit complète et syntaxiquement correcte, et exprimée sous forme de questions (qui, quoi, etc.) et contenant jusqu'à deux triplets. De plus, il ne peut être utilisé que pour une seule ontologie.

- La nouvelle version de AquaLog est **PowerAqua [4]** PowerAqua est un nouveau système de Q/R qui fournit des réponses tirées de plusieurs ontologies, hétérogènes et distribuées sur le Web. Il a évolué à partir de Aqualog, afin de résoudre le problème de la limitation de Aqualog à une seule ontologie. PowerAqua prend en entrée des requêtes exprimées en langage naturel et les traduit en un ensemble de requêtes formelles. Il retourne enfin des réponses pertinentes tirées des ressources distribuées sur le web sémantique.

§ **SemSearch [5]** est un système basé concept avec une interface de requête similaire à Google. L'interface d'interrogation de SemSearch étend la recherche traditionnelle par mot clé, en permettant la spécification explicite de l'objet demandé et la combinaison de mots-clés. Le processus de recherche dans SemSearch, vise à trouver la signification sémantique des mots clés spécifiés dans la requête de l'utilisateur de telle sorte que le moteur de recherche sache ce que l'utilisateur recherche et la façon de satisfaire sa requête. Parmi les limites de SemSearch, on peut citer, la nécessité d'une bonne connaissance du domaine de l'ontologie et l'absence d'un moyen pour préciser la relation entre les termes recherchés, ce qui peut réduire la précision des résultats de ce dernier.

Le principal avantage du système proposé se situe dans la façon de réunir la simplicité d'utilisation de SemSearch et l'efficacité de traitement de Aqualog, c'est-à-dire donner aux utilisateurs plus de liberté lors de l'acquisition de leur requêtes, tout en préservant la performance du processus de conversion de ces requêtes vers nRQL. Donc, le système proposé est constitué d'une interface en langage naturel pour l'interrogation de l'ontologie, qui permet aux utilisateurs ordinaires de formuler leur requête en langage naturel libre sans aucune formation pré-requise. Le formalisme de représentation sous-jacent, à savoir les logiques de description, permet la formulation des questions supportant la quantification, la conjonction, la disjonction et la négation. Ceci permet ainsi d'apporter de la sémantique à la simple représentation par triplets utilisés par les systèmes précédents. Par ailleurs, pour une meilleure prise du domaine médical, nous utiliserons UMLS, en plus de Wordnet. Nous proposons également, un processus léger mais efficace pour le traitement des requêtes sur l'ontologie en faisant appel à des analyseurs performants tels que TreeTagger, Stanford et en exploitant différentes mesures de similarité.

3 Architecture du système

L'architecture proposée s'articule principalement sur six composants principaux : une interface utilisateur, un module d'analyse et de traitement des requêtes des utilisateurs, un module d'appariement, un module de traitement d'ontologies, un générateur de requêtes nRQL et un moteur d'inférence. (cf. figure 1).

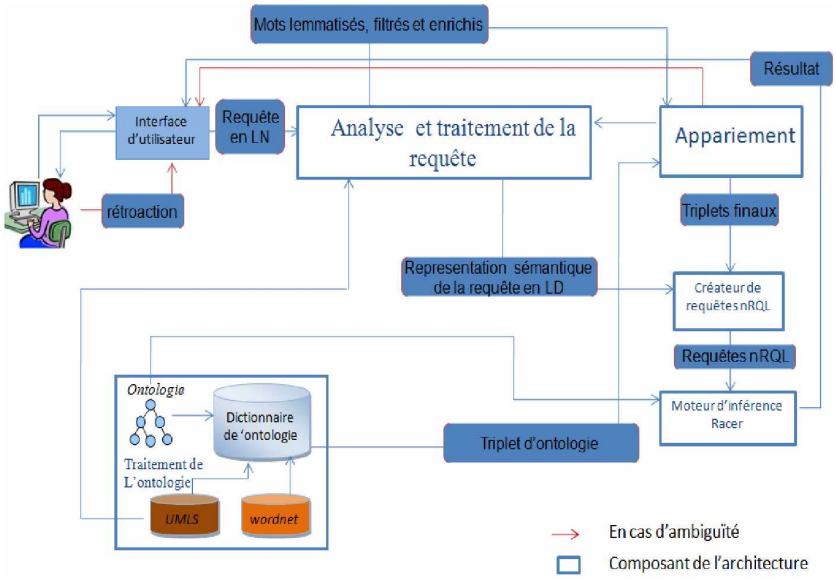


Fig.1. Architecture du système proposé

3.1 Composants de l'architecture

1. L'interface utilisateur

Permet aux utilisateurs de poser leur requête en langage naturel et de choisir l'ontologie à interroger. Elle permet également d'afficher les résultats et en cas de besoin, la requête nRQL générée. En plus, elle offre aux utilisateurs l'opportunité d'interagir avec le système à travers la rétroaction (feedback) pour la clarification du mot ambigu qui n'a pas trouver de correspondance avec les entités des triplets d'ontologies ou avec leurs synonymes.

2. Le composant de traitement de l'ontologie

Accepte en entrée une ontologie de domaine (RDF(S), OWL) qui encapsule plusieurs types d'information : taxonomie entre des concepts de même type modélisé par la relation is-a, le thésaurus des concepts de type différent liés par les relations Object_property et data_type_property, et des données formelles spécifiées par des axiomes.

A chaque fois qu'une nouvelle ontologie est chargée dans le système, le dictionnaire d'ontologie est automatiquement construit par l'extraction de toutes les entités de l'ontologie (classes, propriété, instances) et leur mettre sous forme de triplets « sujet, prédicat, objet » où le sujet et l'objet du triplet peuvent être soit des classes ou des instances, et le prédicat peut être une propriété. Nous avons opté pour cette représentation parce que la plupart des requêtes d'utilisateurs peuvent être compatibles avec la représentation avec des triplets.

Toutes les ontologies (RDF(S), OWL) peuvent être écrites sous forme de triplets. Cette représentation a l'avantage d'être extensible et paramétrable à volonté. Elle permet ainsi de décrire les choses simplement et sans ambiguïté. Donc, utiliser les triplets de cette manière, offre une structure de données très souple et flexible.

Afin d'augmenter la rapidité et l'efficacité de l'appariement des composants des triplets avec les mots de la requête, chaque composant du triplet sera simplifié pendant la reconnaissance et la séparation des mots qui utilisent les tirets/tirets bas et les mots Camel Case [9]. Le mot sera ensuite lemmatisé. Pour chaque composant de triplets, on identifie la liste de ses synonymes en faisant appel à WordNet [6]. Afin de bien cerner le domaine de l'ontologie médicale et d'augmenter la richesse et la précision du vocabulaire nous faisons appel à UMLS **(Unified Medical Language System) [7]**. Ce module fournit en sortie, une représentation de l'ontologie sous forme de triplets, dans lesquels chaque composant est lemmatisé, et enrichi par ses synonymes.

3. Le composant d'analyse et de traitement de la requête d'utilisateur

Supporte deux fonctionnalités essentielles :

Ø *Analyse morpho-syntaxique*: dans le but de pallier aux problèmes des variations morphologiques par la lemmatisation et l'identification de la catégorie grammaticale de chaque composant de la requête, nous utilisons l'outil TreeTagger [8].

L'étape de filtrage des résultats détenus consiste à éliminer les mots vides fréquents et inutiles, en s'aidant d'une liste préétablie des mots vides, pour ne pas filtrer ce qui est utile pour notre application.L'utilisation du filtre permet de réduire considérablement le nombre de termes extraits, et le traitement des situations exceptionnelles [9] par la reconnaissance et la séparation des mots qui utilisent les tirets/tirets bas, les camelcase, par exmple : QuickTime : quick time, suffer_from : suffer from.

Nous faisons également appel à UMLS pour résoudre certains problèmes liés au domaine médical comme la complexité et la grande variabilité de ses termes, et aussi pour éviter toute altération ou changement du sens de ces termes lors des différentes phases du traitement de la requête. A partir des résultats fournis par Tree Tagger, les termes de la requête ainsi que leurs catégories grammaticales, sont extraits en mettant l'accent sur les noms. Par ailleurs, l'outil MetaMap identifie les différents termes médicaux dans la requête ainsi que leurs catégories en utilisant le Métathesaurus et le réseau sémantique d'UMLS.

Les résultats envoyés au module d'appariement consistent en des termes de la requête lemmatisés, et filtrés ainsi que les termes médicaux identifiés et enrichis par leurs catégories.

Pour passer à l'étape suivante nous aurons besoin non seulement de reconnaitre la structure syntaxique de la requête mais aussi d'extraire les relations syntaxiques des questions. Ces relations définissent les dépendances grammaticales entre les mots de la requête. Dans ce but, nous utilisons l'analyseur syntaxique de Stanford[10]. Les arbres syntagmatiques produits par cet analyseur sont traduits dans des structures de dépendances par un module symbolique. La sortie finale est un graphe de dépendances dont les nœuds sont étiquetés avec les mots de la phrase analysée et dont les arcs sont orientés et étiquetés avec des relations grammaticales tels que objet, sujet, ou modificateurs. L'accent a été mis sur les phrases nominales, qui sont souvent beaucoup plus riches en sens que les groupes verbaux, et ils sont facilement reconnaissables à partir d'un arbre.

Ø *Représentation sémantique de la requête*: Un apport sémantique peut servir à réduire les ambiguïtés syntaxiques, mais sa finalité globale est de représenter formellement la requête de l'utilisateur. Dans l'intention de rapprocher la représentation de la requête exprimée en langage naturel à la requête formelle utilisée pour l'interrogation d'ontologie exprimée en nRQL (*new Racer* Query Language), nous avons opté pour les logiques de description (LD) qui sont une famille de langages de représentation de connaissances formels et structurés.

Le système suit une approche compositionnelle, pour élaborer la formule logique correspondante à la question d'entrée. Compositionnalité signifie ici que la signification de toute phrase est fonction des significations de ses parties. Cela veut dire que le sens de chaque unité lexicale non ambiguë est fixée dans le lexique, et que la combinaison de ces sens est guidée par la structure syntaxique de la phrase, et aboutit à l'interprétation de la phrase complète.
Une telle approche exige une sorte de traitement syntaxique par groupement de mots en des unités syntaxiques plus larges et en les ordonnant ainsi sous la forme d'arbre pour guider le calcul récursif. Ceci est accompli par l'analyseur de Stanford.
Le lien entre le langage naturel et les LD a été établi d'une façon formelle dans [11]. Cette connexion a été formalisée dans [12] en se basant sur deux observations:

- La sémantique du langage naturel a été formellement représentée par l'algèbre relationnelle [13].
- Le lien a été établi entre l'algèbre relationnelle et les LD [14].

Nous rappelons d'abord la représentation algébrique des constructeurs des LD comme présentée dans [14]. Nous montrons ensuite comment les algèbres relationnelles ont été utilisées pour l'analyse sémantique du langage naturel. Enfin, nous montrons comment l'approche algébrique est prise comme base pour trouver la représentation en LD. Pour mieux comprendre ce passage, nous reprenons les définitions cités dans [12].

- ***Passage de la LN vers l'algèbre relationnelle***

A partir de l'arbre syntaxique généré par l'analyseur de Stanford, nous extractons les règles de production qu'il contient (comme celles indiquées dans la deuxième colonne du tableau 1). Ensuite nous procédons à l'analyse sémantique de la requête par l'utilisation de l'algèbre relationnelle a travers l'annotation des représentations syntaxique avec des expressions algébriques en se basant sur le travail de P.Suppess [16].

Tableau 1. Associations sémantiques dans les grammaires relationnelles.

	Règle de production	Association sémantique
1.	P→ SN + SV	$[SN] \cap [SV] \neq \emptyset$
2.	P→ NP + SV	$[SN] \subseteq [SV]$
3.	SN →Art + N	$[SN] = [N]$
4.	SN→ Art + N + Adj	$[SN] = [N] \cap [Adj]$
5.	SV→ VT + SN	$[SV] = [VT] : [SN]$

Les symboles P, SN, SV, NP, N, Adj, Art and VT correspondent respectivement à : 'phrase', 'syntagme nominal', 'syntagme verbal', 'nom propre', 'nom', 'adjectif', 'article' et 'verbe transitif'. Pour plus de détails, référez au [15].
La représentation sémantique nous aide à définir et à extraire les relations entre les termes de la requête et entre les différents triplets finaux déjà établis.
Nous procédons ensuite à un parcours ascendant le long de l'arbre sémantique jusqu'au nœud racine, en exécutant le processus suivant :

- Identifier et extraire les valeurs de chaque nœud.
- Identifier la valeur qui lui correspond à partir des entités des triplets finaux valides.
- Remplacer les valeurs des nœuds par les entités des triplets correspondants
- Le procédé est réitéré jusqu'à la racine de l'arbre produisant la sémantique de la phrase entière.

- ***Lien entre les logiques de description et l'algèbre relationnelle***

Dans [16], La sémantique des opérateurs des LD peut être définie en termes d'opérations algébriques. Une interprétation $\mathcal{I}$ est une paire $(\Delta^{\mathcal{I}}.^{\mathcal{I}})$ où $\Delta^{\mathcal{I}}$ est le domaine de l'interprétation et $.^{\mathcal{I}}$ la fonction d'interprétation. Un concept C est interprété par l'ensemble $A^{\mathcal{I}} \sqsubseteq \Delta$ et un rôle r par une relation binaire $r^{\mathcal{I}}$ sur l'ensemble U.L'interprétation algébrique des descriptions de concepts de la LD est présentée dans la figure 2.

$$\top^{\mathcal{I}} = U$$
$$\bot^{\mathcal{I}} = \phi$$
$$(\neg A)^{\mathcal{I}} := (A^{\mathcal{I}})'$$
$$(\neg C)^{\mathcal{I}} = (C^{\mathcal{I}})'$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$
$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\exists r.C)^{\mathcal{I}} = r^{\mathcal{I}} : C^{\mathcal{I}}$$
$$(\forall r.C)^{\mathcal{I}} = (r^{\mathcal{I}} : (C^{\mathcal{I}})')'$$

Fig.2. Interprétation algébrique de quelques descriptions de concepts.

Le top et le bottom, les opérateurs de conjonction, disjonction et négation sont définis de la même manière qu'en LD. La quantification existentielle est assignée au produit de Peirce. Appliqué à une relation R et à un ensemble C, le produit de Peirce correspond à l'ensemble (1) :

$$R : C = \{x \mid \exists y : (x, y) \in R \wedge y \in C\} \quad (1)$$

La quantification universelle est assignée à une variante du produit de Peirce appelée involution (2) :

$$(R : C')' = \{x \mid \forall y : (x, y) \in R \Rightarrow y \in C\} \quad (2)$$

Les restrictions sur les nombres ne peuvent pas être exprimées algébriquement. Pour pallier à cette insuffisance, le travail présenté dans [15] augmente l'algèbre de Peirce avec les équivalents algébriques des opérateurs de quantification numérique. La figure 2 est extraite de la table décrite dans [14].

Donc, étant donné la correspondance exacte entre les opérations algébriques et les constructeurs de LD expliquée précédemment, le passage vers la représentation terminologique d'une phrase est direct.
La représentation finale est envoyée au constructeur de requêtes nRQL.

4. L'appariement

Consiste à combler le fossé sémantique entre la terminologie de l'utilisateur et le vocabulaire de l'ontologie. Ce composant a pour rôle d'apparier les termes lemmatisées et filtrées de la requête, et les termes médicaux identifiés avec les composants des triplets du dictionnaire d'ontologie enrichis par leurs synonymes. Les différentes étapes du processus d'appariement sont les suivantes :

1. Réception et sélection des lemmes des différents termes de la requêtes envoyés par le module d'analyse et traitement de la requête d'utilisateur en traitant en priorité les termes médicaux identifiés.

2. Chaque lemme extrait de la requête, est apparié avec les lemmes des composants des triplets du dictionnaire d'ontologie; Si ce lemme correspond à l'un des lemmes des composants d'un triplet du dictionnaire d'ontologie, le composant correspondant est extrait. Il est placé dans un nouveau triplet que l'on nomme triplet final. Puis pour chaque composant apparié nous vérifions si les composants restants du triplet sont présents dans la question et on procède à leurs appariement après on passe au prochain lemme de terme de la requête et ainsi de suite.

3. Si le lemme ne correspond à aucun lemme des composants des triplets du dictionnaire d'ontologie, nous devons donc chercher parmi ses synonymes tirées de WordNet. Les synonymes seront appariés selon un système d'annotation basé sur des mesures de similarité telles que celle basée sur la distance de Levenshtein et la mesure de similarité sémantique de LIN [17].

4. L'appariement des termes médicaux suit l'algorithme précédent, en respectant pour chaque terme, sa catégorie identifiée. Si le terme ne trouve pas de correspondance avec les composants des triplets nous faisons appel à UMLS pour lever l'ambigüité.

5. Dans le cas ou le lemme ne trouve aucune correspondance, le système permet a l'utilisateur de clarifier les mots ambigus à travers un feedback. Après avoir identifié tous les triplets finaux, ces derniers sont envoyés au composant générateur de requêtes nRQL. Pour ne garder que les triplets pertinents, nous procédons au calcul de la somme des scores des distances de Levenshtein qui mesure la similarité entre les lemmes des entités du triplet et les lemmes de la requête. Nous choisissons ainsi ceux qui ont marqué le plus d'appariement selon une valeur allant de 0/3 jusqu'à 1.

5. La génération des requêtes nRQL

La représentation sémantique de la requête générée par le module d'analyse et de traitement de la requête utilisateur est traduite en une requête nRQL en exploitant les triplets finaux produits par le module d'appariement. Une requête nRQL se compose d'une tête et d'un corps : (retrieve <head> <body>).

La tête de la requête spécifie le format de la réponse et le corps de la requête contient des atomes (unaires ou binaires) de requête qui sont utilisés pour spécifier les conditions de récupération sur les liaisons des variables de requête. Les requêtes nRQL sont soit simples ou complexes. Une requête nRQL simple se compose d'un seul atome dans le corps de la requête. Exemple d'une requête nRQL simple : la requête (retrieve (?x) (?x patients)) a la tête (?x) et le corps (?x patient). Elle retourne tous les individus de patient.

Par contre, une requête nRQL complexe se compose de deux ou plusieurs atomes de requête qui sont combinés entre eux à l'aide des constructeurs de corps de la requête (par exemple, and, union, neg).

Pour générer la requête nRQL, nous faisons appel à la représentation sémantique de la requête d'utilisateur, plus les triplets identifiés. La syntaxe et la sémantique de langage nRQL sont décrites en détail dans [18].

Trouver la cible (les mots qui correspondent aux résultats recherchés par la requête nRQL résultante) de la requête, nous aide à savoir quel genre de réponse peut être attendue et donc à mieux spécifier les requêtes nRQL. Ce processus est le suivant :

D'abord trouver les mots interrogatifs autorisés [19] comme « what », « who », et « how » ou les verbes de commande comme «find», «give » puis prendre les noms qui sont a revoir si en peut les changer par suivants comme cibles, Exemple : Give me the name …..
Le verbe *"give"* signifie que son objet (*"name"*) doit être sélectionné comme étant la cible. Le verbe lui-même ainsi que son agent (ici le *locuteur*, ou *"me"*) sont éliminés dans la phase de filtrage. Les règles détaillées varient selon les différents mots interrogatifs / de verbes de commande, et sont généralement communs pour différents domaines.

Dans notre travail, la requête nRQL est déterminée par quelques règles de génération en se basant sur les triplets et la représentation sémantique de la requête en LD.
Enfin, le système supprime les doublons qui sont sémantiquement équivalents et transmet la requête nRQL au moteur d'inférence RACER.

6. Le moteur d'inférence

Pour exécuter la requête nRQL générée, notre système utilisera le raisonneur RACER, et jena comme couche d'accès au dictionnaire d'ontologie.

4 Etude de cas

Pour valider les phases précédentes, nous avons utilisé l'ontologie médicale qui formalise quelques informations du domaine de la médecine préventive.

1. L'utilisateur entre la requête : « who are the patients suffering from macrocytic anemia» avec le nom de l'ontologie médicale.

2. Analyse morpho-syntaxique à travers l'utilisation de l'analyseur Treetagger qui procède à la lemmatisation et à l'attribution d'une catégorie grammaticale à chaque mot de la requête :

who	WP	who
are	VBP	be
the	DT	the
patients	NNS	patient
suffering	VVG	suffer
from	IN	from
macrocytic	JJ	macrocytic
anemia	NN	anemia

Où les symboles WP, VBP, DT,NNS, VVG, IN, JJ et NN, correspondent respectivement à : wh-pronoun/ verbbe, sing. present, non-3d/ determiner/ Noun, plural/verb, gerund/ participle/IN preposition, subordinating conjunction/JJ adjective/NN noun, singular or mass.

3. Le filtrage (suppression des mots vides, fréquents et inutiles) et identification des termes médicaux à travers l'utilisation de l'outil MetaMap et UMLS. On sait à partir de Wh-questions who que la cible de la requête est une personne (patients).

4. L'analyse de la requête à travers l'utilisation de l'analyseur de stanford.

```
(ROOT
  (SBARQ
    (WHNP (WP who))
    (SQ (VBP are)
      (NP (DT the) (NNS patients))
      (VP (VBG suffering)
        (PP (IN from)
          (NP (JJ macrocytic) (NN anemia)))))))
```

5. Construction de la représentation algébrique et transformation des expressions algébriques obtenues en expressions de logiques de description (3) :

$$\text{Patient} \sqcap \exists \text{suffer}.(\text{macrocytic} \sqcap \text{anemia}). \quad (3)$$

6. Appariement entre les termes de la requête et les composants des triplets de dictionnaire de l'ontologie médicale, en faisant appel à Wordnet et à UMLS:
   < patient, suffer_from, anemia >,< AnemiaMacrocytic , is_a, anemia>

7. Substitution des termes de la requête utilisés dans la représentation sémantique en logique de description de la requête par les composants correspondants des triplets finaux (4) :

$$\text{Patient} \sqcap \exists \text{suffer\_from}.(\text{anemia} \sqcap \text{AnemiaMacrocytic}) \quad (4)$$

8. Traduction en nRQL (5) en appliquant les règles de génération :

```
(Retrieve (?x) (and (?x patient anemia suffer_from)
     (AnemiaMacrocytic anemia is_a) ))
```
(5)

6 Conclusion

Dans cet article nous avons proposé l'architecture d'un système qui agit en tant qu'interface en langue naturelle pour l'interrogation d'ontologies, en mettant l'accent sur son processus de conversion de requêtes en langage naturel vers Nrql. L'interface prend en entrée des requêtes exprimées en langage naturel et les convertit vers nRQL, elles seront exécutés par le moteur d'inférence RACER, épargnant ainsi l'utilisateur d'apprendre un langage formel complexe, et à connaître rigoureusement la structure de l'ontologie. Les travaux futurs consisteront à voir comment surmonter la limitation de notre système à une seule ontologie et l'extension de la représentation à des langages de LD plus expressifs.

Références

1. Kosseim, L., SibliniI, R., Baker, C., et Bergler, S.:"Using Selectional Restrictions to Query an OWL Ontology". International Conference on Formal Ontology in Information Systems (FOIS 2006)
2. Kaufmann, E., Bernstein, A., Zumstein, R.: "Querix: A natural language interface to query ontologies based on clarification dialogs". In: 5th International Semantic Web Conference (ISWC 2006), Springer (November 2006).
3. Lopez, V., Pasin,M., and Motta, E.: "AquaLog: An Ontology-Portable Question Answering System for the Semantic Web", In: Proceedings European Semantic Web Conference (ESWC), Heraklion, Crete, Greece, (2005).
4. Lopez, V., Motta, E., Uren, V.:"Poweraqua: Fishing the semantic web". In: Proceedings European Semantic Web Conference (ESWC), Montenegro (2006).
5. Lei, Y., Uren, V., Motta, E.: "Semsearch: a search engine for the semantic web". In: Managing Knowledge in a World of Networks, Springer Berlin / Heidelberg (2006).
6. Miller, G.:"Wordnet: A lexical database". Communication of the ACM, 38(11):39--41, (1995).
7. *Lindberg, C.: The Unified Medical Language System* (UMLS) of the National Library of Medicine. 61(5): 40-2; JAm Med Rec Assoc (1990).
8. Schmid, H.: "Probabilistic Part-of-Speech Tagging Using Decision Trees", Conference on New Methods in Language Processing, (1994).
9. D. Damljanovic, V. Tablan, and K. Bontcheva. A text-based query interface to owl ontologies. In 6th Language Resources and Evaluation Conference (LREC), Marrakech, Morocco, (May 2008).
10. Marneffe, M.-C., Manning, C. D.: The Stanford typed dependencies representation, COLING Workshop on Cross-framework and Cross-domain Parser Evaluation, (2008).
11. SCHMIDT, R. A. :Terminological Representation, Natural Language Relation Algebra, GWAI, p. 357-371, (1992).
12. Naouel, K. : Comparaison sémantique de textes en langage Naturel,Une approche par les logiques de description, Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (2003).
13. BOETTNER, M.: Natural language, C. BRINK, W. K., SCHMIDT, G., Eds., Relationnal methods in computer science, Advances in Computing, Springer,Wien, p. 226–246, (1997).
14. SCHMIDT, R. A.: Algebraic Terminological Representation, rapport n_ MPI-I-91-216, Saarbrucken, (1991).
15. SCHMIDT, R. A.: Relational Grammars for Knowledge Representation, (1997).
16. SUPPES, P.: Direct inference in English, Teaching Philosophy 4, p. 405–418, (1981).
17. Yang, D., Powers, D. M. W.: "Measuring semantic similarity in the taxonomy of wordnet". In ACSC '05 : Proceedings of the Twenty-eighth Australasian conference on Computer Science, Darlinghurst, Australia, Australia, pp. 315–322. Australian Computer Society, Inc, (2005).
18. RacerPro User's Guide Version 1.9. Racer Systems GmbH & Co. KG, http://www.racer-systems.com December, (2005).
19. Quirk, R., et al. In: A Comphrehensive Grammar of the English Language. Longman, London (1985)

Nouvelle version d'une mesure de similarité pour un meilleur calcul de la distance sémantique entre concepts d'une ontologie

Abdeslem DENNAI[1], Sidi Mohammed BENSLIMANE[2]

[1]Laboratoire FIMAS Université de Béchar, Algérie,
[2]Laboratoire EEDIS Université Sidi Bel Abbes, Algérie
[1]De_selam@yahoo.fr, [2] Benslimane@univ-sba.dz

Résumé. *Les méthodes de calcul de la similarité sémantique se retrouvent dans diverses applications, avec comme objectif de donner à ces dernières des connaissances supplémentaires afin d'effectuer un raisonnement adéquat sur leurs données. Le choix d'une telle mesure de similarité est tout à fait crucial pour une bonne exécution de ce raisonnement. Dans ce papier, nous présentons une mise à jour ou plutôt une nouvelle version d'une méthode de calcul de similarité présentée par Wu et Palmer qui est considérée comme la plus rapide en termes de temps de génération de la similarité. Les résultats obtenus montrent bien que la mesure produite assure une amélioration de la pertinence des valeurs produites pour la similarité de deux concepts dans une ontologie.*

Mots clés : Ontologie, mesure de similarité, distance sémantique, web sémantique, association sémantique.

1 Introduction

L'identification de la similarité a été considérée comme un sujet de recherche fortement recommandé dans les domaines du Web sémantique, de l'intelligence artificielle et de la littérature linguistique. Le choix d'une mesure de similarité est tout à fait crucial pour une bonne exécution du raisonnement [3]. Il s'agit en effet de trouver la meilleure adéquation entre le but à atteindre et le type de connaissances manipulées. L'identification de la similarité entre les données issues de l'extraction et les concepts d'une ontologie de domaine est une phase fondamentale dans une approche de rétro-ingénierie et qui est adoptée par plusieurs techniques telles que le regroupement, la fouille de données (data mining), le Web sémantique et en particulier, le domaine de la recherche de l'information. Cette dernière repose largement sur des mesures pour l'identification de la similarité entre les documents [2] [24].

La majorité des approches de la recherche de l'information ne prennent en compte que des mots simples et/ou des fragments des mots pour la recherche des documents

pertinents et ignorent l'idée essentielle qui prend en compte les rapports ontologiques des mots.

Ces derniers peuvent être détectés par un processus de calcul de similarité entre des paires d'objets. Dans le domaine du Web sémantique où les ontologies interviennent pour la modélisation des connaissances, la mesure de [25], par exemple, a l'avantage d'être simple à implémenter et d'avoir aussi de bonnes performances comparativement à d'autres mesures de similarité [16].

En fouillant dans les différentes méthodes de mesure de similarité existantes, on peut déduire la limite de ces méthodes dans quelques domaines d'application ce qui nous a conduit à faire une synthèse générale de ces méthodes, achevée par notre contribution dans la mise à jour d'une méthode de calcul de similarité entre les concepts d'une ontologie.

Le reste du papier est organisé comme suit : La section 2 consiste à présenter quelques usages d'application de la mesure de similarité, la section 3 cite un état d'art des travaux déjà réalisés dans le domaine de calcul de similarité et enfin, en section 4 nous présentons notre contribution avant de conclure dans la section 5.

2 Usages d'application

Pour justifier l'importance de la mesure de similarité, nous présentons ici ses quelques domaines d'application.

2.1 Web Services

La détermination de la similarité des services sémantiques permet d'obtenir des informations utiles concernant leurs comptabilités. Dans le travail de [11], il y a une proposition des métriques pour mesurer la similarité des services sémantiques annotés avec une ontologie OWL. La mesure de similarité proposée est basée sur l'intuition que les objets similaires partagent les informations descriptives les plus communes.

2.2 Traitement automatique de la langue (TAL)

Plusieurs travaux sur la mesure de similarité ont été motivés par le traitement automatique de la langue (TAL). Parmi les travaux dans ce domaine on peut citer : le travail de [20] qui utilise la métrique de la similarité sémantique pour la mesurer entre tous les sens de mot d'une paire donnée de mots et les désambiguïser ainsi dans un contexte donné. [18] a combiné l'utilisation d'un thesaurus automatiquement acquis à partir des corpus textuels bruts et Wordnet (basé sur la métrique de la similarité) pour trouver des sens prédominants des mots dans des textes non structurés. Les auteurs du travail [7] ont appliqué les mesures de similarité sémantique de WordNet pour évaluer la pertinence des expressions, étant donné un dialogue spécifique, et de construire automatiquement les sommaires du dialogue parlé.

2.3 Bioinformatique

Une variation de mesure de similarité basée sur le contenu informationnel est adoptée pour trouver une meilleure façon d'organiser et d'interroger les données d'une ontologie de gène (GO). GO a été créé en 1998. GO résulte d'une collaboration entre plusieurs bases de données: FlyBase (drosophile), the Saccharomyces Genome Database et des bases de données de génomes (homme et souris), etc.
GO comprend 3 parties axées sur :

- La fonction moléculaire (fonction des gènes exprimés ex: ATPase activity).
- Le processus biologique (rôles biologique généraux de fonctions moléculaire complexes ex: la mitose).
- les composants cellulaires (structures subcellulaires, localisation des complexes macromoléculaires ex: le noyau, le télomère).

Le travail de [17] s'intéresse à la similarité sémantique entre les protéines, plutôt que les termes de l'ontologie GO, c'est pour cette raison qu'il a combiné entre trois mesures de similarités [22] [16] [12].

Remarque 1. Notre contribution est démontrée dans ce domaine en utilisant un extrait de cette ontologie de gène (GO) dans la dernière partie de ce travail.

3 Taxinomie des techniques de mesure de similarité

Dans cette section nous classifions les principales approches de mesure de similarité. La table 1 (voir section 3.5) montre une partie de notre arrangement de classification ainsi que quelques techniques d'échantillon.

3.1 Techniques basées sur les arcs

La mesure de similarité la plus intuitive des objets dans une ontologie est leurs distances [21] [14] [25]. Evidemment, un objet X est plus similaire à un objet Y qu'un objet Z. Cette similarité est évaluée par la distance qui sépare les objets dans l'ontologie.

Ces mesures se servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts. Le calcul des distances dans l'ontologie est basé sur un graphe de spécialisation des objets. Dans chaque graphe, la distance de l'ontologie doit être caractérisée par le plus court chemin qui fait intervenir un ancêtre commun ou le plus petit généralisant (PPG), connectant potentiellement deux objets à travers des descendants communs. Parmi les travaux classifiés sous cette bannière on peut citer :

3.1.1 Mesure de Wu & Palmer

La mesure de [25] a été utilisée par [8] pour organiser des documents web dans des clusters. Elle a aussi servi dans [1] pour évaluer la proximité sémantique de deux

concepts d'une page HTML relativement à un thésaurus dans le cadre d'une indexation d'un site web par des ontologies.

La mesure de similarité de [25] est basée sur le principe suivant :
Etant donnée une ontologie formée par un ensemble de nœuds et un nœud racine R (Root) (voir figure 1). Soit C1 et C2 deux éléments de l'ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (D1+D et D2+D) qui séparent les nœuds C1 et C2 du nœud R et la distance (D) qui sépare le concept subsumant ou le PPG de C1 et de C2 du nœud R.

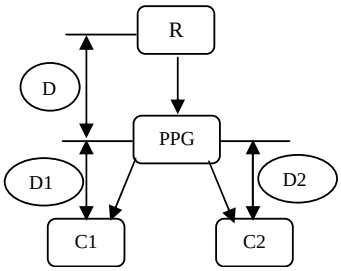


Fig. 1. Exemple d'un extrait d'une ontologie.

La mesure de Wu et Palmer est définie par la formule suivante :

$$\boldsymbol{SimWP(C1, C2)} = \frac{\boldsymbol{2 * D}}{\boldsymbol{D1 + D2 + 2 * D}}$$

[16] a effectué une comparaison entre les méthodes des mesures de similarité. Il en ressort que la mesure de Wu et Palmer [25] a l'avantage d'être simple à calculer en plus des performances qu'elle présente, tout en restant aussi expressive que les autres.

3.1.2 Mesure de Rada et al

Cette mesure [21] est adoptée dans un réseau sémantique et elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques «is-a». Pour calculer la similarité de deux concepts dans une ontologie, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure, basée sur le calcul de la distance entre les nœuds par le chemin le plus court, présente un moyen des plus évidents pour évaluer la similarité sémantique dans une ontologie hiérarchique.

3.1.3 Mesure d'Ehrig et al

Un travail de mesure de similarité pour les ontologies a été introduit par [4]. Ce travail introduit trois couches : les données, l'ontologie et le contexte. La similarité des entités est mesurée au niveau des données en considérant les valeurs de données de type simple ou complexe (entiers, caractères). Les relations sémantiques entre les entités sont mesurées au niveau de la couche de l'ontologie. Finalement la couche du contexte spécifie comment les entités de l'ontologie sont utilisées dans un certain contexte externe, plus spécifiquement, le contexte de l'application.

3.2 Techniques basées sur les nœuds

Ces techniques adoptent une nouvelle mesure en termes de la mesure entropique de la théorie de l'information [16] [23]. La probabilité P(.) pour l'identification de l'utilisation d'une classe ou de ses descendants dans un corpus désigne l'information de la classe. On définit l'entropie d'une classe par la formule suivante :

$$\boldsymbol{E(c) = -\log\,(P(c))}$$

Où P est la probabilité de trouver une instance du concept c. La probabilité d'un concept c est calculée en divisant le nombre des instances de c par le nombre total des instances. En associant des probabilités aux concepts d'une taxonomie, il est possible d'éviter le manque de fiabilité des distances des arcs. Plus l'information est partagée par deux concepts, plus ils sont similaires. Parmi les travaux, recensés dans la littérature, sous cette bannière on peut citer :

3.2.1 Mesure de Resnik

La notion du Contenu Informationnel (CI) a été initialement introduite par [22] qui a prouvé qu'un objet (mot) est défini par le nombre des classes spécifiées et que la similarité sémantique entre deux concepts est mesurée par la quantité d'information qu'ils partagent. Pour évaluer la pertinence d'un objet, il faut calculer le contenu informationnel. Le contenu informationnel est obtenu en calculant la fréquence de l'objet dans le corpus (Wordnet). La formule proposée par Resnik est définie par:

$$\boldsymbol{SimR(C1,C2) = Max[E(CS(C1,C2))] = Max[-\log(p(CS(C1,C2))]}$$

Où CS (C1, C2) représente le concept le plus spécifique (qui maximise la valeur de similarité) qui subsume (situé à un niveau hiérarchique plus élevé) les deux concepts C1 et C2 dans l'ontologie. Cette mesure est un peu sommaire car elle ne dépend que du concept le plus spécifique.

3.2.2 Mesure de Lin

Lin a défini une mesure de similarité légèrement différente de celle de Resnik :

$$\boldsymbol{SimL(C1,C2) = \frac{2 * log\,(P(AC(C1,C2)))}{log(P(C1)) + log\,(P(C2))}}$$

Cette mesure utilise une approche hybride qui combine deux sources de connaissances différentes (Thesaurus, corpus). En plus, elle représente la similarité comme degré probabiliste de chevauchement des concepts descendants de C1 et C2. Les travaux de [19] ont évalué cette mesure à travers une expérience qui utilise des sujets humains pour évaluer la similarité entre 30 paires de noms, il en ressort que cette méthode offre une amélioration significative.

3.2.3 Mesure de Hirst

L'idée de cette mesure est que deux concepts lexicalisés sont sémantiquement étroits si leurs ensembles synonymes (synsets) de WordNet sont reliés par un chemin qui n'est pas trop long et qui "ne change pas la direction trop souvent". Avec cette mesure,

toutes les relations contenues dans un réseau Wordnet sont prises en considération. Dans le travail de [10], les auteurs ont classé la direction des liens en lien haut (superclasse), lien bas (sous-classe) et lien horizontal (antonyme). Le calcul de la similarité, avec cette méthode, s'effectue entre objets (mots) par le poids du plus court chemin allant d'un terme à un autre, en plus des classifications qui indiquent les changements de direction. La force du rapport est donnée par :

$$SimH = T - PCC - K * nd$$

Où T et K sont des constantes, PCC est la distance du plus court chemin en nombre d'arc et nd le nombre de changements de direction.

3.3 Techniques hybrides

Ces techniques sont fondées sur un modèle qui combine entre les approches basées sur les arcs (distances) en plus du contenu informationnel qui est considéré comme facteur de décision.

3.3.1 Mesure de Jiang et Conrath

Pour remédier au problème présenté au niveau de la mesure de Resnik, [12] a apporté une nouvelle formule qui consiste à combiner l'entropie (Contenu Informationnel) du concept spécifique à ceux des concepts dont on cherche la similarité (combine entre les techniques basées sur les arcs et les techniques basées sur les nœuds qui consistent à compter les arcs afin d'améliorer les résultats par des calculs basés sur les nœuds). La mesure adoptant cette méthode est basée sur la combinaison d'une source de connaissance riche (thesaurus) avec une source de connaissance pauvre (corpus).
Notons que cette formule est définie par l'inverse de la distance sémantique.

$$SimJC(C1, C2) = \frac{1}{Distance(C1, C2)}$$

Sachant que la distance entre C1 et C2 est calculée par la formule suivante :

$$Distance(C1, C2) = E(C1) + E(C2) - (2 * E(CS(C1, C2)))$$

3.3.2 Mesure de Leacock et Chodorow

Une autre méthode présentée par [15] qui combine entre la méthode de comptage des arcs et la méthode du contenu informationnel. La mesure proposée par Leacock et Chodorow [15] est basée sur la longueur du plus court chemin entre deux synsets de Wordnet. Les auteurs ont limité leur attention à des liens hiérarchiques «is-a» ainsi que la longueur de chemin par la profondeur globale P de la taxinomie. La formule est définie par :

$$SimLC(C1, C2) = -\log\left(\frac{cd(C1, C2)}{2 * M}\right)$$

Où M est la longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas. On dénote par cd (C1, C2) la longueur du chemin le plus court qui sépare C1 de C2.

3.4 Techniques basées sur l'espace vectoriel

Dans le domaine de la recherche de l'information, les modèles de l'espace vectoriel sont largement adoptés [2] [24]. Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité en se basant sur la mesure de cosine ou la distance euclidienne. Le modèle de l'espace vectoriel est employé pour un arrangement des objets complexes en les représentants comme des vecteurs de k-dimensions. La définition de la similarité entre deux vecteurs d'objets est obtenue par leurs contenus internes. Parmi les approches citées dans la littérature on peut citer :

3.4.1 Similarité de Jaccard

La mesure de similarité de Jaccard est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs :

$$\boldsymbol{SimJ}(\boldsymbol{X},\boldsymbol{Y}) = \frac{\boldsymbol{x} * \boldsymbol{y}}{||\boldsymbol{x}||_2^2 + ||\boldsymbol{y}||_2^2 - \boldsymbol{x} * \boldsymbol{y}}$$

Tels que x et y sont des vecteurs extraits à partir des concepts C1 et C2.

$||\boldsymbol{x}|| = \boldsymbol{\varphi}^{\boldsymbol{x}}{}_{\mathbf{i=1}}$ désigne la norme du vecteur x et $||\boldsymbol{x}||_2 = \sqrt{\sum_{i=1}^{i=n} |\boldsymbol{x}_i|^2}$.

3.4.2 Similarité de Cosine

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets (documents) sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs x et y et le produit de la norme de x et de y.

$$\boldsymbol{SimC}(\boldsymbol{X},\boldsymbol{Y}) = \mathbf{cos}(\boldsymbol{X},\boldsymbol{Y}) = \frac{\boldsymbol{x} * \boldsymbol{y}}{||\boldsymbol{x}||_2 * ||\boldsymbol{y}||_2}$$

La mesure de Cosine quantifie donc la similarité entre les deux vecteurs comme le cosinus de l'angle entre les deux vecteurs.

3.4.3 Similarité euclidienne

La similarité euclidienne est basée sur le ratio de la distance euclidienne augmenté de 1. La distance euclidienne est définie par la formule suivante : $\boldsymbol{dE} = ||\boldsymbol{x} - \boldsymbol{y}||_2$
La mesure de similarité est donc définie par : $\mathbf{Sime}(\mathbf{C1},\mathbf{C2}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{dE}}$

3.4.4 Similarité de Dice

La similarité de Dice [16] est définie par le nombre des objets communs multipliés par 2 sur le nombre total d'objets. Cette mesure est donc définie par la formule :

$$\boldsymbol{SimD}(\boldsymbol{C1},\boldsymbol{C2}) = \frac{\mathbf{2} * \boldsymbol{x} * \boldsymbol{y}}{||\boldsymbol{x}||_2^2 + ||\boldsymbol{y}||_2^2}$$

3.5. Synthèse des techniques de mesure de similarité

Nous reprenons ici, sous forme d'une table, les techniques de mesure de similarité ainsi leurs références :

<table>
<tr><td colspan="4">Techniques de mesure de similarité</td></tr>
<tr><td colspan="3">Similarité sémantique : Utilise des relations partielles</td><td>Rapport sémantique : Utilise la totalité des relations</td></tr>
<tr><td colspan="2">Techniques hybrides</td><td>Techniques basées sur l'espace vectoriel</td><td rowspan="4">[10]
[20]
[18]
[7]
[9]</td></tr>
<tr><td>Techniques basées sur les arcs</td><td>Techniques basées sur les nœuds</td><td rowspan="2">Calcul vectoriel</td></tr>
<tr><td>Comptage des arcs</td><td>Théorie de l'information</td></tr>
<tr><td>[12]
[15]</td><td>[16]
[23]</td><td>Jaccard
Cosine
Dice
Euclide</td></tr>
</table>

Table 1. Taxinomie des techniques de mesure de similarité.

4 Notre contribution

4.1 Exposé du problème

La mesure de Wu & Palmer est intéressante mais présente une limite car elle vise essentiellement à détecter la similarité entre deux concepts par rapport à leur distance de leur PPG. Plus ce subsumant est général, moins ils sont similaires (et inversement). Cependant, elle ne capte pas les mêmes similarités que la similarité conceptuelle symbolique. Ainsi on peut avoir Sim(A, f) < Sim(A,B), f étant un des fils de A et B un des frères de A. Ce qui est à notre sens inadéquat dans le cadre de recherche d'information où il faut ramener tous les fils d'un concept (i.e requête) avant son voisinage. Cette mesure présente l'avantage de la rapidité du temps d'exécution, mais l'inconvénient de la production d'une valeur de similarité de deux concepts voisins qui dépassent la valeur de deux concepts dans la même hiérarchie.

On prend comme référence la figure 2 qui représente un graphe représentant une partie d'une hiérarchie des concepts d'une ontologie de gènes en biologie. Les concepts contenus dans cette ontologie représentent intuitivement un ensemble des distances conceptuelles variées s'ils sont comparés deux à deux.

A titre d'exemple, le concept « cellular process » et « cellular component organization » présentent une valeur de similarité égale à 0 dans le cas de l'utilisation des mesures de similarité traditionnelles qui incluent des informations externes à la hiérarchie telles que les mesures de [5] [6]. Par contre, l'adoption d'une approche reposant sur l'hiérarchie donne une valeur de similarité différente de 0 pour ces deux mêmes concepts. En plus, la valeur de la similarité des deux concepts « cellular

process » et « cellular component organization » est moins élevée que celle des concepts « cellular process » et « cell cycle ».

Cependant, nous jugeons que le concept « cellular process » est plus proche au concept « cell cycle » que le concept « cellular component organization ». Ces précisions sont très intéressantes pour la recherche des similarités sémantiques d'un ensemble de concepts contenus dans une ontologie. Ces distances intuitives peuvent être utilisées, par exemple, pour l'amélioration des moteurs de recherche au niveau de l'efficacité et de la précision des réponses aux requêtes clients. La structure la plus simple supportant le raisonnement sur l'hiérarchie des types est celle que l'on peut trouver dans un support de graphes conceptuels. Dans cette structure, les liens de subsomption groupent les types suivant les caractéristiques définitionnelles qu'ils partagent.

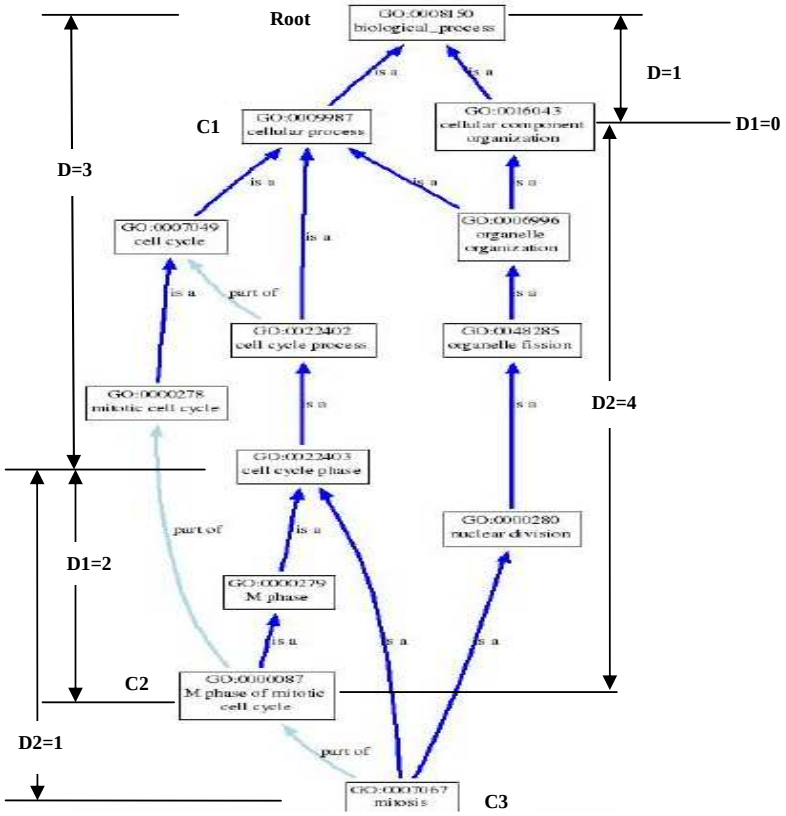


Fig. 2. Graphe représentant une partie d'une hiérarchie des concepts d'une ontologie de gènes en biologie.

4.2 Solution proposée

A titre d'exemple, on peut obtenir avec la mesure de Wu et Palmer, une valeur de similarité entre le concept « M phase of mitotic cell cycle » et « mitosis» qui dépasse la valeur de similarité entre « cellular process » et « M phase of mitotic cell cycle ». Cependant, cette mesure offre une similarité plus élevée entre un concept et son voisinage par rapport à ce même concept et un concept fils (voir exemple d'application ci dessous).

Soit l'ontologie de la Figure 2, on dénote par C1, C2 et C3 respectivement les concepts « cellular process », « M phase of mitotic cell cycle » et « mitosis ». En appliquant la mesure de Wu et Palmer, la valeur de similarité est ainsi calculée :

$$SimWP(C1, C2) = \frac{2 * 1}{0 + 4 + 2 * 1} = 0.33$$

$$SimWP(C2, C3) = \frac{2 * 3}{2 + 1 + 2 * 3} = 0.66$$

Les valeurs obtenues par la mesure de Wu et Palmer montrent que les concepts voisins C2 et C3 sont plus similaires que les concepts C1 et C2 situés dans une même hiérarchie ce qui est inadéquat dans le cadre de la recherche des informations sémantiques.

Nous proposons une nouvelle mesure qui mettre à jour la mesure de Wu et Palmer, dont l'expression est représentée par la formule suivante :

$$SimDB(C1, C2) = \frac{2*D}{D1+D2+2*D+FPP\_PPG(C1,C2)}$$ **, avec :**

$$FPP\_PPG(C1, C2) = \begin{cases} \textbf{0} & \textbf{Si C1 est ancêtre de C2 ou inversement.} \\ \textbf{(D+D1)*(D+D2)} & \textbf{Si C1 et C2 sont voisins par un CS.} \end{cases}$$

FPP_PPG (Fonction Produit de Profondeurs par un Plus Petit Généralisant) est une fonction qui permet de pénaliser la similarité de deux concepts voisins qui ne sont pas situés dans une même hiérarchie. Dans le cas de concepts voisins, ***FPP_PPG*** donne la distance en nombre d'arcs égale au produit de profondeurs des deux concepts par rapport à la racine de l'ontologie (root) en passant par un CS). De plus en plus que les distances D ou D_i (où D est la distance entre le CS et la racine et D_i représente la distance entre un concept C_i et son CS) sont éloignés, de plus en plus SimDB diminue. Avec cette fonction, la mesure de similarité entre deux concepts hiérarchiques est plus élevée que la similarité entre deux concepts voisins par un PPG (CS).

4.3 Exemple d'application

Prenant le même exemple précédent avec les mêmes concepts C1, C2 et C3. En appliquant notre mesure et la mesure de Wu et Palmer, les valeurs de similarité entre C1 et C2 et entre C2 et C3 sont indiquées dans les tables ci-après :

Cas : Concepts hiérarchiques			
C1	C2	**SimWP**	**SimDB**
cellular process	M phase of mitotic cell cycle	**0.33**	**0.33**

Table. 2. Valeurs de similarité calculées par Wu & Palmer et notre mesure (SimDB).

Cas : Concepts voisins			
C2	C3	**SimWP**	**SimDB**
M phase of mitotic cell cycle	Mitosis	**0.66**	**0.20**

Table. 3. Valeurs de similarité calculées par Wu & Palmer et notre mesure (SimDB).

$$\boldsymbol{SimDB(C1, C2) = \frac{2 * 1}{0 + 4 + 2 * 1 + 0} = 0.33}$$

$$\boldsymbol{SimDB(C2, C3) = \frac{2 * 3}{2 + 1 + 2 * 3 + ((3 + 2) * (3 + 1))} = 0.20}$$

4.4 Propriétés de notre mesure

Soit trois concepts C1, C2 et C3 d'une ontologie quelconque. Voici quelques propriétés vérifiées par notre mesure.

- Le non négativité : SimDB (C1, C2) ≥ 0.
- L'identité : SimDB (C1, C1) = SimDB (C2, C2) = SimDB (C3, C3) = 1.
- La symétrie : SimDB (C1, C2) = SimDB (C2, C1).
- L'unicité : SimDB (C1, C2) = 1 ➔ C1 = C2.
- Le différent : SimDB (C1, C2) = 0 ➔ C1 ≠ C2.
- Intervalle de définition : SimDB (C1, C2) ⊂ [0..1].

4.5 Comparaison de notre mesure avec celle de Wu et Palmer

L'objectif de ce papier est de mettre en œuvre et tester une mise à jour d'une méthode de mesure de similarité pouvant faire avancer les recherches dans le domaine des ontologies et de simulation des distances conceptuelles. L'ontologie sur la quelle nous avons fait ces mesures est une ontologie de domaine biologique gene ontology[1] (voir figure 2). Cette ontologie est utilisée pour décrire la fonction moléculaire, le processus biologique et les composants cellulaires. Elle a été crée avec l'éditeur Protégé-OWL qui permet aux utilisateurs de construire des ontologies pour le web sémantique en OWL.

Dans une ontologie OWL, chaque objet est décrit par certains rapports RDF [13]. Soit O un objet dans une ontologie OWL. O est caractérisé par un ensemble de descriptions qui contient tous les rapports qui lui décrit. Un ensemble de description

[1] http://www.geneontology.org/

pour O est défini par : Descr(O) = {(s, p, o) ϵ O}. Où s, p et o est un triplet RDF qui dénote le sujet (Subject), le prédicat (Predicate) et l'objet (Object). RDF (Resource Description Framework) est aujourd'hui utilisé comme un standard pour l'échange des métadonnées entre différentes applications. Il permet de faciliter le travail des moteurs de recherche pour chercher les documents d'une manière efficace.

Pour vérifier la validité de notre mesure, il est judicieux de tester sa pertinence de calcul par rapport à la mesure de Wu et Palmer qui a été jugée la plus rapide en termes de temps de génération de la similarité [16]. L'impact de la modification de la mesure de Wu et Palmer ainsi que l'aboutissement à notre mesure doivent être évalués pour juger sa pertinence.

Dans les tables 4 et 5, on a choisi une représentation par paires de concepts contenus dans l'ontologie afin de calculer les valeurs de similarités. Le calcul s'effectue respectivement par la mesure de Wu & Palmer et par notre mesure.

La table 4 concerne des concepts hiérarchiques tandis que la table 5 examine des concepts voisins.

Concepts Hiérarchiques		SimWP	SimDB
C1	C2	(C1, C2)	(C1, C2)
cellular process	M phase of mitotic cell cycle	0.33	0.33
cellular process	cell cycle	0.66	0.66
cell cycle phase	M phase	0.54	0.54
cellular process	cell cycle process	0.66	0.66
cell cycle phase	mitosis	0.54	0.54

Table. 4. Représentation par paires de concepts (cas : Concepts hiérarchiques).

Concepts Voisins		SimWP	SimDB
C2	C3	(C2, C3)	(C2, C3)
M phase of mitotic cell cycle	mitosis	0.66	0.20
cell cycle	cell cycle process	0.50	0.25
M phase	mitosis	0.75	0.25
cell cycle process	organelle organization	0.50	0.25
Mitosis	M phase of mitotic cell cycle	0.66	0.20

Table. 5. Représentation par paires de concepts (cas : Concepts voisins).

La pertinence de notre mesure par rapport à la mesure de Wup & Palmer est localisée au niveau de deux concepts situés dans une hiérarchie dont le concept subsumant[2] est différent. De plus en plus que la distance entre les concepts subsumant directs est éloignée de plus en plus que la valeur de la similarité diminue. Une comparaison entre la pertinence des valeurs trouvées dans la table 5 est

[2] Un concept C1 est subsumé par C2 si C2 est le père et C1 est le fils.

représentée par la figure 3. Les résultats obtenus montrent qu'il y a une augmentation de la pertinence apportée par notre mesure.

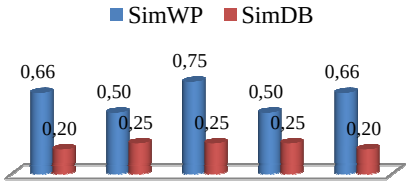


Fig. 3. Pertinence de notre mesure (Cas: Concepts voisins)

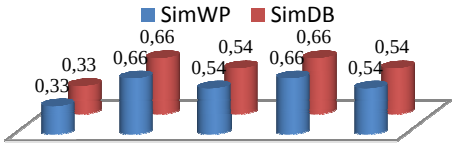


Fig. 4. Mesure Wu & Palmer inchangée par notre mesure (Cas: Concepts hiérarchiques)

Remarque 2. La table 4 et la figure 4 montrent que notre mesure n'a apporté aucun changement sur la mesure de Wu et Palmer dans le cas des concepts hiérarchiques.

5 Conclusion

Dans ce travail, nous avons présenté une mise à jour de calcul de similarité présentée par Wu et Palmer. On a comparé notre mesure avec celle de Wu et Palmer considérée comme la plus rapide. Les résultats obtenus montrent bien que la mesure produite, assure la pertinence des valeurs produites pour la similarité de deux concepts.

La pertinence de cette mesure s'accroît, en plus, dans le cas d'une ontologie hiérarchique qui présente des liens « is-a » ce qui permet de donner une précision plus claire pour les relations. Ceci peut être adopté dans le domaine de l'identification des associations sémantiques où les approches actuelles portent sur les associations ne donnant pas une précision sur le degré de justesse d'une association.

6 Références bibliographiques

1. Desmontils E. & Jacquin C, ''Des ontologies pour indexer un site Web'', *Dans actes des journées francophones d'Ingénierie des Connaissances, IC' 2001*.
2. Baeza-Yates R. et B.Ribeiro-Neto, ''Modern Information Retrieval'', *ACM Press, Addison-Wesley*: New York, Harlow, England Reading, Mass., 1999.

3. Bisson G. La similarité: une notion symbolique/numérique. *Apprentissage symbolique-numérique (tome 2), Editions CEPADUES*, 2000.
4. Ehrig M., P.Haase, M.Hefke et N.Stojanovic, Similarity for ontology-a comprehensive framework, In Workshop *Enterprise Modelling and Ontology: Ingredients for Interoperability*, 2004.
5. Eiter T. et H. Mannila, ''Distance measures for point sets and their computation'', *In Acta Informatica Journal*, 34, 1997.
6. Green J., N.Horne, E.Orlowska et P. Siemens, A Rough Set Model of Information Retrieval, *Theoretica Infomaticae 28*, pp 273-296, 1996.
7. Gurevych I. et M. Strube, Semantic similarity applied to spoken dialogue summarization, In Proceedings of the *20th International Conference on Computational Linguistics*, Geneva, Switzerland, 23 -27, pp. 764-770, 2004.
8. Halkidi M. & Nguyen B & Varlamis I. & Vazirgiannis M, Thesus: ''Organising Web Document Collections based on Semantics and Clustering, *Journal on Very Large Databases, Special Edition on the Semantic Web, Novembre 2003*
9. Hirst G. et A. Budanitsky, Correcting real-word spelling errors by restoring lexical cohesion, *Natural Language Engineering*, 2004.
10. Hirst G. et D.St Onge, Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum (editor), *WordNet: An electronic lexical database*, Cambridge, MA:The MIT Press .1998.
11. Jeffrey H., L. William et D. John, A Semantic Similarity Measure for Semantic Web Services. In proceedings of WSS05. 2005.
12. Jiang J. et D. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research in Computational Linguistics, Taiwan, 1997.
13. Klyne G. et J.Carroll, Web services description language (wsdl)1.1. http://www.w3.org/TR/rdfconcepts/, 2004.
14. Lee J.H., M.H.Kim et Y.J.Lee, ''Information Retrieval Based on Conceptual Distance in IS-A Hierarchy'', *Journal of Documentation* 49, pp. 188-207, 1993.
15. Leacock C. et M. Chodorow, *Combining Local Context and WordNet Similarity for Word Sense Identification, In WordNet: An Electronic Lexical Database*, C. Fellbaum, MIT Press, 1998.
16. Lin D., An Information-Theoretic Definition of similarity, In Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98), Morgan-Kaufmann: Madison, WI, 1998.
17. Lord P.W., R.D. Stevens, A. Brass et C.A.Goble, *Semantic Similarity Measures as Tools for Exploring the Gene Ontology. Pacific Symposium on Biocomputing 8*, pp.601-612, 2003.
18. Diana M., R. Koeling, J. Weeds et J. Carroll, Finding predominant senses in untagged text. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp. 280 – 287, 2004.
19. Miller G. A., R. Beckwith, C. Fellbaum, D. Gross, et K. Miller, Introduction to WordNet: An On-line Lexical Database. Cognitive Science Laboratory, Princeton University, Princeton, Technical Report 1993.
20. Siddharth P., S. Banerjee et T. Pedersen, Using measures of semantic relatedness for word sense disambiguation. In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, pp. 241-257. 2003.
21. Rada R., H. Mili, E. Bichnell et M. Blettner, Development and application of a metric on semantic nets, IEEE Transaction on Systems, Man, and Cybernetics: pp 17-30. 1989.
22. Resnik P., Using information content to evaluate semantic similarity in taxonomy. In Proceedings of 14th International Joint Conference on Artificial Intelligence, Montreal, 1995.

23. 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, 11:95-130, 1999.
24. Salton G. et M. J.McGill, Introduction to modern information retrieval. McGraw-Hill. New York, 1983.
25. Wu Z. et M. Palmer, Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp 133-138. 1994.

Toward an Efficient and Scalable Architecture Based on SKOS Ontology for Resource Discovery in Grid

Nabila Chergui, Salim Chikhi

MISC laboratory
Mentouri University
Constantine, Algeria
chergui.nabila@gmail.com, slchikhi@yahoo.com

Abstract. Grid technology enables the sharing and collaborating of a wide variety of resources. To fully utilize these resources, effective discovery techniques are necessities. The efficiency of these techniques depends on two major factors: the architecture of the system and the mechanism of querying adopted for this system, and the structure of representation of both information resources and queries. This paper focuses on the architecture of the system and query processing mechanism. It introduces a comprehensive semantic federations' cluster into a hybrid grid in which intra-federation adopts centralized management and inter-federation form a distributed one. In the process of constructing the federations and three layers overlay network, a new construction method based on Semantic Web technologies using a **SKOS** (***S**imple **K**nowledge **O**rganization **S**cheme)* lightweight ontology for describing domains of applications in grid computing, and for semantically handles queries is presented. And an efficient process to leaders' federations' election is discussed.

Keywords: Resource discovery, SKOS, Semantic federation clustering, Query processing.

1 Introduction

Grid computing is emerging as an infrastructure to provide collaborative resource sharing over multiple geographically distributed organizations [1]. In this, Resource discovery mechanism is one of the fundamental requirements for grid computing systems, as it aids in resource management and applications scheduling. An efficient resource discovery mechanism depends on two factors, first, the architecture of the system and hence the mechanism used to query processing across this architecture; second, the structure used to represent resources and consequently, the structure of queries.

In this work, we focus on the first factor of resource discovery; in this issue, various kinds of solutions to grid resource discovery have been proposed, the centralized one is efficient, although has some limitations, firstly, the central server in this architecture has a single point of failure, secondly, it can create a bottleneck due to the sent messages to a single point which may render such a system poor scalability. The introduction of P2P and DHT techniques into grids brings some benefits like adaptation, self organization and fault tolerance; however, they are less efficient and have several shortages such as a risk of network congestion and

overhead, due to the sent messages for updating dynamic data on resources and while finding a resource, and the risk of churn effect if a large number of nodes want to update their data at the same time.

To address these problems, we propose a mechanism based on semantic nodes clustering into federations using a **SKOS** ***(Simple Knowledge Organization Scheme)*** [2] lightweight ontology to regroup nodes having the same domain of interest and to process queries between nodes, which performs an effective searching according to the semantic distribution of nodes into federations and thus their resources. We will use a hybrid architecture composed of three layers, which combines the advantages offered by the above types of architectures. Clustering, is the most popular technique for creation of hybrid overlay networks [3], the main aim of clustering is to keep such desirable properties of distributed and centralized architectures [3]. Our approach supports a semantic organization of nodes in a hybrid way, and uses semantic to handle queries between federations. By the integration of Semantic Web techniques and hybrid architecture, this approach speeds up the information query and it guaranties the scalability and the flexibility of the system.

By analogy to the real-world, countries are gathered in federations according to their interests. Under one federation, these last countries work on collaboration, share their resources to achieve their objectives. We draw inspiration from this to organize nodes. Nodes in the grid environment correspond to countries on the real-world; each node has a domain of interest such, mathematics, biology. To construct our federations, we need to extract the domain of interest from node, and then a measure of semantic similarity is applied to affect the node to its adequate federation.

The remainder of this paper is organized as follows: Section 2 presents related works in this topic. Section 3 provides in a clear manner an explanation of the system, the construction of semantic federations based on SKOS and the three layered architecture. Section 4 proposes algorithms to explain how query processing is semantically performed, the leader is elected and federations are maintained. Theoretical performance evaluation is given in Section 5. Conclusion and future works are provided in Section 6.

2 Related Works

In the literature, it exists many different approaches addressing the problem of resource discovery on grid environment; we can classify them on non semantic approaches and semantic ones.

In non semantic approaches we find, centralized techniques like Condor [4], which uses a matchmaker with a centre server to match search resources; it has a single point of failure and scale poorly. P2P techniques like in [5] organize information nodes into an unstructured P2P network and random-walk based methods are used for query forwarding. Random-walks are not efficient in response time for a very large system. [6] Proposes a hierarchical structure to organize nodes to reduce redundant messages. However, the global hierarchy is hard to maintain in a dynamic environment.

Semantic techniques are those that use Semantic Web technologies. Semantic Web [7] attempts to define the metadata information model for the World Wide Web

to aid in information retrieval and aggregation. It improves the effectiveness of resource and query representation and the efficiency of searching. [8] Is a P2P network for searching Semantic Web metadata. Each peer can make its metadata information available as a set of RDF statements. The distributed peers register the queries they may be asked through the query service, then queries are sent through the network to the subset of peers who have registered with the service to be interested in this kind of query. To process queries between nodes, it uses JXTA to broadcast queries to a HyperCup topology. Similarly, [9, 10] also use broadcast/flooding to search semantic metadata. The P2P broadcast used by these systems makes them very difficult to scale to large-scale networks. Our system solves this problem by topology adaptation and semantics-based routing.

Semantic clustering or semantic hybrid approaches have appeared with the idea of grouping nodes with similar contents together to facilitate searching. [11, 12] use a centralized server or super-peers to cluster nodes. However, the efficient communication mechanism between super-peers is absent in these systems. [13] Proposes to cluster nodes with similar interest together into communities, without discussing how to define the interest similarity among peers and how to form clusters. [14, 15] add semantic short-cuts to group nodes. The short-cut approach relies on the presence of interest-based locality. Each peer builds a shortcut list of nodes that answered previous queries. To find content, a peer first queries the nodes on its shortcut list and only if unsuccessful, floods the query. [16] Uses semantic clustering to organize the network topology and reduce search space to semantically related clusters; instead it uses a complex and costly mechanism to construct clusters, each time it needs to add new node to the system.

3 Overview of the System

This section illustrates how to provide efficient construction of federations, and gives a detailed explanation of the system architecture.

3.1 Semantic Federations Construction Based on SKOS OntDD

As we have mentioned above, the computing grids can create federations in scientific domains, such as physics, earth science, mathematics; each federation is formed by a collection of nodes with the same domain of interest because we believe that more nodes share the same interest, more their resources tend to be similar. A federation is managed by a leader and consists of members that serve as workers. Communication and collaboration can operate on top of the federations. With federations, grid users can easily share resources and knowledge within the federation.

To create grid federations, we need a classification technique to classify nodes. Since each node has a specific domain of interest, **Ont**ology of **D**omains **D**escription **OntDD** is used to classify grid domain applications in general. This ontology is a lightweight ontology; we used SKOS [2] vocabulary, SkosEd editor [17], Skos API [18] and Protégé 4 [19] to formalize it.

SKOS is used to represent term lists and controlled vocabularies. It provides a simple machine-understandable; Technologies such as RDF and OWL [20] are seen as key elements for building a Semantic Web. The SKOS model is built in accordance with these technologies and has a serialization to the Resource Description Framework (RDF). In general, KOS differs significantly from formal ontologies, as represented using OWL, as they do not contain detailed intentional descriptions of concepts [21] SKOS provides looser semantics than OWL [22].

The SKOS model can be used to structure and represent any knowledge that contains statements about concepts and the relationships between them. The shared features of these KOS are primarily in the form of a lexical resource along with some semantic relationships between each resource. The semantic relationships between resources are typified by broader, narrower, and related. SKOS provides a data model that can be used to express these kinds of relationships between resources and is designed to be extensible and modular. Central to SKOS is the core vocabulary deemed sufficient to represent most of the common features found in concept schemes. A *Concept* can be considered any unit of cognitive thought. *Lexical labels* allow the association of lexical forms (preferred labels, alternative labels and so on) with each concept. *Semantic relations* capture relationships between concepts, including hierarchical broader-narrower relationships and general associative relationships [23]. For example, a domain "*Biology*" is an individual of *Skos:Concept* in this ontology. We refer to individuals in this work by concepts. "*Biology*" may have subbed domains like "*Ecology*" and "*Botany*". Each of them is an individual of *Skos:Concept* too. They are related to "*Biology"* by narrower (more specific) and broader (more general) relations. We consider each concept as one federation.

The following example, demonstrates how the concept "*Ecology"* is defined using SKOS in OntDD, the example is encoding in Turtle [24] notation.

```
<#Ecology>
 a skos:Concept;
 skos:altLabel "Bionomics"@en, "Environmental
 science"@en;
 skos:broader <#Biology>;
 skos:definition "the branch of biology concerned with
 the relations between organisms and their
 environment."@en;
 skos:narrower <#Paleoecology>;
 skos:prefLabel "Ecology"@en, "Ecologie"@fr;
```

We enhance and enrich the semantic meaning of concepts by the creation of different alternate label relations to represent its synonyms, and another property assertion called *associated_term,* which represents terms associated to the concept other than its synonyms.

A classification technique will classify each node according to its interest into a concept of OntDD; it means it will affect each node to the appropriate federation. We believe that if concepts are well defined, the use of a simple measure of similarity will be efficient and precise. Concepts in OntDD are already enriched by their semantic

synonyms according to their context extracted from *WordNet* [25]; The *Levenshtein* measure will be used to calculate similarity between the domain of interest of node *N* wishes to join the grid and each federation *F* of OntDD. The similarity between *N* and *F* is the max of similarities between *N* and the set of all labels and associated terms of *F*. The node will be affected to the federation with the high similarity.

After a classification algorithm has been determined, the system can classify nodes and create federations. Fig. 1 shows the system architecture.

3.2 Layered Architecture based Semantic Federation

As illustrated in Fig. 1, we propose a hybrid layered architecture in which, each federation is structured following leader-workers paradigm to perform data retrieval of available resources.

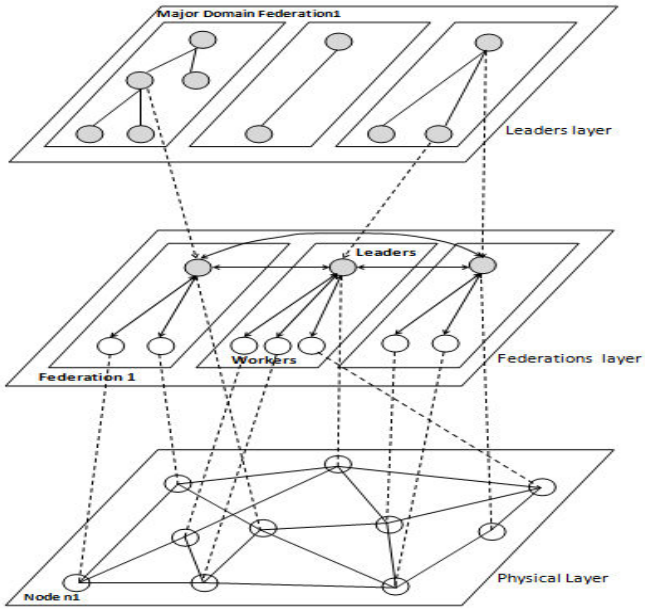


Fig. 1. The hybrid layered architecture.

From the bottom to the up, we have:

1. The physical layer: represents nodes in a real network as unstructured network architecture. Edges in this layer represent physical connections.
2. The federations' layer: in this layer, we adopt a centralized management, it represents the overlay network applied in this work to maintain federations and process queries, each federation captures a concept defined in OntDD, and has a leader and workers, a leader is a representative of its federation, which is selected

among the other nodes, each leader has links to all the other leaders, and has links to all of its own worker nodes, communication is limited on two sorts, between leaders and between a leader and its workers, which reduce the overhead.

3. The leaders' layer: since each concept in OntDD represents one node, which is the leader of the federation, leaders can be organized as a hierarchical structure. This hierarchy between leaders' federations is generated by traversing the configured properties *Skos:narrower* and *Skos:broader*, which are used to express the hierarchical relations between concepts in our case federations' leaders. In addition with SKOS we could organize federations (which are individuals of *Skos:Concept*) into categories using subclasses of *Skos:Concept*, we called them *MajorDomainFederation.* A category serves as grouping mechanism for concepts (leaders' federations) of the same inherent category, concepts being instances of one of those categories. E.g.: "*MajorBiologyFedation"* is a category of all federations with their domain is one field of *Biology*.

This hierarchy organization aids to limit the query search space from the entire grid to a federation through a single step and resource location inside the federation in the next step. If the federation is unable to respond to the query, it forwards the query to its relative hierarchy leaders that may satisfy the query. This forwarding mechanism between leaders' federations achieves high resource discovery efficiency by keeping resource discovery scope at the federation leader level.

4 Algorithms for the System

In this section, we will present the algorithmic details of this system. We will discuss in detail how federations are maintained in terms of add and remove nodes. How a leader is elected and how queries are handled.

4.1 Nodes Joining and Leaving

Algorithm 1 Join (*N*, *X*): Node *N* joins the Grid through node *X*

```
If X.isleader = true then
    Calculate the similarity between N and concepts of OntDD;
    Assign it to its appropriate federation;
  If N ∈ this-fedration then
     Update (addition) knowledge base of resources;
  Else
    If N ∈ other-federation then
      Send a message to its leader's federation;
    Else /* there is no adequate match*/
      N creates a new federation with itself as leader;
    End if
  End if
 Else /* X is not a leader*/
     Transfer a subscribe message to the leader;
End if
```

When a node joins the network, it connects to any existing node in the network by sending a subscribe message, if this last node is not a leader, it transfers the request to its leader. The leader calculates the similarity between a domain of interest of node and concepts of OntDD, and then assigns it to the appropriate federation following the Algorithm 1.

To leave the grid, the node just sends a message of unsubscribing to its leader. If the leader wants to leave, a replacement of a leader occurs by selecting a new leader to preserve the federation's knowledge, and then the leader can unsubscribe.

4.2 The Process of Leader's Election

A good resource discovery mechanism based on leader-workers should be able to select the best node to be a leader, to periodically check if the actual leader is the most pertinent and to prevent leader failure to make the mechanism fault-tolerant. It should have the ability to detect the failure and to replace dynamically a failed leader. It exists several works on the leader election problem. [26] Proposed an election method where each node is assigned a unique steady ID and the node with the highest ID wins the election. The stability of ID even if the resources' node changes may render the current leader performances less than existing nodes. [27] Used a voting based system, where each node casts a vote for which node it prefers for the leader role. In such a system, a mechanism for determining when the election begins and ends must be designed, since distributed election algorithms depend on a clearly defined exchange of information between nodes in order for each node to unanimously agree on the new leader [28, 29]. [30] Used a distributed mechanism of election where all nodes should agree on the future leader, it doesn't handle the case when a leader node was failed, and it uses gossip messages to elect a leader and to inform all the other leader's groups about a new leader in order to update their information. This method is very expensive.

These mechanisms may generate a lot of traffic, particularly if the elections need to be restarted due to corrupted packets of intermittent network failures. A central leader election algorithm [31] is more striking to distributed leader election, since the leader choice is performed by a single node. In our approach, we will adopt this mechanism.

Previous to describe the election method used in this work, we first define functions associated to each leader. It is important to denote that the hierarchical structure of leaders doesn't imply any hierarchy in their functions; on the contrary, each leader has to accomplish the following tasks:

1. It receives and publishes resource information periodically of and to its workers;
2. It receives, sends and formulates query messages;
3. It manages the OntDD, it receives messages from new nodes wish to enter the grid, and affects them to the appropriate federation.
4. It designates the future leader and safeties nodes by the periodic execution of the algorithm of leader election.

The election of leaders takes place periodically to check if the current leader is the most suitable. At the first time, the node that triggered the creation of the federation will be a leader of this federation. Later, each node in the federation can participate to

this process; it has to calculate its proper reputation score. Every time the process is started nodes send their reputation score to the leader. This last selects the three nodes with the highest scores. The leading will be the new leader, and the others will become the safeties' nodes, then it informs the whole federation about them. Safety nodes act as workers. They are introduced to be as a secure in case where the actual leader was failed or want to leave. This mechanism avoids performance degradation and federation dissolution if a leader fails, because the whole system has already prepared its future leaders, which makes the system fault tolerant. The leader sends copies of information to safety nodes every time it makes an update. Once the failure of a leader is detected, a safety node with the high reputation score will be a new leader. The reputation score is calculated based on nodes' characteristics such as stability (it means that node doesn't leave or fail frequently) and the interne characteristics like CPU speed, RAM and HDD sizes, and bandwidth.

To overcome the problem of communication overhead between leader's federations each time a new leader is elected. We give for each federation, independently on the real address of its leader, a virtual static address, using another asserted property into OntDD to assign to federations their virtual addresses. The leader then must associate its own address to the virtual one of its federation. The virtual address is stable, if a new leader is elected; all it has to do is to assign its proper address to the virtual one of the federation, without needing to communicate and to publish it to the other leaders, for them the address of the leader federation is not modified, even if the leader was changed.

4.3 Semantic Query Processing Mechanism

The mechanism used in this work, uses OntDD to semantically propagate the query between semantic related federations. It decides where the query must be sent in the next step using the different semantic relationships seen in Section 3. This mechanism divides the space of query search on three spaces. It limits the query search space from the entire grid to these three spaces.

1. Space 1: it represents the federation itself, the leader and its workers. In this space the leader supports the search of resources in respond of the query in its knowledge base.
2. Space 2: it represents federations inside the MajorDomainFederation. Where federations are related by hierarchical relationships narrower/broader and they are members of the same MajorDomainFederation. These federations are likely capable to respond to query since they have close domains of interest as the leader who sent the query.
3. Space 3: it represents federations related to the actual federation by the associative relationship *Skos:related*. In SKOS, an associative link between two concepts indicates that the two are inherently related, but that one is not in any way more general than the other.eg. "*Business"* is related to "*Statistics"*. "*Biology"* is related to "*Medicine"* and "*Business"*. These federations are possibly capable to respond to the query because their domains of interest are related to the leader who sent the query.

Algorithm 2 Handle_Query (Q): Q is sent to leader L from node n

```
If n.isWorker = false then
    Find node(s) N in the federation that satisfy Q;
    If N ≠ φ then
        The search is succeeded; send a response to a requester;
    End if
Else /* n is one worker of L*/
    Find node(s) N in the federation that satisfy Q;
    If N ≠ φ then
        The search is succeeded; send a response to a requester;
    Else /* no node was found*/
        Forward the query, direct Q to leaders' in MajorDomainFederation;
        Wait responses for a time T;
        If T=0 and no response then
            Forward the query, direct Q to the related leaders';
            Wait responses for a time T';
            If T'=0 and no response then
                The search is failed;
            Else /*one or more related leader could satisfy the query*/
                The search is succeeded; send a response to a requester;
            End if
        Else /*one or more leader from the MajorDomainFederation could satisfy the
                query */
            The search is succeeded; send a response to a requester;
        End if
    End if
End if
```

These three layers of search spaces achieve high resource discovery efficiency by keeping resource discovery scope at the federation layer and its related leaders, and reduce the network traffic and the number of messages compared to the query flooding, or random-walk. A query request is submitted to a leader node from one of its workers or another leader node. The leader follows two behaviors depending on the source of the request, with its workers it tries to find in its federation a worker node able to satisfy the query based on the leader's knowledge. If such a worker is not available, the leader sends the query to leaders' federations in its MajorDomainFederation, which they are likely to satisfy the query. If it doesn't receive any response after a while, it forwards the query to its related leaders' federations as the last resort. If a leader is solicited, it tries to find workers that respond to the request; otherwise it ignores the query. This strategy of semantic query processing reduces the search time and decreases the network traffic by minimizing the number of messages circulating among nodes and federations. Algorithm 2, resumes this strategy.

5 Performance Evaluation

In this section, we present a theoretical study to evaluate the performance efficiency and scalability of our algorithm of query processing. With semantic federations' topology, resource discovery can be efficiently performed. In most cases, a resource can be located, within querying nodes with the same domain, and semantically related nodes that are within the neighborhood of the querying node in terms of related federations and MojorDomainFederation.

Supposing the number of grid nodes is N and the resource searched is r. We divide the whole grid according to our algorithm into M federations, K MajorDomainFederations, each MajorDomainFederation has P federations, L nodes per federation, Q related federations (if they exist) for each federation, and R messages as responses on queries if they exist.

We evaluate the performance of our algorithm by comparisons with flooding based algorithm, [32], Random walk and [33]. We evaluate these algorithms by estimating the number of messages propagated in the network, and the number of hops needed to find a resource during one cycle of searching, and we discuss the theoretical efficiency of each one.

With flooding-based, node X that searches for a resource r checks its resource list, and if the resource is not found there, X contacts all its neighbors. In turn, X's neighbors check their resource lists and if the resource is not found locally, they propagate the search message to all their neighbors. The method ends when either the resource is found or a TTL is expired, in this case, the number of messages increases exponentially to the number of nodes. The number of hops is estimated as $T>>N$, thus it is not scalable.

With randomize walk strategy in pure P2P model, the number of nodes visited during the searching process is $\log(N)$, the number of hops is $O(\log(N))$, so a good performance is expected; However, this kind of algorithm is slow and no guarantee of actually finding the resource even if it exists, thus not efficient.

For [32], it uses a super-peer topology.

1. Best case: one hop with one message. The resource is inside the requested cluster.
2. Average case: 1+ O (log (P)) hops, as it uses a random walk for searching. The number of messages is logarithmic in the number of clusters P in the super cluster called Resource Classified Space, P corresponds to the number federations per MajorDomainFederation in our case.
3. Worst case: 1+ O (log (K)) + O (log (P)) hops. In this stage, it uses a random walk and chord algorithms for searching. The number of messages is logarithmic in the number of clusters P in the super cluster, and the number of nodes K in the routing table of entry nodes, K corresponds to the number MajorDomainFederation in our case.

For [33], it uses a super-peer topology; it organizes nodes into groups with a leader-worker approach using KNN algorithm.

1. Best case: one hop with one message. The resource is inside the requested group.

2. Worst case: two hops with $1+M+R$ messages. It forwards the query to all the other groups in the grid. M is the number of groups in the grid; it corresponds to the number of federations in our approach.

Our strategy divides the space of search on three; this will conduct us to these situations:

1. Best case: one hop with one message. The resource is inside the requested federation.
2. Average case: two hops with $1+P+R$ messages. The resource is inside the MajorDomainFederation.
3. Worst case: three hops with $1+P+Q+R$ messages. The resource is in the related federations.

By comparing the estimated performance efficiency (number of messages and hops) of several algorithms, we could assume that our algorithm theoretically outperforms the flooding-based algorithm, randomize walk algorithm and algorithms presented in [32] and [33].

6 Conclusion and Future Works

As more and more the scale of grid growing, there is a convincing need to find an effective and efficient way to organize nodes in order to facilitate the discovering and the querying of resources of these nodes. In this paper, we have presented a novel semantic approach of regrouping nodes into federations using SKOS ontology, to construct a three layered architecture. As shown, the propagation of queries in this architecture is scalable since the space of querying is diminished from the entire grid to a smaller range consisting of three semantically related spaces in the worst case, which decreases the cost of resources searching. In addition, this architecture is helpful to enlarge the scale of grid, since it is based on OntDD ontology, which is flexible by nature. We have discussed the problem of leader election, and proposed an efficient process that rendered our system more scalable and fault tolerant.

However, this work is limited to theoretical discussion; the study to evaluate the performance of our algorithm in practice is our future work.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of Grid:Enabling Scalable Virtual Organisations. International Journal of Supercomputer Applications,(2001)
2. Miles, A., Bechhofer,S.: SKOS Simple Knowledge Organization System Reference. W3C, available at http://www.w3.org/TR/skos-reference, January 25.(2008)
3. E. Meshkova, E., Riihiarvi, J., Petrova, M., Mahonen, P.: A Survey on Resource Discovery Mechanisms, Peer-to-Peer and Service Discovery Frameworks. Computer Networks 52, pp. 2097–2128.(2008)
4. Raman, R., Livny, M., Solomon, M.: Matchmaking: Distributed Resource Management for High Throughput Computing. In: 7th IEEE HPDC, pp.140-146. IEEE Computer Society Press, Washington DC (1998)
5. Iamnitchi, A., Foster, I.: On Fully Decentralized Resource Discovery in Grid Environments. In: 2nd IEEE/ACM International Workshop on Grid Computing, Denver, November (2001)

6. Lican, H., Zhaohui, W., Yunhe. P.: A Scalable and Effective Architecture for Grid Services Discovery. In: 1st Workshop on Semantics in Peer-to-Peer and Grid Computing, in conjunction with the 12th International World Wide Web Conference (2003)
7. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284(5), pp.34–43.(2001)
8. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: Edutella: A P2P Networking Infrastructure based on RD. In: International World Wide Web Conference,WWW, pp. 604_15. Honolulu, Hawaii, USA (2002)
9. Arumugam, M., Sheth, A., Arpinar, I.B.: Towards Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web. In: International World Wide Web Conference,WWW, Honolulu, Hawaii, USA(2002)
10. Halevy, A., Ives, Z., Madhavan, J., Mork, P. Suciu, D.: The Piazza Peer Data Management System, pp.787-798. (2004)
11. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M.T., Brunkhorst, I., Lser, A.: Super-peer-based Routing and Clustering Strategies for RDF-based Peer-to-Peer Networks.In: International World Wide Web Conference,WWW, pp. 536-543.(2003)
12. Crespo, A., Garcia-Molina, H.: Semantic Overlay Networks for p2p Systems. Technical report, Stanford University (2002)
13. Iamnitchi, A., Ripeanu, M., Foster, I.T.:Locating data in Peer-to Peer Scientific Collaborations. In: International Workshop on P2P Systems, IPTPS, pp. 232-241.(2002)
14. Tempich, X., Staab, S., Wranik, A.: REMINDIN:Semantic Query Routing in Peer to Peer Networks Based on Social Metaphors.In: International World Wide Web Conference, WWW, pp. 640-649. New York, USA,(2004)
15. Castano, S., Ferrara, A., Montanelli, S., Zucchelli, D., Helios.: A General Framework for Ontology-based Knowledge Sharing and Evolution in P2P Systems. In: DEXA WEBS Workshop, IEEE, pp.597-603. Prague, Czech Republic(2003)
16. Li, J.: Grid Resource Discovery based on semantically linked Virtual Organizations. Future Generation Computer Systems vol 26, pp.361-373.(2010)
17. http://code.google.com/p/skoseditor/
18. https://sourceforge.net/projects/skosapi/
19. http://protege.stanford.edu
20. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F. :From SHIQ and RDF to OWL: The making of a Web Ontology Language. Journal of Web Semantics, pp.7–26.(2003)
21. Bechhofer, S., Yesilada, Y., Stevens, R., Jupp, S., Horan, B.:Using Ontologies and Vocabularies for Dynamic Linking. Internet Computing, pp. 32–39.(2008)
22. McGuinness, D.L., Van Harmelen, F.:OWL:Web Ontology Language Overview, W3C, available at http://www.w3.org/TR/owl-features/. February 10.(2004)
23. Jupp, S., Bechhofer, S., Stevens, R.:A Flexible API and Editor for SKOS. ESWC, Springer, pp. 506–520.(2009)
24. Beckett, D., Berners-Lee, T.:Turtle - Terse RDF Triple Language. Team submission available at http://www.w3.org/TeamSubmission/turtle/, W3C , (2008)
25. http://wordnet.princton.edu/
26. Garcia-Molina.:Elections in a Distributed Computing System. IEEE Trans. Computers, pp. 48–59, (1982)
27. Singh, S., Kurose, J.:Electing Leaders based upon Performance: The delay model.In: 11th International Conference on Distributed Computing Systems, pp. 464–471.(1991)
28. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, G., Schopf, J., Shao, G., Smallen, S., Spring, S. , Su, A., Zagorodnov, D.: Adaptive Computing on the Grid using apples.(2003)
29. Kim, J.L., Belford, G.G. :A Robust, Distributed Election Protocol. Symposium on Reliable Distributed Systems, pp. 54–60.(1988)
30. Padmanabhan, A., Ghosh, S., Wang J, S. :A Self-Organized Grouping (SOG) Framework for Efficient Grid Resource Discovery. Grid Computing, Springer(2009)
31. Kim, T.W., Kim, E. H., Kim, J. K.., Kim, T.Y.:A Leader Election Algorithm in a Distributed Computing System. FTDCS, pp. 481–487.(1995)
32. Wang, X., Kong, L.F.:Resource Clustering Based Decentralized Resource Discovery Scheme in Computing Grid.In: 6th International Conference on Machine Learning and Cybernetics, IEEE, pp. 19-22. Hong Kong, August (2007)
33. Zhang, Y., Jia, Y., Huang, X., Zhou, B., Gu, J.:A grid Resource Discovery Method Based on Adaptive k-Nearesr Neighbors Clustering. COCOA, Springer, pp. 171-181. (2007)

Graphes et optimisation

Optimization of problem Min-Max

Samira Tichefatine , Mohamed Aidene

Department of Mathematics, Faculty of Sciences, University Mouloud Mammeri, Tizi-Ouzou, Alegria.

Abstract. A problems of optimization for linear end nonlinear functions have many practical applications, particularly in automatics, in signal theory where we try to minimize the output error signal.
This paper presents a primal and dual methods for the resolution of minimax problems of functions in absolute value[1,4]. The methods presented are based on the concepts and operations of the adaptive method of linear programming[1,2]. To improve the dual method we use a new concepts "coordinator Support" and "long dual step"[1]. The results are illustrated with an example.

keywords : Minimax problems, Adaptive method, coordinator Support, suboptimality.

1 Introduction :

Minimax (L_∞ norm) is known as one of the principles of optimal parameter estimation. The first parameter estimation procedure based on L_∞ norm was proposed by laplace in 1786 [4,5]. Later L_∞ norm approximation problems have been a topic of research in various applications of mathematics. The traditional approach to solve linear minimax problems is to formulate an equivalent linear programming problem and to solve it by structure exploiting modifications of primal or dual simplex method. The aim of the paper is to realize the adaptive method (Method devloped by R.Gabasov and F.M.Kirillova) originated from an appreach to the solution of linear programming problems, which based on the concept of the support matrix [1]. The characteristic of this method is the fact that it allows the starting of the iteration from an interior point and allows to obtain an ε-optimal solution with a precision $\varepsilon \geq 0$ chosen in advance.

2 Statement of the problem

We Consider the following Min-Max problem :

$$\begin{aligned} f(x) = \max_{l \in L} |c_l^t x + d_l| \rightarrow \min \, , \\ Ax = b \, , \\ d_* \leq x \leq d^* \, , \end{aligned} \tag{1}$$

where $A = A(I,J) = \begin{pmatrix} a_i^t \\ i \in I \end{pmatrix} = \begin{pmatrix} a_{ij}, j \in J \\ i \in I \end{pmatrix}$ is an $m \times n$-matrix with

$rangA = m \leq n$; $b = b(I) \in \mathbb{R}^m$; $x = x(J)$, $d_* = d_*(J)$, $d^* = d^*(J) \in \mathbb{R}^n$;

$c_l(J) \in \mathbb{R}^n$, $d_l \in \mathbb{R}$, $l \in L$; $I = \{1, 2, ..., m\}$, $J = \{1, 2, ..., n\}$, $L = \{1, 2, ..., l^*\}$.

The corresponding linear programming problem has the form :

$$\begin{array}{c} x_0 \to \min , \\ -x_0 \leq c_l^t x + d_l \leq x_0 \ , \ \ l \in L \ ; \\ a_i^t x = b_i \ , \ i \in I \ ; \\ d_* \leq x \leq d^* \ . \end{array} \tag{2}$$

with $x_0 = f(x)$.

3 support

-Let J_{sup} be an arbitrary subset of J and $\overline{J}_{sup} = \{J_{sup}\} \cup \{0\}$, $J_n = J \backslash J_{sup}$.
-Let L_{sup} be an arbitrary subset of L with $|L_{sup}| + |I| = |J_{sup}| + 1$, $L_n = L \backslash L_{sup}$.
-Partition L_{sup} into two subsets L_{sup}^+ and L_{sup}^- with $L_{sup} = L_{sup}^+ \cup L_{sup}^-$ and $L_{sup}^+ \cap L_{sup}^- = \emptyset$.
-Let be $e(L) = (e_l = 1, l \in L)$.

Now we compose the matrix $B_{sup} = B(L_{sup} \cup I, \overline{J}_{sup})$:

$$B_{sup} = \begin{pmatrix} e(L_{sup}^-) & c_l^t(J_{sup}), \\ -e(L_{sup}^+) & l \in L_{sup} \\ 0(I) & a_i^t(J_{sup}) \ , \\ & i \in I \end{pmatrix} .$$

Construct the vector of multipliers $u(L) = (u(L_{sup}), u(L_n))$, $\pi(I)$:

$(u(L_{sup}) \ , \ \pi(I)) = c_0^t(\overline{J}_{sup}) B_{sup}^{-1}$, $u(L_n) = 0$, $c_0 = (\ c_{00} = -1 \ , \ c_{0j} = 0 \ , \ j \in J_{sup} \ )$,

and reduced cost $E(J)$:

$$E_j = u^t(L_{sup}) c(L_{sup}, j) + \pi^t(I) A(I, j) \ , \ j \in J_n \ ,$$

by construction , we have :

$$E_j = 0 \ , \ \ j \in J_{sup}.$$

Definitions :

-Any vector x that satisfies the constraints $Ax = b$ and $d_* \leq x \leq d^*$ of problem (1) is called a feasible solution .

-A pair $k_{sup} = \{L_{sup}, \overline{J}_{sup}\}$ is called a support of problem (1) if $|B_{sup}| \neq 0$ and the following inequalities are true for the multipliers :

$u(L^-_{sup}) \leq 0$, $u(l^+_{sup}) \geq 0$.

-A pair $\{x, K_{sup}\}$ of a feasible solution x of problem (1) and support K_{sup} is called a support feasible solution .

-A support feasible solution $\{x, K_{sup}\}$ is called primal nondegenerate if :

$$\begin{array}{c} d_{*j} < x_j < d^*_j \ , \ j \in J_{sup} \ ; \\ |c^t_l x + d_l| < f(x) \ , \ l \in L_n \ . \end{array}$$

-A support K_{sup} is said to be coordinated with a feasible solution x if :

$$\begin{array}{c} L^+_{sup} \subseteq L^+(x) = \{l \in L : c^t_l x + d_l = f(x)\} \ ; \\ L^-_{sup} \subseteq L^-(x) = \{l \in L : c^t_l x + d_l = -f(x)\}. \end{array}$$

4 Increment of performance index

Let x be a nondegenerate feasible solution of problem (1), K_{sup} be a support coordinated with x and $\overline{x} = x + \triangle x$ another feasible solution of problem (1).

At point x we have :

$$\left\{ \begin{array}{ll} c^t_l x - x_0 = w_l - d_l & , l \in L^+_{sup} \\ c^t_l x + x_0 = w_l - d_l & , l \in L^-_{sup} \\ a^t_i x \qquad = w_i + b_i & , i \in I \ , \end{array} \right. \tag{3}$$

Analogous relations hold at point $\overline{x}$:

$$\left\{ \begin{array}{ll} c^t_l \overline{x} - \overline{x}_0 = \overline{w}_l - d_l & , l \in L^+_{sup} \\ c^t_l \overline{x} + \overline{x}_0 = \overline{w}_l - d_l & , l \in L^-_{sup} \\ a^t_i \overline{x} \qquad = \overline{w}_i + b_i & , i \in I \ , \end{array} \right. \tag{4}$$

for the residuals $w = (w(L_{sup}), w(I))$, $\overline{w} = (\overline{w}(L_{sup}), \overline{w}(I))$ with :
$(\overline{w}(L_{sup}), \overline{w}(I)) = (w(L_{sup}), w(I)) + (\triangle w(L_{sup}), \triangle w(I))$,
$\overline{w}(I) = w(I) = \triangle w(I) = 0_I$.

The difference between formulas (3) and (4) gives :

$$\left\{\begin{array}{ll} c_l^t\triangle x - \triangle x_0 = \triangle w_l \ , \ l \in L_{sup}^+ \\ c_l^t\triangle x + \triangle x_0 = \triangle w_l \ , \ l \in L_{sup}^- \\ a_i^t\triangle x \qquad\quad = \triangle w_i \ , \ i \in I \ . \end{array}\right.$$

Then we have :

$$\left\{\begin{array}{rl} \triangle x_0 + c[l, J_{sup}]\triangle x(J_{sup}) = -c[l, J_n]\triangle x(J_n) + \triangle w_l \ , \ l \in L_{sup}^- \\ -\triangle x_0 + c[l, J_{sup}]\triangle x(J_{sup}) \ = -c[l, J_n]\triangle x(J_n) + \triangle w_l \ , \ l \in L_{sup}^+ \\ A[i, J_{sup}]\triangle x(J_{sup}) = -A[i, J_n]\triangle x(J_n) + \triangle w_i \ , \ i \in I \ . \end{array}\right.$$

By developing the last equations, we obtain :

$$\begin{pmatrix} \triangle x_0 \\ \triangle x(J_{sup}) \end{pmatrix} = -B_{sup}^{-1}\left(\begin{pmatrix} c[L_{sup}, J_n] \\ A[I, J_n] \end{pmatrix}\triangle x(J_n) - \begin{pmatrix} \triangle w(L_{sup}) \\ \triangle w(I) \end{pmatrix}\right),$$

thus, we obtain :

$$\begin{pmatrix} \triangle x_0 \\ \triangle x(J_{sup}) \end{pmatrix} = -B_{sup}^{-1}\left(\begin{pmatrix} c[L_{sup}, J_n] \\ A[I, J_n] \end{pmatrix}\triangle x(J_n) - \begin{pmatrix} \triangle w(L_{sup}) \\ 0(I) \end{pmatrix}\right), \quad (5)$$

for all $\triangle x(J_n)$ and $\triangle w(L_{sup})$.

The increment of the cost function of problem (1) is given by :

$$\triangle x_0 = \sum_{j \in J_n} E_j \triangle x_j - \sum_{l \in L_{sup}} u_l \triangle w_l. \quad (6)$$

The maximum of the functional $-\triangle f(x) = -[f(x+\triangle x) - f(x)] = -\triangle x_0$ under the constraints :

$$\left\{\begin{array}{lll} d_{*j} - x_j \leq \triangle x_j \leq d_j^* - x_j \ , \ j \in J_n \ ; \\ \triangle w_l \leq -w_l \quad , \quad l \in L_{sup}^+ \quad ; \\ \triangle w_l \geq -w_l \quad , \quad l \in L_{sup}^- \quad , \end{array}\right. \quad (7)$$

is reached for :

$$\triangle x_j = \left\{\begin{array}{l} d_{*j} - x_j \ , \ E_j > 0 \ , \\ d_j^* - x_j \ \ , \ E_j < 0 \ , \\ 0 \ , \ E_j = 0 \ , \ j \in J_n, \end{array}\right. ; \ \triangle w_l = \left\{\begin{array}{l} -w_l \ , \ l \in L_{sup}^+ \ ; \\ -w_l \ , \ l \in L_{sup}^- \ , \end{array}\right. \quad (8)$$

and is equal to :

$$\beta(x,K_{sup}) = \sum_{E_j>0, j\in J_n} E_j(x_j - d_{*j}) + \sum_{E_j<0, j\in J_n} E_j(x_j - d_j^*) + \sum_{l\in L^+_{sup}} u_l(f(x) - c_l^t x - d_l) + \sum_{l\in L^-_{sup}} u_l(-f(x) - c_l^t x - d_l), \quad (9)$$

called a suboptimality estimate of the support feasible solution $\{x, K_{sup}\}$.

Thus, there is always the inequality :

$$f(x) - f(\overline{x}) \leq \beta(x, K_{sup}) \ , \ \forall\ \overline{x}\ . \quad (10)$$

From this inequality, we deduce the following optimality criterion.

5 Optimality criterion.

5.1 Theorem [1]

Let x be a feasible solution of problem (1), K_{sup} be a support coordinated with x such that the relations :

$$\left\{ \begin{array}{l} x_j = d_{*j} \quad , \text{si} \quad E_j > 0 \quad ; \\ x_j = d_j^* \quad , \text{si} \quad E_j < 0 \quad ; \\ d_{*j} \leq x_j \leq d_j^* \ , \text{si } E_j = 0 \ , \ j \in J_n, \end{array} \right. \quad (11)$$

are sufficient and in the case of nondegeneracy they are necessary for the optimality of the support feasible solution $\{x, K_{sup}\}$.

6 Dual problem

The dual problem to (1) has the form :

$$\begin{array}{l} \phi(\lambda) = d^t\gamma - b^t y + d_*^t v - d^{*t} w \to \max \ , \\ \quad c^t\gamma + A^t y - v + w = 0 \ , \\ \quad y = y_1 - y_2 \ , \ y_1 \geq 0 \ , \ y_2 \geq 0 \ , \\ \quad v \geq 0 \ , \quad w \geq 0 \ , \quad \sum_{l\in L} |\gamma_l| = 1 \ , \end{array} \quad (12)$$

where $c = \begin{pmatrix} c_l^t(J), \\ l \in L \end{pmatrix}$.

A triaple $\lambda = (\gamma = u(L), y = \pi(I), y_1, y_2, v, w)$ constructed with support K_{sup} of problem (1) and satisfies the relations :

$$\left\{ \begin{array}{l} \gamma(L) = u(L) \ ; \\ v_j = E_j \ , \ w_j = 0 \ , \text{si } E_j \geq 0 \ , \\ v_j = 0 \ , \ w_j = -E_j \ , \text{si } E_j < 0 \ , \ j \in J \ ; \\ y_{1i} = y_i \ , \ y_{2i} = 0 \ , \text{si } y_i \geq 0 \ , \\ y_{1i} = 0 \ , \ y_{2i} = -y_i \ , \text{si } y_i \leq 0 \ , \ i \in I \ . \end{array} \right. \quad (13)$$

is a dual feasible solution of problem (12) .

Using relations (12), function (9) will be :

$$\begin{aligned}\beta(x, K_{sup}) &= \sum_{E_j>0, j\in J_n} E_j(x_j - d_{*j}) + \sum_{E_j<0, j\in J_n} E_j(x_j - d_j^*) \\ &\quad + \sum_{l\in L_{sup}^+} u_l(f(x) - c_l^t x - d_l) + \sum_{l\in L_{sup}^-} u_l(-f(x) - c_l^t x - d_l) \\ &= E^t x - v^t d_* + w^t d^* + f(x) - E^t x - u^t d \\ &= f(x) - u^t d - v^t d_* + w^t d^* \\ &= f(x) - \phi(\lambda) \\ &= f(x) - f(x^0) + \phi(\lambda^0) - \phi(\lambda) \\ &= \beta(x) + \beta(K_{sup}) \ ,\end{aligned}$$

where λ^0 : is an optimal dual feasible solution corresponds to the optimal primal x^0 of problem (1) ,

$\beta(x) = f(x) - f(x^0)$: is the measure of nonoptimality of a feasible solution x ,

$\beta(K_{sup}) = \phi(\lambda^0) - \phi(\lambda)$: is the measure of nonoptimality of a support K_{sup} .

As

$$\beta(x, K_{sup}) = \beta(x) + \beta(K_{sup}) \ , \tag{14}$$

then we deduce the suboptimality criterion that we present thereafter :

7 Suboptimality criterion

7.1 Theorem [1]

For any $\varepsilon \geq 0$ a feasible solution x of problem (1) is ε-optimal if and only if there exists a support K_{sup} such that the suboptimality estimate $\beta(x, K_{sup})$ of the support feasible solution $\{x, K_{sup}\}$ satisfies the inequality :

$$\beta(x, K_{sup}) \leq \epsilon.$$

7.2 Corollary :

A feasible solution x is optimal in problem (1) if and only if there exists a support K_{sup} such that the suboptimality estimate of the support feasible solution $\{x, K_{sup}\}$ is equal to zero :

$$\beta(x, K_{sup}) = 0 \ . \tag{15}$$

7.3 Remark :

Equality (15) is true if and only if relations (11) hold and the support K_{sup} is coordinated with x .
Let us describe an iterative method for solving problem (1). The method starts from a support feasible solution $\{x, K_{sup}\}$ with a support coordinated with x .

8 Iteration of the adaptive method

It follows from (14) that the suboptimality estimate can be decreased, first, by decrease of the measure of the feasible solution x nonoptimality by changing x and, second, by decrease of the measure of the support K_{sup} nonoptimality by changing K_{sup} .

8.1 Changing x ($x \longrightarrow \overline{x}$)

The first procedure starts with the constructing of a pseudosolution accompanying the support K_{sup} by the following rules :

$$\kappa_j = \begin{cases} d_{*j} & if \quad E_j \geq 0 \ , \\ d_j^* & if \quad E_j < 0 \ , \ j \in J_n \ ; \end{cases}$$

$$\begin{pmatrix} \kappa_0 \\ \kappa(J_{sup}) \end{pmatrix} = -B_{sup}^{-1} \left(\begin{pmatrix} c(L_{sup}, J_n) \\ A(I, J_n) \end{pmatrix} x(J_n) + \begin{pmatrix} d(L_{sup}) \\ -b(I) \end{pmatrix} \right) \tag{16}$$

a/ If the relations :

$$\begin{aligned} & d_{*j} \leq \kappa_j \leq d_j^* \quad , \ j \in J_{sup} \quad ; \\ & |c_l^t \kappa + d_l| \leq \kappa_0 \ , \quad l \in L_n \quad ; \\ & \kappa_0 \geq 0 \qquad\quad if \ \ L_n = \emptyset \quad , \end{aligned} \tag{17}$$

are true then κ is optimal and $f(\kappa) = \kappa_0$.

b/ If relations (17) are violated then we constuct the direction $q \in \mathbb{R}^n$ by:

$$q = \kappa - x \ . \tag{18}$$

Therefore, according to the principle of decreasing nonoptimality estimate , we construct a new feasible solution by the formula :

$$\overline{x} = x + \theta q \ ,$$

where the steplength $\theta = \min\{1, \theta_{j_0}, \theta_f\}$ is the greatest step :
. 1 is the step allowed by the nonsupport constraints .
. θ_{j_0} is the step allowed by the support constraints .
. θ_f is the step defined by the constraints derived from the cost function of problem (1).

The various steps are obtained by the following formulas :

$$.\theta_{j_0} = \min \theta_j \quad , \quad j \in J_{sup} \quad ; \quad \theta_j = \begin{cases} (d_{*j} - x_j)/q_j & \text{if } q_j < 0 \ , \\ (d_j^* - x_j)/q_j & \text{if } q_j > 0 \ , \\ \infty & \text{if } q_j = 0 \ , j \in J_{sup} \ . \end{cases}$$

$$.\theta_f = \min\{\theta_0, \theta_{l_0}\} \ ; \ \theta_0 = f(x)/B(x, K_{sup}) \ , \ \theta_{l_0} = \min\{\theta_l^+, \theta_l^-\} \ , \ l \in L_n; \quad (19)$$

$$\theta_l^+ = \begin{cases} (-c_l^t x - d_l + x_0)/(c_l^t q + B(x, K_{sup})) & \text{if } c_l^t q + B(x, k_{sup}) > 0 \ ; \\ \infty & \text{else} \ , \quad l \in L_n \ . \end{cases}$$

$$\theta_l^- = \begin{cases} (-c_l^t x - d_l - x_0)/(c_l^t q - B(x, K_{sup})) & \text{if } c_l^t q - B(x, K_{sup}) < 0 \ , \\ \infty & \text{else} \ , \quad l \in L_n \ . \end{cases}$$

The suboptimality estimate of the support feasible solution $\{\overline{x}, K_{sup}\}$is equal to:

$$\begin{aligned} \beta(\overline{x}, K_{sup}) &= \sum_{j \in J_n, E_j > 0} E_j(\overline{x_j} - d_{*j}) + \sum_{j \in J_n, E_j < 0} (\overline{x}_j - d_j^*) - \sum_{l \in L_{sup}} u_l w_l \\ &= \beta(x, K_{sup}) + \theta \sum_{j \in J_n} E_j q_j \\ &= (1 - \theta)\beta(x, K_{sup}). \end{aligned}$$

If the support feasible solution $\{x, K_{sup}\}$ is primal nondegenerate , then $\theta > 0$ consequently :

$$\beta(\overline{x}, K_{sup}) < \beta(x, K_{sup}).$$

We calculate θ , we have the various following cases :

`.First case :` $\theta = \theta_0$, then $\overline{x} = x + \theta q$ is optimal and $f(\overline{x}) = 0$ or

$$\overline{x} = x + \theta q \text{ is } \varepsilon\text{-optimal with } f(\overline{x}) \leq \varepsilon \ .$$

`.Second case :` $\theta = 1$, then $\overline{x} = \kappa$ is optimal .

`.Third case :` $(1 - \theta)\beta(x, K_{sup}) > \epsilon$, then we perform the change of the support.

8.2 Changing K_{sup} ($K_{sup} \longrightarrow \overline{K}_{sup}$)

The change of support $K_{sup} \longrightarrow \overline{K}_{sup}$ involves the change of the dual feasible solution $\lambda \longrightarrow \overline{\lambda}$, i.e:

$$\begin{cases} \overline{u}(L) = u(L) + \sigma_0 t(L) \ ; \\ \overline{E}(J) = E(J) + \sigma_0 t(J) \ , \end{cases}$$

where the vector $t(J, L)$ is the direction of the dual cost function increase and σ_0 is the greatest step along this direction .

This change of the support is achieved according to the value of θ . Hence there are two cases :

a/ $\theta = \theta_{j_0}$, $j_0 \in J_{sup}$:

Calculate :

$$\mu_* = \begin{cases} d_{*j_0} - \kappa_{j_0} & \text{if } \kappa_{j_0} < d_{*j_0} \ ; \\ d^*_{j_0} - \kappa_{j_0} & \text{if } \kappa_{j_0} > d^*_{j_0} \ , \end{cases} \tag{20}$$

and assume :

$$\begin{cases} t_{j_0} = sign\mu_* \ \ ; \\ t(J_{sup} \backslash j_0) = 0 \ \ ; \\ t(L_n) = 0 \ \ , \end{cases} \tag{21.1}$$

And from the admissibility of $\overline{u}$, $\overline{E}$ we obtain :

$$\begin{cases} (t(L_{sup}), t(L_{sup})) = (0, t(J_{sup}))^t B^{-1}_{sup} \ ; \\ t^t(J_n) = t^t(L_{sup}) c(L_{sup}, J_n) + t^t(I) A(I, J_n) \ . \end{cases} \tag{21.2}$$

We calculate the steps σ_j , $j \in J_n$; σ_k , $k \in L_{sup}$:

$$\sigma_j = \begin{cases} -E_j/t_j \text{ for } E_j t_j < 0 \ ; \\ 0 \qquad \text{for } E_j = 0,\ t_j > 0,\ \kappa_j \neq d_{*j} \text{ or } E_j = 0,\ t_j < 0,\ \kappa_j \neq d^*_j; \\ \infty \qquad \text{in other cases , } j \in j_n. \end{cases} \tag{22.1}$$

$$\sigma_k = \begin{cases} -u_k/t_k \text{ for } t_k < 0 \ , \ k \in L^+_{sup} \ \ ; \\ -u_k/t_k \text{ for } t_k > 0 \ , \ k \in L^-_{sup} \ \ ; \\ \infty \qquad \text{in other cases , } k \in L_{sup} \ \ . \end{cases} \tag{22.2}$$

Put the finite values σ_j , $j \in J_n$; σ_k , $k \in L_{sup}$ in to nondecreasing order :

$$\sigma_{j_1} \leq \sigma_{j_2} \leq \ldots \leq \sigma_{j_p} \ \ , \qquad j_k \in \{J_n \cup L_{sup}\} \ .$$

For each j_k , we calculate the jump of the dual objective function :

$$\triangle \mu_k = -|t_{j_k}|(d^*_{j_k} - d_{*j_k}) \ .$$

The dual cost function behaves along the direction t as a concave continuous piecewise-linear function . Its slope in the interval $[\sigma_k, \sigma_{k+1}]$ is equal to :

$$\begin{cases} \mu_k = \mu_{k-1} - |t_{j_k}|(d^*_{j_k} - d_{*j_k}) \ , \ \ k = 1, ..., \overline{p} \ \ ; \\ \mu_0 = |\mu_*| \ , \end{cases} \tag{23}$$

where $\overline{p} \leq p$ is a number such that :

$$\begin{cases} \overline{p} = p \qquad \text{if } j_k \in J_n \ \ , \ \ \forall i = \overline{1, p} \ ; \\ or \\ \overline{p} < p \qquad \text{if } j_{\overline{p}+1} \in L_{sup} \ . \end{cases}$$

Remark : μ_0 is the initial speed of change of the dual cost function .

We move along the direction t until the dual cost function is increasing (the principle of full relaxation).

1- If $\mu_{\overline{p}} > 0$, then $\overline{p} < p$. Assume $\sigma_0 = \sigma_{j_{\overline{p}+1}}$, $s_0 = j_{\overline{p}+1}$, $s = \overline{p}+1$.

2- If $\mu_{\overline{p}} \leq 0$, then we find an index ν , $0 < \nu \leq \overline{p}$ such that : $\mu_{\nu-1} > 0$ and $\mu_\nu \leq 0$. Assume $\sigma_0 = \sigma_\nu$, $s_0 = j_\nu$, $s = \nu$.

Therefore we construct the reduced costs and the multipliers :

$$\begin{cases} \overline{E}(J) = E(J) + \sigma_0 t(J) \ ; \\ \overline{u}(L) = u(L) + \sigma_0 t(L) \ \ . \end{cases} \tag{24}$$

using the new support $\overline{K}_{sup} = \{\overline{\overline{J}}_{sup}, \overline{L}_{sup}\} : \overline{\overline{J}}_{sup} = \{0\} \cup \overline{J}_{sup}$,

$$\begin{cases} \overline{J}_{sup} = J_{sup} \backslash j_0, \\ \overline{L}_{sup} = L_{sup} \backslash s_0, \text{ if } s_0 \in L_{sup} \ \ , \end{cases} \quad , \quad \begin{cases} \overline{J}_{sup} = (J_{sup} \backslash j_0) \cup s_0, \\ \overline{L}_{sup} = L_{sup}, \qquad\qquad \text{if } s_0 \in J_n \ . \end{cases}$$

b/ $\theta = \theta_{l_0}$, $l_0 \in L_n$:

Calculate :

$$\mu_* = \begin{cases} -\kappa_0 + c^t_{l_0}\kappa + d_{l_0}, & \text{if } \theta_{l_0} = \theta^+_{l_0} \ ; \\ \kappa_0 + c^t_{l_0}\kappa + d_{l_0}, & \text{if } \theta_{l_0} = \theta^-_{l_0} \ , \end{cases} \tag{25}$$

and assume :

$$\begin{cases} t_{l_0} = sign \mu_*; \\ t(L_n \backslash l_0) = 0; \\ t(J_{sup}) = 0, \end{cases} \tag{26.1}$$

and from the admissibility of $\overline{u}$, $\overline{E}$ we obtain :

$$\begin{cases} (t(L_{sup}), t(I)) = (1, -t_{l_0} c^t_{l_0}(J_{sup})) A^{-1}_{sup} \ \ , \\ t^t(J_n) = t^t(L_{sup}) c(L_{sup}, J_n) + t_{l_0} c^t_{l_0}(J_n) + t^t A(I, J_n) \ . \end{cases} \tag{26.2}$$

We apply the same method used of (a) in order to find the index s_0 and the step σ_0.

Therefore we construct the reduced costs and the multipliers : $\overline{E}(J)$, $\overline{u}(L)$ defined by the formula (24) using the new support $\overline{K}_{sup} = \{\overline{\overline{J}}_{sup}, \overline{L}_{sup}\}$:
$\overline{\overline{J}}_{sup} = \overline{J}_{sup} \cup \{0\}$,

$$\begin{cases} \overline{J}_{sup} = J_{sup}, \\ \overline{L}_{sup} = (L_{sup} \backslash s_0) \cup l_0, \ \text{ if } \ \ s_0 \in L_{sup} \ \ , \end{cases} \qquad \begin{cases} \overline{J}_{sup} = J_{sup} \cup s_0, \\ \overline{L}_{sup} = L_{sup} \cup l_0, \ \text{ if } \ \ s_0 \in J_n \ , \end{cases}$$

such that :

$$\begin{cases} l_0 \in \overline{L^+}_{sup} & \text{if } \ \ \theta_{l_0} = \theta^+_{l_0} \ \ ; \\ l_0 \in \overline{L^-}_{sup} & \text{if } \ \ \theta_{l_0} = \theta^-_{l_0} \ \ . \end{cases}$$

By construction, the new support is coordinated with the feasible solution $\overline{x}$.

The suboptimality estimate of the new support feasible solution $\{\overline{x}, \overline{K}_{sup}\}$ is equal to :

$$\beta(\overline{x}, \overline{K}_{sup}) = (1-\theta)\beta(x, K_{sup}) - \sum_{k=0}^{s-1} \mu_k(\sigma_{j_{k+1}} - \sigma_{j_k}) \ , \quad \sigma_{j_0} = 0 \ .$$

9 Numerical Example

$$\max(\ |2x_1 - 3x_2 + x_3| \ , \ |x_1 + 4x_2 - x_3 - 1| \ , \ |x_1 - x_2 - x_3 + 3|\ ) \ \rightarrow \ \min \ ,$$

$$\begin{pmatrix} 2 & -1 & 3 \\ -1 & 4 & 1/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{1}$$

$$-1 \leq x_1 \leq 3 \ , \ -1 \leq x_2 \leq 1 \ , \ -1 \leq x_3 \leq 1 \ .$$

9.1 First iteration :

$x = (6/7, 5/7, 0)$ is a feasible solution of problem (1) and $K_{sup} = \{\overline{J}_{sup}, I, L_{sup}\}$ is the support coordinated with x such that : $\overline{J}_{sup} = J_{sup} \cup \{0\}$, $J_{sup} = \{1, 2\}$, $I = \{1, 2\}$, $L_{sup} = \{3\} = L^+_{sup}$.
$\{x, K_{sup}\}$ is a support feasible solution nondegenerate .
$\beta(x, K_{sup}) = 31/14 > \varepsilon$, Then the support feasible solution $\{x, K_{sup}\}$ is not optimal.

9.1.1 Changing of feasible solution : $x \longrightarrow \overline{x} = x + \theta q$
$q = (-25/14, -4/7, 1)$; $\theta = 41/51 = \theta^-_2$, $2 \in L_n$.
Then $\overline{x} = (-59/102, 13/51, 41/51)$ and $\{\overline{x}, K_{sup}\}$ is not optimal.

9.1.2 Changing of support : $K_{sup} \longrightarrow \overline{K}_{sup}$
$\sigma_0 = 31/102 = \sigma_3$, $3 \in J_n$. Then the new support is $\overline{K}_{sup} = \{\overline{\overline{J}}_{sup}, \overline{L}_{sup}\}$ such that $\overline{\overline{J}}_{sup} = \overline{J}_{sup} \cup \{0\}$, $\overline{J}_{sup} = \{1, 2, 3\}$; $I = \{1, 2\}$; $\overline{L}_{sup} = \{2, 3\}$, $2 \in L^-_{sup}$ and $3 \in L^+_{sup}$.
$\beta(\overline{x}, \overline{K}_{sup}) = 0$, then the support feasible solution $\{\overline{x}, \overline{K}_{sup}\}$ is optimal , with : $x^0 = \overline{x} = (-59/102, 13/51, 41/51)$ and $x^0_0 = f(x^0) = 139/102$.

10 Conclusion :

In the paper we have used the principles of the adaptive method, at the same time specific features of minimax problems have been taken into consideration. After giving some definitions, we have constructed the support, then calculated the increment of the objective functional. Optimality and suboptimality criteria have been formulated to describe the iteration of the adaptive method. This method is based on two procedures. One is the change of the feasible solution, the other is the coordinator support. The results are illustrated with a numerical example.

References

1. R.Gabasov, F.M.Kirillova and E.A.Kostina. "Adaptive methods for solving minimax problems". Received 28 May 1997; In final from 21 December 1999.
2. R.Gabasov, F.M.Kirillova, E.A.Kostyukova and V.M.Raketsky. "Constructive methods of optimization". Volume 4: Convex problms. University Prees, Minsk, 1987.
3. E.A.kostina. "Algorithms of solving nonsmooth problems of minimax-type". Ph.D.Dissertation. Minsk(in Russian). 1990.
4. N.Z.Shor. "Minimization methods for nondifferentiable functions". Springer-verlag, Berlin 1985.
5. G.A.Watson. "Approximation theory and numerical methods". Wiley, New york. 1980.

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

On the dominator colorings in trees

Boumediene Merouane Hocine and Mustapha Chellali

LAMDA-RO Laboratory, Department of Mathematics

University of Blida

B.P. 270, Blida, Algeria.

e-mail: m_chellali@yahoo.com

Abstract. In a graph G, a vertex is said to dominate itself and all its neighbors. A dominating set of a graph G is a subset of vertices that dominates every vertex of G. The domination number $\gamma(G)$ is the minimum cardinality of a dominating set of G. A dominator coloring is a coloring of the vertices of a graph such that every vertex is either alone in its color class or adjacent to all vertices of at least one other class. The dominator chromatic number $\chi_d(G)$ is the minimum number of color classes in a dominator coloring of G. Gera showed that every nontrivial tree T satisfies $\gamma(T) + 1 \leq \chi_d(T) \leq \gamma(T) + 2$. In this note we characterize nontrivial trees T attaining each bound.

Keywords: Dominator coloring, domination, trees.
AMS subject classification: 05C15, 05C69

1 Introduction

Let $G = (V, E)$ be a simple graph. A vertex in a graph G is said to dominate every vertex adjacent to it. A set D of vertices in G is a dominating set if every vertex not in D is adjacent to at least one vertex in D. The *domination number* $\gamma(G)$ is the minimum cardinality among the dominating sets of G.

A *proper coloring* of a graph $G = (V, E)$ is a function from the vertices of the graph to a set of colors such that any two adjacent vertices have different colors. A *dominator coloring* of a graph G is a proper coloring such that every vertex of V dominates all vertices of at least one color class (possibly its own class). The *dominator chromatic number* $\chi_d(G)$ is the minimum number of color classes in a dominator coloring of G. A dominator coloring of G with $\chi_d(G)$ colors will be called $\chi_d(G)$-DC. The concept of dominator coloring was introduced by Gera, Horton and Rasmussen [4] and studied further in [2, 3], and recently in [1].

It is shown in [2, 3] that for every nontrivial tree T, $\gamma(T) + 1 \leq \chi_d(T) \leq \gamma(T) + 2$, however computing the exact value of the dominator coloring number of a tree remains an open problem. Our aim in this note is to characterize all nontrivial trees T attaining each bound. To this end we will focus only on trees T with $\chi_d(T) = \gamma(T) + 1$.

Let us introduce some notations and definitions. The *open neighborhood* $N(v)$ of a vertex v consists of the vertices adjacent to v, and $N[v] = N(v) \cup \{v\}$ is the *closed neighborhood.* For a vertex set $S \subseteq V(G)$, $N(S) = \cup_{v \in S} N(v)$ and $N[S] = \cup_{v \in S} N[v]$. A *leaf* of a graph G is a vertex of degree 1, and its neighbor is called a *stem.* For a set $S \subseteq V$, the *private neighborhood* $pn(v, S)$ of $v \in S$ is defined by $pn(v, S) = N[v] - N[S - \{v\}]$. If D is a minimum dominating set of G, then let $D_I = \{v \in D : pn(v, D) = \{v\}\}$ and $D_R = D - D_I$. Let $V_1, V_2, \ldots, V_{\chi_d(G)}$ be the color classes of a dominator coloring of V. We denote by C_P the set of color classes containing a single vertex, by C_S the set of dominated color classes containing at least two vertices and by C_G the set of color classes containing at least two vertices and dominated by no vertex of V. Clearly C_P, C_S, C_G are disjoint sets and $C_G \cup C_P \cup C_S = \{V_1, V_2, \ldots, V_{\chi_d(G)}\}$. A vertex of V is called *solitary* if it belongs to color class of size one. Let A be the set of all solitary vertices and B the set of all vertices belonging to color classes in C_G. Clearly $|C_P| = |A|$. We denote by x_S a vertex dominating the color class S and let $DS = \{x_S \in V : S \in C_S\}$. Recall that a subset of vertices $S \subseteq V$ is *independent* if no edge of G has its two endvertices in S.

We shall prove:

Theorem 1 *Let T be a nontrivial tree. Then $\chi_d(T) = \gamma(T) + 1$ if and only if T admits a minimum dominating set $D = D_I \cup D_R$ such that $V(T) - (D_R \cup N[D_I])$ is an independent set.*

2 Proof of Theorem 1

We begin by the following straightforward observation.

Observation 2 *Let T be a nontrivial tree. Then for every $\chi_d(T)$-DC of T, either each stem is solitary or it is adjacent to exactly one leaf and that leaf is solitary.*

Lemma 3 *Every tree T of order at least three admits a dominator coloring with $\chi_d(T)$ colors such that all leaves of T have the same color.*

Proof. Consider a dominator coloring of T with $\chi_d(T)$ colors such every stem is solitary. Such a dominator coloring exists for otherwise if some stem u is not solitary, then by Observation 2 its leaf neighbor would be solitary but then we can swap the colors between the two vertices. Now clearly all leaves can be colored by the same color. □

Lemma 4 *For every $\chi_d(T)$-DC of a tree T, every color class $S \in C_S$ is dominated by exactly one vertex. In that case we have $|C_S| = |DS|$.*

Proof. Consider a dominator coloring of T with $\chi_d(T)$ colors and suppose that a color class $S \in C_S$ is dominated by two vertices x_S and y_S. Then x_S, y_S and any two vertices of S induce a cycle C_4, a contradiction. Clearly $|C_S| = |DS|$ follows immediately. □

Lemma 5 *For every $\chi_d(T)$-DC of a tree T, $A \cup DS$ is a dominating set of T.*

Proof. Consider a dominator coloring of T with $\chi_d(T)$ colors. Every vertex x of T dominates at least one class color, say H_x. If $H_x \in C_p$, then x is dominated by A and if $H_x \in C_S$, then x belongs to DS. Hence $A \cup DS$ dominates all vertices of T. □

Lemma 6 *Let T be a nontrivial tree. If $\chi_d(T) = \gamma(T) + 1$, then for every $\chi_d(T)$-DC of T, there is at most one color class dominated by no vertex.*

Proof. Let T be a nontrivial tree with $\chi_d(T) = \gamma(T) + 1$. Consider any dominator coloring of T with $\chi_d(T)$ colors. We have to prove that $|C_G| \leq 1$. By Lemma 5, $A \cup DS$ is a dominating set of T and so $\gamma(T) \leq |A \cup DS|$. It follows that

$$\begin{aligned} \gamma(T) + |C_G| &\leq |A \cup DS| + |C_G| \\ &\leq |A| + |DS| + |C_G| . \end{aligned}$$

Using the fact that $|C_S| = |DS|$ (see Lemma 4) we obtain $\gamma(T) + |C_G| \leq |A| + |C_S| + |C_G| = \gamma(T) + 1$ and so $|C_G| \leq 1$. □

Lemma 7 *Let T be a nontrivial tree different from a star. If $\chi_d(T) = \gamma(T) + 1$, then for every dominator coloring with $\chi_d(T)$ colors such that all leaves of T have the same color we have:*

a) $|C_G| = 1$.

b) *$A \cup DS$ is a minimum dominating set.*

c) $A \cap DS = \emptyset$.

d) *Every color class $S \in C_S$ is dominated by a vertex of B.*

Proof. Consider a dominator coloring with $\gamma(T) + 1$ colors such that all leaves of T have the same color.

(a)- Since T is not a star, all leaves of T form a color class dominated by no vertex, that is $|C_G| \geq 1$. Equality follows from 6.

(b)- $\gamma(T) + 1 = \chi_d(T) = |C_P| + |C_S| + |C_G| = |C_P| + |C_S| + 1$, implying that $\gamma(T) = |C_P| + |C_S|$. Since $|C_P| = |A|$ and $|C_S| = |DS|$ it follows that $\gamma(T) = |A| + |DS|$ and so $A \cup DS$ is a minimum dominating set.

(c)- follows from (b).

(d)- Let S be a color class of C_S dominated by some vertex $x_S \in DS$. Suppose to the contrary that $x_S \notin B$. Then by item (c) $x_S \notin A$ and hence $x_S \in V(T) - (A \cup B)$, that is x_S

belongs to some color class in C_S. We shall prove that there is a color class, say $S^* \in C_S$ such that each of all its vertices dominates a color class in C_P. Since every vertex of T must dominate a color class, if $S \neq S^*$, then a vertex of S, say x_{S_1} dominates a color class $S_1 \in C_S$. Now if $S_1 \neq S^*$, then a vertex x_{S_2} of S_1 dominates a color class $S_2 \in C_S$, and so on. Since T is finite, the process stops by providing either a cycle induced by vertices $x_S, x_{S_1}, x_{S_2}, ..$ or the existence of S^*. Clearly since T contains no cycle, we conclude that S^* exists. It follows that every vertex of S^* is dominated by A. Now let x_{S^*} be the vertex of DS that dominates all vertices of S^*. Then $A \cup DS - \{x_{S^*}\}$ is a dominating set of T of size $\gamma(T) - 1$, a contradiction. Therefore every color class $S \in C_S$ is dominated by a vertex of B. □

Now we are ready to prove Theorem **1.**

Proof of Theorem 1. Let T be a nontrivial tree and assume that T admits a minimum dominating set $D = D_I \cup D_R$ such that $V(T) - (D_R \cup N[D_I])$ is an independent set. To see that $\chi_d(T) = \gamma(T) + 1$, we color the vertices of G as follow:

- give a different color to every vertex of D_R.

- for every vertex $y \in D_I$ give a new color to the vertices of $N(y)$.

- give the same color (but new) to the remaining vertices.

Obviously the previous coloring is a dominator coloring. Hence $\gamma(T) + 1 \leq \chi_d(T) \leq |D_R| + |D_I| + 1 = |D| + 1 = \gamma(T) + 1$, and the equality follows.

Conversely, let T be a nontrivial tree with $\chi_d(T) = \gamma(T) + 1$. Suppose T is a star of center vertex, say x. Then $D = \{x\}$ is a minimum dominating set, where $D_I = \emptyset$ and clearly the set $V(T) - (D_R \cup N(D_I))$ that consists of the set of leaves of the star is independent. Therefore the theorem is valid. Now assume that T is a tree different from a star and let us consider a dominator coloring with $\chi_d(T)$ colors such that all leaves of T have the same color. Note that such a dominator coloring exists by Lemma 3. Also by Lemma 7, $A \cup DS$ is a minimum dominating set of T. Let $D = A \cup DS$, $D_I = \{v \in D : pn(v, D) = \{v\}\}$ and $D_R = D - D_I$. We shall show that every vertex in $V(T) - (A \cup B)$ is adjacent to a vertex of D_I, that is $V(T) - (A \cup B) \subset N(D_I)$. Let x be any vertex of a color class $S \in C_S$. Let x_S be a vertex of DS that dominates S. By Lemma 7-(d), $x_S \in B$. Recall that $x_S \in D$ since $x_S \in DS$. It is well known by Ore's theorem (see [7]) that every vertex in a minimum dominating set has a private neighborhood. Suppose that $x^* \neq x_S$ is a private neighbor of x_S with respect to D. Clearly $x^* \notin D$ for otherwise $D - \{x^*\}$ would be a dominating set smaller than D, a contradiction. Therefore x^* does not dominate a color class of C_S (else $x^* \in DS \subset D$). Also since $x^* \in pn(x_S, D)$, x^* has no neighbor in A but then x^* does not dominate any color class, a contradiction. Consequently x_S has no private neighbor other than itself, that is $x_S \in D_I$. Thus $V(T) - (A \cup B) \subset N(D_I)$. It follows now that all vertices of $V(T) - (D_R \cup N[D_I]) \subseteq B$ and since B is an independent set we are done. □

3 Caterpillars

A caterpillar is a tree in which every vertex of degree at least three has at most two non-leaf neighbors. As was noted in the introduction, there is no polynomial time algorithm that computes the dominator chromatic number for the class of trees. It was even mentioned by Gera et al. in [4] that an efficient algorithm for computing χ_d of an arbitrary caterpillar would be a worthwhile contribution. Our aim in this section is to give a characterization of caterpillars T with $\chi_d(T) = \gamma(T) + 1$. Using a result of Volkmann [8] (see Theorem 8), one can check easily whether a caterpillar satisfies $\chi_d(T) = \gamma(T) + 1$ or $\chi_d(T) = \gamma(T) + 2$.

A *vertex cover* in a graph G is a set of vertices that covers all edges of G. The minimum cardinality of a vertex cover in a graph G is called the covering number of G and is denoted by $\alpha_0(G)$. It is well known that a set D of vertices of G is a vertex cover if and only if $V(G) - D$ is independent. Also every vertex cover set is a dominating set. The following result of Volkmann gives a characterization of nontrivial trees T with equal domination and vertex covering numbers.

Theorem 8 (Volkmann [8]) *A nontrivial tree satisfies $\gamma(T) = \alpha_0(T)$ if and only if each component in the graph resulting from G by removing the set of leaves and their stems is an isolated vertex or a star, where the centers of these stars are not adjacent to any stem in T.*

Now we are ready to state the following result.

Proposition 9 *let T be a nontrivial caterpillar. Then $\chi_d(T) = \gamma(T) + 1$ if and only if $\gamma(T) = \alpha_0(T)$.*

Proof. Let T be a caterpillar with $\gamma(T) = \alpha_0(T)$. Let Dbe any minimum transversal. Then color the vertices of D so that each vertex has a unique color and the remaining vertices of T by a new color. Then $\gamma(T)+1 \leq \chi_d(T) \leq |D|+1 = \gamma(T)+1$, and the equality follows.

Now assume that T is a caterpillar with $\chi_d(T) = \gamma(T) + 1$. By Theorem 1, T admits a minimum dominating set D such that $T\backslash(D_R, N[D_I])$ is independent. Assume that $V(T) - D$ is not independent and let u, v be any two adjacent vertices in $V(T) - D$. Clearly since D contains either a leaf or its stem, neither u nor v is a leaf. First, assume that u and v are not stems. Let d_1 and d_2 be two vertices in D such that d_1, u, v, d_2 induce a path P_4. Then d_1 is the unique neighbor of u in D for otherwise u would be a support vertex. Likewise d_2 is the unique neighbor of v in D. Hence d_1 and d_2 belong to D_R and so $T\backslash(D_R, N[D_I])$ is not an independent set, contradicting Theorem 1. Hence at least u or v is a stem. Without loss of generality, assume that u is a stem and let f be its leaf. Since f belongs to D, f is the unique leaf adjacent to u. Let us modify D as follows: $D' = \{u\} \cup D\backslash\{f\}$. Clearly D' remains a minimum dominating set for T with less edges in $V(T) - D'$. This procedure

can be repeated for every two adjacent vertices not in the current $\gamma(T)$-set until we obtain a $\gamma(T)$-set S for which $V(T) - S$ has no two adjacent vertices. Therefore $\gamma(T) = \alpha_0(T)$. □

According to Theorem 8, Proposition 9 can be also stated as follows.

Proposition 10 *let T be a nontrivial caterpillar. Then $\chi_d(T) = \gamma(T) + 1$ if and only if T is a star or the distance between any two consecutive stems is $1, 2$ or 4.*

References

[1] M. Chellali and F. Maffray, Dominator colorings in some classes of graphs. *Graphs and Combinatorics*, to appear.

[2] R. Gera. On the dominator colorings in bipartite graphs. *Information Technology: New Generations* (2007) 947–952.

[3] R. Gera. On dominator colorings in graphs. *Graph Theory Notes of New York* LII (2007) 25–30.

[4] R. Gera, S. Horton, C. Rasmussen. Dominator colorings and safe clique partitions. *Congressus Numerantium* 181 (2006) 19–32.

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

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

[7] O. Ore, Theory of Graphs, *Amer. Math. Soc. Colloq. Publ.* **38** 1962.

[8] L. Volkmann, On graphs with equal domination and covering numbers. *Discrete Applied Mathematics* 51 (1994) 211–217.

La b-coloration des graphes Spider complets

Zoham Zemir et Mostafa Blidia
LAMDA-RO, Dept. Mathematics,
University of Blida, B.P. 270, Blida, Algeria.
E-mail: zohaze@yahoo.fr, m_blidia@yahoo.fr

Résumé

Soit $G = (V, E)$ un graphe simple. Une coloration dominante est une coloration propre des sommets de G telle que toute classe de couleur i contient un sommet adjacent à au moins un sommet de chaque classe de couleur $j \neq i$. Ce sommet est dit sommet b-dominant pour la couleur i. Le nombre b-chromatique $b(G)$ est le nombre maximum de classes de couleurs dans une coloration dominante.

On désigne par $L(G)$ et $T(G)$, le graphe des lignes et le graphe total de G respectivement. Le nombre b-arête-chromatique de G est le nombre b-chromatique du graphe $L(G)$ et le nombre b-chromatique total de G est le nombre b-chromatique de $T(G)$.

Un Spider est un arbre avec au plus un sommet de degré supérieur à deux, appelé centre du Spider noté o. Une branche d'un Spider est une chaîne issue du centre o vers un sommet de degré un. Si $m \geq 3$ est un entier on définit un graphe de Spider S_m comme un graphe obtenu à partir d'un Spider avec m branches $P_1, P_2, ..., P_m$ toutes de longueur au moins deux telles que deux sommets x et y appartenant à deux branches différentes P_i et P_j sont adjacents si $|i - j| \in \{1, m - 1\}$, et $d(x, o) = d(y, o)$ où $d(x, y)$ est la distance (longueur d'une plus courte chaîne) de x à y. Si toutes les branches sont de même longueur n, alors le graphe de Spider S_m est dit graphe de Spider complet qu'on notera $S_{m,n}$ où $m \geq 3$ et $n \geq 2$ sont des entiers.

Nous déterminons dans ce papier, le nombre b-chromatique, le nombre b-arête-chromatique et enfin le nombre b-chromatique total pour les graphes Spider complets $S_{m,n}$ où $m \geq 1$ et $n \geq 1$.

Keywords: b-coloration, Graphe des lignes, Graphe total, Graphe de Spider.

1 Introduction

Soit $G = (V, E)$ un graphe simple, avec V et E ensemble des sommets et arêtes respectivement. Le voisinage d'un sommet $v \in V$ est $N_G(v) = \{u \in V \ / \ uv \in E\}$. Le degré d'un sommet v de G est $d(v) = |N(v)|$.On note une chaîne sans cordes avec n sommets par P_n. La distance du sommet x au sommet y dans le graphe G est la longueur d'une plus courte chaîne de x à y, $d(x, y)$ désigne la distance de x à y. On désigne par $L(G)$, le graphe dont l'ensemble des sommets est l'ensemble des arêtes et deux sommets sont adjacentes dans $L(G)$ si les arêtes correspondantes sont adjacentes dans G.

Une coloration propre est une application c de V dans IN telle que si deux sommets x et y sont adjacents, alors leurs couleurs correspondantes sont différentes (i.e $c(x) \neq c(y)$). Une classe de couleur i est un ensemble de sommets stable de V colorés avec la même couleur i. Le nombre minimum de classes de couleurs qui partitionnent l'ensemble V est le nombre chromatique, noté $\chi(G)$ et une $\chi(G)$-coloration est une coloration propre des sommets avec $\chi(G)$ couleurs.

Le fait qu'on ne peut fusionner deux classes de couleurs différentes dans une coloration minimum, a inspiré Irving et Manlove [2, 3] à introduire un autre procédé de coloration, en effet une autre manière de réduire le nombre de couleurs est d'essayer à partir d'une coloration propre donnée du graphe de réduire le nombre de classes de couleurs en transférant les sommets d'une même classe de couleur dans les autres classes de couleurs. D'où la définition de la coloration dominante.

Une coloration dominante est une coloration propre telle que toute classe de couleur i contient un sommet adjacent à au moins un sommet de chaque classe de couleur $j \neq i$. Ce sommet est dit sommet b-dominant pour la couleur i.

Le nombre b-chromatique $b(G)$ est le nombre maximum de classes de couleurs dans une coloration dominante. Une coloration dominante avec $b(G)$ couleurs est dite b-coloration.

Ce concept fùt introduit en 1999 par Irving et Manloveoù ils montrent que la détermination de $b(G)$ est un problème NP-dur en général mais polynomial dans le cas des arbres. D'autre part Kratochvil, Tuza et Voigt [6] montrent que la détermination de $b(G)$ est un problème NP-dur même dans le cas des graphes bipartis.

Aussi, une borne supérieure triviale de $b(G)$ est $\Delta(G) + 1$ où $\Delta(G)$ est le degré maximum de G.

Il est important de rappeler que si $v_1, v_2, ..., v_n$ sont tels que $d(v_1) \geq d(v_2) \geq ... \geq d(v_n)$ et si on note par $m(G) = max\{i \ / \ d(v_i) \geq i - 1\}$,

alors $b(G) \leq m(G)$; dans d'autres termes pour qu'un graphe G admette une coloration dominante avec k couleurs, il faut qu'il ait au moins k sommets de degré supérieur ou égal à $k-1$ [2, 3].

On appelle graphe total du graphe G, noté $T(G)$, le graphe dont l'ensemble des sommets est l'ensemble $V \cup E$ et deux sommets dans $T(G)$ sont adjacents s'ils sont deux sommets adjacents dans G, ou bien deux arêtes adjacentes dans G ou bien un sommet et une arête incidente à ce sommet dans G. Si v est un sommet de G, alors le degré de v dans $T(G)$ est $d_T(v) = 2d(v)$. Si $x = uv$ est une arête de G, alors on a $d_T(x) = d(u) + d(v)$. Si G est un graphe à p sommets et q arêtes, tels que le sommet noté v_i est de degré d_i; $i = 1, ..., p$, alors $T(G)$ est le graphe à $p_T = p + q$ sommets et à $q_T = 2q + \frac{1}{2}\sum d_i^2$ arêtes.

Une coloration totale du graphe G est une application $c : V \cup E \longmapsto K$, où K définit un ensemble de couleurs, c est une coloration totale de G si non seulement deux sommets voisins et deux arêtes ajacentes ont des couleurs différentes mais aussi la couleur d'une arête quelconque est différente de celle de ses extrémités. G est dit k-total colorable s'il admet une coloration propre totale c où l'application; $c : V \cup E \longmapsto K$ est telle que $|K| = k$. Le nombre total chromatique noté $\chi_T(G)$ est le plus petit k tel que G soit k-total colorable. Une coloration total dans G correspond à une coloration propre dans le graphe total $T(G)$. Un sous ensemble de sommets et d'arêtes colorés avec la même couleur correspond à un stable dans le graphe total $T(G)$ et à l'union d'un stable et un couplage dans le graphe initial G.

Une coloration dominante des sommets (resp. arêtes) de G ou b-coloration (resp. b'-coloration) de G est une coloration propre c des sommets (resp. arêtes) telle que V (resp. E) peut être partitionné en stables (resp. couplage) dites classes de couleurs et que toute classe de couleur i contient au moins un sommet (resp. une arête) admettant un sommet voisin (resp. une arête voisine) dans toute autre classe, un tel sommet (resp. arête) est dit(resp. dite) sommet b-dominant (resp. arête b-dominante). G est dit k-b-sommet (resp. k-b-arête) colorable s'il admet une coloration dominante des sommets (resp. arêtes) telle que l'application : $c : V \longmapsto K$, est telle que $|K| = k$ (resp. $c : E \to K$,est telle que $|K| = k$). Le nombre b-sommet chromatique est aussi le nombre b-chromatique noté $b(G)$. Le nombre b-arête chromatique dit aussi le nombre $b'(G)$-chromatique est le nombre maximum k de couleurs utilisées pour que G soit k-b-arête colorable. Déterminer le nombre b'-chromatique dans le graphe G, revient à déterminer le nombre b-chromatique $b(L(G))$ dans le graphe des arêtes $L(G)$.

Une coloration dominante totale de G est une coloration propre dans $T(G)$ qui est dominante; c'est à dire telle que toute classe de couleur i dans

le graphe total $T(G)$ contient un sommet adjacent à au moins un sommet de chaque classe de couleur $j \neq i$, ce sommet est dit sommet b-dominant pour la couleur i dans $T(G)$. Le nombre b_t-chromatique dit aussi b-chromatique total est le nombre maximum de classes de couleurs dans une coloration dominante dans $T(G)$. Une coloration dominante avec $b_t(G)$ couleurs est notée b_t-coloration.

L'absence d'algorithme polynomial pour la détermination d'un paramètre donné dans un graphe ne cesse d'inciter les chercheurs à établir des bornes qui encadrent le plus possible le paramètre en question ou de déterminer la valeur exacte de ce paramètre pour certaines classes de graphes .

Un Spider est un arbre avec au plus un sommet de degré supérieur à deux, appelé centre du Spider noté o. (Si aucun sommet n'est de degré plus de deux, tout sommet du spider peut être un centre). Une branche d'un Spider est une chaîne issue du centre o vers un sommet de degré un. Une étoile avec k branches est un Spider avec k branches toutes de longueur 1.

Si $m \geq 3$ est un entier, on définit un graphe de Spider S_m comme un graphe obtenu à partir d'un Spider avec m branches $P_1, P_2, ..., P_m$ toutes de longueur au moins deux telles que deux sommets x et y appartenant à deux branches différentes P_i et P_j sont adjacents si $|i-j| \in \{1, m-1\}$, et $d(x,o) = d(y,o)$ où o est le centre du Spider.

Nous considérons ici une sous classe de la classe des graphes S_m. Soit S_m un graphe de Spider ayant m branches $P_1, P_2, ..., P_m$. Si toutes les branches sont de même longueur n et si pour tous sommets v_i, v_j appartenant successivement à P_i, P_j tels que $|i-j| \in \{1, m-1\}$, on a v_i, v_j sont adjacents, alors le graphe Spider S_m est dit graphe Spider complet qu'on notera $S_{m,n}$ où $m \geq 3$ et $n \geq 2$ sont des entiers (voir Figure 1).

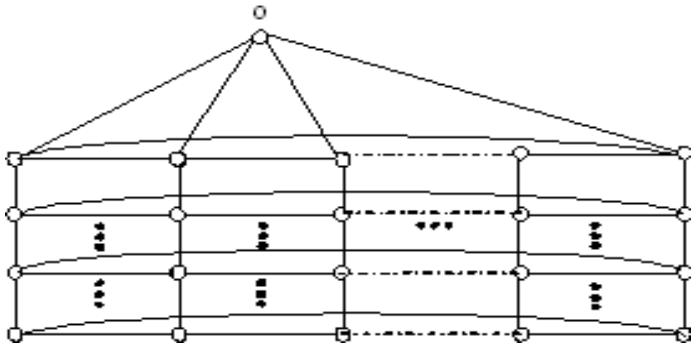


Figure 1: $S_{m,n}$

Notons que dans la suite, on généralise la définition de $S_{m,n}$ pour le cas où $m \geq 1$ et $n \geq 1$. Sans compter le sommet centre o qui est dans le niveau zéro, on a n niveaux (lignes) et m colonnes. On note par v_{ij} le sommet de $S_{m,n}$ qui est dans le niveau i et la colonne j. On note par $m(S_{m,n})$ le m-degré de $S_{m,n}$. On utilisera souvent le fait que $b(G) \leq m(G)$.

$S_{1,n}$ est la chaîne P_{n+1}. $S_{2,1}$ est le triangle et $S_{2,n}$ est le graphe de la Figure 2

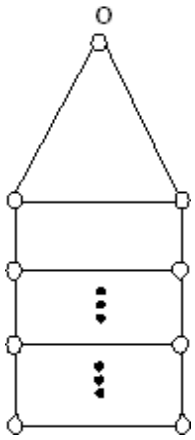


Figure 2: $S_{2,n}$

Sadegh Rahimi Sharebaf dans [7] ont déterminés le nombre sommet chromatique (le nombre chromatique classique), le nombre arête chromatique (l'indice chromatique classique), et enfin le nombre chromatique total dans les graphes de Spider complets.

Theorem 1 *[7]* $\chi(S_{m,n}) = 3$ *si* n *est pair, et* $\chi(S_{m,n}) = 4$ *si* m *est impair.*

Theorem 2 *[7]* $\chi'(S_{m,n}) = \Delta(G)$.

Theorem 3 *[7]* $\chi_t(S_{m,n}) = \Delta(G) + 2$ *si* $m = 3, 4$.

Theorem 4 *[7]* $\chi_t(S_{m,n}) = \Delta(G) + 1$ *si* $m \geq 5$.

Dans ce papier, nous déterminons le nombre b-chromatique (b-sommet-chromatique), le nombre b'-chromatique (b-arête chromatique) et enfin le nombre b_t-chromatique (b-chromatique total) pour les graphes Spider complets.

2 Le nombre b-chromatique dans les graphes Spider complets.

Theorem 5 *Soit $S_{m,n}$ un graphe Spider complet où m et n sont des entiers positifs. Alors on a :*

$$b(S_{m,n}) = \begin{cases} \mathbf{2} & si \ \ (m,n) \in \{(m,n) \ / \ m = 1 \ et \ 1 \leq n \leq 3\} \\ \mathbf{3} & si \ (m,n) \in \{(m,n) \ / \ m = 1 \ et \ n \geq 4\} \cup \\ & \qquad \{(2,1),(2,2),(2,3),(4,1)\} \\ \mathbf{4} & si \ \ (m,n) \in \{(m,n) \ / \ m = 2 \ et \ n \geq 4\} \cup \\ & \qquad \{(m,n) \ / \ m \geq 5 \ et \ n = 1 \ \} \cup \\ & \qquad \{(m,n) \ / \ m = 3 \ et \ 1 \leq n \leq 4 \ \} \ \cup \\ & \qquad \{(3,2),(3,3),(3,4)\} \\ \mathbf{5} & si \ (m,n) \in \{(m,n) \ / \ m = 3 \ et \ n \geq 5\} \cup \\ & \qquad \{(m,n) \ / \ m \geq 4 \ et \ n \geq 2\} \end{cases}$$

Preuve: Si $m = 1$, le graphe $S_{1,n}$ est réduit à une chaîne P_{n+1}, où $b(S_{1,n}) = 2$,pour $1 \leq n \leq 3$ et $b(S_{1,n}) = 3$, pour $n \geq 4$ (voir [**?**]).

Si $m = 2$, $m(S_{2,1}) = m(S_{2,2}) = b\,(S_{2,1}) = b(S_{2,2}) = 3$, il suffit de choisir les sommets du triangle dont un est le centre o, comme sommets b-dominants. $m(S_{2,3}) = 4$, supposons qu'on puisse faire une b-coloration utilisant quatre couleurs, les seuls sommets pouvant être b-dominants sont $v_{i1}, v_{i2}, 1 \leq i \leq 2$, il est simple de vérifier que seuls trois d'entre eux peuvent l'être. On peut voir que dans ce cas $b\ (S_{2,3}) = 3$, il suffit de choisir les sommets du triangle dont un est le centre o, comme sommets b-dominants. $m(S_{2,n}) = b(S_{2,n}) = 4$ si $n \geq 4$, choisir pour cela les sommets $v_{i1}, v_{i2}, 2 \leq i \leq 3$ comme sommets b-dominants, en donnant la couleur 1 à v_{21} et v_{42}, la couleur 2 à v_{22} et v_{41}, la couleur 3 à v_{12} et v_{31} et la couleur 4 à v_{11} et v_{32}. Ensuite on étend la coloration aux autres sommets non colorés, ceci est toujours possible vu que dans le voisinage de chaque sommet il y a au moins une couleur manquante.

Si $m = 3$, remarquer que si $n = i, 1 \leq i \leq 4$, le graphe $S_{3,n}$, contient i triangles ne contenant pas le centre o.

- $n = 1$ et $2, S_{3,1} = K_4$, $m(S_{3,n}) = b\,(S_{3,n}) = 4$, les sommets du K_4 sont choisis comme sommets b-dominants.

- $n = 3$, $m(S_{3,n}) = 5$, supposons qu'on puisse faire une b-coloration utilisant cinq couleurs , il est simple de vérifier qu'on ne peut avoir plus d'un sommet b-dominant dans le niveau 1, donc on ne peut avoir plus de 4 sommets b-dominants (en comptant au plus trois sommets b-dominants dans le niveau deux), par conséquent $b\,(S_{3,n}) = 4$, il suffit pour celà de prendre les

sommets du K_4 (formé par o et les sommets du triangle du premier niveau) comme sommets b-dominants.

- $n = 4$, $m(S_{3,n}) = 5$, supposons qu'on puisse faire une b-coloration avec cinq couleurs, comme dans le cas précédent on ne peut avoir plus d'un sommet b-dominant dans le niveau 1. Supposons qu'on a trois sommets b-dominants dans le niveau 2, on peut voir que le troisième sommet a toujours deux sommets de même couleur dans son voisinage, ce qui conduit à une impossibillité. En utilisant le même argument on peut vérifier aussi que dans le niveau trois on ne peut avoir plus de deux sommets b-dominants. Enfin en examinant d'une manière exhaustive tous les cas possibles lorsque le premier niveau contient un sommet b-dominant, le second niveau contient deux sommets b-dominants et le niveau trois contient deux sommets b-dominants, on arrive à une impossibilité. Donc $b\,(S_{3,4}) = 4$.Choisir les sommets du K_4 comme sommets b-dominants.

- $n \geq 5$, $m(S_{3,n}) = b\,(S_{3,n}) = 5$, le centre ne peut être b-dominant, car il est de degré 3 , il suffit de donner la couleur 1 au sommet v_{11}, la couleur 2 au sommet v_{21}, la couleur 3 au sommet v_{31}, la couleur 4 au sommet v_{41}, et enfin la couleur 5 au sommet v_{32} et pour les rendre b-dominants on colore successivement les sommets $o, v_{12}, v_{13}, v_{22}, v_{23}, v_{33}, v_{42}, v_{43}, v_{51}$ avec les couleurs $5, 3, 4, 4, 5, 1, 2, 5, 1$. Ensuite on étend la coloration aux autres sommets non colorés, ceci est toujours possible vu que dans le voisinage de chaque sommet il y a au moins une couleur manquante.

Si $m = 4$.

- $n = 1$, $m(S_{4,n}) = 4$, supposons qu'on puisse faire une b-coloration du graphe en utilisant quatre couleurs. Si on colore les sommets $v_{1i}, 1 \leq i \leq 4$ avec les couleurs $1, 2, 3, 4$,alors le centre o ne peut prendre aucune couleur. Si on choisit le centre o et 3 autres sommets parmi les $v_{1i}, 1 \leq i \leq 4$, ceci est impossible car dans le niveau un deux sommets seulement peuvent être b-dominants. Donc $b(S_{4,n}) = 3$. Pour faire une b-coloration utilisant trois couleurs choisir trois sommets sur un même triangle pris arbitrairement.

- $n \geq 2$, $m(S_{4,n}) = b(S_{4,n}) = 5$, il suffit de prendre le centre o et les sommets $v_{i2}, 1 \leq i \leq 4$, comme sommets b-dominants.

Si $m = 5$.

- $n = 1, m(S_{5,n}) = b(S_{5,n}) = 4$, choisir les sommets $o, v_{11}, v_{12}, v_{13}$ comme sommets b-dominants et leur donner les couleurs $4, 1, 2, 3$ successivement; ensuite colorer respectivement les sommets v_{14}, v_{15} avec les couleurs $1, 3$.

- $n = 2$, $m(S_{5,n}) = 5$, supposons qu'on puisse faire une b-coloration du graphe en utilisant cinq couleurs.Si on prend comme sommets b-dominants $v_{1i}, 1 \leq i \leq 5$, le centre o ne peut prendre aucune couleur. Supposons que le centre o est un sommet b-dominant, on peut vérifier qu'au plus trois

sommets dans le niveau un peuvent être b-dominants, donc il est impossible d'avoir cinq sommets b-dominants vu que les sommets $v_{2i}, 1 \leq i \leq 5$ sont de degré 3, donc ne peuvent pas être choisis b-dominants. Donc $b(S_{5,n}) \leq 4$ et on donne une b-coloration utilisant quatre couleurs, il suffit de prendre comme sommets b-dominants $o, v_{11}, v_{12}, v_{13}$ en leur donnant respectivement les couleurs $4, 1, 2, 3$. Colorer respectivement les sommets v_{14}, v_{15} avec les couleurs $1, 3$ et ensuite étendre aisément la coloration.

- $n \geq 3$, $m(S_{5,n}) = b(S_{5,n}) = 5$, dans ce cas les sommets $o, v_{1i}; 1 \leq i \leq 2, v_{2i}; 1 \leq i \leq 2$, sont choisis b-dominants de couleurs respectives $1, 2, 3, 5, 4$. On donne aux sommets $v_{13}, v_{15}, v_{23}, v_{25}, v_{31}, v_{32}$ respectivement les couleurs $4, 5, 1, 3, 1, 2$. On peut étendre ensuite cette coloration partielle à tout le graphe puisque tout autre sommet est de degré au plus quatre.

Si $m \geq 6$ et $n = 1$, $m(S_{m,n}) = b(S_{m,n}) = 4$, il suffit de prendre comme sommets b-dominants $o, v_{1i}; 1 \leq i \leq 3$ en leur donnant respectivement les couleurs $1, 2, 3, 4$ et en donnant les couleurs 2 et 4 aux sommets v_{14} et v_{1m} respectivement. On peut étendre ensuite cette coloration partielle à tout le graphe puisque tout autre sommet est de degré au plus trois.

Si $m \geq 6$ et $n \geq 2$, $m(S_{m,n}) = b(S_{m,n}) = 5$, il suffit de prendre comme sommets b-dominants $o, v_{1i}; 1 \leq i \leq 4$ en leur donnant respectivement les couleurs $1, 2, 3, 4, 5$ et en donnant aux sommets $v_{15}, v_{1m}, v_2, v_{22}, v_{23}, v_{24}$ respectivement les couleurs $2, 5, 4, 5, 2, 3$. On peut étendre ensuite cette coloration partielle à tout le graphe puisque tout autre sommet est de degré au plus quatre. ■

3 Le nombre b'-chromatique (ou b-indice chromatique) dans les graphes Spider complets

Dans la suite on note par $m'(G)$ le m-degre du graphe $L(G)$.

Theorem 6 *Soit $S_{m,n}$ un graphe Spider complet où m et n sont des entiers positifs. Alors on a:*

$$b'(S_{m,n}) = \begin{cases} \mathbf{1} & si\ (m,n) = (1,1) \\ \mathbf{2} & si\ \ (m,n) \in \{(m,n)\ /\ m = 1\ et\ 2 \leq n \leq 4\} \\ \mathbf{3} & si\ \ \ (m,n) \in \{(m,n)\ /\ m = 1\ et\ n \geq 5\} \cup \{(2,1),(3,1)\} \\ \mathbf{4} & si\ \ \ \ (m,n) = \{(2,2),(2,3),(2,4)\} \\ \mathbf{5} & si\ (m,n) \in \{(m,n)\ /\ m = 2\ et\ n \geq 5\} \cup \{(4,1),(5,1)\} \\ \mathbf{6} & si\ \ \ \ (m,n) \in \{(3,2),(3,3),(3,4),(4,2),(5,2),(6,1)\} \\ \mathbf{7} & si \begin{cases} (m,n) \in \{(m,n)\ /\ m = 3\ et\ n \geq 5\} \cup \\ \quad \{(m,n)\ /\ m = 4\ et\ n \geq 3\} \cup \\ \quad \{(m,n)/m = 5\ et\ n \geq 3\} \cup \\ \quad \{(m,n)/m = 6\ et\ n \geq 2\} \end{cases} \\ \mathbf{m} & si\ \ \ \ (m,n) \in \{m \geq 7\ et\ n \geq 1\} \end{cases}$$

Preuve: Si $m = 1$, le graphe $S_{1,n}$ est reduit à une chaîne P_{n+1}. $b'(S_{1,1}) = 1$ et $b'(S_{1,n}) = 2$ si $n = 2, 3, 4$; $b'(S_{1,n}) = 3$ si $n \geq 5$.

Si $m = 2$.

- $n = 1$, $S_{2,1}$ est un triangle et on a: $m'(S_{2,1}) = b'(S_{2,1}) = 3$.

- $n = 2$ ou 3, $m'(S_{2,n}) = b'(S_{2,n}) = 4$, il suffit de colorer les arêtes du triangle $1, 2, 3$ arbitrairement, les arêtes (v_{11}, v_{21}) et (v_{12}, v_{22}) avec la couleur 4 et l'arête (v_{21}, v_{22}) avec la couleur 2, ainsi les arêtes du triangle avec l'arête (v_{11}, v_{21}) sont b'-dominantes.

- $n = 4$, $m'(S_{2,n}) = 5$, en citant d'une manière exhaustive tous les cas possible, on peut voir qu'on ne peut faire une 5-b'-coloration. Donc $b'(G) = 4$, il suffit pour celà de prendre la même b'-coloration que précédemment.

- $n \geq 5$, $m'(S_{2,n}) = b'(S_{2,n}) = 5$, il suffit de colorer les arêtes (o, v_{12}) , (v_{41}, v_{42}) et (v_{11}, v_{21}) avec la couleur 1,les arêtes (o, v_{12}) , (v_{12}, v_{22}) et (v_{31}, v_{32}) avec la couleur 2, les arêtes (v_{11}, v_{12}) , (v_{31}, v_{41}) et (v_{42}, v_{52}) avec la couleur 3, les arêtes (v_{21}, v_{22}) , (v_{32}, v_{42}) et (v_{41}, v_{52}) avec la couleur 4, les arêtes (v_{21}, v_{31}) et (v_{22}, v_{32}) avec la couleur 5, ainsi les arêtes (v_{11}, v_{21}), $(v_{12}, v_{22}), (v_{31}, v_{41}), (v_{32}, v_{42})$ et (v_{21}, v_{31}) sont b'-dominantes.

Si $m = 3$.

- $n = 1$, $S_{3,1}$ est le graphe complet K_4 et $m'(S_{3,1}) = 5$, supposer qu'on fait une une b'-coloration utilisant 5 couleurs, colorer 5 arêtes, sans perdre de généralité $(o, v_{1i}), 1 \leq i \leq 3$, (v_{11}, v_{12}) et (v_{12}, v_{13}) sont colorées successivement avec les couleurs $1, 2, 3, 4, 5$ remarquer que seule l'arête de couleur 2 peut être b'-dominante. Alors que $b'(S_{3,1}) \leq 4$, mais là aussi par un raisonnement similaire, si on fait une b'-coloration utilisant 4 couleurs, on colore sans perdre de generalité $(o, v_{1i}), 1 \leq i \leq 3$, (v_{11}, v_{12}) arêtes arbitrairement avec les couleurs $1, 2, 3, 4$, remarquer que l'arête de couleur 3 ne peut être b'-dominante. Donc $b'(S_{3,1}) = 3$.Une b'-coloration avec 3 couleurs est donnée en choisissant trois arêtes b'-dominantes dans un même triangle, et sans

perdre de généralité, soit $(o, v_{1i}), 1 \leq i \leq 2$ et (v_{11}, v_{12}) arêtes b'-dominantes colorées respectivement $1, 2, 3$, l'arête (o, v_{13}) colorée 3, (v_{12}, v_{13}) colorée 1 et (v_{11}, v_{13}) colorée 2.

- $n = 2$, $m'(S_{3,2}) = b'(S_{3,2}) = 6$, soient les arêtes $(o, v_{1i}), 1 \leq i \leq 3, (v_{11}, v_{21}), (v_{12}, v_{22}), (v_{13}, v_{23})$ colorées respectivement $1, 2, 3, 4, 6, 5$ et on complète la coloration pour les rendre b'-dominantes comme suit: on donne la couleur 5 à l'arête (v_{11}, v_{12}), la couleur 4 à l'arête (v_{12}, v_{13}), la couleur 6 à l'arête (v_{11}, v_{13}), la couleur 3 à l'arête (v_{21}, v_{22}), la couleur 1 à l'arête (v_{22}, v_{23}) et la couleur 2 à l'arête (v_{21}, v_{23}).

- $n = 3$, $m'(S_{3,3}) = 7$, supposons qu'on puisse faire une b'-coloration utilisant 7 couleurs, seules les 9 arêtes peuvent être b-dominantes; les arêtes du niveau 1 et 2 et celles qui sont entre les deux. Constater que deux au plus des 3 arêtes du niveau 1 peuvent être b-dominantes; et deux au plus dans le niveau 2, ou entre les deux niveaux. Donc $b'(S_{3,3}) = 6$; choisir pour cela la même coloration que le cas précédent et comme les arêtes restantes sont adjacentes à au plus 5 autres alors il est possible de finir la coloration proprement.

- $n = 4$, $m'(S_{3,n}) = 7$, les seules arêtes qui peuvent être b'-dominantes dans une coloration à 7 couleurs, sont celles des niveaux $1, 2, 3$ et celles qui sont entre ces trois niveaux; de manière similaire à celle adoptée dans le cas précédent on fait remarquer que deux arêtes peuvent être b'-dominantes dans le niveau 1, deux autres dans le niveau 2 ou entre les niveaux 1 et 2 et deux autres dans le niveau 3 ou entre les niveaux 2 et 3. Donc $b'(S_{3,n}) = 6$, choisir pour cela la même coloration que le cas précédent, dans ce qui suit on colore les arêtes ayant 6 autres arêtes dans leurs voisinage de la manière suivante: affecter la couleur 1 aux arêtes (v_{21}, v_{31}) et (v_{32}, v_{33}), la couleur 2 aux arêtes (v_{22}, v_{32}) et (v_{31}, v_{33}), la couleur 3 aux arêtes (v_{23}, v_{33}) et (v_{31}, v_{32}) et comme les arêtes restantes sont adjacentes à au plus 5 autres alors il est possible de finir la coloration proprement.

- $n \geq 5$, $m'(S_{3,n}) = b'(S_{3,n}) = 7$; dans une coloration b'-dominante utilisant 7 couleurs, il suffit de choisir les arêtes suivantes comme arêtes b'-dominantes (v_{11}, v_{12}), (v_{12}, v_{13}), (v_{21}, v_{22}), (v_{22}, v_{23}), (v_{32}, v_{33}), (v_{41}, v_{42}), (v_{41}, v_{43}) de couleurs respectives $1, 2, 3, 5, 4, 6, 7$ et compléter comme suit pour les satisfaire: (v_{21}, v_{23}) et (v_{31}, v_{33}) colorées avec la couleur 1, (v_{22}, v_{32}) et (v_{41}, v_{51}) colorées avec la couleur 2, $(o, v_{12}), (v_{32}, v_{42})$ et (v_{43}, v_{53}) colorées avec la couleur 3, $(o, v_{11}), (v_{13}, v_{23}), (v_{21}, v_{31})$ et (v_{42}, v_{43}) colorées avec la couleur 4, $(v_{11}, v_{13}), (v_{33}, v_{43})$ et (v_{42}, v_{52}) colorées avec la couleur 5, $(o, v_{13}), (v_{11}, v_{21})$ et (v_{23}, v_{33}) colorées avec la couleur 6, (v_{12}, v_{22}) et (v_{31}, v_{32}) colorées avec la couleur 7. Comme les arêtes restantes sont adjacentes à au plus 6 autres alors il est possible d'étendre la coloration pour tout le graphe

puisqu'il existe toujours une couleur disponible pour toute arête considérée.

Si $m = 4$

- $n = 1$, $m'(S_{4,n}) = b'(S_{4,n}) = 5$.Pour faire une b'-coloration utilisant 5 couleurs, il suffit de prendre $(o, v_{1i}), 1 \leq i \leq 4, ,$ et (v_{12}, v_{13}) comme arêtes b'-dominantes respectivement colorées $1, 2, 3, 4, 5$ et pour les satisfaire, on colore $(v_{11}, v_{12}), (v_{13}, v_{14}), (v_{11}, v_{14})$ respectivement $4, 1, 5$.

- $n = 2$, $m'(S_{4,n}) = 7$, supposons qu'on puisse faire une b'-coloration utilisant 7 couleurs, les arêtes pouvant être b'-dominantes sont $(o, v_{i1}), 1 \leq i \leq 4, (v_{11}, v_{12}), (v_{12}, v_{13}), (v_{13}, v_{14}), (v_{11}, v_{14})$; si on choisit trois arêtes b'-dominantes parmi les arêtes $(o, v_{i1}), 1 \leq i \leq 4$, alors aucune arête du niveau 1 ne peut être b'-dominante. Donc $b'(S_{4,n}) = 6$.On donne une b'-coloration utilisant 6 couleurs où les arêtes b'-dominantes sont $(o, v_{i1}), 1 \leq i \leq 4$, $(v_{12}, v_{13}), (v_{12}, v_{22})$, colorées respectivement $1, 2, 3, 4, 5, 6$; pour les satisfaire on colore 4 l'arête (v_{11}, v_{12}); 5 l'arête (v_{11}, v_{14}); 1 les arêtes (v_{13}, v_{23}),(v_{14}, v_{24}), (v_{21}, v_{31}) et ; 6 les arêtes (v_{13}, v_{14}) et (v_{11}, v_{21}) et colorer l'arête (v_{22}, v_{23}) avec la couleur 3; on peut ensuite finir la coloration de manière propre.

- $n \geq 3$, $m'(S_{4,n}) = b'(S_{4,n}) = 7$ et on donne une b'-coloration utilisant 7 couleurs où les arêtes b'-dominantes sont $(o, v_{i1}), 1 \leq i \leq 4$, (v_{13}, v_{23}), (v_{14}, v_{24}), (v_{21}, v_{22}) colorées respectivement $1, 2, 3, 4, 6, 7, 5$ et pour les satisfaire on colore avec la couleur 1 les arêtes $(v_{23}, v_{24}), (v_{22}, v_{32})$, avec la couleur 2 les arêtes $(v_{21}, v_{24}), (v_{23}, v_{33})$; avec la couleur 3 les arêtes $(v_{21}, v_{31}), (v_{24}, v_{34})$; avec la couleur 4 l'arête (v_{22}, v_{23}), avec la couleur 5 les arêtes $(v_{11}, v_{12}), (v_{13}, v_{14})$, avec la couleur 6 les arêtes (v_{12}, v_{22}) et (v_{11}, v_{14}) ; avec la couleur 7 les arêtes (v_{12}, v_{13}) et (v_{11}, v_{21}). Comme les arêtes restantes sont adjacentes à au plus 6 autres alors il est possible d'étendre la coloration pour tout le graphe puisqu'il existe toujours une couleur disponible pour toute arête considérée.

Si $m = 5$.

- $n = 1$, $m'(S_{5,n}) = b'(S_{5,n}) = 5$, dans une b'-coloration utilisant 5 couleurs on considère les arêtes b'-dominantes : $(o, v_{i1}), 1 \leq i \leq 5$; colorées respectivement $1, 2, 3, 4, 5$; pour completer la coloration on colore les arêtes $(v_{11}, v_{12}), (v_{12}, v_{13}), (v_{13}, v_{14}), (v_{14}, v_{15}), (v_{11}, v_{15})$ respectivement avec les couleurs $3, 4, 5, 1, 2$.

- $n = 2$, $m'(S_{5,n}) = 7$, seules les arêtes $(o, v_{i1}) = i, 1 \leq i \leq 5$ et $(v_{1i}, v_{1(i+1)}), (v_{11}, v_{15}), 1 \leq i \leq 4$, peuvent être b'-dominantes. Si on choisit les arêtes $(o, v_{i1}) = i, 1 \leq i \leq 3$, b'-dominantes dans une b'-coloration utilisant 7 couleurs, alors aucune arête de couleur autres que celles déjà choisies ne sera b'-dominante. On a donc $b'(S_{5,2}) = 6$.Il est simple de trouver une b'-coloration utilisant 6 couleurs, en choisissant $(o, v_{i1}), 1 \leq i \leq 5$ colorées successivement $1, 2, 3, 4, 5$ et (v_{13}, v_{14}) colorée 6 comme arêtes b'-dominantes, afin de les satisfaire on colore l'arête (v_{12}, v_{13}) avec la couleur 1, l'arête

(v_{14}, v_{15}) avec la couleur 2, l'arête (v_{13}, v_{23}) avec la couleur 5, les arêtes (v_{11}, v_{15}) et (v_{12}, v_{22}) avec la couleur 6, pour completer cette coloration on colore les arêtes $(v_{11}, v_{12}), (v_{11}, v_{21})$, (v_{14}, v_{24}) et (v_{15}, v_{25}) respectivement avec les couleurs $3, 2, 1, 1$; puis finir la coloration proprement puisqu'il existe toujours une couleur disponible pour toute arête restante considérée.

- $n \geq 3$, $m'(S_{5,n}) = b'(S_{5,n}) = 7$. On donne une b'-coloration utilisant 7 couleurs où les arêtes b'-dominantes sont $(o, v_{i1}), 1 \leq i \leq 5, (v_{14}, v_{24}), (v_{15}, v_{25})$ colorées respectivement $1, 2, 3, 4, 5, 6, 7$, afin de les satisfaire on colore l'arête (v_{14}, v_{15}) avec la couleur 1, les arêtes (v_{23}, v_{24}) et (v_{21}, v_{25}) avec la couleur 2, l'arête (v_{24}, v_{25}) avec la couleur 3, l'arête (v_{25}, v_{35}) avec la couleur 4, l'arête (v_{24}, v_{34}) avec la couleur 5, les arêtes (v_{12}, v_{13}) et (v_{11}, v_{15}) avec la couleur 6, les arêtes (v_{11}, v_{12}) et (v_{13}, v_{14}) avec la couleur 7. Maintenant les arêtes restantes et qui ont 6 autres arêtes dans leur voisinage sont : (v_{11}, v_{21}), $(v_{21}, v_{22}), (v_{12}, v_{22}), (v_{22}, v_{23}), (v_{13}, v_{23})$, colorées respectivement $2, 1, 3, 5, 4$; puis finir la coloration proprement puisqu'il existe toujours une couleur disponible pour toute arête restante considérée.

Si $m = 6$.

- $n = 1$, $m'(S_{6,1}) = b'(S_{m,1}) = 6$. Les arêtes b'-dominantes sont données par $(o, v_{i1}), 1 \leq i \leq 6$.colorées respectivement $1, 2, 3, 4, 5, 6$, puis finir la coloration proprement puisqu'il existe toujours une couleur disponible pour toute arête restante considérée.

- $n \geq 2$, $m'(S_{6,n}) = b'(S_{m,n}) = 7$. Les arêtes b'-dominantes sont données par $(o, v_{i1}), 1 \leq i \leq 6$ et (v_{14}, v_{15}), colorées respectivement $1, 2, 3, 4, 5, 6, 7$, puis finir la coloration proprement puisqu'il existe toujours une couleur dispo-nible pour toute arête restante considérée.

Si $m \geq 7, n \geq 1, m'(S_{m,n}) = b'(S_{m,n}) = m$, Les arêtes b'-dominantes sont données par $(o, v_{i1}), 1 \leq i \leq m$, puis finir la coloration proprement puisqu'il existe toujours une couleur disponible pour toute arête restante considérée.

■

Corollary 1 *Si $m \geq 7, n \geq 1$, alors on a $b'(S_{m,n}) = m = \Delta(L(S_{m,n}))$.*

4 Le nombre b_t-chromatique (ou b-total chromatique) dans les graphes Spider complets

Theorem 7 *Soit $S_{m,n}$ un graphe Spider complet où m et n sont des entiers positifs. Alors on a:*

$$b_t(S_{m,n}) = \begin{cases} \mathbf{3} & si\ (m,n) = \{(1,1),(1,2),(2,1)\} \\ \mathbf{4} & si\ (m,n) \in \{(1,3)\} \\ \mathbf{5} & si\ (m,n) \in \{(m,n)/m = 1\ et\ n \geq 4\} \cup \{(2,2)\} \\ \mathbf{6} & si\ (m,n) \in \{(4,1),(2,3)\} \\ \mathbf{7} & si \begin{cases} (m,n) \in \{(m,n)/m = 2\ et\ n \geq 4\} \\ \cup\{(2,3),(3,1),(3,2),(5,1),(6,1)\} \end{cases} \\ \mathbf{8} & si\ (m,n) \in \{(3,3),(4,2),(5,2),(7,1)\} \\ \mathbf{9} & si \begin{cases} (m,n) \in \{(m,n)/m = 3\ et\ n \geq 4\} \cup \\ \{m = 4\ et\ n \geq 3\} \cup \{m = 5\ et\ n \geq 3\} \cup \\ \{m = 6\ et\ n \geq 2\} \cup \{m = 7\ et\ n \geq 2\} \end{cases} \\ \mathbf{m+1} & si\ (m,n) \in \{(m,n)/m \geq 8\ et\ n \geq 1\} \end{cases}$$

Preuve: Les tableaux établis dans la suite donnent la b_t-coloration de $S_{m,n}$ pour les différentes valeurs de m et n, on utilisera * pour indiquer les sommets ou arêtes b_t-dominants.

Si $m = 1$, le graphe $S_{1,n}$ est réduit à une chaîne P_{n+1} et on a:

- $n = 1$, $m_t(S_{1,n}) = b_t(S_{1,n}) = 3$

sommets ou arêtes	o^*	v_{11}^*	$(o, v_{11})^*$
b_t-coloration	1	2	3

- $n = 2$, $m_t(S_{1,n}) = b_t(S_{1,n}) = 3$

sommets ou arêtes	o^*	v_{11}^*	v_{21}	$(o, v_{11})^*$	(v_{11}, v_{21})
b_t-coloration	1	3	2	2	1

- $n = 3$, $m_t(S_{1,n}) = b_t(S_{1,n}) = 4$

sommets ou arêtes	o	v_{11}^*	v_{21}^*	$(o, v_{11})^*$	$(v_{11}, v_{21})^*$	(v_{21}, v_{31})	v_{31}
b_t-coloration	1	3	1	2	4	2	3

- $n = 4$, $m_t(S_{1,n}) = b_t(S_{1,n}) = 5$

sommets ou arêtes	o	v_{11}^*	v_{12}^*	v_{13}^*	v_{14}	(o, v_{11})	$(v_{11}, v_{21})^*$
b_t-coloration	3	1	5	2	4	2	4

$(v_{21}, v_{31})^*$	(v_{31}, v_{41})
3	1

Si $n \geq 5$, $m_t(S_{1,n}) = b_t(S_{1,n}) = 5$, adopter la même coloration, finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m = 2$.

- $n = 1$, $m_t(S_{2,n}) = 5$ et $b_t(S_{2,n}) = 3$, car $S_{2,n}$ est un triangle, où à un sommet correspond une arête opposée et on ne peut prendre que l'un des deux; arête ou sommet; comme élément b_t-dominant, donc $b_t(S_{2,n}) = 3$.

sommets ou arêtes	o^*	v_{11}^*	v_{12}^*	(o, v_{11})	(o, v_{12})	(v_{11}, v_{12})
b_t-coloration	1	2	3	3	2	1

- $n = 2$, $m_t(S_{2,n}) = 6$. Si on suppose qu'il existe une coloration dominante totale utilisant 6 couleurs, les sommets et arêtes pouvant être b_t-dominantes sont $ov_{11}, ov_{12}, v_{11}, v_{12}$, $(v_{11}, v_{12}), (v_{11}, v_{21}), (v_{12}, v_{22})$ et en examinant tous les cas il est impossible de faire une b_t-coloration avec 6 couleurs. Donc $b_t(S_{2,2}) = 5$.

sommets ou arêtes	o	v_{11}^*	v_{12}	v_{21}	v_{22}	$(o, v_{11})^*$	(o, v_{12})
b_t-coloration	5	1	2	5	4	2	1

$(v_{11}, v_{12})^*$	$(v_{11}, v_{21})^*$	$(v_{12}, v_{22})^*$
3	4	5

- $n = 3$, $m_t(S_{2,n}) = 7$. Supposons qu'il existe une coloration dominante totale utilisant 7 couleurs, les sommets et arêtes pouvant être b_t-dominantes sont $v_{11}, v_{12}, v_{21}, v_{22}$, $(v_{11}, v_{12}), (v_{11}, v_{21}), (v_{21}, v_{22})$,(v_{12}, v_{22}) et si on colore 7 d'entre eux avec les couleurs $1, 2, 3, 4, 5, 6, 7$ et on veut les rendre b_t-dominants il est simple de voir que deux au moins parmi eux ne peuvent pas être b_t-dominants. Donc $b_t(S_{2,3}) = 6$.

sommets ou arêtes	v_{11}^*	v_{21}^*	v_{12}^*	v_{22}^*	v_{32}	$(v_{11}, v_{21})^*$	$(v_{11}, v_{12})^*$
b_t-coloration	1	2	3	4	6	6	5

(o, v_{11})	(o, v_{12})	(v_{21}, v_{31})	(v_{22}, v_{32})	(v_{12}, v_{22})	(v_{21}, v_{22})
4	6	3	1	2	5

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

- $n \geq 4$, $m_t(S_{2,n}) = b_t(S_{2,n}) = 7$.

sommets ou arêtes	o	(o, v_{11})	(o, v_{12})	v_{11}^*	v_{12}^*	(v_{11}, v_{12})	v_{21}	v_{22}^*
b_t-coloration	3	2	7	1	5	4	7	6

(v_{11}, v_{21})	$(v_{12}, v_{22})^*$	v_{42}	(v_{21}, v_{22})	(v_{21}, v_{31})	$(v_{22}, v_{32})^*$	v_{31}	v_{32}^*
6	2	1	1	6	3	2	4

$(v_{31}, v_{32})^*$	(v_{31}, v_{41})	(v_{32}, v_{42})
7	1	5

Pour $n \geq 5$, on choisira les mêmes sommets et arêtes b_t-dominants puis on étend la coloration à tout le graphe.

Si $m = 3$.

- $n = 1$, $m_t(S_{3,n}) = 7$. Remarquer que $S_{3,n}$ est un K_4, et que les sommets $o, v_{11}, v_{12}, v_{13}$ jouent le même rôle, si on choisit un sommet quelconque b_t-dominant dans une coloration dominante totale utilisant 7 couleurs, on peut voir qu'au moins une classe de couleur ne peut avoir de sommet ou arête b_t-dominante donc $o, v_{11}, v_{12}, v_{13}$ ne peuvent être b_t-dominants. Il restera cependant 6 arêtes, ce qui n'est pas suffisant pour faire une coloration dominante totale avec 7 couleurs; alors $b_t(S_{3n}) = 6$. Avec un raisonnement analogue les sommets ne peuvent être b_t-dominants, donc les arêtes sont

toutes b_t-dominantes mais ceci ne peut être réalisé. Donc $b_t(S_{3,n}) = 5$; on donne une b_t-coloration

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	(v_{11}, v_{12})
b_t-coloration	1	2	3	4	1

(v_{12}, v_{13})	(v_{11}, v_{13})	v_{11}	v_{12}^*	v_{13}
2	5	4	5	3

- $n = 2$, $m_t(S_{3,n}) = 8$. Remarquer que seuls les sommets $v_{1i}, 1 \leq i \leq 3$ et les arêtes $(o, v_{1i}), 1 \leq i \leq 3, (v_{11}, v_{21}), (v_{12}, v_{22}), (v_{13}, v_{23}), (v_{11}, v_{12}), (v_{12}, v_{13}), (v_{11}, v_{13})$ peuvent être b_t-dominants. Pour toute b_t-coloration utilisant 8 couleurs, on ne peut arriver à rendre b_t-dominants qu'au plus 5 sommets ou arêtes. Donc $b_t(S_{3,2}) = 7$ et on donne une b_t-coloration utilisant 7 couleurs :

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	v_{11}	v_{12}	v_{13}^*
b_t-coloration	1	2	3	4	5	6	7

(v_{11}, v_{12})	$(v_{12}, v_{13})^*$	(v_{11}, v_{13})	(v_{11}, v_{21})	(v_{12}, v_{22})	(v_{13}, v_{23})	v_{21}^*
1	5	6	7	7	2	6

v_{22}	v_{23}	(v_{21}, v_{22})	(v_{21}, v_{23})	(v_{22}, v_{23})
2	3	1	4	6

- $n = 3$, $m_t(S_{3,n}) = 9$; Remarquer que les seuls sommets et arêtes pouvant être b_t-dominants dans une b_t-coloration utilisant 9 couleurs sont $v_{ij}, 1 \leq i \leq 2, 1 \leq j \leq 3$, $(v_{11}, v_{12}), (v_{12}, v_{13}), (v_{11}, v_{13}), (v_{11}, v_{21}), (v_{12}, v_{22}), (v_{13}, v_{23}), (v_{21}, v_{22}), (v_{22}, v_{23}), (v_{21}, v_{23})$. Si on colore arbitrairement 9 sommets ou arêtes parmi ces derniers, et on veut les rendre b_t-dominants alors on ne peut avoir que 5 sommets b_t-dominants; donc $b_t(S_{3,n}) = 8$ et on donne une b_t-coloration utilisant 8 couleurs :

sommets ou arêtes	o	(o, v_{11})	(o, v_{12})	(o, v_{13})	v_{11}^*	v_{12}^*	v_{13}^*
b_t-coloration	5	4	7	8	1	2	3

(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{11}, v_{13})	$(v_{11}, v_{21})^*$	(v_{12}, v_{22})	(v_{13}, v_{23})
6	4	7	8	8	6

v_{21}^*	v_{22}^*	v_{23}^*	(v_{21}, v_{22})	(v_{22}, v_{23})	(v_{21}, v_{23})	(v_{21}, v_{31})	$(v_{22}, v_{32})^*$
5	4	7	3	1	4	3	6

(v_{23}, v_{33})	v_{31}	v_{32}	v_{33}	(v_{31}, v_{32})	(v_{32}, v_{33})	(v_{31}, v_{33})
2	6	3	8	5	7	1

- $n \geq 4$, $m_t(S_{3n}) = b_t(S_{3n}) = 9$.

sommets ou arêtes	o	(o, v_{11})	(o, v_{12})	(o, v_{13})	v_{11}^*	v_{12}^*	v_{13}^*
b_t-coloration	5	4	7	8	1	2	3

(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{11}, v_{13})	$(v_{11}, v_{21})^*$	(v_{12}, v_{22})	$(v_{13}, v_{23})^*$	v_{21}
6	9	7	8	4	4	9

v_{22}	v_{23}^*	(v_{21}, v_{22})	(v_{22}, v_{23})	(v_{21}, v_{23})	(v_{21}, v_{31})	(v_{22}, v_{32})
8	6	5	1	2	3	6

$(v_{23}, v_{33})^*$	v_{31}	v_{32}	v_{33}^*	(v_{31}, v_{32})	$(v_{32}, v_{33})^*$	(v_{31}, v_{33})
5	1	2	7	4	9	8

(v_{32}, v_{42})	(v_{33}, v_{43})	v_{43}
1	3	4

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m = 4$.

- $n = 1$, $m_t(S_{4,n}) = 7$. Supposons qu'on puisse faire une coloration b_t-dominante utilisant 7 couleurs, on colore le centre avec la couleur 7. Si le centre est dominant pour cette couleur alors on colore les arêtes (o, v_{1i}), $(i = 1, 2, 3, 4)$ choisies b_t-dominantes avec les couleurs $1, 2, 3, 4$ et choisir deux sommets b_t-dominants parmis les quatre sommets adjacents au centre, sans perte de généralité on peut prendre ou bien v_{11}, v_{12} ou v_{11}, v_{13} colorés respectivement avec les couleurs $5, 6$. Alors il est simple de voir que trois arêtes ou sommets ne peuvent être b_t-dominants. Même raisonnement pour le cas où on prend quatre sommets b_t-dominants et deux arêtes b_t-dominantes adjacents au centre. Si le centre est non b_t-dominant pour cette couleur alors cette couleur va être portée par une des arêtes $(v_{11}, v_{12}), (v_{12}, v_{13}), (v_{13}, v_{14}), (v_{11}, v_{14})$; alors sans perdre de généralité on choisit comme arête b_t-dominante colorée 7, (v_{12}, v_{13}). Il est simple de vérifier qu'au moins une classe de couleur ne contient de sommet ou arête b_t-dominant. Donc $b_t(S_{4,n}) = 6$; et une b_t-coloration utilisant 6 couleurs est donnée.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(v_{12}, v_{13})^*$	v_{11}	v_{12}	v_{13}	v_{14}	(v_{11}, v_{12})	(v_{13}, v_{14})	(v_{11}, v_{14})
6	3	2	5	2	1	1	6

- $n = 2$, $m_t(S_{4,n}) = 9$. Les sommets et arêtes pouvant être b_t-dominants dans une coloration dominante totale utilisant 9 couleurs sont: $o; v_{1i}, (o, v_{1i}), 1 \leq i \leq 4; (v_{11}, v_{12}), (v_{12}, v_{13}), (v_{13}, v_{14}), (v_{11}, v_{14})$. On raisonne de manière similaire au cas précédent, en considérant le cas où le centre est b_t-dominant et le cas où il ne l'est pas et dans chacun des cas on peut vérifier qu'on n'arrive pas à une 9-b_t-coloration. Donc $b_t(S_{4,2}) = 8$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(v_{13}, v_{23})^*$	v_{23}	v_{24}	(v_{22}, v_{23})	(v_{21}, v_{24})	v_{11}	v_{12}	v_{13}^*	v_{14}
7	2	4	5	2	6	5	8	7

(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{13}, v_{14})	(v_{11}, v_{14})	(v_{11}, v_{21})	(v_{12}, v_{22})
7	6	3	8	1	8

$(v_{14}, v_{24})^*$	(v_{23}, v_{24})	v_{21}	v_{22}	(v_{21}, v_{22})
6	1	3	4	7

- $n \geq 3$, $m_t(S_{4,n}) = b_t(S_{4,n}) = 9$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(v_{13}, v_{23})^*$	v_{11}	v_{12}	v_{13}	v_{14}	(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{13}, v_{14})
9	6	7	8	9	9	6	7

(v_{11}, v_{14})	$(v_{11}, v_{21})^*$	$(v_{12}, v_{22})^*$	$(v_{14}, v_{24})^*$	(v_{23}, v_{24})	v_{21}	v_{22}
8	7	8	6	2	3	2

(v_{21}, v_{22})	(v_{21}, v_{24})	(v_{21}, v_{31})	(v_{22}, v_{32})	(v_{23}, v_{33})	(v_{24}, v_{34})
5	1	4	4	3	3

(v_{22}, v_{23})	v_{23}	v_{24}	(v_{21}, v_{24})	(v_{21}, v_{31})	(v_{22}, v_{32})	(v_{23}, v_{33})
1	5	4	1	4	4	3

(v_{24}, v_{34})
3

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m = 5$.

- $n = 1$, $m_t(S_{5,n}) = b_t(S_{5,n}) = 7$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(o, v_{15})^*$	v_{11}	v_{12}	v_{13}^*	v_{14}	v_{15}	(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{13}, v_{14})
6	3	2	7	6	5	7	5	3

(v_{14}, v_{15})	(v_{11}, v_{15})
7	1

- $n = 2$, $m_t(S_{5,n}) = 9$. Seules les arêtes $(o, v_{1i}), 1 \leq i \leq 5$, $(v_{1i}, v_{1(i+1)}); 1 \leq i \leq 4$, (v_{11}, v_{15}) et les sommets o, $v_{i1}, 1 \leq i \leq 5$, peuvent être b_t-dominants. En citant de manière exhaustive toutes les manières de choisir les 9 sommets b_t-dominants, il est impossible de faire une 9-b_t-coloration. On a alors $b_t(S_{5,2}) = 8$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(o, v_{15})^*$	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}^*	(v_{11}, v_{12})	(v_{12}, v_{13})	$(v_{11}, v_{15})^*$
6	5	7	8	2	8	8	1	7

(v_{13}, v_{14})	(v_{14}, v_{15})	(v_{11}, v_{21})	(v_{12}, v_{22})	(v_{13}, v_{23})	(v_{14}, v_{24})
7	4	1	2	2	8

(v_{15}, v_{25})	v_{25}
3	5

Le reste des sommets et arêtes sont de degré 6, on finit la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

- $n \geq 3$, $m_t(S_{5,n}) = b_t(S_{5,n}) = 9$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(o, v_{15})^*$	v_{11}^*	v_{12}	v_{13}	v_{14}	v_{15}	(v_{11}, v_{12})	(v_{12}, v_{13})	(v_{11}, v_{15})
6	9	5	8	2	3	8	9	7

(v_{13}, v_{14})	(v_{14}, v_{15})	(v_{11}, v_{21})	$(v_{12}, v_{22})^*$	(v_{13}, v_{23})	(v_{14}, v_{24})	(v_{15}, v_{25})
7	8	4	7	7	9	9

v_{21}	(v_{21}, v_{22})	v_{22}	(v_{22}, v_{23})	(v_{23}, v_{24})	v_{24}	v_{25}^*	(v_{24}, v_{25})
6	1	2	6	4	1	8	7

(v_{21}, v_{25})	v_{35}	(v_{22}, v_{32})	(v_{25}, v_{35})
2	5	4	4

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m = 6$.

- $n = 1$, $m_t(S_{6,n}) = b_t(S_{6,n}) = 7$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$	$(o, v_{15})^*$
b_t-coloration	1	2	3	4	5	6

$(o, v_{16})^*$
7

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

$n \geq 2$, $m_t(S_{6n}) = b_t(S_{6n}) = 9$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$	$(o, v_{15})^*$
b_t-coloration	1	2	3	4	5	6

$(o, v_{16})^*$	v_{11}^*	v_{12}	v_{13}^*	v_{14}	v_{15}	v_{16}	v_{21}	v_{23}	(v_{11}, v_{12})	(v_{12}, v_{13})
7	8	6	9	7	5	4	3	5	9	2

(v_{11}, v_{16})	(v_{13}, v_{14})	(v_{14}, v_{15})	(v_{15}, v_{16})	(v_{11}, v_{21})	(v_{12}, v_{22})
5	8	1	8	7	8

(v_{13}, v_{23})	(v_{14}, v_{24})	(v_{15}, v_{25})	(v_{16}, v_{26})
3	9	9	9

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m = 7$.

- $n = 1$, $m_t(S_{7,n}) = b_t(S_{7,n}) = 8$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(o, v_{15})^*$	$(o, v_{16})^*$	$(o, v_{17})^*$
6	7	8

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

- $n \geq 2$, $m_t(S_{7,n}) = b_t(S_{7,n}) = 9$.

sommets ou arêtes	o^*	$(o, v_{11})^*$	$(o, v_{12})^*$	$(o, v_{13})^*$	$(o, v_{14})^*$
b_t-coloration	1	2	3	4	5

$(o, v_{15})^*$	$(o, v_{16})^*$	$(o, v_{17})^*$	v_{11}	v_{16}	v_{17}^*	(v_{11}, v_{12})	(v_{11}, v_{17})
6	7	8	5	2	9	9	4

(v_{13}, v_{14})	(v_{15}, v_{16})	(v_{16}, v_{17})	(v_{17}, v_{27})	v_{27}
9	9	3	7	6

Finir la coloration proprement puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants considérés.

Si $m \geq 8$ et $n \geq 1$, on a $m_t(S_{m,n}) = b_t(S_{m,n}) = m + 1$. Il suffit de prendre comme sommet et arêtes b_t-dominants le centre o et les arêtes $(o, v_{1i}), 1 \leq i \leq m$ en leur affectant les couleurs $1, ..., m + 1$ respectivement puis étendre cette coloration à tout le graphe ceci est possible puisqu'il existe toujours une couleur disponible pour tous arête ou sommet restants. ■

References

[1] F. Harary, S. Hedetniemi. The achromatic number of a graph. *Journal of Combinatorial Theory* 8 (1970) 154–161.

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

[3] D.F. Manlove. Minimaximal and maximinimal optimisation problems: a partial order-based approach. PhD thesis. Tech. Rep. 27, Comp. Sci. Dept., Univ. Glasgow, Scotland, 1998.

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

[5] P.Erdos, A.L. Rubin, H. Taylor, Choosability in graphs, PROC. West Coast Conference on Combinatorics, Graph Theory, and Computing, Arcata, California(1979), p. 125-157, Congruessus umerantium XXVI.

[6] J. Kratochvil, Z. Tuza, M. Voigt, On the b-chromatic number of graphs, *Lecture Notes in Comput. Sci.* 2573 (2002) 310

[7] Sadegh Rahimi Sharebaf;Vertex, Edge and Total Coloring in Spider Graphs,Applied Mathematical science, Vol. 3, 2009, no. 18, 877-881

[8] S. Corteel, M. Valencia-Pabon, J.-C. Vera. On approximating the b-chromatic number. *Disc. Appl. Math.* 146 (2005) 106–110.

Traitement d'images

Une méthode rapide et efficace pour le cryptage évolutionnaire d'images

Ismahane SOUICI [1], Hamid SERIDI [2, 3]

[1]Département d'informatique, Université de Jijel, Algérie
[2]Département d'informatique, Université de Guelma, Algérie
[3]Crestic, Université de Reims, France
{souici.ismahane, seridihamid}@yahoo.fr

Résumé : Les transmissions croissantes des informations dans les réseaux publics soulèvent un nombre conséquent de problèmes qui ne sont pas tous encore résolus. Nous citons, par exemple, la sécurité, la confidentialité, l'intégrité et l'authenticité des données pendant leur transmission. Alors, le cryptage des données transmises se trouve l'une des solutions les plus prometteuses. Dans ce contexte, un schéma de transfert sécurisé des données images par exploitation des algorithmes évolutionnaires est proposé dans cet article. Les aptitudes du schéma proposé pour la confusion, la sensibilité à l'image nette et la clef ont été testées. De même, les résultats obtenus montrent l'efficacité du schéma contre les attaques avancées.

Mots clés : Sécurité, confidentialité, cryptage, chiffrement, optimisation, algorithme évolutionnaire, attaques avancées.

1 Introduction

L'évolution des techniques de traitement, de partage et de communication des images, s'accompagne d'une évolution des risques pour l'information alors sous forme numérique. Les possibilités d'accès distants et de communication de l'information sont améliorées mais les possibilités de fuites, de détournement et de modification de l'information sont aussi plus importantes, voire même facilitées par la mise à disposition d'outils de surveillance des réseaux et d'outils d'édition avancée de l'information comme les logiciels d'imagerie.

Cependant, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information où la **cryptographie**, qui est la science du chiffrement, s'est imposée comme passage incontournable dans le transit des informations sensibles. Ainsi, un crypto-système doit assurer un maximum de brouillage de l'information à chiffrer en utilisant une méthode difficilement contournable. Mais avec la puissance montante des ordinateurs, la cryptographie reste un domaine en plein mouvement pour suivre les nombreux progrès des cryptanalyses. Ainsi, certains algorithmes ont été remis en cause, tel que le DES [1] [2] [3] qui a été remplacé par 3DES [1] [4] puis récemment par AES [5]. De plus, la sécurité de RSA [6] est directement liée au progrès de la factorisation d'entiers qui peut être révélé un jour.

En parallèle, ces dernières années ont vu l'émergence de techniques de vie artificielle imitant les processus de l'évolution naturelle pour résoudre des problèmes complexes. Ces méthodes d'optimisation ou d'apprentissage permettent de résoudre des problèmes auxquels les méthodes classiques n'apportent pas de réponses satisfaisantes. Parmi ces méthodes, on distingue les **algorithmes évolutionnaires** de l'évolution darwinienne des populations biologiques qui leur permit d'évoluer au cours du temps en créant des systèmes biologiques très complexes adaptés à de nombreuses conditions [7]. C'est d'ailleurs ces algorithmes que nous avons utilisé pour développer notre nouvel algorithme de chiffrement, motivé dans ce cas, par l'adaptabilité et l'efficacité d'application de ces algorithmes dans un tel domaine à cause de leur large utilisation de l'aléatoire.
Dans les sections suivantes nous présentons les différentes étapes du nouveau processus cryptographique.

2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires représentent une famille assez riche et très intéressante d'algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Les champs d'application sont fort diversifiés et leur efficacité a été démontrée sur plusieurs domaines (voir par exemple [8]). Ainsi, on les retrouve aussi bien en théorie des graphes qu'en compression d'images numérisées ou encore en programmation automatique [9], [10], [11], [12]. Les raisons de ce nombre d'applications sont claires. Leur principe est d'opérer une recherche stochastique sur un important espace à travers un ensemble – une population – de pseudo-solutions. Ces algorithmes sont simples et très performant dans leur recherche d'amélioration. De plus, ils ne sont pas limités par des hypothèses contraignantes sur le domaine d'exploration.

Utilisant globalement le processus évolutionnaire dont le principe sera décrit dans la section qui suit, les algorithmes évolutionnaires (AEs) couvrent un ensemble d'algorithmes et de techniques. On retrouve alors quatre classes principales :

- La programmation évolutionnaire proposée par Fogel [13];
- Les stratégies d'évolution proposées indépendamment par Rechenberg [14] et Schwefel.
- Les algorithmes génétiques proposés par Holland [15];
- Beaucoup plus tard, une autre classe révolutionnaire vit le jour. Il s'agit de la programmation génétique proposée par Koza [16].

La différence entre ces catégories est essentiellement d'ordre historique. Actuellement, la distinction entre elles est de plus en plus floue puisqu'elles ne diffèrent que sur le type de codage des individus, les détails d'implémentation des opérateurs génétiques (croisement et mutation) et sur les procédures de sélection et de remplacement de la population.

3 Algorithme développé (OIEEA)

Le principal avantage des algorithmes évolutionnaires vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.
Ainsi, le problème considéré qui est celui de chiffrement de données, est résolu par une procédure d'optimisation (algorithme évolutionnaire) qui retient la structure générale décrite dans la section précédente.

Dans cette section, nous allons présenter les deux phases essentielles de notre algorithme de chiffrement évolutionnaire à savoir la phase de chiffrement et celle de déchiffrement. La figure 1 résume le principe de fonctionnement de l'algorithme proposé. La phase de chiffrement est assurée par exploitation du cycle évolutionnaire. La phase inverse corresponde à l'opération de déchiffrement.

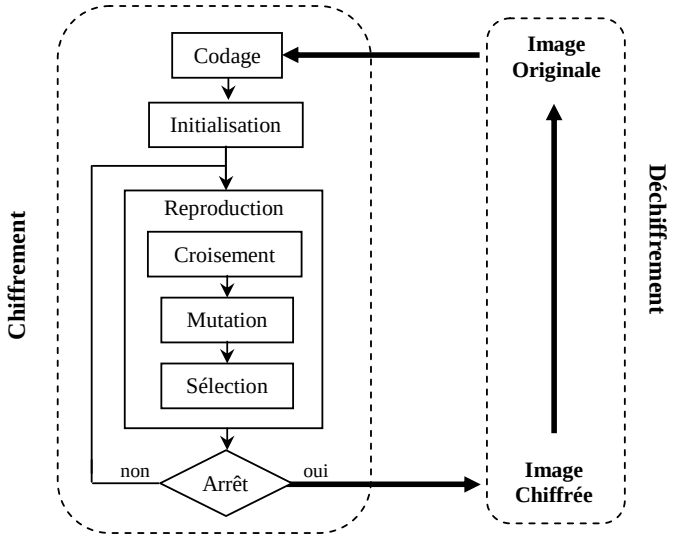


Fig. 1. Principe de fonctionnement de l'algorithme développé.

3.1 Chiffrement

Les grandes lignes des différentes étapes de l'opération de chiffrement de l'algorithme crypto-évolutionnaire proposé sont présentées en ce qui suit :

1) Codage : Cette opération consiste à transformer l'image en code particulier. En choisissant le RGB (Red Green Blue) comme espace de représentation d'images, le codage qu'on propose consiste à calculer le nombre d'occurrence des 256 valeurs

possibles que peut prendre les trois composantes R, G et B. Ainsi, le génotype résultat regroupe 768 gènes (voir figure 2).

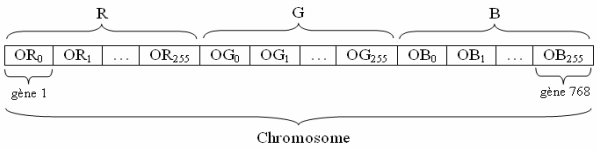


Fig. 2. Codage des individus.

Où : OR_i (respectivement OG_i et OB_i) est le nombre d'occurrence des valeurs de la matrice composante R (respectivement G et B) qui égale à *i*.

Remarque :
Si pour une valeur donnée de l'intervalle [0,255] aucune des valeurs de la composante R (G ou B) codant l'image n'est égale à cette valeur, alors son nombre d'occurrences est nul. Ainsi, la valeur du gène correspondant vaut 0.

2) Initialisation : La population initiale est celle qui servira de population souche à l'algorithme génétique. Sa composition peut influencer la vitesse de convergence de l'algorithme et par conséquent elle ne peut pas être constituée de manière arbitraire.

Dans notre cas, les individus formant la population initiale sont obtenus par perturbations aléatoires du chromosome initial représentant le codage de l'image originale. Ainsi, le processus évolutionnaire aura le sens d'un processus cryptographique en partant du codage de l'image originale pour arriver à la solution finale représentant le codage de l'image chiffrée correspondante.

3) Reproduction : Pour assurer la reproduction de nouveaux individus, on fait appel à des opérateurs génétiques (croisement et mutation). Le croisement permet la création de nouveaux individus à partir du patrimoine génétique de parents. Cette reproduction a pour but d'engendrer des individus enfants mieux adaptés que leurs parents. Le plus souvent, deux enfants sont crées par le croisement de deux parents sélectionnés.

La mutation d'un individu consiste à modifier localement son patrimoine génétique afin de générer un nouvel individu. Ainsi, un ou plusieurs gènes sont tirés aléatoirement puis sont modifié en assurant que la résultante est une solution admissible au problème.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX «Order Cross-over» proposé par Davis [17] qui consiste à générer des descendants avec un taux de 60% à 100% [18]. Quand à la mutation, nous appliquons une simple permutation aléatoirement de deux gènes d'un chromosome et cela avec un taux de 0,1% à 5% [18].

4) Evaluation : Aucune hypothèse n'existe pour définir la fonction fitness à maximiser, le calcul de la fonction d'adaptation peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation. C'est d'ailleurs ce que nous avons fait où la fonction d'évaluation que nous avons défini (expression 1) représente l'écart qu'on cherche à maximiser entre les nombres d'occurrence des différentes valeurs possibles du codage des pixels de l'image originale et ceux des pixels de l'image chiffrée ; c'est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \quad (1)$$

Cette fonction est équivalente à celle donnée à travers l'expression 2 et c'est celle qui est finalement adoptée.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad (2)$$

Avec : $I_i = [O_1, O_2, \ldots, O_{768}]$, $Pop = \{I_1, I_2, \ldots, I_m\}$ où m est un paramètre de réglage et O_{j_i} est le $j^{ème}$ gène du $i^{ème}$ individu.

5) Sélection : Le rôle de la sélection est d'assurer majoritairement la survie des meilleurs éléments à travers les générations. Il est tout de même important de ne pas se limiter seulement aux meilleurs individus et ce afin de conserver une diversité génétique suffisante. En effet, même les individus les moins bien adaptés peuvent, par croisement ou mutation, engendrer une descendance pertinente par rapport au critère d'optimisation. L'opérateur de sélection a pour mission de choisir dans la population présente les futurs parents nécessaires à l'étape de remplacement. Dans notre cas, nous avons choisi d'utiliser la sélection proportionnelle [19].

6) Arrêt : Il peut être défini en fonction du nombre de génération, de l'adaptation du meilleur individu, etc.
La convergence de notre algorithme est assurée par une double condition d'arrêt dont l'une est celle donnée par l'expression (3) en plus d'un nombre maximal d'itérations fixé expérimentalement et à ne pas dépasser.

$$0 \le F(I_i) \le 768 \times N \quad (3)$$

3.2 Déchiffrement

Le processus de déchiffrement permet la reconstitution de l'image originale. Pour le récepteur qui possède l'image cryptée et la clé, le processus de déchiffrement est très simple et efficace. Après l'introduction de la bonne clé, il permet la permutation des positions des nombres d'occurrences des pixels de l'image cryptée pour obtenir ceux des pixels de l'image originale. De ce fait, elle varie d'une instance du problème à une autre.

4 Résultats expérimentaux et interprétation

L'application de l'algorithme décrit ci-dessus requiert le réglage d'un certain nombre de paramètres. Dans la littérature et depuis les travaux pionniers de Grenfenstette [18], plusieurs travaux ont traité de ce sujet tels que [20], [21], [22], [23] et [24]. Cependant, en pratique les paramètres des AEs sont, souvent, réglés approximativement par tâtonnement.

À partir des images originales des figures 3.a et 4.a, nous avons appliqué notre algorithme pour obtenir les images chiffrées des figures 3.b et 4.b. Nous constatons que les images originales contiennent plus ou moins de zones homogènes ou textures (figures 3.a et 4.a). Cela se vois sur leurs histogrammes correspondants (Figures 3.c-e pour l'image Smile et Figures 4.c-e pour l'image Lena). Cependant, ce phénomène n'apparaît pas sur les histogrammes correspondants aux images chiffrées (figures 3.f-h pour l'image Smile et 4.f-h pour l'image Lena). Le cryptage est alors bon parce qu'il est difficile de deviner la nature de l'image à partir de l'image chiffrée (image médicale échographie par exemple qui contient de grandes zones homogènes).

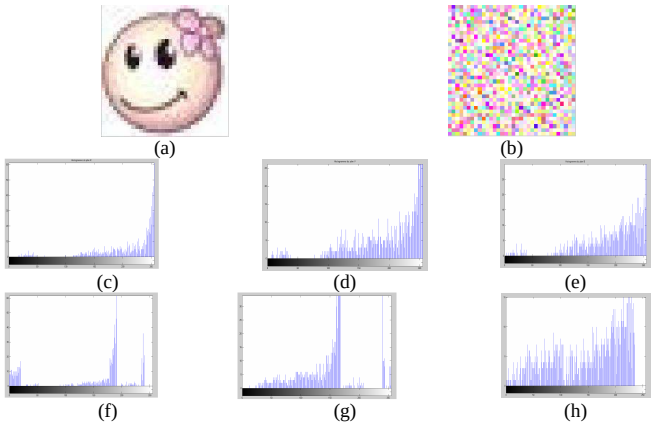


Fig. 3. a) Image originale Smile, b) Image cryptée, c) Histogramme de la composante R de l'image (a), d) Histogramme de la composante V de l'image (a), e) Histogramme de la composante B de l'image (a), f) Histogramme de la composante R de l'image (b), g) Histogramme de la composante V de l'image (b), h) Histogramme de la composante B de l'image (b).

Le tableau 1 présente un résumé des résultats de chiffrement des images test en termes de valeurs de convergence représentative du niveau de confusion de l'algorithme et de temps d'exécution. D'après ces résultats, nous constatons que le temps d'exécution est indépendant de la taille de l'image où la différence en temps de chiffrement de l'image Smile de taille égale à 1190 pixels et le temps de chiffrement de l'image Lena ayant une taille égale à 17161 pixels est très rudimentaire en comparaison avec la grande différence en taille entre les deux images. La toute petite différence correspond aux opérations de codage (représentation chromosomique) et de

décodage (génération de l'image chiffrée à partir de la représentation chromosomique).

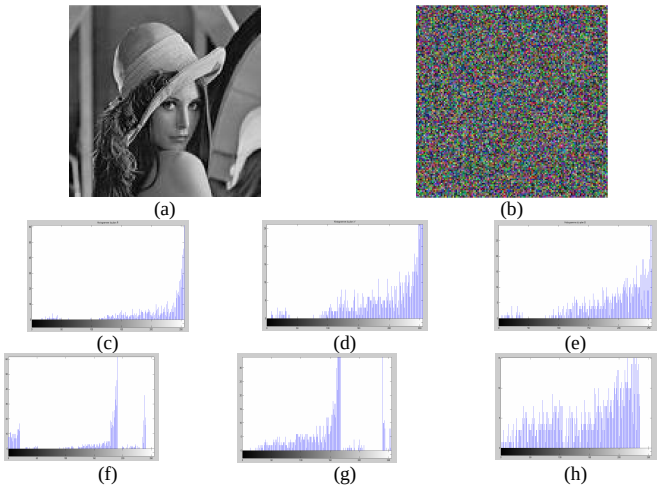


Fig. 4. a) Image originale Lena, b) Image cryptée, c) Histogramme de la composante R de l'image (a), d) Histogramme de la composante V de l'image (a), e) Histogramme de la composante B de l'image (a), f) Histogramme de la composante R de l'image (b), g) Histogramme de la composante V de l'image (b), h) Histogramme de la composante B de l'image (b).

Tableau 1. Résultats de chiffrement des images tests Smile et Lena.

	Taille image	**VC**	**T(s)**
L'image Smile	34 X 35	5832	2.92
L'image Lena	131 X 131	56674	3.37

VC : valeur de convergence
T : temps d'exécution

Il est utile de noter que l'image reconstruite est identique à l'image originale, donc il n'y a aucune perte d'information et la valeur du PSNR tend vers l'infini.

5 Analyse de la sécurité

De façon générale, un crypto-système est qualifié de bon s'il assure la confidentialité des données secrètes durant leur durée de validité. Cependant, la résistance aux différentes tentatives d'attaques ou de cryptanalyses est la principale mesure de la qualité d'un crypto-système. Cette mesure s'appelle la sécurité [25]. Les critères utilisés pour évaluer la sécurité de notre crypto-système proposé sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

5.1 Vitesse de l'algorithme

En plus du paramétrage, la vitesse de l'algorithme dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de notre problème, par exemple, si le codage dépend de la taille de l'image à chiffrer cela affectera grandement la vitesse de l'algorithme puisque le temps de traitement des images de grandes tailles sera beaucoup plus grand que celui de traitement des images de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Notre codage proposé unifie la taille des chromosomes codant des images de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des images traitées. En plus et d'après les résultats expérimentaux présentés dans le tableau 1, nous constatons que le temps de convergence de l'algorithme proposé est très raisonnable (de l'ordre de 3 secondes dans le cas de l'image Smile et de 4 secondes dans le cas de l'image Lena).

5.2 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour se faire, nous avons utilisés les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :
Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (4), l'erreur absolue moyenne (*MAE : Mean*
Absolute Error) donnée par l'expression (6) et l'erreur quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (7). Le tableau 2 résume les valeurs des différentes mesures obtenues après les testes qui ont été effectués sur les images Smile et Lena et leurs images chiffrées correspondantes (voir figure 3 et 4).

Tableau 2. Niveaux de confusion

	NPCR	**MAE**	**MSE**
L'image Smile	0.8489	8.7283×10^{-4}	1.2385×10^{-6}
L'image Lena	0.9955	9.4892×10^{-4}	1.3788×10^{-6}

$$NPCR = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j) \qquad (4)$$

$$D(i,j) = \begin{cases} 0 & Si\ Im_o(i,j)=Im_c(i,j) \\ 1 & Si\ Im_o(i,j)\neq Im_c(i,j) \end{cases} \qquad (5)$$

$$MAE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \qquad (6)$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \qquad (7)$$

5.3 Attaque différentielle

Les algorithmes de chiffrement par blocs, tels que la majorité des algorithmes s'inscrivant sous le mode de chiffrement symétrique (DES, 3DES, AES), permettent à la cryptanalyse différentielle d'être appliquée en parallèle sur les différents blocs. Et comme OIEEA opère sur l'image entière pour obtenir l'image chiffrée, donc il sera difficilement cassé, voir même hors porté de ce type d'attaque.

5.4 Attaque exhaustive

L'attaque exhaustive est l'attaque la plus destructive que peut subir un crypto-système. Elle réussit toujours à remettre en cause des crypto-systèmes utilisant des clés de petites tailles (DES). Notre clé utilisée est la concaténation de 768 nombres représentant des permutations de positions des nombres d'occurrences, donc elle est de taille égale à 6144 bits pour des images de taille ne dépassant pas 255 pixels sinon la taille de la clé est de 12288 bits si l'image est de taille comprise entre 256 et 65535 pixels. Il est clair que ces tailles de clé sont largement suffisantes pour pénaliser toute attaque exhaustive.
Ce point est bien pris en considération dans la conception de notre algorithme. Premièrement, par le fait que la clé utilisée est une clé calculée à partir de l'image originale et de l'image chiffrée correspondante, donc elle change d'une instanciation du problème à une autre puisque le cryptage successif d'une même image donne généralement lieu à un ensemble varié d'images chiffrées.

5.5 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de l'image originale et de l'image chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même image donne lieu à un ensemble d'images chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement de la même image originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même image donc elle ne sera plus utile par la suite.

6 Conclusion

Dans ce travail nous avons présenté un schéma efficace de cryptage d'images dans lequel nous avons développé un nouvel algorithme de chiffrement d'images par exploitation des algorithmes évolutionnaires. Leur application nécessite d'une part un bon réglage de paramètres et d'autre part un codage adéquat vis-à-vis du problème à résoudre. Les résultats obtenus montrent que le mode de codage que nous avons mis au point présente des aptitudes dans la confusion et dans la sensibilité à l'image originale et à la clé générée. De plus, l'algorithme proposé peut servir au cryptage de tout type d'images à cause du temps réduit de chiffrement et de déchiffrement.

REFERENCES

1. Stinson. D., *Cryptographie, théorie et pratique*, International Thomson Publishing, France, 1996.
2. Biham. E. and Shamir. A., « Differential Cryptanalysis of DES-like cryptosystems », *Journal of Cryptology*, 4(1):3_72, 1991.
3. Matsui. M., **«** Linear cryptanalysis method for DES cipher **»**. Advances in Cryptology**,** EUROCRYPT'93, volume 765 de *Lecture Notes in Computer Science*, Springer-Verlag, 1994.
4. Ganteaut. A. and Lévy. F., « La cryptologie moderne », L'Armement, 73:76_83, mars 2001.
5. Leprévost. F., « Les standards cryptographiques du XXIe siècle : AES et IEEE-P1363 », Gazette des Mathématiciens - n°85, Juillet 2000.
6. Menezes. A.J., Oorschot. P.C., and Vanstone. S.A., *Handbook of Applied Cryptography*, CRC Press, 1996.
7. Darwin. C., The origin of species by means of natural selection, or the preservation of favored races in the struggle for life, New York: D. Appleton and Company, 443 & 445 Broadway.
8. Yu. T., Davis. L., Baydar. C., and Roy. R., editors. *Evolutionary Computation in Practice*. Studies in Computational Intelligence 88, Springer Verlag, 2008.
9. Goldberg. D.E., Computer-aided gas pipeline operation using genetic algorithms and rule learning. PhD thesis, University of Michigan, 1983.
10. Gefenstette. J.J., Fitzpatrick. J.M. and Van Gucht. D., Image registration by genetic search. In *proceeding of IEEE Southeast conference*, 1984.
11. Starkweather. T., Whitley. D. and Bogard. C., Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Computing, 14(3):347-361, august 1990.
12. Robertson. G., Parallel implementation of genetic algorithms in classifier system. Genetic algorithms and simulated annealing, 129-140, 1987.
13. Fogel. L., Owens. A. and Walsh. M., (1966). *Artificial Intelligence Through Simulated Evolution*. Wiley, J., Chichester, UK. In [Jour, 2003].
14. Rechenberg. I., (1973). Evolutions Strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart. In [Jour, 2003].
15. Holland. J., Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975.
16. Koza. J., Genetic Programming. Cambridge, MA: MIT Press, 1992.
17. Davis. L., « Applying Adaptive Algorithms to Epistatic Domains », *Proceedings of the International Joint Conference on Artificial Intelligence*, pp162-164, 1985.
18. Grenfenslette. J.J, « Optimization of control parameters for genetic algorithms », *IEEE translation on system Man and cybernetics*, Vol 16 №1, pp122-128, 1986.
19. Coueque. Y., J. Ohler et S. Tollari, « Algorithmes génétiques pour résoudre le problème du commis voyageur », Avril 2002.
http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/node5.html
20. Davis. L., Adapting operator probabilities in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*. P61-69, George Mason University, Fairfax, Virginia, USA, June 1989.
21. Bäck. T., Mutation parameters. Handbook of Evolutionary Computation. 97/1, E1.2, IOP Publishing Ltd. 1997.
22. Hutter. F., Hamadi. Y., Hoos. H.H., and Leyton-Brown. K., Performance prediction and automated tuning of randomizedand parametric algorithms. In CP 2006, number 4204 in lncs, pages 213–228. Springer Verlag, 2006.
23. Eiben. A.E., Michalewicz. Z., Schoenauer. M. and Smith. J.E., Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*. 2007.
24. Bibai. J., Savéant. P. and Schoenauer. M., On the Generality of Parameter Tuning in Evolutionary Planning, *Genetic and Evolutionary Computation Conference (GECCO-2010)*, Portland, Oregon : United States, 2010.

25. Behnia. S., Akhshani. A., Mohmodi. H., and Akhavan. A., ”A novel algorithm for image encryption based on mixture of chaotic maps,” ELSEVIER, Chaos, Solitons and Fractals, in press.

Compression des images basée sur les essaims particulaires et la recherche taboue

MANSOURI Douelkefel, BENAMRANE Nacéra

Département d'Infomatique, Faculté des Sciences
Université des Sciences et de la Technologie d'Oran,-Mohamed BOUDIAF-USTOMB
B. P 1505 EL-Mnaouer Bir El Djir 31000 Oran Algerie
nabenamrane@yahoo.com; douelkifl31@hotmail.com

Résumé. Bien que la transformée en ondelettes est efficace pour la compression des images numériques, elle ne préserve pas les contours, car elle est souvent utilisée de manière séparable sur l'axe horizontal et vertical de l'image. Nous proposons dans ce papier une technique de compression des images basée sur les bandelettes et l'optimisation des bases de bandelettes par les essaims particulaires et la recherche taboue. Une transformée qui permet de capturer les singularités le long des contours. Cette technique d'optimisation par essaim de particules PSO est appliquée sur un dictionnaire de bases de bandelettes pour sélectionner la base qui donne la meilleure représentation de l'image. Afin d'éviter le problème de minima locaux, nous proposons la PSO hybridée avec la méthode de recherche taboue.

Mots-clés : compression, ondelettes, bandelettes, essaims particulaires, recherche taboue.

1 Introduction

Les nouveaux systèmes d'imagerie servent à fournir des données de plus en plus précises, détaillées et aussi plus complexe. La nécessité de stockage de ces informations est donc évidente, mais les espaces de stockage sont considérables et ne cessent d'augmenter suivant l'évolution des technologies. La réponse à la gestion de cette quantité importante de données et le seul moyen de réduction de la taille de ces informations est la compression.

La régularité géométrique des images naturelles est une caractéristique intéressante et un élément important dans la compression. Malgré le succès des ondelettes [1] [2] pour la représentation de singularités ponctuelles, elles ne sont pas, dans les images, adaptées à la représentation des singularités le long des contours à cause de leur schéma séparable qui n'est pas adapté à la représentation des contours. L'exploitation de cette régularité a suscité beaucoup de chercheurs de trouver des transformées directionnelles qui surmontent le problème de régularité. Les transformées directionnelles sont divisées d'après les chercheurs en approches adaptatives et non adaptatives .Dans les approches non-adaptatives la plupart des transformées ne sont pas utilisées en compression à cause de leur redondance (il y a

plus d'information en sortie qu'en entrée) .Parmi les transformées utilisées dans ce type non-adaptative, nous citons entres autres, la transformée de Radon [3][4], La transformée Ridgelet [5][6] et les curvelets [7]. Pour les approches adaptatives, ce sont des approches qui dépendent du signal et qui offrent une plus grande flexibilité que les approches non adaptatives, les transformées que l'on trouve dans ce type sont: le Matching pursuit [8],Brushlets [9], les bandelettes [10], et les Beamlets [11].

Dans ce papier, nous avons proposé une nouvelle transformée introduite par Le Pennec et Mallat, à bases adaptées qui est, la transformée en bandelettes.

Les bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique. On distingue deux générations des bandelettes, la première génération proposée par Le Pennec en 2002 [12] et une seconde génération développée par Peyré en 2005 [13][14].

2 Première génération de bandelettes

Selon Le Pennec et Mallat [12] les bandelettes sont des frames qui garantissent une représentation optimale des fonctions de θ_α. Elles forment un dictionnaire de frames adaptées aux images de θ_α. Pour tout $f \in \theta_\alpha$. il existe un frame de bandelettes B dans lequel l'erreur d'approximation non linéaire est bornée par

$$\|f - f_M\|_2^2 \leq CM^{\alpha}. \qquad \textbf{(1)}$$

Au départ la construction des bandelettes se fait localement au voisinage d'une singularité sur des domaines en forme de tubes ou de bandes. La représentation de la bande donnée par la formule:

$$B=\{(x1,x2) : x1 \in [a1,b1],\ x2 \in [g(x1)+a2, g(x1)+b2]\}. \qquad \textbf{(2)}$$

La représentation d'une déformation locale de l'image est donnée par Wf (x1,x2) = f(x1,x2+ g(x1)). Avec g la géométrie locale. WB est un rectangle qui a une base orthogonale adaptée à la régularité de Wf, voici la base B de bandelettes, son support est dans une bande autour de la courbe g.

$$\mathbf{B}=\{\varphi_{l,m1}(x1)\varphi_{j,m2}(x2-g(x1));\ \varphi_{j,m1}(x1)\phi_{j,m2}(x2-g(x1))\ \ \varphi_{j,m1}(x1)\varphi_{j,m2}(x2g(x1))\}$$
$$(j, l> j ,m1,m2)\in \mathbf{I_B}. \qquad \textbf{(3)}$$

3 Seconde génération de bandelettes

La transformée en bandelettes par groupements est une version de la transformée

en bandelette pour la compression, son objectif est d'exploiter les régularités géométriques le long des contours dans les sous-bandes de la transformée en ondelettes. Cette seconde génération considère un champ d'orientations plutôt qu'un nombre restreint de courbes. Ce champ indique l'orientation locale à l'intérieur de blocs de l'image. Comme nous l'avons déjà cité, les ondelettes ne peuvent pas exploiter la régularité géométrique surtout sur une longueur d'une dizaine de pixels, pour cela on divise les sous-bandes de détails de la transformée en ondelettes en blocs de taille 4×4 (16 coefficients), et par la suite on transforme les coefficients de ces blocs en des coefficients de bandelettes en projetant les coefficients d'ondelettes de chaque bloc dans une base polynomiale. Pour chaque bloc on effectue un groupement des pixels (coefficients) selon un ensemble de directions. Il existe des dictionnaires qui contiennent un ensemble de bases directionnelles, chaque base contient 16 blocs de 16 coefficients, par une simple projection des blocs de coefficients d'ondelette sur les blocs des bases de bandelettes de ces dictionnaires on trouve les coefficients de bandelettes.

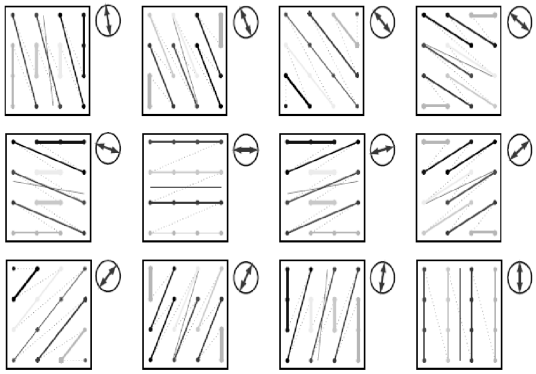


Fig. 1: Dictionnaire des bases directionnelles.

La construction d'une image (après la compression en appliquant la transformée en bandelettes) avec des contours bien clairs est motivé par le choix de base. On a appliqué, dans ce papier, une technique d'optimisation sur un dictionnaire de bases afin de trouver la meilleure base qui donne la meilleure représentation aux contours.

Le suivant paragraphe détaille la méthode d'optimisation par essaims particulaires.

4 Optimisation par essaims particulaires

L'optimisation par essaim de particules PSO [15][16][17][18] est une métaheuristique née en 1995 aux États-Unis sous le nom de Particle Swarm Optimization, elle a été proposée par R. Eberhart et J. Kennedy[19][20][21]. La technique a vu le jour sous la forme d'une simulation simplifiée d'un milieu social, tel

que le déplacement des oiseaux à l'intérieur d'une volée, elle se base sur la collaboration des individus entre eux, l'individu est appelé particule, l'essaim désigne l'ensemble des particules. Le processus de recherche est basé sur les deux règles suivantes:

1. Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.
2. Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

Dans un système PSO, initialement, chaque particule est positionnée aléatoirement dans l'espace de recherche, Elle va être pilotée dans l'espace multidimensionnel de recherche en ajustant sa position $X_i(t)$ selon sa propre expérience et celle des particules voisines. Chaque particule dans l'essaim est déplacée vers le point optimal en ajoutant une vitesse $V_i(t)$ à sa position , mise à jour de la position par

$$X_i(t) = X_i(t-1) + V_i(t) . \quad \textbf{(4)}$$

La vitesse de chaque particule est mise à jour par :

$$V_i(t) = V_i(t-1) + \varphi_1 (P_i - X_i(t-1) + \varphi_2 (P_g - X_i(t-1)) \quad \textbf{(5)}.$$

φ_1 et φ_2 sont des paramètres utilisés pour ajuster l'importance des termes $(P_i - X_i(t-1)$ et $\varphi_2 (P_g - X_i(t-1)$, $V_i(t) \in [V_{min}(t), V_{max}(t)]$.Si une particule sort de l'intervalle $[X_{min}(t), X_{max}(t)]$, on lui attribue la valeur du point frontière le plus proche.

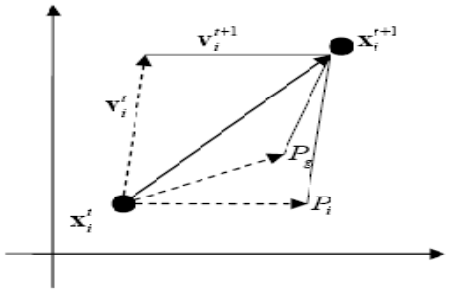


Fig. 2: Mise à jour d'une particule.

Nous récapitulons les définitions précédentes par l'algorithme PSO suivant :

```
Algorithme: Optimisation par essaims particulaires
Initialisation des paramètres Xi,Vi,Vmax,Xmin,Xmax
For  t=1 au temps maximum
For i=1 au nombre de particules
If  G(Xi)> G(pi)            //G() évaluer la qualité
pi= Xi                      // pi meilleure position
EndIf
g=i                         //arbitraire
```

```
For  j =   index des voisins de la particule i
 If G(pj)>G(pg) then g=j        //index de la meilleure
                                  particule globale
Next  j
Vi(t)= Vi(t-1)+φ1 (Pi - Xi(t-1)+ φ2 (Pg - Xi(t1)),
                                        Vi∈ (-Vmax, +Vmax)
Xi (t) = Xi (t-1)+ Vi (t) , Xi ∈  (Xmin, Xmax)
Next i
next t, jusqu'à obtention du critère d'arrêt
```

Cette technique d'optimisation a été, dans ce papier, appliquée sur l'ensemble de bases de bandelettes, nous l'avons également hybridé avec la technique de recherche taboue pour éviter les minima locaux.

5 PSO basée sur le concept de la recherche taboue TL-PSO [23][24][25][26]

La recherche taboue est une métaheuristique développée en 1986 par Fred Glover, l'objectif est de pouvoir mémoriser l'historique de solutions en passant d'une solution valide à une autre. Nous avons hybridé la technique PSO avec celle de recherche taboue pour pouvoir éviter le problème de minima locaux. L'algorithme hybridé procède de la manière suivante : pour chaque particule sa meilleure information sera stockée dans une liste taboue, Quand une particule veut se déplacer vers une autre particule, pour chercher la solution, elle consulte la liste taboue (historique des solutions), et on met à jour sa vitesse grâce à l'historique des informations données par les autres particules.

L'algorithme TL-PSO est comme suite :

```
Algorithme : TL-PSO
Initialisation des paramètres Xi ,Vi ,Vmax, Xmin, Xmax
For  t=1 au temps maximum
    For i=1 au nombre de particules
        If  G(Xi)> G(pi)      //G() évaluer la qualité
                 pi= Xi        // pi meilleure position
        EndIf
                g=i                          //arbitraire
        For  j=index des voisins de la particule i
                 If  G(pj)> G(pg) then g=j  //index de la
                       meilleure particule globale
                 Tabu = pg
         Next  j
         Consulter Tabu
            Vi(t)=Vi(t-1)+φ1(Pi -  Xi(t-1)+  φ2(Pg -  Xi(t-1)),
                          Vi∈  (-Vmax, +Vmax)
```

$X_i(t) = X_i(t-1) + V_i(t)$, $X_i \in (X_{min}, X_{max})$

Next i

next t, jusqu'à obtention du critère d'arrêt

6 Expérimentation

Nous avons choisi un dictionnaire de 10 bases avec des directions différentes, pour cela notre essaim va contenir 10 particules.

Après avoir divisé les sous-bandes de détail de la transformée en ondelettes en des blocs de 16 coefficients, on effectue un produit scalaire entre chaque bloc de la transformée en ondelettes et les vecteurs de la base choisie pour avoir des blocs de coefficients de bandelettes. On répète la même opération pour les autres bases (on projette les blocs de coefficients d'ondelettes sur chaque base), dans ce cas nous aurons 10 blocs de coefficients de bandelettes qui peuvent représenter un bloc de coefficients d'ondelettes (chaque bloc de coefficients de bandelettes appartient à une base). Pour la sélection des blocs (bases), les particules se déplacent dans un espace de recherche en testant les meilleurs blocs entre elles suivant la technique PSO. Le PSNR est utilisé pour tester l'efficacité des blocs :

$$PSNR = 10 \log_{10} \frac{X^2 \max}{MSE} \quad \textbf{(6)}.$$

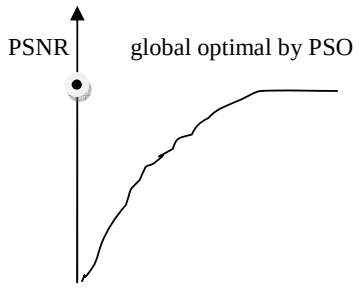
Avec :
$$MSE = \frac{1}{M.N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left[I(m,n) - \hat{I}(m,n) \right]^2 \quad (\textbf{7}).$$

X_{max} : la luminance maximale possible.

I : bloc original.

$\hat{I}$: bloc d'approximation.

Après l'optimisation par essaims particulaires nous avons dressé les deux schémas suivants :



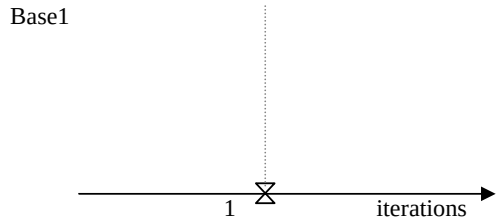


Fig. 3: Meilleure particule (base1) en terme de PSNR après une seule itération

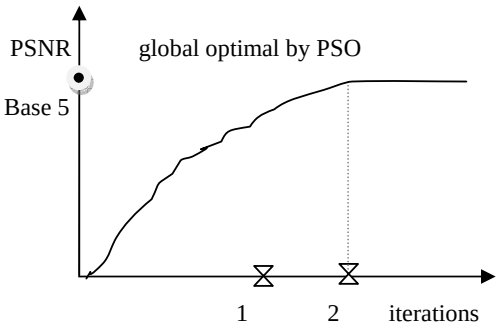


Fig. 4: Meilleure particule(base5) en terme de PSNR après deux itérations

Le schéma suivant montre les résultats de PSO-TL

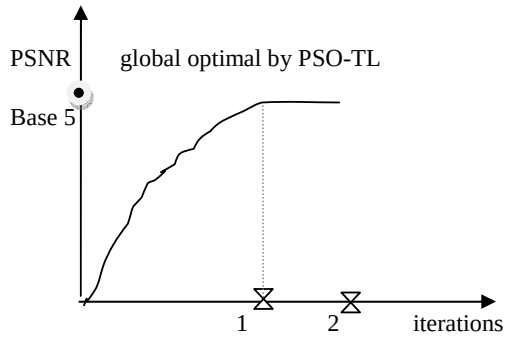


Fig. 5: Meilleure particule (base5) en terme de PSNR après une seule itération

Table 1. PSNR des images en utilisant PSO et PSO-TL .

Image	**PSNR/ itération**	
	(1itération)	***(2 itérations et plus)***

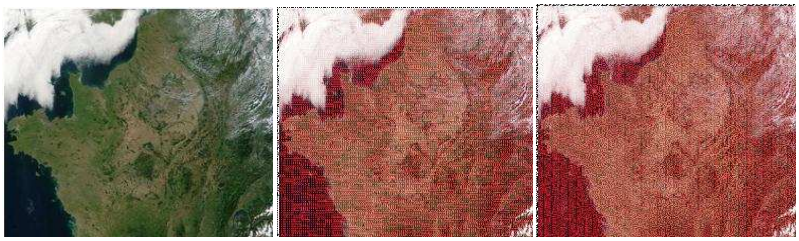
Lena **(PSO)**	28,30	29,30
Lena **(PSO-TL)**	29,30	29,30

(a) (b) (c)

Fig. 6 : "(a) image originale", "(b) image reconstruite par la base1 « PSNR=28,75 trouvée par PSO après une seule itération »", "(c) image reconstruite par la base5 « PSNR=29,30 trouvée par PSO-TL après une seule itération et par PSO après deux itérations »"

Les trois schémas montrent les meilleures bases à travers lesquelles nous pouvons construire des images avec des meilleurs PSNR. D'après le premier schéma l'algorithme PSO a donné, après une seule itération, la base1 (particule) comme étant la solution optimale (c.-à-d. celle qu'à travers laquelle on peut atteindre le meilleur PSNR); le deuxième schéma montre qu'on peut atteindre l'optimum global (base 5) après deux itérations. Quant au troisième schéma, nous pouvons constater qu'avec l'algorithme PSO-TL on atteint l'optimum global après une seule itération.

Si on applique l'approche proposée sur des images satellitaires, en utilisant le même dictionnaire, on obtient les résultats suivants :



(a) (b) (c)

Fig.7 : "(a) image originale", "(b) image reconstruite par la base10 « PSNR=18,91 »", "(c) image reconstruite par la base1 « PSNR=18,31 »"

7 Conclusion

La transformée en bandelette est très efficace pour la régularité géométrique le long des contours. Dans ce papier on a essayé de trouver une base parmi les bases de bandelettes d'un dictionnaire donné qui, après la transformée en bandelettes, permet de nous offrir une image avec des contours bien claires, la sélection de cette base a été effectuée en utilisant la technique d'optimisation par essaims particulaires afin de gagner du temps en cherchant la meilleure base. L'idée d'hybrider la méthode PSO avec la technique recherche taboue TL est très intéressante, cette dernière améliore le résultat et évite le problème de minima locaux.

References

1. B.K. Alpert. Wavelets and Other Bases for Fast Numerical Linear Algebra, pages 181–216. C. K. Chui, editor, Academic Press, San Diego, CA, USA, 1992.
2. M. Vetterli, "Wavelets, approximation and compression," in IEEE Signal Processing Magazine, Sept. 2001, pp. 59-73.
3. J. Radon, "Über die bestimmung von funktionen durch ihre integralwertelängs gewisser mannigfaltigkeiten," in Berichte Saechsische Akademie der Wissenschaften,Leipzig. Math. Nat, 1917, vol. 69, pp. 262-277.
4. F. Matús and J. Flusser, "Image representation via a finite radon transform," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15(10), pp. 996-1006, Oct. 1993.
5. E. J. Candes, Ridgelets : Theory and Applications, Ph.D. thesis, Department of Statistics, Stanford University, 1998.
6. M. N. Do and M. Vetterli, "Orthonormal finite ridgelet transform for image compression," in IEEE International Conference on Image Processing, 2000, vol. 2,pp. 367-370.
7. E. Candès and D. Donoho. Curvelets : A surprisingly effective nonadaptive representation of objects with edges. Vanderbilt University Press, 1999.
8. S. G. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries,"in IEEE Transactions on Signal Processing, Dec. 1993, vol. 41, pp. 3397-3415
9. F. G. Meyer and R. R. Coifman, "Brushlets : a tool for directional image analysisand image compression," in Applied and Computational Harmonic Analysis, 1997,vol. 4, pp. 147-187
10. Erwan Le Pennec, Stéphane Mallat :Représentation d'Image par Bandelettes et Application à la Compression.
11. D. L. Donoho and X. Huo, « Beamlet pyramids : a new form of multiresolution analysis, suited for extracting lines, curves and objects from very noisy image data, » in SPIE Conference on Wavelet Applications in Signal and Image Processing, 2000, vol. 4119, pp. 434-444.
12. Erwan Le Pennec, Stéphane Mallat : Bandelettes et représentation géométrique des images.
13. G. Peyré. Géométrie multi-échelle pour les images et les textures. Thèse de doctorat, École Polytechnique, 2005.
14. Gabriel Peyré, Stéphane Mallat : DISCRETE BANDELETS WITH GEOMETRIC ORTHOGONAL FILTERS, CMAP, Ecole Polytechnique 91128 Palaisau Cedex, France.
15. Kennedy, J., Everhart, R.: Particle Swarm Optimization. Proc. Of IEEE international Conference on Neural Networks (ICNN).(1995) 1942-1948

16. Merwe, D.W.: Engelbrecht A P. Data Clustering Using Particle Swarm Optimization. Evolutionary Computation, 2003. CEC '03. The 2003 Congress on. (2003) 215 – 220
17. Maurice Clerc and James Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1):58–73, 2002.
18. Mudassar Iqbal and Marco A. Montes de Oca. An estimation of distribution particle swarm optimization algorithm. In Marco Dorigo, Luca M. Gambardella, Mauro Birattari, Alcherio Martinoli, Riccardo Poli, and Thomas St¨utzle, editors, LNCS 4150. Ant Colony Optimization and Swarm Intelligence. 5th International Workshop, ANTS 2006, pages 72–83, Berlin, Germany, 2006. Springer-Verlag.
19. James Kennedy and Russell Eberhart. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, pages 4104 – 4108, Piscataway, NJ, USA, 1997. IEEE Press.
20. Rui Mendes, James Kennedy, and José Neves. The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation, 8(3):204–210, 2004.
21. Yuhui Shi and Russell Eberhart. Empirical study of particle swarm optimization. In Proceedings of the 1999 IEEE Congress on Evolutionary Computation, pages 1945–1950, Piscataway, NJ, USA, 1999. IEEE Press.
22. Ioan C. Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. Information Processing Letters, 85(6):317–325, 2003.
23. NAKANO et al., Particle swarm optimization based on the concept of tabu search (japonais) 2007.
24. ALLAHVERDI , AL-ANZI , A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application,2006
25. SHEN et al., Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, 2007
26. XIANG et al.,Energy transmission modes based on Tabu search and particle swarm hybrid optimization algorithm, 2007

Optimisation multicritères

Weak pseudo-invexity in multiobjective programming

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 introduce new classes of generalized invex vector functions and we study Fritz-John type optimality for multiobjective problems. Relationships between these classes of vector functions are established by giving several examples. Furthermore, optimality conditions and a characterization of weakly efficient solutions are obtained under weak pseudo-invexity and by using a new concept of generalized Fritz-John vector critical point. The results obtained in this paper generalize and extend previously known results in this area.
Keywords: Multiple objective programming; Weak pseudo-invexity / $(\eta_i)_i$; Weak FJ-pseudo-invexity / $(\eta_i)_i$ and $(\theta_j)_j$; Generalized Fritz-John vector critical point; (Weakly) efficient solution.

1 Introduction

Convexity and generalized convexity play a central role in mathematical economics, engineering, management science, and optimization theory. Therefore, the research on convexity and generalized convexity is one of the most important aspects in mathematical programming. Several new concepts concerning a generalized convex function have been proposed. Among these, the concept of invexity, for differentiable functions, introduced by Hanson in [9] has received more attention. A function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be *invex* at $x_0 \in D$ with respect to η, if there exists a vector function $\eta : D \times D \to \mathbb{R}^n$ such that for each $x \in D,\ f(x) - f(x_0) \geqq [\nabla f(x_0)]^t \eta(x, x_0)$. Craven and Glover [7] and Ben-Israel and Mond [5] stated that the class of invex functions are all those functions whose stationary points are global minima. Hanson [9] noted that there are simple extensions of invexe functions, the pseudo-invex and quasi-invex functions. However, in the scalar case, Ben-Israel and Mond [5] proved that the class of invex and pseudo-invex functions coincide.
For the classical mathematical programming problem (P), defined by

$$(P) \quad \begin{array}{c} \text{Minimize } f(x), \\ \text{subject to } \ g_j(x) \leqq 0,\ j = \overline{1,k}, \end{array}$$

with $f, g_j : D \subseteq \mathbb{R}^n \to \mathbb{R},\ j = \overline{1,k}$, Hanson [9] showed that, under the invexity requirement for f and $g_j,\ j = \overline{1,k}$ (with respect to a same η), every Kuhn-Tucker critical point is a global minimizer of (P). Martin [12] remarked that the

converse is not true in general, and he proposed a weaker notion, called KT-invexity, which assures that every Kuhn-Tucker critical point is a minimizer of problem (P) if and only if problem (P) is KT-invex.

Later, researchers have extended these results to multiobjective problems. So, Ruíz-Canales and Rufián-Lizana [16] have characterized weakly efficient solutions in the case of nondifferentiable functions. In the differentiable case, Osuna-Gómez et al. [14, 15] have defined new kind of vector pseudo-invex functions and they have characterized the weakly efficient solutions for unconstrained and constrained multiobjective programming problems. Arana-Jiménez et al. [3, 4] have extended the study of Osuna-Gómez et al. [14, 15] to provide necessity and sufficiency results for efficient solutions under new kind of functions. They called these functions pseudo-invexity II in difference to pseudo-invexity of Osuna-Gómez et al. which is called pseudo-invexity I by Arana-Jiménez et al. Sufficient optimality conditions to multiobjective problems have been obtained, with different approaches, under generalized invexity with respect to a same η by Antczak [1], Giorgi and Guerraggio [8], Hanson et al. [10], Kaul et al. [11] and Mishra et al. [13]. By considering the invexity with respect to different $(\eta_i)_i$ (each function occurring in the studied problem is considered with respect to its own function η_i instead of a same function η), Slimani and Radjef [17–19] have obtained necessary and sufficient optimality conditions and duality results for nonlinear and multiobjective problems.

In the present paper, we introduce new concepts of generalized vector invex functions with respect to different $(\eta_i)_i$ and we extend the studies of Osuna-Gómez et al. [14, 15] and Arana-Jiménez et al. [3]. We establish relationships between these classes of vector functions and we obtain necessary and sufficient optimality conditions for a feasible point to be weakly efficient solution. Moreover, we introduce a new concept of Fritz-John type vector critical point and we establish a characterization of weakly efficient solutions.

2 Preliminaries and definitions

The following conventions for equalities and inequalities will be used. If $x = (x_1, ..., x_n)$, $y = (y_1, ..., y_n) \in \mathbb{R}^n$, then

$x = y \Leftrightarrow x_i = y_i$, i=1,...,n;
$x < y \Leftrightarrow x_i < y_i$, i=1,...,n;
$x \leqq y \Leftrightarrow x_i \leqq y_i$, i=1,...,n;
$x \leq y \Leftrightarrow x \leqq y$ and $x \neq y$.

We also note $\mathbb{R}^q_{\geqq}$ (resp. $\mathbb{R}^q_{\geq}$ or $\mathbb{R}^q_{>}$) the set of vectors $y \in \mathbb{R}^q$ with $y \geqq 0$ (resp. $y \geq 0$ or $y > 0$).

Invex functions were introduced to optimization theory by Hanson [9] (and called by Craven [6]) as a very broad generalization of convex functions.

Definition 1. *[9] 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:

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

(ii) def: pseudo-invex,
cond:

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

(iii) def: quasi-invex,
cond:

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

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

Definition 2. *[2] Let D be a nonempty subset of $\mathbb{R}^n$, $\eta : D \times D \to \mathbb{R}^n$ and let x_0 be an arbitrary point of D. The set D is said to be invex at x_0 with respect to η, if for each $x \in D$,*

$$x_0 + \lambda\eta(x, x_0) \in D, \ \forall \ \lambda \in [0, 1]. \tag{4}$$

D is said to be an invex set with respect to η, if D is invex at each $x_0 \in D$ with respect to the same η.

Definition 3. *[5] Let $D \subseteq \mathbb{R}^n$ be an invex set with respect to $\eta : D \times D \to \mathbb{R}^n$. A function $f : D \to \mathbb{R}$ is called pre-invex on D with respect to η, if for all $x, x_0 \in D$,*

$$\lambda f(x) + (1 - \lambda) f(x_0) \geqq f(x_0 + \lambda\eta(x, x_0)), \ \forall \ \lambda \in [0, 1]. \tag{5}$$

Definition 4. *Let $D \subseteq \mathbb{R}^n$ be an invex set with respect to $\eta : D \times D \to \mathbb{R}^n$. A m-dimensional vector valued function $\Psi : D \to \mathbb{R}^m$ is pre-invex with respect to η, if each of its components is pre-invex on D with respect to the same function η.*

In the following example, we give two scalar functions f_1 and f_2 such that each function f_i is invex at a point x_0 with respect to its own η_i, $i = 1, 2$. However, there exists no a function η for which the vector function $f = (f_1, f_2)$ is invex at x_0.

Example 1. The function $f_1 :]0, \frac{\pi}{2}[\to \mathbb{R}$ defined by $f_1(x) = x + \sin x$ is invex at $x_0 = \frac{\pi}{3}$ with respect to $\eta_1(x, x_0) = (\sin x - \sin x_0)/\cos x_0$, but f_1 is not invex at x_0 with respect to $\eta_2(x, x_0) = (\cos x_0 - \cos x)/\sin x_0$ (take $x = \frac{\pi}{6}$).
On the other hand, the function $f_2 :]0, \frac{\pi}{2}[\to \mathbb{R}$ defined by $f_2(x) = \cos x$ is invex at $x_0 = \frac{\pi}{3}$ with respect to η_2, but f_2 is not invex at x_0 with respect to η_1 (take $x = \frac{\pi}{6}$). Furthermore, it is not difficult to prove that there exists no a function $\eta :]0, \frac{\pi}{2}[\times]0, \frac{\pi}{2}[\to \mathbb{R}$ for which the functions f_1 and f_2 are both invex at $x_0 = \frac{\pi}{3}$ (take $x = \frac{\pi}{6}$).

Now, we define the invex and weakly pseudo-invex vector functions with respect to different $(\eta_i)_{i=\overline{1,N}}$.

Definition 5. *Let D be a nonempty open set of $\mathbb{R}^n$ and $\eta_i : D \times D \to \mathbb{R}^n,\ i = 1, ..., N$ be vector functions. A function $f : D \to \mathbb{R}^N$ is said to be invex at $x_0 \in D$ with respect to $(\eta_i)_{i=\overline{1,N}}$, if the function f is differentiable at x_0 and for each $x \in D$:*

$$f_i(x) - f_i(x_0) \geqq [\nabla f_i(x_0)]^t \eta_i(x, x_0),\ \textit{for all } i = 1, ..., N. \tag{6}$$

In other terms, f is invex at $x_0 \in D$ with respect to $(\eta_i)_{i=\overline{1,N}}$, if each of its components f_i is invex at x_0 with respect to its own $\eta_i,\ i = \overline{1, N}$. f is said to be invex on D with respect to $(\eta_i)_{i=\overline{1,N}}$, if f is invex at each $x_0 \in D$ with respect to the same $(\eta_i)_{i=\overline{1,N}}$. If the inequalities in (6) are strict, we say that f is strictly invex at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$.

Arana-Jiménez et al. [3, 4], have defined two classes of functions generalizing the class of scalar pseudo-invex functions. They call them pseudo-invex I, pseudo-invex in the sense of Osuna-Gómez et al. [14, 15], and pseudo-invex II (with respect to a same η). In the same manner, we introduce new kinds of functions which we will designate as weak pseudo-invex I and weak pseudo-invex II (with respect to different $(\eta_i)_{i=\overline{1,N}}$).

Definition 6. *Let D be a nonempty open set of $\mathbb{R}^n$ and $\eta_i : D \times D \to \mathbb{R}^n,\ i = 1, ..., N$ be vector functions. A function $f : D \to \mathbb{R}^N$ is said to be (def) at $x_0 \in D$ with respect to $(\eta_i)_{i=\overline{1,N}}$, if the function f is differentiable at x_0 and for each $x \in D$, (cond) holds.*

(i) def: weakly pseudo-invex I,
cond:

$$f(x) - f(x_0) < 0 \Rightarrow \exists\ \bar{x} \in D,\ [\nabla f_i(x_0)]^t \eta_i(\bar{x}, x_0) < 0,\ \textit{for all } i = 1, ..., N. \tag{7}$$

(ii) def: weakly pseudo-invex II,
cond:

$$f(x) - f(x_0) \leq 0 \Rightarrow \exists\ \bar{x} \in D,\ [\nabla f_i(x_0)]^t \eta_i(\bar{x}, x_0) < 0,\ \textit{for all } i = 1, ..., N. \tag{8}$$

If $\bar{x} = x$, in the relation (7) (resp. (8)), we say that f is pseudo-invex I (resp. II) at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$. f is said to be (weakly) pseudo-invex I (resp. II) on D with respect to $(\eta_i)_{i=\overline{1,N}}$, if f is (weakly) pseudo-invex I (resp. II) at each $x_0 \in D$ with respect to the same $(\eta_i)_{i=\overline{1,N}}$.

Remark 1. (i) In the definition 6, it is easy to see that if the vector functions $\eta_i,\ i = \overline{1, N}$ are equal to a same function η and $\bar{x} = x$, we obtain equivalently, with the condition (7), the pseudo-invexity given by Osuna-Gómez et al. [14, 15] and, with the condition (8), the pseudo-invexity II given by Arana-Jiménez et al. [3, 4].

(ii) In the definition 6, if $N = 1$ then weak pseudo-invexity I and II are equivalent and we obtain the weak pseudo-invexity of scalar function. If further $\bar{x} = x$, we deduce the pseudo-invexity of the definition 1.

In the following example, we give a vector function which is not pseudo-invex with respect to a same η (in the sense of Osuna-Gómez et al. [14, 15] and in the sense of Arana-Jiménez et al. [3, 4]) but it is weakly pseudo-invex I (and II) with respect to different $(\eta_i)_i$.

Example 2. Consider the function $f : \mathbb{R}^2 \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x_1 - x_2 + x_1^2, -x_1 + x_2 - x_2^2)$. There exists no a function η for which the vector function f is pseudo-invex (in the sense of Osuna-Gómez et al. [14, 15] and in the sense of Arana-Jiménez et al. [3, 4]) at $x_0 = (0, 0)$ (take $x = (0, 2)$). But f is weakly pseudo-invex I at x_0 with respect to $\eta_1(x, x_0) = (x_1, -x_1)$ and $\eta_2(x, x_0) = (-x_2, x_2)$ (take $\bar{x} = f(x) - f(x_0) \in \mathbb{R}^2$). Furthermore, f is weakly pseudo-invex II at x_0 with respect to the same η_1 and η_2 (take $\bar{x} = (a, b) < 0$).

We have seen that a vector function may be invex or weakly pseudo-invex I (II) with respect to different $(\eta_i)_{i=\overline{1,N}}$ without it be with respect to a same η (examples 1 and 2). However, conversely, if a vector function is invex or weakly pseudo-invex I (II) with respect to an η then it is invex or weakly pseudo-invex I (II) with respect to different $(\eta_i)_{i=\overline{1,N}}$.

Proposition 1. *Let D be a nonempty open set of $\mathbb{R}^n$. If a function $f : D \to \mathbb{R}^N$ is invex or weakly pseudo-invex I (II) at $x_0 \in D$ with respect to an η then it is invex or weakly pseudo-invex I (II) at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$ with $\eta_i(x, x_0) = \eta(x, x_0) - \nabla f_i(x_0),\ i = \overline{1, N}$.*

Remark 2. From proposition 1, we conclude that the invex (resp. weakly pseudo-invex I (II)) functions set with respect to a same η is included in the invex (resp. weakly pseudo-invex I (II)) functions set with respect to different $(\eta_i)_{i=\overline{1,N}}$ and from examples 1 and 2, we deduce that the inclusions are strict.

3 Relationships between the classes of vector functions

In this section, we present relationships between the introduced classes of functions namely invex and weakly pseudo-invex I (II) functions with respect to different $(\eta_i)_i$.

Proposition 2.

(i) It is clear that if f is invex at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$, then it is pseudo-invex I at x_0 with respect to the same $(\eta_i)_{i=\overline{1,N}}$.
(ii) If f is (weakly) pseudo-invex II at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$, then f is (weakly) pseudo-invex I at x_0 with respect to the same $(\eta_i)_{i=\overline{1,N}}$.

(iii) If f is pseudo-invex I (resp. II) at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$, then it is weakly pseudo-invex I (resp. II) at x_0 with respect to the same $(\eta_i)_{i=\overline{1,N}}$ (with $\bar{x} = x$). However, if f is weakly pseudo-invex I (resp. II) at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$, then f may not be pseudo-invex I (resp. II) at x_0 with respect to the same $(\eta_i)_{i=\overline{1,N}}$ but it will be pseudo-invex I (resp. II) at x_0 with respect to $(\tilde{\eta}_i)_{i=\overline{1,N}}$ with $\tilde{\eta}_i(x, x_0) = \eta_i(\bar{x}, x_0),\ \forall\ x \in D,\ \forall\ i = \overline{1,N}$. (see examples 3 and 4). Thus the classes of pseudo-invex I (resp. II) functions and weakly pseudo-invex I (resp. II) functions are the same class of functions.

Example 3. (f weakly pseudo-invex I $\Rightarrow$ f pseudo-invex I). Consider the function $f : \mathbb{R}^2 \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x_1 + \sin x_2, \sin x_2)$. f is weakly pseudo-invex I at $x_0 = (\frac{\pi}{6}, \frac{\pi}{3})$ with respect to $\eta_1(x, x_0) = x - x_0$ and $\eta_2(x, x_0) = x$ (take $\bar{x} = f(x) - f(x_0) \in \mathbb{R}^2$). But, f is not pseudo-invex I at x_0 with respect to the same $(\eta_i)_{i=1,2}$ because for $x = (\frac{\pi}{3}, 0)$, $f(x) - f(x_0) < 0$ and $[\nabla f_i(x_0)]^t \eta_i(x, x_0) = 0,\ \forall\ i = 1, 2$. However, f is pseudo-invex I at x_0 with respect to $\tilde{\eta}_1(x, x_0) = f(x) - f(x_0) - x_0$ and $\tilde{\eta}_2(x, x_0) = f(x) - f(x_0)$.

Example 4. (f weakly pseudo-invex II $\Rightarrow$ f pseudo-invex II). Consider the function $f : \mathbb{R}^2 \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (-x_1^2, x_2)$. f is weakly pseudo-invex II at $x_0 = (1, 0)$ with respect to $\eta_1(x, x_0) = x_0 - x$ and $\eta_2(x, x_0) = -x$ (take $\bar{x} = (0, 1) \in \mathbb{R}^2$). But f is not pseudo-invex II at x_0 with respect to the same $(\eta_i)_{i=1,2}$ because for $x = (1, -1)$, $f(x) - f(x_0) \leq 0$ and $[\nabla f_i(x_0)]^t \eta_i(x, x_0) \geqq 0,\ \forall\ i = \overline{1,2}$. However, f is pseudo-invex II at x_0 with respect to $\tilde{\eta}_1(x, x_0) = (1, -1),\ \forall\ x \in \mathbb{R}^2$ and $\tilde{\eta}_2(x, x_0) = (0, -1),\ \forall\ x \in \mathbb{R}^2$.

Let us continue the relationships between the concepts of invex and weakly pseudo-invex I (II) functions by giving the following examples.
From proposition 2 (ii), we have the class of weakly pseudo-invex II functions is included in the class of weakly pseudo-invex I functions w.r.t. $(\eta_i)_{i=\overline{1,N}}$. The converse in not true, as it is shown in example 5.

Example 5. (f weakly pseudo-invex I $\not\Rightarrow$ f weakly pseudo-invex II). Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x^2, 0)$. f is weakly pseudo-invex I with respect to any functions $(\eta_i)_{i=1,2}$ because $f(x) - f(x_0) \not< 0,\ \forall\ x, x_0 \in \mathbb{R}$. On the other hand, by choosing $x = 0$ and $x_0 = 1$, we have $f(x) - f(x_0) \leq 0$ and since $\nabla f_2(x_0) = 0$, it follows that $[\nabla f_2(x_0)]u = 0,\ \forall\ u \in \mathbb{R}$. Hence, there does not exist a function η_2 and $\bar{x} \in \mathbb{R}$ such that $[\nabla f_2(x_0)]\eta_2(\bar{x}, x_0) < 0$, and in consequence f is not weakly pseudo-invex II.

As in Arana-Jiménez et al. (2008a), the examples 6 and 7 show that the classes of invex functions and weakly pseudo-invex II functions w.r.t. $(\eta_i)_{i=\overline{1,N}}$ are different.

Example 6. (f weakly pseudo-invex II $\not\Rightarrow$ f invex). Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x^2, -x^2)$. We have f_2 is not invex because $\nabla f_2(0) = 0$ and $x_0 = 0$ is not a minimum for this function. We conclude that f is not invex.
We now prove that f is weakly pseudo-invex II. We have $f(x) - f(x_0) = (x^2 -$

$$x_0^2, x_0^2 - x^2) \leq 0 \Leftrightarrow \begin{cases} (i) \;\; x^2 - x_0^2 < 0 \;\text{ and }\; x_0^2 - x^2 \leqq 0; \\ \text{or} \\ (ii) \;\; x^2 - x_0^2 \leqq 0 \text{ and } \; x_0^2 - x^2 < 0. \end{cases}$$

If $x^2 - x_0^2 < 0$ then $x_0^2 - x^2 > 0$ and (i) is not verified. In the same way we prove that (ii) is not verified. Therefore, the inequality $f(x) - f(x_0) \leq 0$ is not verified, and we conclude that f is weakly pseudo-invex II with respect to any functions $(\eta_i)_{i=1,2}$.

Example 7. (f invex $\not\Rightarrow f$ weakly pseudo-invex II). Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x^2, 0)$. From example 5, we know that f is not weakly pseudo-invex II.
However, we have f_1 is convex and then it is invex with respect to $\eta_1(x, x_0) = x - x_0$, f_2 is invex with respect to any function $\eta_2(x, x_0)$. Therefore, the vector function f is invex with respect to $\eta_1(x, x_0) = x - x_0$ and $\eta_2(x, x_0)$ any function.

From proposition 2 (i), we conclude that the class of weakly pseudo-invex I functions contains the class of invex functions w.r.t. $(\eta_i)_{i=\overline{1,N}}$. The converse in not true, as it is shown in example 8.

Example 8. (f weakly pseudo-invex I $\not\Rightarrow f$ invex). Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x^2, -x^2)$. From example 6, we know that f is not invex. Besides, as f is weakly pseudo-invex II with respect to any functions $(\eta_i)_{i=1,2}$, it follows that, from proposition 2, f is weakly pseudo-invex I with respect to any functions $(\eta_i)_{i=1,2}$.

Let
$WPSI = \{f : D \subseteq \mathbb{R}^n \to \mathbb{R}^N / \; f$ is weakly pseudo-invex I w.r.t. $(\eta_i)_{i=\overline{1,N}}\}$,
$WPSII = \{f : D \subseteq \mathbb{R}^n \to \mathbb{R}^N / \; f$ is weakly pseudo-invex II w.r.t. $(\eta_i)_{i=\overline{1,N}}\}$,
$INV = \{f : D \subseteq \mathbb{R}^n \to \mathbb{R}^N / \; f$ is invex w.r.t. $(\eta_i)_{i=\overline{1,N}}\}$.
From (i) and (ii) of proposition 2, we conclude the following result.

Theorem 1. *$INV \cup WPSII \subset WPSI$.*

The above inclusion is strict and $INV \cup WPSII \neq WPSI$. To show this, the following example give a weakly pseudo-invex I function which is neither invex nor weakly pseudo-invex II.

Example 9. Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x^3, 0)$. f is weakly pseudo-invex I with respect to any functions $(\eta_i)_{i=1,2}$ because $f(x) - f(x_0) \not< 0, \; \forall \; x, x_0 \in \mathbb{R}$.
On the other hand, f_1 is not invex because $\nabla f_1(0) = 0$ and $x_0 = 0$ is not a minimum for this function. We conclude that f is not invex. Furthermore, by choosing $x = 0$ and $x_0 = 1$, we have $f(x) - f(x_0) \leq 0$ and since $\nabla f_2(x_0) = 0$, it follows that $[\nabla f_2(x_0)]u = 0, \; \forall \; u \in \mathbb{R}$. Hence, there does not exist a function η_2 and $\bar{x} \in \mathbb{R}$ such that $[\nabla f_2(x_0)]\eta_2(\bar{x}, x_0) < 0$, and in consequence f is not weakly pseudo-invex II.

The intersection between invex functions set and weakly pseudo-invex II functions set (w.r.t. $(\eta_i)_{i=\overline{1,N}}$) is a nonempty set, since a linear function is invex, weakly pseudo-invex I and weakly pseudo-invex II.

Example 10. Consider the function $f : \mathbb{R} \to \mathbb{R}^2$ with $f(x) = (f_1(x), f_2(x)) = (x, -x)$. We have $f(x) - f(x_0) = (x - x_0, x_0 - x) \leq 0 \Leftrightarrow (i)$ "$x - x_0 < 0$ and $x_0 - x \leqq 0$" or (ii) "$x - x_0 \leqq 0$ and $x_0 - x < 0$".
If $x - x_0 < 0$ then $x_0 - x > 0$ and (i) is not verified. In the same way we prove that (ii) is not verified. Therefore, the inequality $f(x) - f(x_0) \leq 0$ is not verified, and we conclude that f is weakly pseudo-invex II (then weakly pseudo-invex I) with respect to any functions $(\eta_i)_{i=1,2}$.
On the other hand, f is invex with respect to $\eta_1(x, x_0) = x - x_0 - 1$ and $\eta_2(x, x_0) = x - x_0 + 1$.

Consequently, the relationship between invex, w-pseudo-invex I and w-pseudo-invex II functions with respect to $(\eta_i)_{i=\overline{1,N}}$ is as given in Fig 1.

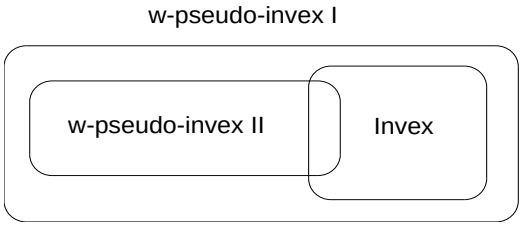


Fig. 1.

According to remark 2 and proposition 2 (iii), the figure 1 above extends the figure 1 given in Arana-Jiménez et al. [3] to the wide classes of functions.

4 Optimality conditions

We consider the following multiobjective optimization problem

$$(VP) \qquad \begin{array}{c} \text{Minimize } f(x) = (f_1(x), ..., f_N(x)), \\ \text{subject to } \ g(x) \leqq 0, \end{array}$$

where $f : D \to \mathbb{R}^N$ and $g : D \to \mathbb{R}^k$ with D is an open set of $\mathbb{R}^n$.
Let $X = \{x \in D : g(x) \leqq 0\}$ be the set of feasible solutions of (VP).
For $x_0 \in X$, we denote $J(x_0) = \{j \in \{1, ..., k\} : g_j(x_0) = 0\}$, $J = |J(x_0)|$.

We recall some optimality concepts, the most often studied in the literature, for the problem (VP). For other notions and their connections, see Yu [21].

Definition 7. *A point $x_0 \in X$ is said to be a weakly efficient (an efficient) solution of the problem (VP), if there exists no $x \in X$ such that*

$$f(x) < f(x_0) \ (f(x) \leq f(x_0)). \tag{9}$$

Weir and Mond [20] proved the following alternative lemma which will be used for establishing a characterization of weakly efficient solutions for (VP).

Lemma 1. *Let S be a nonempty invex set in $\mathbb{R}^n$ with respect to $\eta : S \times S \to \mathbb{R}^n$ and let $\psi : S \to \mathbb{R}^m$ be a pre-invex function on S with respect to the same η. Then either*

(i) $\psi(x) < 0$ has a solution $x \in S$,
or
(ii) $p^t\psi(x) \geqq 0$ for all $x \in S$, for some $p \in \mathbb{R}^m_{\geq}$,

but both alternatives are never true.

Using a Fritz-John type condition, we establish sufficient conditions for a feasible point to be weakly efficient for (VP) under weak invexity with respect to different $(\eta_i)_{i=\overline{1,N}}$.

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

1. *f is weakly pseudo-invex I at x_0 with respect to $\eta_i : X \times X \to \mathbb{R}^n,\ i = \overline{1,N}$;*
2. *g is differentiable at x_0 and for all $j \in J(x_0)$, there exists a function $\theta_j : X \times X \to \mathbb{R}^n$ such that $[\nabla g_j(x_0)]^t\theta_j(x, x_0) < 0,\ \forall\ x \in X$.*

If there exists a vector $(\mu, \lambda) \in \mathbb{R}^{N+J}_{\geq}$ such that $(x_0, \mu, \lambda, (\eta_i)_i, (\theta_j)_j)$ satisfies the following generalized Fritz-John condition

$$\sum_{i=1}^{N} \mu_i[\nabla f_i(x_0)]^t\eta_i(x, x_0) + \sum_{j\in J(x_0)} \lambda_j[\nabla g_j(x_0)]^t\theta_j(x, x_0) \geqq 0,\ \forall\ x \in X, \quad (10)$$

then x_0 is a weakly efficient solution for (VP).

Proof. Let us suppose that x_0 is not a weakly efficient solution of (VP). Then there exists a feasible point x such that $f(x) - f(x_0) < 0$.
Since f is weakly pseudo-invex I at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$, it follows that

$$\exists\ \bar{x} \in X,\ [\nabla f_i(x_0)]^t\eta_i(\bar{x}, x_0) < 0,\ \forall\ i = \overline{1,N}. \quad (11)$$

By hypothesis, we have

$$[\nabla g_j(x_0)]^t\theta_j(\bar{x}, x_0) < 0,\ \forall\ j \in J(x_0). \quad (12)$$

As $(\mu, \lambda) \in \mathbb{R}^{N+J}_{\geq}$ and from (11) and (12), it follows that

$$\sum_{i=1}^{N} \mu_i[\nabla f_i(x_0)]^t\eta_i(\bar{x}, x_0) + \sum_{j\in J(x_0)} \lambda_j[\nabla g_j(x_0)]^t\theta_j(\bar{x}, x_0) < 0,$$

which contradicts (10), and therefore, x_0 is a weakly efficient solution of (VP). ■

In the following theorem, we prove that the generalized Fritz-John condition (10) is not only sufficient for optimality but also a necessary condition.

Theorem 3. *(Fritz-John type necessary optimality condition) Suppose that*

1. *x_0 is an efficient solution for (VP);*
2. *the functions $f_i,\ i=\overline{1,N},\ g_j,\ j\in J(x_0)$ are differentiable at x_0.*

Then there exist vector functions $\eta_i : X\times D\to\mathbb{R}^n,\ i=\overline{1,N},\ \theta_j : X\times D\to\mathbb{R}^n,\ j\in J(x_0),\ (\eta_i\not\equiv 0,\ \forall\ i=\overline{1,N},\ \theta_j\not\equiv 0,\ \forall\ j\in J(x_0))$, and vector $(\mu,\lambda)\in\mathbb{R}^{N+J}_{\geqq}$ such that $(x_0,\mu,\lambda,(\eta_i)_{i=\overline{1,N}},(\theta_j)_{j\in J(x_0)})$ satisfies the generalized Fritz-John condition (10).

Proof. It suffices to take $\eta_i,\ i=1,...,N,\ \theta_j,\ j\in J(x_0),\ \mu$ and λ as follows:

- If f is a constant on X, then $\eta_i(x,x_0)$ can be any non-identically null function. If f is not a constant on X, then there exists $\bar{x}\in X,\ f(\bar{x})\neq f(x_0)$, it follows that there exists $i_0\in\{1,...,N\},\ f_{i_0}(\bar{x})>f_{i_0}(x_0)$ because x_0 is efficient for (VP).
  For all $x\in X$, consider the set $I_x=\{i\in\{1,...,N\} : f_i(x)-f_i(x_0)>0\}$. Note that I_x can be empty. Thus, $\eta_i(x,x_0)=\phi(x,x_0)t^i(x_0),\ t^i(x_0)\in\mathbb{R}^n$ with
  - $\phi(x,x_0)=\begin{cases} f_{i_x}(x)-f_{i_x}(x_0), \text{ if } I_x\neq\emptyset \text{ (with } i_x=\min\limits_{i\in I_x} i);\\ f_{i_0}(\bar{x})-f_{i_0}(x_0), \text{ otherwise.}\end{cases}$
  - $t^i_l(x_0)=\begin{cases}1, & \text{if } \frac{\partial f_i}{\partial x_l}(x_0)\geqq 0,\\ -1, \text{ otherwise,}\end{cases}$ for all $l=1,...,n$;

  (an other choice of η_i is: $\eta_i(x,x_0)=\phi(x,x_0)[\nabla f_i(x_0)]$ if $\nabla f_i(x_0)\neq 0$);
- $\theta_j(x,x_0)=-g_j(x)s^j(x_0),\ x\in X,\ s^j(x_0)\in\mathbb{R}^n$
  with $s^j_l(x_0)=\begin{cases}1, & \text{if } \frac{\partial g_j}{\partial x_l}(x_0)\geqq 0,\\ -1, \text{ otherwise,}\end{cases}$ for all $l=1,...,n$;
  (an other choice of θ_j is: $\theta_j(x,x_0)=-g_j(x)[\nabla g_j(x_0)]$ if $\nabla g_j(x_0)\neq 0$);
- $\mu_i=\frac{1}{N}$, for all $i=1,...,N$;
- $\lambda_j=\frac{1}{J}$, for all $j\in J(x_0)$. ■

5 Characterization of weakly efficient solutions

Osuna-Gómez et al. [14, 15] and Arana-Jiménez et al. [3, 4] characterized the weakly efficient and efficient solutions of (VP) by using the concepts of Kuhn-Tucker (Fritz-John) vector critical points under generalized invexity. In this section, we characterize the weakly efficient solutions by defining a new concept of generalized Fritz-John vector critical point and new kinds of invex functions (with respect to different $(\eta_i)_i$ and $(\theta_j)_j$) and which we present below.

Definition 8. *Let x_0 be a feasible point of (VP) and $\eta_i : X\times X\to\mathbb{R}^n,\ i=\overline{1,N}$, $\theta_j : X\times X\to\mathbb{R}^n,\ j\in\ J(x_0)$ be vector functions. x_0 is said to be a generalized Fritz-John (resp. Kuhn-Tucker) vector critical point with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j\in J(x_0)}$, if the functions f and g are differentiable at x_0 and there exists a vector $(\mu,\lambda)\in\mathbb{R}^{N+J}_{\geqq}$ (resp. there exist vectors $\mu\in\mathbb{R}^N_{\geqq},\ \lambda\in\mathbb{R}^J_{\geqq}$), such that $(x_0,\mu,\lambda,(\eta_i)_{i=\overline{1,N}},(\theta_j)_{j\in J(x_0)})$ satisfies the relation (10) of theorem 2.*

Osuna-Gómez et al. [14, 15] have characterized the weakly efficient solutions for (VP) by using the concept of KT-pseudo-invexity (with respect to a same η) defined in the following way.

Definition 9. *Let $\eta : X \times X \to \mathbb{R}^n$ be a vector function. The problem (VP) is said to be KT-pseudo-invex at $x_0 \in X$ with respect to η, if the functions f and g are differentiable at x_0 and for each $x \in X$:*

$$f(x) - f(x_0) < 0 \Rightarrow \begin{cases} [\nabla f_i(x_0)]^t \eta(x, x_0) < 0, \ \forall\ i = \overline{1, N}, \\ [\nabla g_j(x_0)]^t\, \eta(x, x_0) \leqq 0, \ \forall\ j \in J(x_0). \end{cases} \tag{13}$$

The problem (VP) is said to be KT-pseudo-invex on X with respect to η, if it is KT-pseudo-invex at each $x_0 \in X$ with respect to the same η.

For the study of weakly efficient solutions and the generalized Fritz-John vector critical points we need a new kind of function which we define as follows.

Definition 10. *Let x_0 be a feasible point of (VP) and $\eta_i : X \times X \to \mathbb{R}^n,\ i = \overline{1, N}$, $\theta_j : X \times X \to \mathbb{R}^n,\ j \in J(x_0)$ be vector functions. The problem (VP) is said to be weakly FJ-pseudo-invex I at $x_0 \in X$ with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j \in J(x_0)}$, if the functions f and g are differentiable at x_0 and for each $x \in X$:*

$$f(x) - f(x_0) < 0 \Rightarrow \exists\ \bar{x} \in X, \begin{cases} [\nabla f_i(x_0)]^t \eta_i(\bar{x}, x_0) < 0, \ \forall\ i = \overline{1, N}, \\ [\nabla g_j(x_0)]^t\, \theta_j(\bar{x}, x_0) < 0, \ \forall\ j \in J(x_0). \end{cases} \tag{14}$$

If $\bar{x} = x$, in the relation (14), we say that (VP) is FJ-pseudo-invex I at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j \in J(x_0)}$. The problem (VP) is said to be (weakly) FJ-pseudo-invex I on X with respect to $(\eta_i)_i$ and $(\theta_j)_j$, if it is (weakly) FJ-pseudo-invex I at each $x_0 \in X$ with respect to the same $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j \in J(x_0)}$.

Now, we establish the following theorem which characterizes the weakly efficient solutions of (VP) by using the weak FJ-pseudo-invexity I with respect to different $(\eta_i)_i$ and $(\theta_j)_j$.

Theorem 4. *Let $X \subseteq \mathbb{R}^n$ be a nonempty invex set with respect to $\eta : X \times X \to \mathbb{R}^n$ and f, g are differentiable on X. Further, let $\eta_i : X \times X \to \mathbb{R}^n,\ i = \overline{1, N}$ and $\theta_j : X \times X \to \mathbb{R}^n,\ j = \overline{1, k}$ be functions, such that for all $x_0 \in X, [\nabla f_i(x_0)]^t \eta_i(x, x_0),\ i = \overline{1, N}$ and $[\nabla g_j(x_0)]^t \theta_j(x, x_0),\ j \in J(x_0)$ are pre-invex of x on X with respect to η. Then, every generalized Fritz-John vector critical point with respect to $(\eta_i)_i$ and $(\theta_j)_j$ of problem (VP) is a weakly efficient solution if and only if (VP) is weakly FJ-pseudo-invex I on X with respect to $(\eta_i)_i$ and $(\theta_j)_j$.*

Proof. (**1**) Let $x_0 \in X$ be a generalized Fritz-John vector critical point with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j \in J(x_0)}$ for (VP). If (VP) is weakly FJ-pseudo-invex I at x_0 with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j \in J(x_0)}$, then, in the same manner as in the theorem 2, we obtain that x_0 is a weakly efficient solution for (VP).
(**2**) For the converse, suppose that every generalized Fritz-John vector critical

point with respect to $(\eta_i)_i$ and $(\theta_j)_j$ of problem (VP) is a weakly efficient solution.
Let us suppose that there exist two feasible points $\tilde{x}$ and x_0 such that

$$f(\tilde{x}) - f(x_0) < 0. \tag{15}$$

This means that x_0 is not a weakly efficient solution, and by using the initial hypothesis, x_0 is not a generalized Fritz-John vector critical point with respect to $(\eta_i)_{i=\overline{1,N}}$ and $(\theta_j)_{j\in J(x_0)}$ for (VP), ie:

$$\left(\sum_{i=1}^{N}\mu_i[\nabla f_i(x_0)]^t\eta_i(x,x_0) + \sum_{j\in J(x_0)}\lambda_j[\nabla g_j(x_0)]^t\theta_j(x,x_0) \geqq 0,\ \forall\ x\in X.\right)$$

is not satisfied for all $(\mu,\lambda)\in\mathbb{R}^{N+J}_{\geq}$. Therefore, by lemma 1, the system

$$\begin{cases} [\nabla f_i(x_0)]^t\eta_i(x,x_0) < 0,\ \forall\ i=\overline{1,N}, \\ [\nabla g_j(x_0)]^t\,\theta_j(x,x_0) < 0, \quad \forall\ j\in J(x_0). \end{cases}$$

has a solution $x=\bar{x}\in X$. In consequence, (VP) is weakly FJ-pseudo-invex I on X with respect to $(\eta_i)_i$ and $(\theta_j)_j$. ■

Remark 3. In definition 10, it is easy to see that if the vector functions $\eta_i,\ i=\overline{1,N}$ and $\theta_j,\ j\in J(x_0)$ are equal to a same function η and $\bar{x}=x$, we obtain kind of functions which is contained in the KT-pseudo-invexity class given by Osuna-Gómez et al. [14, 15]. On the other hand, the set of generalized Fritz-John vector critical points is wider that the set of usual Kuhn-Tucker vector critical points. Thus, in this sense, the theorem 4 can be considered as a generalization of theorem 3.7 (resp. theorem 2.3) given by Osuna-Gómez et al. [14] (resp. [15]).

In the following example, we show that there exist weakly efficient solutions which are not characterized by the theorem 3.7 (resp. theorem 2.3) given by Osuna-Gómez et al. [14] (resp. [15]) but they are characterized by the theorem 4.

Example 11. We consider the following multiobjective optimization problem

$$\begin{array}{rl} \text{Minimize} & f(x)=(-(x_1+1)^2,-x_1^2-x_1), \\ \text{subject to} & g_1(x)=x_1^3-x_2\leqq 0, \\ & g_2(x)=x_2\leqq 0, \\ & g_3(x)=-x_1-2\leqq 0, \end{array} \tag{16}$$

where $f:\mathbb{R}^2\to\mathbb{R}^2$ and $g=(g_1,g_2,g_3):\mathbb{R}^2\to\mathbb{R}^3$. The set of feasible solutions of problem is $X=\{x=(x_1,x_2)\in\mathbb{R}^2 : x_1^3-x_2\leqq 0,\ x_2\leqq 0 \text{ and } -x_1-2\leqq 0\}$. We have $x_0=(0,0)\in X$ is not a Kuhn-Tucker vector critical point of problem (16), because the condition of Kuhn-Tucker at x_0 takes a form $\mu_1\nabla f_1(x_0)+\mu_2\nabla f_2(x_0)+\lambda_1\nabla g_1(x_0)+\lambda_2\nabla g_2(x_0) = (-2\mu_1-\mu_2,-\lambda_1+\lambda_2)\neq (0,0),\ \forall\ (\mu_1,\mu_2)\geq 0,\ \forall\ (\lambda_1,\lambda_2)\geqq 0$. Thus, the point x_0 does not belong to the set of weakly efficient solutions characterized by the theorem 3.7 (resp. theorem

2.3) given by Osuna-Gómez et al. [14] (resp. [15]).
However, x_0 is a generalized Fritz-John vector critical point with respect to $\eta_1(x, x_0) = (x_2, 0)$, $\eta_2(x, x_0)$, $\theta_1(x, x_0)$ and $\theta_2(x, x_0)$ such that η_2, θ_1 and θ_2 can be any functions (take $\mu_1 = 1$, $\mu_2 = 0$ and $\lambda_1 = \lambda_2 = 0$). Furthermore, the problem (16) is weakly FJ-pseudo-invex I at x_0 with respect to $(\eta_i)_{i=1,2}$ and $(\theta_j)_{j=1,2}$ because $f(x) - f(x_0) \not< 0$, $\forall\, x \in X$, it follows that, by using the sufficient condition of theorem 4, x_0 is a weakly efficient solution of problem (16).
Note that x_0 is not an efficient solution of problem (16) because there exists $x = (-2, 0) \in X$ such that $f(x) - f(x_0) \leq 0$.

6 Conclusion

In this paper, we have defined new concepts of weak pseudo-invexity (I and II) and weak FJ-pseudo-invexity I to study Fritz-John type optimality for constrained multiobjective programming. Relationships between classes of vector functions are established by giving illustrated examples. New necessary and sufficient optimality conditions for a feasible point to be weakly efficient are obtained under weak invexity with respect to different $(\eta_i)_i$ and $(\theta_j)_j$. Moreover, a new concept of Fritz-John type vector critical point is introduced and a characterization of weakly efficient solutions is established under suitable generalized invexity assumptions. The content of this study extend previously known results in this area.

References

1. Antczak, T. (2003). A New approach to multiobjective programming with a modified objective function. Journal of Global Optimization, **27**, 485-495.
2. Antczak, T. (2005). Mean value in invexity analysis, Nonlinear Analysis TMA, **60**, 1473-1484.
3. Arana-Jiménez, M., Rufián-Lizana, A., Osuna-Gómez, R., & Ruiz-Garzón, G. (2008a). A characterization of pseudoinvexity in multiobjective programming. Mathematical and Computer Modelling, **48**, 1719-1723.
4. Arana-Jiménez, M., Rufián-Lizana, A., Osuna-Gómez, R., & Ruiz-Garzón, G. (2008b). Pseudoinvexity, optimality conditions and efficiency in multiobjective problems; duality. Nonlinear Analysis, **68**, 24-34.
5. Ben-Israel, A, & Mond, B. (1986). What is Invexity ?. Journal of Australian Mathematical Society Series B, **28**, 1-9.
6. Craven, B. D. (1981). Invex functions and constrained local minima. Bulletin of the Australian Mathematical Society, **24**, 357-366.
7. Craven, B. D., & Glover, B. M. (1985). Invex Functions and Duality. Journal of Australian Mathematical Society Series A, **39**, 1-20
8. Giorgi, G., & Guerraggio, A. (1998) The notion of invexity in vector optimization: smooth and nonsmooth case. In J. P. Crouzeix, J. E. Martinez Legaz, & M. Voile (Eds.), Proceedings of the Fifth Symposium on Generalized Convexity, Luminy-Marseille (France), June 17-21, 1996, in: Nonconvex Optimization and Its Applications, (PP. 389-405). Kluwer Academic, Dordrecht 27.

9. Hanson, M. A. (1981). On sufficiency of the Kuhn-Tuker conditions. Journal of Mathematical Analysis and Applications, **80**, 445-550.
10. Hanson, M. A., Pini, R., & Singh, C. (2001). Multiobjective programming under generalized type I invexity. Journal of Mathematical Analysis and Applications, **261**, 562-577.
11. Kaul, R. N., Suneja, S. K., & Srivastava, M. K. (1994). Optimality criteria and duality in multiple-objective optimization involving generalized invexity. Journal of Optimization Theory and Applications, **80**, 465-482.
12. Martin, D. H. (1985). The essence of invexity. Journal of Optimization Theory and Applications, **47**, 65-76.
13. Mishra S. K., Wang S. Y., & Lai K. K. (2005). Nondifferentiable multiobjective programming under generalized d-univexity. European Journal of Operational Research, **160**, 218-226.
14. Osuna-Gómez, R., Beato-Morero, A., & Rufián-Lizana, A. (1999). Generalized convexity in multiobjective programming. Journal of Mathematical Analysis and Applications, **233**, 205-220.
15. Osuna-Gómez, R., Rufián-Lizana, A., & Ruíz-Canales, P. (1998). Invex functions and generalized convexity in multiobjective programming. Journal of Optimization Theory and Applications, **98** (03), 651-661.
16. Ruíz-Canales, P., & Rufián-Lizana, A. (1995). A characterization of weakly efficient points. Mathematical Programming, **68**, 205-212.
17. Slimani, H., & Radjef, M. S. (2009). Duality for nonlinear programming under generalized Kuhn-Tucker condition. Journal of Optimization: Theory, Methods and Applications, **1 (1)**, 75-86.
18. Slimani, H., & Radjef, M. S. (2010a). Multiobjective Programming under Generalized Invexity: Optimality, Duality, Applications. LAP Lambert Academic Publishing AG & Co. KG, Saarbrücken, Germany.
19. Slimani, H., & Radjef, M. S. (2010b). Nondifferentiable multiobjective programming under generalized d_I-invexity. European Journal of Operational Research, **202**, 32-41.
20. Weir, T., & Mond, B. (1988). Pre-invex functions in multiple objective optimization. Journal of Mathematical Analysis and Applications, **136**, 29-38.
21. Yu, P. L. (1985). Multicriteria Decision Making: Concepts, Techniques, and Extensions, Plenum Press, New York.

A leximin linear approach for solving multicriteria Package Upgradability Problem

Aribi Noureddine[1,2] and Yahia Lebbah[2]

[1] Laboratoire LITIO, Université d'Oran Es-Sénia, Oran, Algérie
{aribi.noureddine,ylebbah}@gmail.com
[2] Laboratoire I3S/CNRS, Université de Nice - Sophia Antipolis, Nice, France

Abstract. In this paper we introduce a leximin multicriteria approach for solving the package upgradability problem on GNU/Linux Distributions. In a Linux distribution, there are thousands of packages that can meet the challenges of the interrelationships between packages. In fact, to upgrade a package, it is often necessary to upgrade a whole suite of packages that can get into conflicts with packages already installed. Furthermore, the upgrade process is achieved according to some user preferences. This problem is obviously a problem of multicriteria optimization under constraints, that can be modeled as a linear multicriteria optimization problem with boolean variables. We adopted the Leximin solving approach to address fairness and efficiency requirements of the multicriteria problem. This approach was tested using a database benchmarking of Mancoosi project (www.mancoosi.org). The contribution of our work involves the use of SCIP MIP solver to implement fully and effectively the Leximin approach.

Keywords: packages, upgrade, multicriteria optimization, aggregation operators, leximin, fairness.

1 Introduction

Free and Open Source Software (FOSS) distributions [7] are among the most complex software systems known, being made of tens of thousands deployment units known as packages. These packages evolve rapidly and are developed and released independently without a priori coordination or central authority able to control the involved parties[7, 18]. Thus raise difficult problems both for software editors and system administrators. Fore instance, system upgrade in a GNU/Linux distribution may proceed on different paths, and requires the presence of a set of previously installed packages, and the absence of another set of packages. Hence, very often it is not possible to install or upgrade all the desired packages and a possible failures can occur during upgrades. Thus, a variety of solutions are proposed [2].

Research works developed in the context of the Mancoosi (Managing the Complexity of the Open Source Infrastructure) project (www.mancoosi.org),

aim at developing tools for the system administrator in order to handle complex inter-package dependencies and frequently available package upgrades, required regularly to address security issues, bug fixes or to add new features that respect user's preferences. While the predecessor EDOS[3] project [21] had focused on tools for the distribution editor. User's preferences are expected to be handled in a consistent and efficient way, which is a current hot topic in AI with active research lines in constraint satisfaction and optimization [15, 20].

When upgrading packages, for instance, one can choose to minimize the whole size of the packages to install, to minimize the number of packages to install, to install the recent versions of the packages. All these criteria are objective functions, subject to constraints stating the dependency and avoiding conflicts. By this way, upgrading packages comes back to a multicriteria optimization problem, which has been tackled by the Mancoosi project, and Argelich et al. [3, 22] have succeeded to apply it with a boolean multilevel optimization (BMO) approach. Claude and Rueher have also succeeded to handle this problem with multicriteria ILP approach [17], where they used the sum aggregation operator on the considered criteria.

In optimization problems with a single criterion, the goal is to find an optimal solution such that the objective function is minimized or maximized. Hence, the problem is said to be well posed. Alternatively, if the optimization problem comprises more than one criterion (which is the case of most real world optimization problems), then the solution of the problem becomes difficult to characterize. In fact, we cannot find a feasible solution that optimize the whole criteria at the same time. So, what we need is an efficient ordering method that aggregate all these objective functions into a single and global objective function, which is the strategy adopted in most optimization methods [3, 2].

In this paper, we propose a MIP model of a Leximin multicriteria optimization approach applied on the upgradability problem, where the criteria and constraints are linear on 0-1 domains. At our knowledge, we have not found such a MIP formulation of the Leximin operator in the multicriteria litterature. Experimentation done on the benchmarks provided by Mancoosi are very encouraging.

This paper is organized as follows. In the next section we start by introducing the software upgradabilily problem encoded as a LP problem, and is followed by the description of some refinements of minimum ordering. We then consider how Leximin approach can be adapted to handle fairness and efficiency requirements in linear multicriteria problems. Next, the experimental result section analyses benchmarking results run on Mancoosi upgrade instances with respect to some critera. Finally the paper concludes.

2 Motivating example

Let us detail a simple example taken from the Mancoosi project. Updating constraints for package upgradability problem, denoted by PUP, is defined by a

[3] http://www.edos-project.org

triple (p, D, C) where p is the package to install, D is the set of all dependency clauses of the package p, so that each dependency clause is a disjunction of packages, and C the set of packages in conflict with p.

Let us take the set of package constraints

$$\left\{ \begin{array}{l} (p_1, \{p_2, p_5 \vee p_6\}, \emptyset), \\ (p_2, \emptyset, \{p_3\}), \\ (p_4, \emptyset, \{p_5, p_6\}) \end{array} \right\}$$

Let be some triple of updating constraints (p, D, C). Each package variable p_i is represented by a boolean variable x_i. The variable x_i is *true* if and only if p_i is installed. For each clause $c \in D$, we generate the clause $\neg x_i \vee c$ (coming from the implication $x_i \rightarrow c$, meaning that c should be installed). Similarly for each package x_j belonging to C, we generate the clause $\neg x_i \vee \neg x_j$.

By this way, we obtain both a CNF formula and an LP formulation related to the problem of installing a set of packages in the example described above. We point out that this LP formulation has been already proposed in [17].

$$\begin{array}{lll} (p_1, \{p_2, p_5 \vee p_6\}, \emptyset) & \stackrel{CNF}{\Longrightarrow} \left\{ \begin{array}{l} \neg x_1 \vee x_2 \\ \neg x_1 \vee x_5 \vee x_6 \end{array} \right. & \stackrel{LP}{\Longrightarrow} \left\{ \begin{array}{l} 1 - x1 + x2 \geq 1 \\ 1 - x1 + x5 + x6 \geq 1 \end{array} \right. \\ (p_2, \emptyset, \{p_3\}) & \stackrel{CNF}{\Longrightarrow} \neg x_2 \vee \neg x_3 & \stackrel{LP}{\Longrightarrow} 1 - x2 + 1 - x3 \geq 1 \\ (p_3, \{p_4\}, \{p_1\}) & \stackrel{CNF}{\Longrightarrow} \left\{ \begin{array}{l} \neg x_3 \vee x_4 \\ \neg x_3 \vee \neg x_1 \end{array} \right. & \stackrel{LP}{\Longrightarrow} \left\{ \begin{array}{l} 1 - x3 + x4 \geq 1 \\ 1 - x3 + 1 - x1 \geq 1 \end{array} \right. \\ (p_4, \emptyset, \{p_5, p_6\}) & \stackrel{CNF}{\Longrightarrow} \left\{ \begin{array}{l} \neg x_4 \vee \neg x_5 \\ \neg x_4 \vee \neg x_6 \end{array} \right. & \stackrel{LP}{\Longrightarrow} \left\{ \begin{array}{l} 1 - x4 + 1 - x5 \geq 1 \\ 1 - x4 + 1 - x6 \geq 1 \end{array} \right. \end{array}$$

The Debian distribution has more than 17,000 packages. CEVE-TOOL of EDOS project allows to extract the dependencies within a Debian distribution. For example, a generated instance contains 17,000 variables and 25,000 clauses. Concerning MANCOOSI project, an instance of PUP problem contains 47,956 variables and 254,529 clauses.

In an upgrade process, we can choose to optimize a user provided objective function, such as smaller packages should be preferred to larger ones in term of occupied hard disk space. This is modeled below, where 500 (resp. 100, 200, and 700) is the size of package p_3 (resp. p_4, p_5, and p_6).

$$min\ 500x_3 + 100x_4 + 200x_5 + 700x_6$$

One more criteria aims at minimizing the impact of introducing new packages in the current system, which is a reasonable assumption:

$$min \sum_{i=1}^{6} x_i$$

Several approaches are proposed in the literature in other to solve this multicriteria problem, especially in the synthesis of [15].

After the resolution process, the leximin optimal solution returned by SCIP-Soplex solver was $\langle x_1, x_2, x_5 \rangle$ which corresponds to the optimal objective vector

$\langle f_1(x), f_2(x)\rangle = \langle 200, 3\rangle$. This makes sense because we wanted to install package P_1 which requires packages P_2 and P_5 or P_6, so these packages should be installed too. Here the package P_5 was selected since it requires less disk space ($200Kb$). However, package P_3 was not installed because it falls into conflict with package P_2.

Finally, our package upgradability problem can be formulated as a linear multicriteria problem with r objective functions $f_i(x) = c^i x$ as follows:

$$LMP \begin{cases} min & f_1(x) \\ \ldots & \\ min & f_r(x) \\ \text{Subject to} & Lin : Ax \geq b. \end{cases} \quad (1)$$

Without loss of generality, that the objective functions are to be minimized.

$$min\{Cx : x \in \mathbb{Q}\}$$

where C is an $r \times n$ matrix (consisting of rows c^i) representing the vector-function that maps the decision space $X = \mathbb{R}^n$ into the criterion space $Y = \mathbb{R}^r$: $x \in X$ (the vector of decision variables). $\mathbb{Q} \subset X$ denotes the feasible set defined by a system of linear equations Lin, such that $\mathbb{Q} = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$. Where A is a given $p \times n$ matrix and $b = \langle b_1, \ldots, b_p\rangle^T$ is a given RHS vecteur.

3 Multicriteria optimization

In a multicriteria optimization problem (1), the set of constraints defines the set of feasible points. If there is only one objective minimization function, then the solutions can be totally ordered, and the solving method aims to find the solution which takes the lowest value of the objective function. If there are many objective minimization functions, the solutions can not be ordered, and it is impossible to state the best solution of some set of feasible points. Thus the first step in a multicriteria solving approach is to state a global preference function [4] which enables to order the feasible points. Generaly, the global preference is an aggregation function which aims at transforming the multicriteria problem to a monocriteria problem.

We can distinguish two main approaches related to two opposite points of view: classical utilitarianism and egalitarianism. According to the utilitarian approach, we obtain an aggregate value for each objective vector, so that the optimal objective vector is one that returns the minimal sum of all its components. This corresponds to the *sum* operator. However this kind of aggregation function is not suitable in the context of fair aggregation, because it can lead to huge differences between components. The second approach, amounts to compare the criteria evaluation vectors by comparing the minimum of each vector, thus the optimal vector is one that maximizes the minimum component. This corresponds to the *min* operator. The latter operator is interesting and suitable

[4] We are not concerned with the outranking approaches.

for problems in which fairness is essential. However, this operator has a major drawback known as *drowning effect* [5] since it cannot distinguich between two objective vectors having the same minimal value. To overcome this problem the min operator needs to be refined, since no pairs of objective vectors should remain incomparable. In the following, we discuss two important refinements of the min operator, the *discrimin* and the well-known *leximin* ordering operators[5].

In this paper we limit ourselves to make a presentation of these operators, without being exhaustive. We refer the reader to the following references for further study of various aggregation operators along with all set relations between them [6, 23, 8, 9, 11, 13, 16, 24]. Besides, operator properties which are suitable for this case together with relation between these properties are presented in [13, 16].

4 Refining minimum ordering

We recall the classical Pareto criteria: A solution is said to be Pareto-efficient if and only if we cannot strictly increase the objective value of one function unless we strictly decrease the objective value of atleast one another function. *Min* ordering solution is not Pareto optimal, since it is possible to get *min* optimal solutions which can be ordered.

In the following we use a refinements of the *min* operator, which refines the Pareto ordering.

There exists two noticeable refinements of the min operator, named *Discrimin* and *Leximin*, see e.g. [10, 4] that allow to distinguish between two vectors having the same minimal value. The *Discrimin* applies the *min* ordering once identical components having the same rank have been deleted.

Here we are concerned by the Leximin ordering, which refines the *min* ordering. In *min* ordering, given some feasible solution a, this criteria selects the objective function which takes the lowest value at a.

5 The new leximin multicriteria approach based on linear programming

Leximin ordering is a refinement of the *Discrimin* ordering, it is also based on the idea of deleting identical elements before doing a successive comparison of two solution vectors, until we find a difference that can discriminate the solutions: It is therefore a lexicographic comparison on sorted solution vectors.

Let us take an example of two solutions having the following five objective function values:

- $u = \langle 2, 5, 3, 4, 8 \rangle$ and
- $v = \langle 2, 3, 5, 6, 8 \rangle$

[5] Compromises between these two extreme approaches exists, thus the most common are: OWA aggregators, Choquet inegral and Sugeno integral (see e.g.[25, 13]).

With *Discrimin* ordering, comparing u and v result in the comparison of u' and v' where

- $u' = \langle 5, 3, 4\rangle$ and
- $v' = \langle 3, 5, 6\rangle$

since $u_1 = v_1 = 2$ and $u_5 = v_5 = 8$. Therefore, $u \sim_{discrimin} v$ since $u' \sim_{discrimin} v'$. But $v' \succ_{leximin} u'$ and then $v \succ_{leximin} u$.

A formal definition of the leximin order is given as follow:

Definition 1 (Leximin order [19]). *Let consider two solutions* $x' = \langle x'_1, x'_2, ..., x'_n\rangle$ *and* $x'' = \langle x''_1, x''_2, ..., x''_n\rangle$. *Suppose that* $x^1 = \langle x^1_1, x^1_2, ..., x^1_r\rangle = \langle f_1(x'), ..., f_r(x')\rangle$ *and* $x^2 = \langle x^2_1, x^2_2, ..., x^2_r\rangle = \langle f_1(x''), ..., f_r(x'')\rangle$. *We say that* x^1 *is preferred than* x^2, *i.e.* $x^1 \succ_{leximin} x^2$, *or that* $x' \succ_{leximin} x''$, *under leximin if and only if*

$$\exists i \in \{0, ..., r-1\}, (\forall j \in [1, i], x_j^{1\uparrow} = x_j^{2\uparrow}) \\ \wedge (x_{i+1}^{1\uparrow} > x_{i+1}^{2\uparrow}),$$

where $x^{\uparrow}$ *denotes* x *sorted in ascending order.*

if $x^{1\uparrow} = x^{2\uparrow}$, *then we say that* x^1 *and* x^2 *are indifferent, and we write* $x^1 \sim_{leximin} x^2$, *or that* $x' \sim_{leximin} x''$.

$x^1 \succeq_{leximin} x^2$ *(or as* $x' \succeq_{leximin} x''$*), iff* $x^1 \succ_{leximin} x^2$ *or* $x^1 \sim_{leximin} x^2$.

We can easily prove that $\succeq_{leximin}$ is a total order. For example, let $x^1 = \langle 4, 2, 3, 2\rangle$ and $x^2 = \langle 2, 7, 2, 2\rangle$. We have $x^{1\uparrow} = \langle 2, 2, 3, 4\rangle$ and $x^{2\uparrow} = \langle 2, 2, 2, 7\rangle$. According to the definition, it is clear that $< 4, 2, 3, 2 > \succeq_{leximin} \langle 2, 7, 2, 2\rangle$.

5.1 Leximin algorithm based on linear constraints and linear objective functions

We describe in this section the algorithm of finding a leximin-optimal solution [5], that we will linearize in a MIP scope. Concider the linear multicriteria problem (1). We need to define a single objective function $g(x)$ that combines all the mentioned objective functions. In what follows, we assume that the variable domains are defined in *Lin* inequalities given in (1).

Algorithm (1) is based on an iterative computation of the leximin optimal vector components. In the first iteration, it solves the input constraint system and computes the maximal value $\hat{v}(y_1)$ (or returns *Inconsistent* if the solution does not exist) such that $\sum_i (y_1 \leq f_i(x)) = r$, where r is the number of criteria, and by convention the value of $y_1 \leq f_i(x)$ is 1 if the inequality is satisfied and 0 otherwise [5]. After having fixed this value for y_1, it proceeds with the resolution of the second constraint system, and computes optimal value $\hat{v}(y_2)$ in the second iteration, such that $\sum_i (y_2 \leq f_i(x)) \geq r - 1$, and so on until it finds the optimal value $\hat{v}(y_r)$ which satisfies $\sum_i (y_r \leq f_i(x)) \geq 1$, this constraint is what we called *Atleast* in our algorithm and which is detailed in the following section.

Let us illustrate the behavior of this algorithm with a practical example. The following table contains six feasible solutions to a linear problem according to three criteria $\langle u_1, u_2, u_3\rangle$. shown in the columns of the following table:

Algorithm 1 LEXIMIN ALGORITHM - finding a leximin-optimal solution to a multicriteria problem.

Require: A set of linear constraints Lin, and a vector of objective functions $\langle f_1(X), ..., f_r(X)\rangle$.
Ensure: A leximin-optimal solution of MLP.
1: **if** solve(Lin) = "Inconsistent" **then**
2: **return** "Inconsistent";
3: **end if**
4: $r \leftarrow nbCriteria$;
5: **for** $i \leftarrow 1$ to r **do**
6: $X^i \leftarrow X^{i-1} \cup \{y_i\}$;
7: $Lin_i \leftarrow Lin_{i-1} \cup$ {**Atleast**($\{Y_i \leq f_1(X), ..., Y_i \leq f_r(X)\}$, $r - i + 1$)};
8: $\hat{v}_{Y_i} \leftarrow$ **Maximize**(Y_i, Lin);
9: $Dom(Y_i) \leftarrow \{\hat{v}_{Y_i}\}$; $lb(Y_{i+1}) \leftarrow \{\hat{v}_{Y_i}\}$;
10: **end for**
11: **return** X;

	s_1	s_2	s_3	s_4	s_5	s_6
u_1	3	3	5	5	7	7
u_2	9	8	3	8	3	9
u_3	1	7	1	3	7	3

The algorithm runs in atmost 3 iterative steps (which corresponds to the number of criteria): In **Step 1**, we introduce one variable Y_1, and we look for its maximal value $\hat{v}(Y_1)$ such that each (*at least* 3) objective function u_i gets at least Y_1. We find $\hat{v}(Y_1) = 3$. The variable Y_1 is fixed to this value ($Dom(Y_1) \leftarrow \{\hat{v}_{Y_1}\}$), implicitly removing solutions s_1 and s_3. In **Step 2**, we introduce a second variable Y_2, and we look for the maximal value $\hat{v}(Y_2)$ such that *at least* 2 objective functions get at least Y_2. We find $\hat{v}(Y_2) = 7$. The variable Y_2 is set to this value, implicitly deleting solution s_4. In **Step 3**, we introduce the third variable Y_3, we look for the maximal value $\hat{v}(Y_3)$ such that *at least* 1 objective function gets at least Y_3. We find $\hat{v}(Y_3) = 9$. There exists only one instantiation that maximizes Y_3: s_6. Thus, leading to the leximin-optimal vector s_6, *i.e.* $\langle u_1, u_2, u_3\rangle = \langle 7, 9, 3\rangle$.

Moreover, if we replace the second objective vector $\langle u_1, u_2, u_3\rangle = \langle 3, 8, 7\rangle$ with the objective vector $\langle u_1, u_2, u_3\rangle = \langle 9, 8, 7\rangle$, then we can break the outer loop (*c.f.* Algorithm 1, line 5) and we return the leximin-optimal vector s_2 after only one iteration. Hence, we do not need to find maximal values of Y_2 and Y_3. However, even if we continue the execution of the remaining iterations, the returned optimal solution will be the same.

Note that in our context of package upgradability problem, objective functions are defined in comprehension, so we do not have the solution vectors. Hence, we can not apply sorting in this case. However, sorting constraint (Atleast) is considered in the leximin resolution process.

Global Constraints are one of the important factors behind the success of CP. Especially, Atleast global constraint is useful in symmetry breaking and

searching for leximin optimal solution[6] to multicriteria optimization problems, where fairness and Pareto-effeciency are the major issues.

Definition 2 (Atleast constraint).

Consider a set of p constraints, and $k \in [1, p]$ an integer. The meta-constraint[7] ***Atleast***(Γ, k) *holds on the union of the scopes of the constraints in Γ and allow a tuple if and only if at least k constraints from Γ are satisfied. This constraint is derived from the cardinality combinator introduced in [14].*

Linearizing Atleast constraint

Proposition 1 (Linearizing *Atleast* constraint). ***Atleast***$(\{Y_i \leq f_1(x), ..., Y_i \leq f_r(x)\}, k)$ *is equivalent to the linear constraints, with $D_{p_j} = \{0, 1\}$, $\forall j \in [1, r]$.*

$$Atleast - lp \begin{cases} f_1(x) - Y_i \geq -M \times p_1 \\ f_2(x) - Y_i \geq -M \times p_2 \\ ... \\ f_r(x) - Y_i \geq -M \times p_r \\ \sum_{j \in 1..r}(1 - p_j) \geq k \end{cases} \tag{2}$$

proof. The **Atleast**$(\{Y_i \leq f_1(x), ..., Y_i \leq f_r(x)\}, k)$ constraint holds if and only if atleast k among r constraints are satisfied, where r is the number of criterions, i.e. $\sum_j (y_i \leq f_j(x)) = k$. In order to get the equivalent linear model, we need to keep track of the number of satisfied constraints. This restriction can be fulfilled by introducing new binary variables $P_j, j = 1, \ldots, r$, to take care of the satisfiability of each inequation. These binary variables may have an extra coefficient M, where M is a large enough constant. So, eash inequation $Y_i \leq f_j(x)$ can be formulated as $Y_i \leq f_j(x) + M \times P_j$ which are equivalent constraints because:

$$\begin{cases} P_j = 0 \Rightarrow Y_i \leq f_j(x) \\ \vee \\ P_j = 1 \Rightarrow Y_i \leq f_j(x) + M, \text{ (obsolete constraint)} \end{cases}$$

Consequently, to unsure that atleast k constraints are satisfied we just add one extra constraint $\sum_{j \in 1..r}(1 - p_j) \geq k$, and the condition is fulfilled, since k constraints are enforced to hold.

Now, let's concider the linear program (2) and trying to prove that it is equivalent to Atleast constraint. It is easy to see that the same reasoning can be used. So in this case, the Atleast constraint holds by fixing k binary variables to 0. Hence, this can only be fulfilled if $\sum_{j \in 1..r}(1 - p_j) \geq k$. □

[6] Efficient and effective filtring algorithms for Atleast constraint are proposed in [12].
[7] The prefix *meta* means that this constraint takes as a parameter other constraints.

6 Experimental results

Two fixed optimization criteria[8] are defined in Mancoosi project. Both are combinations of different criteria, and which we describe below. We consider here only the first optimization criteria, the second one can be handled in the same way.

1. `paranoid:` we want to answer the user request: *do less changes as possible*, and is a combination of two objective functions. The first objective function tends to minimize the number of packages removed in the proposed solution w.r.t. the original installation.

$$min \sum_{p \in F_{Installed}} -p \tag{3}$$

   where $F_{Installed}$ is the set of installed functionalities.

   While the second objective function tends to minimize the number of packages with a modified (set of) version(s) in the proposed solution w.r.t. the original installation;

$$min \sum_{p_i \in P_{Installed}} -p_i + \sum_{p_u \in P_{Uninstalled}} p_u \tag{4}$$

   where $P_{Installed}$ is the set of installed versioned packages[9] and $P_{Uninstalled}$ is the set of uninstalled versioned packages.
2. `trendy:` we want to answer the user request: *do more updates as possible*, minimizing the number of packages removed in the solution, minimizing the number of outdated packages in the solution, minimizing the number of unsatisfied recommendations, and minimizing the number of extra packages installed.

The algorithm (1) for finding a leximin-optimal solution, and the proposition (1) have been implemented in SCIP. SCIP solver adopts an approach that integrates Constraint and Integer Programming in a single framework [1][10]. Our program handles the problem of package upgradability management as well as similar problems of equitable resource sharing.

The results are summarized in the table below (1). The experimentations were performed with respect to the following configuration.

[8] http://www.mancoosi.org/misc.html

[9] A versioned package is represented by a variable p_v that gives the status of package p version s.

[10] http://scip.zib.de

Server:	crabe.essi.fr - 4 x CPU x86_64 3Ghz, 16 Gb of RAM
OS:	Linux Mandriva
Solver:	SCIP-SOPLEX-v1.2
Nb Vars/problem	47 956
Nb Cons/problem:	254 529
Nb problems:	100
Source:	~/mancoosi/benchmark/data.mancoosi.org/cudf/10orplus/lp
Criteria:	$Source/crit/paranoid
$Crit_1$:	min changed (45598 vars)
$Crit_2$:	min removed (2352 vars)

The experimental results show that the Leximin (with respect to the following utility vector $u = \langle u_1 = f_1(x), u_2 = 5 \times f_2(x) \rangle$) approach gets an efficient solution compared with a Lexicographic approach where the objective functions are well ordered. This means that Leximin is able to find the good order in most cases. Indeed, it is shown from the given table (1) that leximin is atleast as good as lexicographic ordering in the worst case, and this without having to look for the optimal ordrer between criterions, which is not abvious in general. These experimental results are also competitive with current approaches [17, 22]. We are planning to make a detailled comparison with these approaches. Moreover, objective values sums of Leximin ordering are better compared with Lexicographic ordering sums. Knowing that the minimal values (see e.g. last column of table (1)) correnpond to less changes and hence less system upgrades.

7 Conclusion

We introduced in this paper a leximin approach for upgradability package management problem. We have modeled the problem as a linear multiobjective problem on 0-1 domains. The leximin solution allows to provide a solution that respects the leximin ordering, guaranteeing to get the best solution on sorted objective function values.

The leximin algorithm along with the *AtLeast* meta-constraint are linearized in order to exploit a MIP solver, for instance SCIP. The experimental results compared with a well chosen lexicographic order, show the effectiveness of the leximin approach. We are planning to make a detailled comparison with BMO [22] and sum aggregation operator [17] in order to improve the current performances of our approach.

Table 1. Experimental results for both Leximin and lexicographic ordering

		Leximin				Lexico				
	Package to install	Solving time(mn)	$Crit_1$	$Crit_2$	$\sum_1$	Solving time(sec)	$Crit_1$	$Crit_2$	$\sum_2$	$\sum_1 - \sum_2$
1	hal-cups-utils	1.68	47	9	56	29.63	43	13	56	0
2	hellanzb	1.71	50	10	60	25.08	47	13	60	0
3	help2man	1.68	45	9	54	33.78	41	13	54	0
4	hgsvn	1.65	48	9	57	31.48	44	13	57	0
5	hpijs	1.73	44	9	53	30.46	40	13	53	0
6	hplip	1.74	44	9	53	31.10	40	13	53	0
7	i2e	1.83	85	17	102	29.83	73	29	102	0
8	install-info	1.64	47	9	56	30.64	40	13	155	**99**
9	iotop	2.25	98	19	117	50.91	109	46	154	**37**
					⋮					
91	libalgorithm-dep	1.75	50	9	59	25.80	46	13	59	0
92	libapache2-python	6.52	44	9	53	33.07	80	40	120	**67**
93	libapache2-mod-cgi	2.58	98	19	117	35.45	136	45	181	**64**
94	libaprutil-dbd-mysql	2.36	65	13	78	48.55	64	14	78	0
95	libapt-pkg-dev	2.19	64	12	76	32.16	56	20	76	0
96	libapt-pkg-perl	5.76	44	9	53	30.23	40	13	53	0
97	libarchive-zip-perl	6.49	140	26	166	42.02	135	51	186	**20**
98	libaudio-cd-perl	5.61	141	26	167	42.88	136	51	187	**20**
99	libaudio-flac-header	4.65	140	26	166	45.98	135	51	186	**20**
100	libaudio-musepack	5.43	142	26	168	47.23	137	51	188	**20**

References

1. Tobias Achterberg, Timo Berthold, Thorsten Koch, and Kati Wolter. Constraint integer programming: A new approach to integrate cp and mip. In Laurent Perron and Michael A. Trick, editors, *CPAIOR*, volume 5015 of *Lecture Notes in Computer Science*, pages 6–20. Springer, 2008.
2. Josep Argelich and Ines Lynce. Cnf instances from the software package installation problem. In Marco Gavanelli and Toni Mancini, editors, *15th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*, 2008.
3. Josep Argelich, Inês Lynce, and João P. Marques Silva. On solving boolean multilevel optimization problems. *CoRR*, abs/0904.0019, 2009.
4. Mohand Boughanem, Yannick Loiseau, and Henri Prade. Refining aggregation functions for improving document ranking in information retrieval. In *Proceedings of the 1st international conference on Scalable Uncertainty Management*, SUM '07, pages 255–267, Berlin, Heidelberg, 2007. Springer-Verlag.
5. Sylvain Bouveret and Michel Lemaître. New constraint programming approaches for the computation of leximin-optimal solutions in constraint networks. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 62–67, 2007.

6. Yann Collette and Patrick Siarry. *Optimisation multiobjectif*. Eyrolles, 2002.
7. Roberto Di Cosmo, Stefano Zacchiroli, and Paulo Trezentos. Package upgrades in foss distributions: details and challenges. In *Proceedings of the 1st International Workshop on Hot Topics in Software Upgrades*, HotSWUp '08, pages 7:1–7:5, New York, NY, USA, 2008. ACM.
8. D. Dubois and H. Prade. Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory. In H.J. Zimmermann, L.A. Zadeh, and B.R. Gaines, editors, *Fuzzy Sets and Decision Analysis*, pages 209–240. Studies in the Management Sciences, vol. 20, North-Holland, Amsterdam, 1984.
9. D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36:85–121, 1985.
10. Didier Dubois, Hélène Fargier, and Henri Prade. *Beyond min aggregation in multicriteria decision: (ordered) weighted min, discri-min, leximin*, pages 181–192. Kluwer Academic Publishers, Norwell, MA, USA, 1997.
11. J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Verlag, Boston, Dordrecht, London, 2005.
12. Alan M. Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, and Toby Walsh. Filtering algorithms for the multiset ordering constraint. *Artif. Intell.*, 173(2):299–328, 2009.
13. Michel Grabisch, Sergei A. Orlovski, and Ronald R. Yager. *Fuzzy aggregation of numerical preferences*, pages 31–68. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
14. Pascal Van Hentenryck, Helmut Simonis, and Mehmet Dincbas. Constraint satisfaction using constraint logic programming. *Artif. Intell.*, 58(1-3):113–159, 1992.
15. Ulrich Junker. Preference-based search and multi-criteria optimization. *Annals OR*, 130(1-4):75–115, 2004.
16. J.-L. Marichal. *Aggregation Operators for Multicriteria Decision Aid*. PhD thesis, Institute of Mathematics, University of Liège, Liège, Belgium, 1998.
17. Claude Michel and Michel Rueher. Handling software upgradeability problems with milp solvers. In Inês Lynce and Ralf Treinen, editors, *LoCoCo*, volume 29 of *EPTCS*, pages 1–10, 2010.
18. Martin Michlmayr, Francis Hunt, and David Probert. Release management in free software projects: Practices and problems. In Joseph Feller, Brian Fitzgerald, Walt Scacchi, and Alberto Sillitti, editors, *OSS*, volume 234 of *IFIP*, pages 295–300. Springer, 2007.
19. Hervi Moulin. *Axioms of Cooperative Decision Making*. Number 9780521360555 in Cambridge Books. Cambridge University Press, 1989.
20. Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Preferences in constraint satisfaction and optimization. *AI Magazine*, 29(4):58–68, 2008.
21. Ralf Treinen and Stefano Zacchiroli. Solving package dependencies: from edos to mancoosi. Technical Report arXiv:0811.3620, Nov 2008.
22. Paulo Trezentos, Inês Lynce, and Arlindo L. Oliveira. Apt-pbo: solving the software dependency problem using pseudo-boolean optimization. In Charles Pecheur, Jamie Andrews, and Elisabetta Di Nitto, editors, *ASE*, pages 427–436. ACM, 2010.
23. Ph. Vincke. *Multicriteria Decision-Aid*. J. Wiley, New York, 1992.
24. Ronald R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets Syst.*, 40:39–75, March 1991.
25. R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.

Développement d'une Métaheuristique Hybride pour la Résolution d'un Problème de Job Shop Flexible Multicritère au niveau du Complexe Cevital de Béjaia

M.S. RADJEF[1,2], N. HALIMI[3], K. BOUCHAMA[4], and A. AMER[5]

[1] CRIL-CNRS, Université Lille-Nord de France
[2] Laboratoire de Modélisation et d'optimisation des Systèmes (LAMOS)
Département de Recherche Opérationnelle, Université de Bejaia
radjef@cril.univ-artois.fr [1], yousfi_Na@hotmail.com[3]
kahina_bouchama@hotmail.fr [4]

Mots clés: Job shop flexible, ordonnancement, optimisation multicritère, métaheuristiques, coopération de métaheuristiques, algorithmes génétiques, MOSA, AG-MOSA.

Abstract. Cette étude a pour objectif la résolution d'un problème d'ordonnancement multicritère, modélisant la gestion de l'atelier de production des huiles du complexe agro-alimentaire Cevital de Béjaia.
L'ordonnancement des opérations au niveau de l'unité de raffinage des huiles doit prendre en compte les pertes de temps engendrées par le nettoyage des lignes de production à chaque changement d'huile. Ces dernières influent sur le rendement de l'atelier, ce qui fait que l'on doit les réduire, tout en respectant les délais des livraisons des demandes des clients et minimisant les temps d'achèvement des différents travaux.
Le problème est modélisé sous forme d'un job shop multicritère flexible. Or, un tel problème est connu pour sa nature NP-difficile. Par conséquent, nous avons opté pour la résolution par des métaheuristiques et une méthode hybride. En effet, nous avons appliqué une variante d'algorithme génétique, ainsi que la méthode du récuit simulé multiobjectifs (MOSA). Afin d'affiner les résultats fourni par l'algorithme génétique, nous avons développé une méthode AG-MOSA le faisant coopérer avec la méthode MOSA, et pour affirmer l'efficacité de cette hybridation, une comparaison des résultats a été effectuée.

1 Introduction

Actuellement, les organisations industrielles sont de plus en plus complexes, et la nécessité de produire à moindre coût des biens et services de plus haute qualité et d'avoir des systèmes de production plus flexibles sont des objectifs de tout gestionnaire d'une entreprise de production.

Pour faire face à son environnement concurrentiel, l'entreprise doit recourir aux outils scientifiques afin d'organiser au mieux sa production et mettre en oeuvre un système de gestion informatisé, dont l'une de ses fonctions primordiales est l'ordonnancement.

Un problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement) et de contraintes portant sur la disponibilité des ressources. En gestion de production, l'ordonnancement consiste à déterminer le séquencement des opérations à réaliser sur les différentes machines de l'atelier [7].

L'un des problèmes d'ordonnancement d'atelier le plus fréquent est le job shop flexible. Le job shop consiste à réaliser un ensemble de jobs composés d'une série d'opérations élémentaires, qui s'effectuent sur un ensemble de ressources. Ce problème est un problème d'optimisation combinatoire NP-difficile.

Le job shop flexible représente une généralisation de ce dernier, du fait qu'une opération donnée peut être réalisée par une ou plusieurs ressources et possède une durée de traitement dépendant de la ressource utilisée.

Vue la complexité de cette classe de problèmes, l'utilisation des métaheuristiques fournissant des solutions acceptables en un temps raisonnable s'avère souvent nécessaire.

Ce travail traite la gestion de la production de la raffinerie des huiles du groupe Cevital. La position du problème a été détaillée dans la section 2. Le modèle mathématique établi est présenté dans la section 3. Pour la résolution du problème, nous avons développé une métaheuristique AG-MOSA basée sur une hybridation d'un algorithme génétique avec la méthode du récuit simulé multi-objectifs, les détails concernant ces méthodes figurent dans la quatrième section. La section 5, nous fournie une interprétation des résultats obtenus et on termine l'article par une conclusion sur ce travail.

2 Position du problème

Le problème traité dans cet article concerne l'atelier de production des huiles végétales au niveau du complexe agroalimentaire Cevital. Par conséquent, il est nécessaire de présenter son processus de production à son niveau:

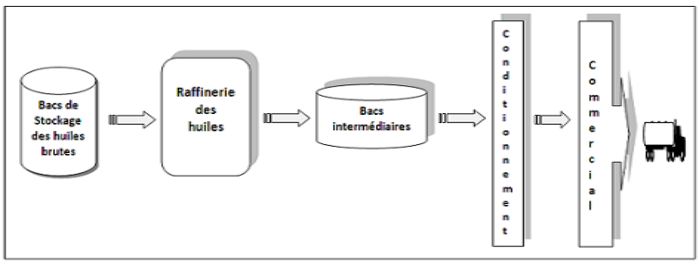


Fig. 1. Processus de production des huiles végétales.

La figure 1 illustre le cheminement des huiles brutes depuis leurs bacs de stockage jusqu'à leur expédition.

L'huile brute passe par l'unité de raffinage où elle subit un traitement sur une des lignes. L'huile raffinée sera affectée dans les bacs de stockage intermédiaires entre la raffinerie et l'unité du conditionnement. La mise en bouteilles s'effectue au niveau de l'unité de conditionnement. Ce processus s'achève par l'expédition de l'huile.

On traite plusieurs qualités d'huiles au niveau de l'unité de raffinage. De ce fait, le passage du raffinage d'un certain type d'huile vers un autre sur une même ligne, nécessite un nettoyage de la ligne afin d'assurer la bonne qualité du produit final (éviter la contaminations des huiles), ce qui engendre un retard sur la durée de réalisation des travaux dans la chaîne de production.

Par conséquent, l'objectif principal de cette étude est de trouver un ordonnancement réalisable qui minimise la date d'achèvement des travaux, le retard sur les jobs et les pertes de temps sur les lignes de raffinage. La spécificité du modèle construit est la prise en compte des opérations de nettoyage des lignes de raffinage des huiles et des bacs des stockage se situant entre les lignes de raffinage et la mise en bouteilles des huiles.

3 Modélisation du problème

Dans le problème considéré, on se fixe l'hypothèse que toutes les machines sont fiables pendant toute la période de l'ordonnancement, c'est-à-dire que les pannes et les arrêts des machines ne sont pas pris en compte dans notre étude.

Notations:

Afin de modéliser mathématiquement le problème posé, nous avons adoptés les notations suivantes:

- (i,j): Opération "j" du job "i";
- t_{ij}: date de début de l'opération (i,j);
- C_i: date d'achèvement du travail "i";
- d_{ijk}: durée de l'opération (i,j) sur la machine "k";
- $p_{jj'k}$: temps perdu lors du passage de l'opération "j" à "j'" sur la la machine "k";
- j_{der}: la dernière opération d'un job;
- D_i: date d'achèvement prévue pour le job "i";
- O_r: ensemble des opérations de raffinage;
- O_c: ensemble des opérations du conditionnement;
- O_s: ensemble des opérations de stockage;
- Qte_i: quantité d'huile 'i' à raffiner durant une semaine;
- E_{ij}: ensemble des lignes pouvant effectuer l'opération (i,j);
- Nb_{job}: le nombre de jobs considérés dans l'ordonnancement;
- $nbout_i$: nombre de bouteilles de format type 'i' (i=1,..,5) à remplir durant la semaine, sachant que Cevital utilise cinq types de bouteilles pour le conditionnement des huiles commercialisées;

- cap_i: capacité de conditionnement en format type 'i' durant la semaine;
- $cap(A)$: capacité de raffinage sur la ligne A durant la semaine;
- $cap(B)$: capacité de raffinage sur la ligne B durant la semaine;
- $cap(C)$: capacité de raffinage sur la ligne C durant la semaine.

Variables de décision

Dans ce modèle, nous avons considéré les variables suivantes:

$$X_{ijk} = \begin{cases} 1, \text{ si l'opération (i,j) est affectée à la ligne k,} \\ 0, \text{ sinon.} \end{cases}$$

$i = 1, \ldots, Nb_{job}; \quad j = 1, \ldots, nbop_i; \quad k = 1, \ldots, m,$

avec $nbop_i$: nombre d'opérations constituant le job "i";

$$Y_{(ij)(i'j')k} = \begin{cases} 1, \text{ si (i',j') succède (i,j) sur la ligne k,} \\ 0, \text{ sinon.} \end{cases}$$

$i = 1, \ldots, Nb_{job}; \;\; i' = 1, \ldots, Nb_{job}; \;\; j = 1, \ldots, nbop_i; \;\; j' = 1, \ldots, nbop'_i.$

Les fonctions objectifs

1. Minimiser la plus grande date d'achèvement des travaux:

$$\min \max_{i \in \{1,\ldots,Nb_{job}\}} C_i,$$

où

$$C_i = t_{ij_{der}} + \sum_{k \in E_{ij_{der}}} d_{ij_{der}k}.X_{ij_{der}k}, \quad \forall i = \overline{1, Nb_{job}}.$$

2. Minimiser le retard maximal sur les travaux:

$$\min \max_{i \in \{1,..,Nb_{job}\}} (C_i - D_i).$$

3. Minimiser les pertes de temps sur les lignes de raffinage:

$$\min \sum_{i=1}^{Nb_{job}} \sum_{j=1}^{nbopi} \sum_{i'=1}^{Nb_{job}} \sum_{j'=1}^{nbopi'} p_{jj'k}.Y_{(ij)(i'j')k}, \quad k = \overline{1,3}.$$

Les contraintes

1. Une opération n'est affectée qu'à une seule ligne:

$$\sum_{k \in E_{ij}} X_{ijk} = 1, \;\; \forall (i,j).$$

2. Une opération ne peut être affectée à une ligne inadéquate:

$$\sum_{k \in \overline{E_{ij}}} X_{ijk} = 0, \ \ \forall (i,j),$$

où $\overline{E_{ij}}$ est l'ensemble des machines qui n'appartiennent pas à l'ensemble E_{ij}.

3. Deux opérations ne peuvent s'effectuer simultanément sur une même ligne:

$$Y_{(ij)(i'j')k}[t_{ij} + \sum_{k \in E_{ij}} (d_{ijk} + p_{jj'k}) X_{ijk}] \leq ti'j',$$

$\forall k, \ \forall (i,j), \ \forall (i',j') \ avec \ (i,j) \neq (i',j')$.

4. Une opération d'un certain job ne peut débuter que lorsque certaines opérations de celui-ci sont achevées:

$$t_{ij} \geq t_{i(j-1)} + \sum_{k \in E_{i(j-1)}} d_{i(j-1)k} X_{i(j-1)k},$$

$\forall (i,j) \in \{i = 1, \ldots, Nb_{job}, \ j \in O_c, \ (j-1) \in O_s\}$.

5. On n'attend pas la fin du raffinage d'une huile pour commencer à la stocker:

$$t_{i(j-1)} + \alpha \sum_{k \in E_{i(j-1)}} d_{i(j-1)k} X_{i(j-1)k} \leq t_{ij},$$

$$t_{ij} \leq t_{i(j-1)} + \sum_{k \in E_{i(j-1)}} d_{i(j-1)k} X_{i(j-1)k},$$

avec: $\alpha \in]0,1[, \quad \forall (i,j)$ tel que $j = Os(l), \ (j-1) = Or(l), \ \forall l = 1, \ldots, 6$.
Notons qu'il y'a six qualités d'huiles brutes raffinées à Cevital.

6. Chaque opération précède une et une seule opération de manière directe:

$$\sum_{i'=1}^{Nb_{job}} \sum_{j'=1}^{nbopi'} \sum_{k \in E_{ij} \cap E_{i'j'}} Y_{(ij)(i'j')k} = 1, \ \forall (i,j).$$

7. Contrainte sur les capacités de raffinage des huiles:

$$\sum_{i=1}^{2} Qte_i \leq cap(A) + cap(B) + cap(C),$$

$$\sum_{i=3}^{6} Qte_i \leq cap(A) + cap(B).$$

8. Contrainte sur les capacités de conditionnement:

$$nbout_i \leq cap_i, \ \forall i = 1, \ldots, 5.$$

où le nombre de type de bouteilles de conditionnement utilisées à Cevital est de cinq.

Contraintes de positivité et de binarité

$$t_{ij} \geq 0.$$
$$X_{ijk} \in \{0,1\}, \ \ \forall i,j,k.$$
$$Y_{(ij)(i'j')k} \in \{0,1\}, \ \ \forall k,(i,j),(i',j').$$

4 Les méthodes de résolution

Le problème du job shop flexible est un problème NP-difficile. Par conséquent, sa résolution par des méthodes exactes s'avère dans la plupart des cas impossible, vu son caractère fortement combinatoire. C'est pour cela que pour un problème de taille industrielle tel que celui étudié dans ce papier, le recours à l'usage des métaheuristiques est inévitable pour obtenir une solution acceptable en un temps raisonnable.

Pour ce job shop flexible multicritère, nous avons opté pour la résolution par un algorithme génétique, la méthode MOSA et l'hybridation de ces deux algorithmes.

4.1 Algorithme Génétique (AG)

Selon la variante d'algorithmes génétiques utilisée, de nombreuses solutions plus ou moins bonnes, du problème donné sont créées au hasard. Ces "solutions" ne sont au départ pas nécessairement très bonnes. La population de solutions est alors soumise à une imitation de l'évolution des espèces: mutations et reproductions par hybridation(croisement). En favorisant la survie des plus "aptes" (les solutions les plus correctes), on provoque l'apparition d'hybrides meilleurs que chacun de leurs parents.

La population initiale donne ainsi naissance à des générations successives, mutées et hybridées à partir de leurs "parents". Le mécanisme d'encouragement des éléments les plus aptes a pour résultat que les générations successives sont de plus en plus adaptées à la résolution du problème [1].

Algorithme génétique classique:

---

(1) choisir un codage pour les individus de la population;
(2) générer la population initiale;
(3) évaluer les individus générés;
Tant que (le critère d'arrêt est non satisfait) **faire**:
(4) sélectionner les meilleurs individus;
(5) appliquer l'opérateur de croisement, puis évaluer les individus résultants;
(6) appliquer l'opérateur de mutation, puis évaluer les mutants;
(7) remplacer l'ancienne population par les meilleurs individus obtenus;
Fin tant que.

Pour la résolution de notre problème, nous nous sommes inspirés de la variante présentée dans l'article de F. Pezzella, G. Morganti et G. Ciaschettib [3]. Cette variante résout un job shop flexible monocritère, où sont intégrées des stratégies modifiant celles connues dans la littérature, et dont la combinaison fournit une meilleure valeur à la fonction objectif à chaque étape de l'algorithme. En fait, il s'agit d'une amélioration des stratégies que Kacem et al ont présenté dans [5] pour la génération de la population initiale, la selection des individus, la reproduction et la mutation.

Afin d'intégrer l'aspect multicritère dans l'algorithme à implementer, nous avons défini une fonction "fitness" pondérant les différents objectifs à optimiser afin de pouvoir évaluer les individus de la population. Par la suite, nous avons fixé un nombre d'itérations de l'algorithme nous permettant d'avoir une solution de bonne qualité en un temps de calcul raisonnable.

4.2 Structure de l'algorithme génétique

Le codage des individus: Chaque chromosome correspond à une solution du problème, c'est-à-dire à un ordonnancement. Les gènes des chromosomes décrivent l'affectation des opérations aux machines ainsi que leurs dates de début. L'ordre dont lequel ils apparaissent reflète le séquencement entre les opérations.

La population initiale: Un chromosome est obtenu en combinant plusieurs procédures:

L'affectation:

Pour résoudre le problème d'affectation, nous avons utilisé les deux règles présentées dans [3].

Le séquencement:

Une fois le problème d'affectation résolu, il faut séquencer les opérations sur les machines tout en respectant les contraintes de précédence pour un même job, et minimisant les pertes de temps au niveau des lignes de raffinage. Pour cela, on pourrait combiner entre trois règles, à savoir:

1. Random: sélectionner aléatoirement un job, puis séquencer ses opérations, répéter ceci jusqu'a ce que tous les travaux soient séquencés.
2. Most Work Remaining (MWR): on commence par séquencer les jobs dont le temps restant pour leurs achèvement est le plus grand.
3. Most Operations Remaining (MOR): on séquence d'abord les jobs dont le nombre d'opérations restantes est le plus grand.

Dans notre travail, on a ajouté une quatrième règle "séquencement" où les opérations de stockage n'attendent pas forcément la fin du raffinage des huiles en question pour débuter, mais elles peuvent commencer dès qu'un certain pourcentage de la quantité à raffiner soit prête à être stockée.

Evaluation des individus: Pour évaluer les individus de la population, on doit définir une fonction fitness ou adaptation, celle-ci peut être formulée comme une pondération des objectifs considérés.

La selection: On a choisi la méthode "Linear Ranking" qui consiste à trier les individus suivant l'ordre décroissant de leurs adaptations. On affecte un rang r_i pour chacun d'entre eux et une probabilité:

$$p_i = \frac{2r_i}{N(N+1)}, \quad i = 1, \ldots, N,$$

où N est la taille de la population.
A la fin de la procédure, on aura sélectionné un certain nombre d'individus.

Le croisement: Le croisement est un opérateur génétique que l'on applique à une paire d'individus parmi ceux sélectionnés. Certains opérateurs agissent sur le séquencement des opérations, alors que d'autres influent sur leur affectation. La procédure que l'on a implémentée est POX (Precedence Preserving Order-based Crossover) [3].

La mutation: Contrairement à l'opérateur de croisement, la mutation ne s'applique pas à un couple d'individus, mais à un seul chromosome. La mutation peut aussi influencer soit sur l'affectation, soit sur le séquencement des opérations. Notre choix s'est porté sur la procédure PPS (Precedence Preserving Shift mutation) [3].

Le Critère d'arrêt: L'algorithme s'arrête lorsque le nombre d'itérations (générations) fixé est atteint, ainsi le meilleur ordonnancement est obtenu.

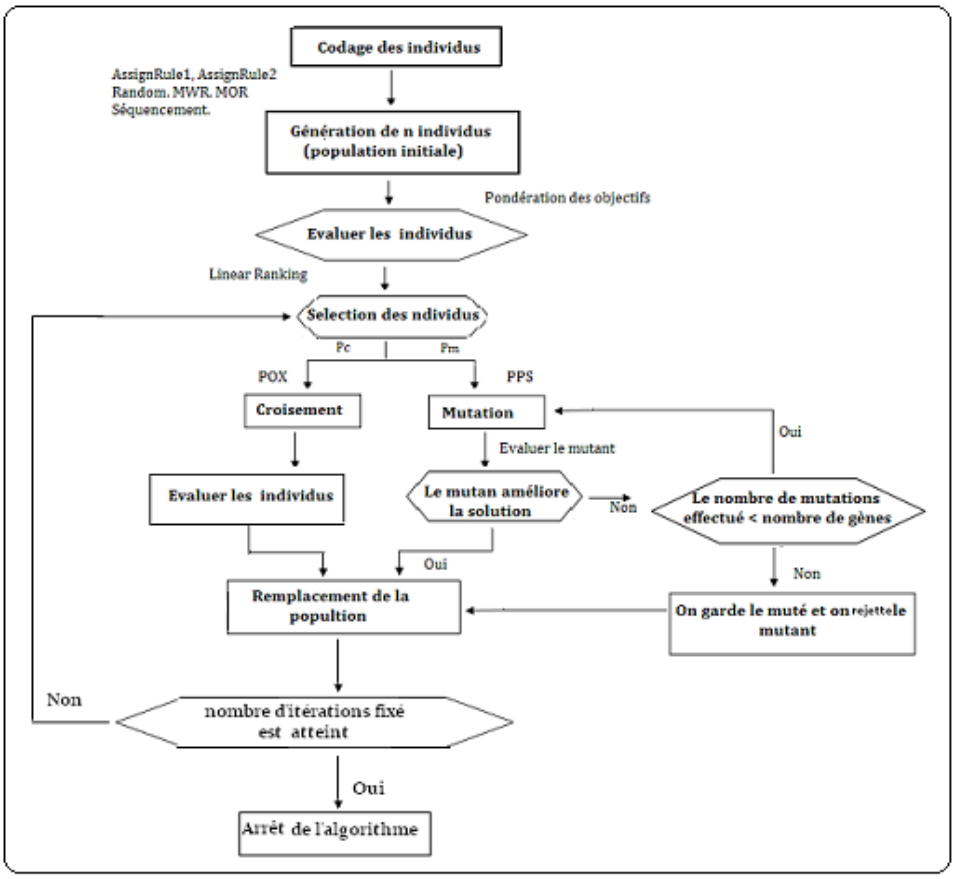


Fig. 2. Organigramme de l'algorithme génétique implémenté

4.3 La méthode MOSA

La méthode MOSA (Multi Objective Simulated Annealing) présentée dans [4], s'appuie sur la méthode du recuit simulé multiobjectif, elle prend en compte plusieurs vecteurs de pondérations des objectifs, puis fournit le vecteur d'agrégation correspondant à la meilleure solution obtenue.

Le principe de cette méthode est de générer aléatoirement une solution initiale x_0, puis de rechercher parmi les solutions voisines à x_0, une bonne approximation de la solution efficace, pour chaque vecteur poids considéré.

L'algorithme de la méthode MOSA:

(a) Initialisation:
- calculer aléatoirement une solution initiale x_0.
- évaluer la solution initiale.
- fixer la température maximale (initiale).
- fixer la température minimale (finale).
- fixer le pas de l'algorithme.
- fixer le paramètre de refroidissement.
- fixer le nombre maximal d'itérations.
- insérer les vecteurs poids à considérer.

(b) Itération (n):
- on génère aléatoirement une solution voisine à la solution calculée à l'itération (n-1).
- on évalue cette solution avec une fonction $\sum_{i=1}^{N} \lambda_i \ f_i$ où $\lambda = (\lambda_1, ..., \lambda_N)$ est le vecteur de pondération choisi à l'itération (n).
- calculer l'écart entre les deux évaluations successives $\triangle s = \sum_{i=1}^{n} \lambda_i f_i(x_n) - \sum_{i=1}^{n} \lambda_i f_i(x_{n-1})$.
- s'il y a une amélioration, alors on garde la nouvelle solution sinon on l'accepte avec une probabilité $e^{\frac{-\triangle s}{T_n}}$.
  Dans le cas d'amélioration, la température décroît avec un certain taux de refroidissement tout en respectant le pas fixé.

(c) Critère d'arrêt: l'algorithme s'arrête lorsque le nombre d'itérations fixé est atteint ou bien la température a diminué en dessous de la température finale.

4.4 L'algorithme hybride: AG-MOSA

Afin de rendre les algorithmes plus performants et robustes, de nombreux travaux proposent de combiner différentes métaheuristiques et/ou méthodes exactes. Depuis quelques années, de nombreuses approches coopératives ont vu le jour dans la littérature. Le but de ces approches est de combiner les avantages des différentes méthodes d'optimisation. Ainsi, de nombreuses coopérations de type métaheuristique/métaheuristique puis métaheuristique/exacte ont prouvé leur efficacité [6] .

Il faut savoir que la coopération entre des algorithmes ne transforme pas un problème NP-difficile en un problème polynomial. Cependant, il s'avère que des événements statistiquement rares pour un algorithme stochastique (par exemple, découvrir un optimum local d'une certaine qualité) ont une probabilité d'occurrence beaucoup moins négligeable quand plusieurs algorithmes coopèrent [2].

Pour affiner la solution fournie par l'algorithme génétique, nous avons développé une méthode faisant coopérer cet algorithme avec la méthode MOSA, que nous avons appelée méthode *AG-MOSA*.

Son principe est de prendre comme solution initiale celle fournie par l'algorithme génétique, l'évaluer, puis exécuter la méthode MOSA à partir de l'étape (b).

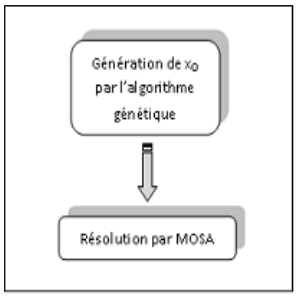


Fig. 3. Hybridation AG-MOSA

5 Interprétation des résultats obtenus avec les algorithmes sur le modèle

Nous présentons les résultats des résolutions à l'aide des trois algorithmes AG, MOSA et AG-MOSA du problème du job shop flexible multicritère, modélisant la gestion de la production de l'unité de raffinage des huiles de Cevital, disposant de trois lignes de production. On prendra en considération les temps de nettoyage à chaque changement de la qualité d'huile sur une ligne.

Nous avons considéré trois jeux de données:

- *Test (1):* la demande a été insérée pour un délai de *deux jours* et la production a été lancée en date du 06/06/2009.
- *Test (2):* la demande a été insérée pour un délai de *quatre jours* et la production a été lancée en date du 06/06/2009.
- *Test (3):* la demande a été insérée pour un délai de *sept jours* et la production a été lancée en date du 02/06/2009.

Les tests sont effectués sur une machine avec un processeur "pentium 4" (1.50 GHz). On a introduit des données communes pour les trois méthodes afin de pouvoir les comparer.

Les données insérées:

	Quantité en palettes										
Produit	Flr(5LB)	Flr(4LB)	Flr(2L)	Flr(1L)	Flr(0.75L)	Frd (5L)	Frd(2L)	Frd(1L)	El(5L)	El(2L)	El(1L)
Test 1	300	400	490	450	520	320	330	100	700	630	600
Test 2	600	800	900	950	520	420	330	200	960	610	600
Test 3	900	1000	1080	1110	720	620	530	200	1260	810	600

	Quantité en Kg				
Produit	PO	HPO	HBO	TN	ODF
Test 1	23450	34590	45400	30000	44050
Test 2	23450	123560	545400	400000	14400
Test 3	634590	703560	1345400	4000000	1114400

Flr(..): huile 'Fleurial';
Frd(..): huile 'Fridor';
El(..): huile 'Elio';
5LB (4LB): format de conditionnement 5(4) litres boxée;
PO: huile de palme;
HPO: huile de palme hydrogénée;
ODF: oléine doublement fractionnée;
TN: huile de tournesol.

Comparaison des résultats: Voici un tableau comparatif, qui récapitule les résultats obtenus (en terme de fonctions objectifs) pour chacune des méthodes implémentées, obtenus pour les trois tests.

Fonctions objectifs	Tests	AG	MOSA	AG-MOSA
Retard maximal	T(1)	3jr/15h/51min	1jr/19h/52min	1jr/12h/17min
	T(2)	2jr/0h/7 min	1jr/3h/43 min	Pas de retard
	T(3)	2jr/6h/45 min	4h36min	Pas de retard
C_{max}	T(1)	11/06 à 15h51min	09/06 à 19h52min	09/06 à 12h17min
	T(2)	15/06 à 0h7min	14/06 à 3h43min	11/06 à 5h15min
	T(3)	11/06 à 00h/37min	09/06 à 04h/36min	08/06 à 22h/19min
Perte de temps sur la ligne C	T(1)	2h	6h	2h
	T(2)	2h	2h	2h
	T(3)	2h	6h	2h
Perte de temps sur la ligne A	T(1)	8h	2h	2h
	T(2)	4h	2h	2h
	T(3)	4h	2h	2h
Perte de temps sur la ligne B	T(1)	4h	2h	2h
	T(2)	8h	6h	2h
	T(3)	8h	2h	2h

A travers ce tableau, il est clair que les résultats de la méthode MOSA sont de meilleure qualité que ceux de l'algorithme génétique. La méthode hybride AG-MOSA fournit les meilleurs résultats parmi les trois méthodes. En effet, les pertes de temps au niveau des lignes de raffinage données par l'algorithme génétique sont plus grandes que celles données par la méthode MOSA, ce qui justifie le retard important sur les jobs pour cet algorithme. La méthode hybride AG-MOSA affine la solution de l'algorithme génétique sur tous les critères, elle minimise les temps de pertes sur les lignes de raffinage et réduit considérablement le retard sur les travaux.

6 Conclusion

Dans cet article, nous avons développé une métaheuristique hybride pour la résolution d'un problème de job shop flexible multicritère modélisant le problème de

planification de la production des huiles de Cevital. Cette métaheuristique fait coopérer un algorithme génétique(AG) et la méthode du récuit simulé multi-objectifs (MOSA). Afin de prouver son efficacité, nous l'avons testé sur des données de l'entreprise et nous avons comparé les résultats obtenus avec ceux fournis par les deux premières méthodes. Après l'étude des résultats de ces métaheuristiques, nous avons conclu que la méthode hybride répond mieux aux objectifs fixés. De plus, cette méthode permet de fournir une solution acceptable en un temps raisonnable pour pouvoir procéder une planification de la production.

References

1. M. Gen, R. Cheng and L. Lin (2008). Network Models and Optimization Multiobjective Genetic Algorithm Approach. Springer Edition.
2. J.K Hao, P. Galinier and M. Habib (1999). Métaheuristiques pour l'Optimisation Combinatoire et l'Affectation sous Contraintes. Revue d'Intelligence artificielle. **13(2)**: 283-324.
3. F. Pezzellaa, G. Morgantia and G. Ciaschettib (2008). A Genetic Algorithm for the Flexible Job-shop Scheduling Problem. Computers and Operations Research. **35**: 3202-3212.
4. T. Loukil, J. Teghem and P. Fortemps (2007). A Multi-objective Production Scheduling case study Solved by Simulated Annealing. European Journal of Operational Research **179**: 709-722.
5. I. Kacem, S. Hammadi and P. Borne (2002). Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-Shop scheduling problems. IEEE Transactions on Systems, Man, and Cybernetics **32**: 1-13.
6. M. Basseur (2005). Conception d'Algorithmes Coopératifs pour l'Optimisation Multi-objectifs: Application aux Problèmes d'Ordonnancement de Type Flow-Shop. Thèse de doctorat, Université des Sciences et Technologie de Lille.
7. V. T'Kindt and J.C. Billant (2006). Multicriteria Schudeling: Theory, Models and Algorithms. Springer.

A Novel Method for Integer Chance Constrained Problems with Multiple Objective

Fatima BELLAHCENE

Operational Research and Mathematics Decision Aid Laboratory (LAROMAD), Faculty of Sciences, Mouloud Mammeri University, Tizi-Ouzou 15000, Algeria.
f_bellahcene@yahoo.fr

Abstract. We deal with obtaining Pareto optimal solutions of a multiobjective stochastic integer linear programming problem when chance constrained programming (CCP) is used to handle with the randomness. Under the assumption that the random variables are independent and normally distributed, it is shown how this problem can be approximated by an equivalent multiobjective integer monotonic programming problem. The algorithm developed here for identifying the Pareto optimal solutions of the considered problem combines the discrete polyblock approach and the notion of level sets. A numerical example is included to illustrate the different steps of the presented algorithm.

keywords: Multiple objective programming, Stochastic Programming, Chance constrained programming, nonlinear programming, level sets.

1 Introduction and background

Much of decision making in the real-world takes place in an environment where the objectives, constraints or parameters are not known precisely (see Lai and Hwang [12] and Liu [13]). Therefore a decision is often made on the basis of vague information or uncertain data. The uncertainty may be interpreted as randomness or fuzziness. The randomness occurring in the multiobjective linear programming (MOLP) problems is categorized as the multiobjective stochastic linear programming (MOSLP) problems. As in the stochastic optimization problems (see, for example, Birge and Louveaux [6], Kall [9], Prékopa [16], Stancu-Minasian [19] and Klein Haneveld &Van der Vlerk [10]) the problem coefficients are assumed as random variables with known distributions in most of cases. (MOSLP) models are appropriate when data evolve over time and decisions need to be prior to observing the entire data stream. Then, the way of modeling the (MOSLP) problems and obtaining efficient solutions depends in large part on the nature of available information about the random parameters.

The (MOSLP) models have been developed for a variety of applications, including portfolio selection (Aouni and al. [2], Ballestero and al. [3], Shing and al. [18], Ogryczak [14]), investment planing (Ben Abdelaziz and al. [4]) and electric power generation (Teghem and al. [21]), to mention a few.

Most previous efforts in this field have been devoted to positive decision variables. In many situations, however, fractional values of the variables are not physically meaningful. Therefore, modeling with multiobjective stochastic integer linear programming

(MOSILP) programs and the development of solution algorithms for such problems are of great interest to management scientists.

Progress has not been substantial on (MOSILP) and the present literature on it is surprisingly thin (see for example, Abbas and Bellahcene [1], Saad and Kittani [17], Teghem [22]). In [1], the Benders decomposition method [5] and four types of cuts are used to develop a generating technique for identifying a compromise solution from a set of available candidates. The stochastic data are treated by a recourse approach to obtain an equivalent multiobjective integer linear programming (MOILP) two-stages problem. Duality proprieties are then used to check for feasibility of the recourse problem. In [17], a solution algorithm is presented for solving (MOILP) problems involving dependent random parameters in the objective functions and linearly independent random parameters in the constraints. The STRANGE-MOMIX method presented by Teghem in [22] is interactive and based on the generalized Tchebycheve norm to generate efficient solutions.

This paper deals with the multiobjective integer problem with probabilistic constraints. The random variables are assumed to be normally distributed with mean and variance that are linear in the decision variables. Due to the randomness, it is almost impossible to solve it directly. It is shown how this problem can be transformed into an equivalent multiobjective integer monotone programming problem. The algorithm presented here combines the polyblock method [15] with the notion of level sets to compute all the Pareto optimal solutions respecting given reservation levels.

The next section describes the considered stochastic model and shows how to convert it into an equivalent multiobjective deterministic monotone program. In section 3, we review some basic properties of monotonic functions followed by a brief description of the polyblock approach in section 4. The level-sets characterization of the pareto optimal solutions is given in section 5. The algorithm development is detailed in section 6. We conclude the paper with an illustrative example and some considerations on possible future research in this field.

2 Problem statement and structural properties

We consider a multiobjective integer linear programming problem involving random variable coefficients in the objectives functions formulated as:

$$\begin{array}{c} "max" \ (f_1(x), f_2(x) \ldots, f_p(x)) \\ subject\ to\ \ P_r[A_i(w)x \leq d_i(w)] \geq q_i,\ i = 1, \ldots, m \\ x \in X = \{x \in \mathbb{Z}_+^n \mid a_j \leq x_j \leq b_j,\ j = 1, ..., n\} \end{array} \qquad \text{(PS)}$$

$f_k,\ k = 1, \ldots, p$ are linear increassing functions of the form $f_k(x) = C_k x$ where $C_k^t \in R^n$ and $x \in R^n$. $A(m \times n)$ and $b(m \times 1)$ represent stochastic parameters. w is a random vector from the probability space (Ω, Σ, P). $P_r\{t\}$ denotes the probability of the event $t \in \Sigma$ under the probability measure P_r and $q_i,\ i = 1, \ldots, m$ are probability levels.

In the following, the basic technique of chance constrained programming (CCP) is used to transform problem (PS) into an equivalent multiobjective integer monotonic problem according to the predefined probabilities $q_i,\ i = 1, \ldots, m$. We assume

that the random variables a_{ij} and d_i are normally and independently distributed $a_{ij} \curvearrowright N(\mu_{ij}, \nu_{ij}^2)$ and $d_i \curvearrowright N(m_i, \sigma_i^2)$ then, the random vector $t_i(x) = A_i x - d_i$ is normally distributed with mean

$$m_i(x) = \sum_{j=1}^{n} \mu_{ij} x_j - m_i$$

and variance

$$\sigma_i^2(x) = \sum_{j=1}^{n} \nu_{ij}^2 x_j^2 + \sigma_i^2.$$

Using this observation, the constraints in (PS) can be written as:

$$\begin{aligned} P_r\left(A_i x \leq d_i\right) \;\geq\; q_i &\Longleftrightarrow P_r\left(\frac{t_i(x) - m_i(x)}{\sigma_i(x)} \leq \frac{-m_i(x)}{\sigma_i(x)}\right) \geq q_i \\ &\Longleftrightarrow m_i(x) + \Phi^{-1}(q_i)\sigma_i(x) \leq 0 \end{aligned}$$

Where $\Phi(\cdot)$ is the distribution function of the standard normal distribution. The values $\Phi^{-1}(q_i)$ which are positive for $q_i \geq \frac{1}{2}$ (see, Ishii [8]) can be obtained from any standard normal distribution table.

Problem (PS) is rewritten as:

$$\begin{gathered} max\left(f_1(x), f_2(x), \ldots, f_p(x)\right) \\ subject\ to \\ g_i(x) = \sum_{j=1}^{n} \mu_{ij} x_j - m_i + \Phi^{-1}(q_i)\sqrt{\sum_{j=1}^{n} \nu_{ij}^2 x_j^2 + \sigma_i^2} \leq 0,\ i = 1, \ldots, m \qquad \text{(PD)} \\ x \in X = \{x \in \mathbb{Z}_+^n \mid a_j \leq x_j \leq b_j,\ j = 1, ..., n\} \end{gathered}$$

where g_i, $i = 1, \ldots, m$ are real increasing functions. Kataoka [11] is credited for formulating problem (PS) and the development of problem (PD).

3 Characteristics of monotonic programs

For any two vectors $x, y \in \mathbb{R}^n$ we write $x \leq y$ to mean $x_i \leq y_i$ for every $i = 1, \ldots, n$. If $a \leq b$ then the box $[a, b]$ is the set of all $x \in \mathbb{R}^n$ satisfying $a \leq x \leq b$. When $x \leq y$ we also say that y dominates x. A function $f : \mathbb{R}_+^n \longmapsto \mathbb{R}$ is said to be increasing if $f(y) \geq f(x)$ whenever $y \geq x \geq 0$. A set $S \subset [a, b]$ is said to be normal if

$$a \leq x \leq y\ ,\ \ y \in S \implies x \in S \tag{1}$$

The normal hull of S is the smallest normal set containing S.

Proposition 1. *The normal hull of S is the set $S^{\urcorner} = \bigcup\limits_{z \in S} [a, z]$. If S is compact so is $S^{\urcorner}$.*

Proof. Let $P = \underset{z \in S}{\cup} [a, z]$. P is normal and $P \supset S$, hence $P \supset S^{\urcorner}$. Conversely, if $x \in P$ then $x \in [a, z]$ for some $z \in S \subset S^{\urcorner}$, hence $x \in S^{\urcorner}$ by normality of $S^{\urcorner}$, so that $P \subset S^{\urcorner}$ and, therefore $P = S^{\urcorner}$. If S is compact then S is contained in a ball B centered at 0, and if $x^k \in S^{\urcorner}$, then since $x^k \in \left[a, z^k\right] \subset B$, there exists a subsequence $\{k_\nu\} \subset \{1, 2, \ldots\}$ such that $z^{k_\nu} \longrightarrow z^0 \in S,\ x^{k_\nu} \longrightarrow x^0 \in \left[a, z^0\right]$, hence $x^0 \in S^{\urcorner}$, proving the compactness of $S^{\urcorner}$.

Definition 1. *A polyblock P is the normal hull of a finite set $T \subset [a, b]$ called its vertex set.*

By proposition 1, $P = \underset{z \in T}{\cup} [a, z]$. The intersection of finitely many polyblocks is a polyblock. A vertex z of a polyblock is proper if there is no vertex $z' \neq z$ "dominating" z i.e. such that $z' \geq z$ and improper otherwise. Improper vertices can be deleted without changing the polyblock, so a polyblock is fully determined by its proper vertices.

Proposition 2. *The maximum of an increasing function f over a polyblock is achieved at a proper vertex of this polyblock.*

Proof. Let x^* be a maximizer of $f(x)$ over a polyblock P. Since a polyblock is the normal hull of its proper vertices, there exists a proper vertex z of P such that $x^* \in [a, z]$. Then $f(z) \geq f(x^*)$ because $z \geq x^*$, so z must be also an optimal solution.

4 The polyblock approach

Consider a single objective optimization program (PD1) derived from (PD) by choosing the first objective. Note that in place of f_1, one can similarly consider the objective function f_k for any $k \in \{2, ..., p\}$.

$$\begin{gathered} max\ f_1(x) \\ \text{subject to} \quad g_i(x) \leq 0,\ \ i = 1, \ldots, m \qquad \text{(PD1)} \\ x \in X = \{x \in \mathbb{Z}_+^n \mid a_j \leq x_j \leq b_j,\ j = 1, ..., n\} \end{gathered}$$

From property (1), the set $S = \{x \in X \mid g_i(x) \leq 0\ for\ i = 1, ..., m\}$ defined above (with m increasing functions g_i) is normal.

Let us define

$$G(x) = \underset{i=1,\ldots,m}{max} \{g_i(x)\} \tag{2}$$

The boundary of the constraints can be expressed as $\Gamma = \{x \in X \mid G(x) = 0\}$.
Let $\langle \alpha, \beta \rangle$ be an integer box in X with $\alpha \in S$ and $\beta \notin S$. Suppose also that $G(\alpha) < 0$. Let x_b be an intersection point of the line $x = \lambda^* \alpha + (1 - \lambda^*)\beta,\ 0 \leq \lambda^* \leq 1$ and the boundary Γ. Since $G(\alpha) < 0$ and $G(\beta) > 0$, there must exist an x_b in X that satisfies $G(x_b) = 0$, i.e., $g_i(x_b) \leq 0$ for $i = 1, ..., m$ and there exists at least one i such that $g_i(x_b) = 0$.

$$\lambda^* = \sup \{\lambda \in [0, 1] \mid \lambda\alpha + (1 - \lambda)\beta \in S\} \tag{3}$$

Bisection method or Newton's method can be used to search for the root of equation (3).

Denote by $\lfloor x \rfloor$ the integer vector with its i-th component being the maximum integer less than or equal to x_j , $j = 1, ..., n$ and denote by $\lceil x \rceil$ the integer vector with its j-th component being the minimum integer greater than or equal to x_j, $j = 1, ..., n$. Let $x^F = \lfloor x_b \rfloor$ and $x^I = \lceil x_b \rceil$. Suppose that x_b is not integral (otherwise $x^F = x^I$). It is easy to see that x^F is a feasible point ($x^F \in S$) and x^I is infeasible ($x^I \notin S$).

Consider the integer boxes $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$. By the monotonicity of f_1 and g_i, there are no feasible points better than x^F in $\langle \alpha, x^F \rangle$ and there are no feasible points in $\langle x^I, \beta \rangle$. Therefore, we can remove integer boxes $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$ from $\langle \alpha, \beta \rangle$ for further consideration after comparing x^F with the incumbent solution.

We will refer the process of cutting non-promising integer boxes and partitioning a revised domain into sub-boxes as domain cut. The domain cut is based on the monotone properties of f_1 and g_i, (see Xun and al. [19, p. 172]).

Theorem 1. *Let $A = \langle \alpha, \beta \rangle$, $B = \langle \alpha, \gamma \rangle$ and $C = \langle \gamma, \beta \rangle$ where $\alpha \leq \gamma \leq \beta$. Then both $A \setminus B$ and $A \setminus C$ can be partitioned into at most n new integer boxes.*

$$A \setminus B = \bigcup_{j=1}^{n} \left(\prod_{k=1}^{j-1} \langle \alpha_k, \gamma_k \rangle \, \langle \gamma_j + 1, \beta_j \rangle \times \prod_{k=j+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{4}$$

$$A \setminus C = \bigcup_{j=1}^{n} \left(\prod_{k=1}^{j-1} \langle \gamma_k, \beta_k \rangle \, \langle \alpha_j, \gamma_j - 1 \rangle \times \prod_{k=j+1}^{n} \langle \alpha_k, \beta_k \rangle \right) \tag{5}$$

Theorem 1 shows that the set of the integer points left in $\langle \alpha, \beta \rangle$ after removing $\langle \alpha, x^F \rangle$ and $\langle x^I, \beta \rangle$ can be partitioned into a union of smaller integer boxes.

5 Characterization of efficient solutions

In the following, we will use the concept of Pareto optimality to define the maximization in (PD).

Definition 2. *A solution $x^* \in S$ is called Pareto optimal if and only if there is no $x \in S$ such that $f_k(x) \geq f_k(x^*)$, $k = 1, \ldots, p$ and $f_k(x) > f_k(x^*)$ for at least one $k \in \{1, \ldots, p\}$. The set of all Pareto optimal solutions is denoted by $Spar$.*

Independent of the properties of the objective functions C_k or the constraint set S, Pareto optimal solutions can be characterized geometrically. In order to state this characterization, we introduce the notion of level sets and level curves.

Definition 3. *Let $\eta_k \in R$ for $k = 1, \ldots, p$*
1. The set $L^k_{\geq}(\eta_k) = \{x \in S \mid f_k(x) \geq \eta_k\}$ is called the level set of f_k with respect to the level η_k
2. The set $L^k_{=}(\eta_k) = \{x \in S \mid f_k(x) = \eta_k\}$ is called the level curve of f_k with respect to the level η_k.

The following characterization of Pareto optimal solutions by level sets and level curves was given by Ehrgott and al. [7].

Lemma 1. *Let $x^* \in S$. Then x^* is Pareto optimal if and only if*

$$\bigcap_{k=1}^{p} L^k_{\geq}(f_k(x^*)) = \bigcap_{k=1}^{p} L^k_{=}(f_k(x^*))$$

i.e. x^* is Pareto optimal if and only if the intersection of all p level sets of f_k with respect to levels $f_k(x^*)$ is equal to the intersection of the level curves of $f_k,\ k = 1, \ldots, p$ with respect to the same levels.

Because we will use the result of Lemma 1 throughout the paper the following notation will be convenient.
For $\eta \in R^p$ let

$$S(\eta) = \{x \in S \mid f_k(x) \geq \eta_k,\ k = 1, \ldots, p\} = \bigcap_{k=1}^{p} L^k_{\geq}(\eta_k)$$

Correspondingly, $S(\eta)_{par}$ will denote the Pareto set of $S(\eta)$.

When the decision maker is not able to specify the levels η_k, these can be replaced by the values $f^N_k = \min\limits_{x \in S} f_k(x)$ where $f^N = (f^N_1, f^N_2, \ldots, f^N_p)$ is the nadir point of problem (PD).

6 Pareto optimal solutions with reservation levels

The goal is to use the characterization given in Lemma 1 to find all Pareto optimal solutions of problem (PD) respecting given reservation levels $\eta_k,\ k = 1, \ldots, p$. Instead of an explicit computation of the intersection of level sets and checking the condition of Lemma 1, we will generate one level set $L_{\geq}(\eta_1)$ (without loss of generality) and then check for each element of this level set if it is also contained in the other level sets and if it dominates or is dominated by a solution found before. The level sets construction uses the polyblock method which consists of finding a feasible point x^F and an infeasible point x^I and generating integer boxes using the formulas (4) and (5). The best feasible solution obtained during the generation of integer boxes is kept as an incumbent solution. Moreover, by the monotonicity of the problem, an integer box with the function value of its upper bound point less than or equal to the function value of the incumbent can be discarded. The partition is successively refined and integer boxes that do not contain an optimal solution are removed. The set of the nondominated solutions respecting the given levels is find in a finite number of iterations.

6.1 Algorithm

(Initialization). Set $(\eta_1, \eta_2, \ldots, \eta_p)$.

Let $a = (a_1, ..., a_n)$, $b = (b_1, ..., b_n)$.

If a is infeasible, then problem (PD1) has no feasible solution. If b is feasible, then b is the optimal solution to (PD1), Otherwise, set $x^1 = a$.

If $f_1(x^1) < \eta_1$ then stop $S(\eta)_{par} = \varnothing$

$k = 1$

$X^k =< a, b >,\ S(\eta)_{Par} = \left\{x^k\right\}$

Step 1 : $k = k + 1$ (Box selection and finding boundary point).

Select an integer box $\langle \alpha, \beta \rangle \in X^k$.

Use the bisection method to find the roots of the equation

$$G\left[\lambda\alpha + (1-\lambda)\beta\right] = 0, \quad \lambda \in [0, 1]$$

where G is defined in (2). Set $x_b = \lambda^*\alpha + (1-\lambda^*)\beta$. Set $x^F = \lfloor x_b \rfloor$. If $x^F = x_b$ then set $x^I = x_b + e_j$, where e_j is the j-th unit vector in R^n with $x_b + e_j \leq \beta$. Otherwise, set $x^I = \lceil x_b \rceil$.

Step 2 : If $f_1(x^F) < \eta_1$, go to step 4.

else If $x^k \in L_{\geq}$ for all $k = 2, \ldots, p$ then go to step 3.

Step 3 : For $1 \leq i \leq k-1$.

If x^k dominates x^i then $S(\eta)_{par} = S(\eta)_{par} \setminus \left\{x^i\right\}$, go to step 4.

else if x^i dominates x^k then go to step 4.

else if $f_1(x^k) = f_1(x^i)$ then $S(\eta)_{par} = S(\eta)_{par} \cup \left\{x^k\right\}$, go to step 4.

Step 4 : (Partition and Remove).

(i) Apply the formula (5) to partition the set $\Omega_1 = \langle \alpha, \beta \rangle \setminus \langle x^I, \beta \rangle$ into a union of integer boxes. Let $x^F \in \langle \widehat{\alpha}, \widehat{\beta} \rangle \in \Omega_1$. Set $\Omega_1 = \Omega_1 \setminus \langle \widehat{\alpha}, \widehat{\beta} \rangle$.

(ii) Apply the formula (4) to partition the set $\Omega_2 = \langle \widehat{\alpha}, \widehat{\beta} \rangle \setminus \langle \widehat{\alpha}, x^F \rangle$.

(iii) Set $Y^k = \Omega_1 \cup \Omega_2$.

(iv) Perform the following for each integer box $\langle \alpha, \beta \rangle$ generated in the above partition process:

(a) If β is feasible, remove $\langle \alpha, \beta \rangle$ from Y^k. Furthermore if $f_1(\beta) > f_1(x^k)$ set $x^k = \beta$.

(b) If α is infeasible, remove $\langle \alpha, \beta \rangle$ from Y^k.

(c) If $f_1(\beta) < f_1(x^k)$, remove $\langle \alpha, \beta \rangle$ from Y^k.

(d) If α is feasible, β is infeasible and $f_1(\alpha) > f_1(x^k)$.

Set $x^k = \alpha$ and $f_1(x^k) = f_1(\alpha)$ go to step 2.

Denote Z^k the set of integer boxes after the above removing process.

Step 5 : (Updating integer boxes)

Remove all integer $\langle \alpha, \beta \rangle$ in X^k with $f_1(\beta) < f_1(x^k)$. Set $X^{k+1} = X^k \cup Z^k$.

If $X^{k+1} = \varnothing$ stop, otherwise set $k = k + 1$ go to step 1.

The finite convergence of the algorithm can be easily seen from the finiteness of X and the fact that at each iteration at least the integer points x^F and x^I are removed from X^k. The algorithm proceeds successively by refining the partition and removing integer boxes that do not contain promising solutions and finally terminates in a finite number of iterations.

7 Conclusion

The main contribution of this study is a level-set-polyblock method for generating Pareto optimal solutions for multiobjective integer linear problems with probabilistic constraints. This method does not require specific mathematical properties to be satisfied by the objectives or the constraints. It exploit only their monotonic properties. It appears on several examples that the algorithm performs faster and since only the Pareto optimal solutions respecting given reservation levels must be found, this method may be usefull for large monotonic programs. However further experimental validation of this observation is needed.

References

1. Abbas M. and Bellahcene F., (2006). Cutting plane method for multiple objective stochastic integer linear programming problem. European Journal of operational research 168 (3), 967-984.
2. Aouni B., Ben Abdelaziz F., Martel J.M., (2005). Decision-maker's preferences modeling in the stochastic goal programming. European Journal of Operational Research 162, 610–618.
3. Ballestero E., (2005). Using stochastic goal programming: Some applications to management and a case of industrial production. Information Systems and Operational Research Journal 43 (2), 63–77.
4. Ben Abdelaziz F., Mejri S., (2001). Application of goal programming in a multi-objective reservoir operation model in Tunisia. European Journal of Operational Research 133, 352-361.
5. Benders J.F, (1962). Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238-252.
6. Birge A.J.R., and Louveaux F.V., (1997). Introduction to Stochastic Programming. Springer Verlag, New York.
7. Ehrgott M., Hamacher H.W., Klamroth K. , Nickel S., Schobel A., and Wiecek. M.M., (1997). A note on the equivalence of balance points and Pareto solutions in multiple-objective programming. Journal of Optimization Theory and Applications, 92(1), 209 - 212.
8. Ishii H., Nishida T. and Nanbu Y., (1978). A generalized chance constrained programming problem. Journal of Operations Research Society of Japan 21(1).
9. Kall P. and Wallace S.W., (1994). Stochastic Programming. Wiley, Chichester, Also available as PDF file at http://www.unizh.ch/ior/Pages/Deutsch/ Mitglieder/Kall/bib/ka-wal-94.pdf.
10. Klein Haneveld W.K. and van der Vlerk M.H., (1999). Stochastic integer programming: General models and algorithms. Ann. Oper. Res., 85, 39-57.
11. Kataoka S., (1963). A stochastic programming model. Econometrica 31, 181-196.
12. Lai Y.J. and Hwang C.L., (1994). Fuzzy Multiple Objective Decision Making: Methods and Applications. Springer-Verlag, Berlin, New York.
13. Liu B., (2002). Theory and Practice of Uncertain Programming; Physica-Verlag, Heidelberg.
14. Ogryczak W., (2000). Multiple criteria linear programming model for portfolio selection. Annals of Operations Research 97, 143-162.
15. Prékopa A., (1995). Stochastic Programming. Kluwer Academic Publishers, Boston.
16. Rubinov A., Tuy H. and Mays H., (2001). An algorithm for monotonic global optimization problems. Optimization (49), 205 221.
17. Saad O.M. and Kittani H.F., (2003). Multiobjective integer linear programming problems under randomness. LAPOR TRANSACTIONS, 28(2), 101-108.

18. Shing C. and Nagasawa H., (1999). Interactive decision system in stochastic multi-objective portfolio selection. International Journal of Production Economics 60-61,187-193.
19. Stancu-Minasian I.M., (1984). Stochastic Programming with Multiple Objective Functions. D. Reidel Publishing Company.
20. Sun X.L. and Li D., (2006). Nonlinear Integer Programming, Springer Science &Business Media, New York.
21. Teghem J. and Kunsch P. L., (1985). Multi-objective decision making under uncertainty: an example for power systems. In Haimes Y.Y. and Chankong V. (Eds), Decision Making with Multiple Objective, Springer, 443-456.
22. Teghem J., (1990). Strange-Momix : an Interactive method for mixed integer linear programming; in R. Slowinski and J. Teghem (eds), Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty; Dordrecht: Kluwer Academic Publishers, 101-115.

Représentation des connaissances et applications

Un Système de Classification basé sur la Logique Floue et la Recherche Locale pour la Détection d'Intrusions

Dalila Boughaci, Samia Bouhali et Selma Ordeche

LRIA/USTHB
BP 32 El-Alia, Beb-Ezzoaur, 16111, Alger, Algérie.
dboughaci@usthb.dz, dalila_info@yahoo.fr

Résumé Dans ce travail, nous proposons une recherche locale utilisant les concepts de la logique floue pour la détection d'intrusion. Le système de classification flou proposé démarre d'un ensemble de règles floues *si-alors* générées aléatoirement. Ensuite, une recherche locale est appliquée afin d'optimiser les règles de base générées initialement par le système. L'algorithme proposé a été implémenté et validé sur les benchmarks DARPA. Les données que nous manipulons sont orientées détection d'intrusions. Plus précisément, nous traitons 10% de la base de donnée $KDD'99$. Nous considérons les cinq classes de la batterie de test KDD à savoir la classe DoS (Denial-of-Service), la classe $R2L$ (Remote to Local Access), la classe $U2R$ (User to Root attacks), la classe $Probing$ (Probing Attack) et la classe $Normale$. Les résultats trouvés sont encourageants et montrent l'intérêt de notre approche.

Mots clés : Recherche locale, Logique Floue, Détection d'Intrusion, Classification, KDD.

1 Introduction

La détection d'intrusions est la capacité d'un système informatique de déterminer automatiquement, à partir d'événements relevant de la sécurité, qu'une violation de sécurité se produit ou s'est produite dans le passé. Pour ce faire, la détection d'intrusions nécessite qu'un grand nombre d'événements de sécurité soient collectés et enregistrés afin d'être analysés [14].

Les systèmes de détection d'intrusion (IDS) sont divisés en deux catégories, suivant l'approche utilisée :

1. L'approche comportementale qui consiste à décrire le comportement (profil) usuel d'un utilisateur et ce, afin de détecter toute action anormale ou inhabituelle de cet utilisateur. Contrairement à l'approche par scénarios, l'approche comportementale permet de détecter des attaques inconnues auparavant ainsi que les abus de privilèges des utilisateurs légitimes du système. Parmi les méthodes proposées pour construire les profils, nous citons :
   – Les méthodes statistiques où le profil est calculé à partir de variables prises aléatoirement et échantillonnées à intervalles réguliers [9]. Ces variables

peuvent être, par exemple, la durée et l'heure des connexions, le temps machine utilisé, etc.
- Les systèmes experts dont la base de règles décrit statistiquement le profil de l'utilisateur au vu de ses précédentes activités. Son comportement courant est comparé aux règles, à la recherche d'une anomalie [16].
- Les réseaux de neurones où le principe consiste à apprendre à un réseau de neurones le comportement normal d'un utilisateur. Par la suite, lorsqu'on lui fournira les actions courantes, il devra décider de leur normalité [6].

2. L'approche par scénarios qui consiste à définir des comportements anormaux et ce, afin d'analyser les données susceptibles d'être des attaques. L'approche utilise souvent une base de scénarios d'attaques. Parmi ces méthodes, nous trouvons :
   - Les systèmes experts où leur base de règles décrit les attaques. Les événements d'audit sont traduits en des faits [13].
   - Les algorithmes génétiques où chaque individu de la population code un sous-ensemble particulier d'attaques qui sont présentes dans les traces d'audit. Cette approche permet d'optimiser de temps de recherche dans le journal d'audit [11].
   - Le pattern matching utilisé pour localiser les signatures d'attaques dans les traces d'audit [10]. L'inconvénient majeur de l'approche par scénarios réside dans la détection des attaques connues uniquement, ce que nécessite une remise à jour de la base de scénarios d'attaque très souvent.

Récemment, de nouvelles approches ont été approfondies, un certain nombre de méthodes ont été proposées et de nombreux systèmes ont été conçus pour détecter les intrusions. Parmi eux, nous citons les systèmes de détection d'intrusions à base d'agents [4] qui peuvent fournir de nombreux avantages pour les solutions existantes en raison de la mobilité des agents et de leurs aspects coopératifs. D'autres techniques ont été largement appliquées pour la détection d'intrusions comme les approches de DataMining [12, 7], le clustering [15], les réseaux bayésiens [2] et les algorithmes génétiques flous [1].

Dans ce travail, on s'intéresse à la recherche locale basée sur les concepts de base de la logique floue. Le système que nous proposons est un classificateur flou, dont la base de connaissances est modélisée sous forme de règles floues *si-alors* et qui sont améliorées par une recherche locale. L'objectif principal est la conception d'un système capable de distinguer entre les connexions normales et les attaques.

Les données que nous manipulons sont celles de KDD'99 et sont orientées détection d'intrusions. Plus précisément, nous traitons 10% de la base de donnée KDD'99 correspondant à 494019 de connexions d'apprentissage. La base de données test contient 311029 connexions. Dans ce travail, on considère les cinq classes de la batterie KDD. Plus précisément, on considère les quatre catégories d'attaques (Denial-Of-Service (DOS), Remote to Local Access ($R2L$), User to Root attacks ($U2R$), Reconnaissance Probing) et bien sûr la classe normale. Dans la base d'apprentissage il y a 19.65% (resp. 79.07%, 0.23%, 0.22%, 0.83%) de connexions d'apprentissage normales (resp. DOS, $R2L$, $U2R$, $Probing$) et

19.48% (resp. 73.90%, 5.21%,0.07%, 1.34%) de connexions de test normales (resp. DOS, $R2L$, $U2R$, $Probing$). La cinquième classe est la classe Normale qui ne contient aucune attaque.

L'article est organisé comme suit. La deuxième section propose une recherche locale basée sur les concepts de la logique floue pour la détection d'intrusions. L'implémentation et quelques résultats numériques sont donnés en troisième section. Enfin, la quatrième section porte sur la conclusion et dresse quelques perspectives de ce travail.

2 Présentation de notre approche

2.1 Phase de prétraitement et normalisation des données

Chaque ligne de la base de données $KDD'99$ code un flot de données entre une source et une destination identifiées par leur adresse IP, sous un protocole donné (TCP, UDP...). Chaque ligne est donc une *connexion* caractérisée par 41 attributs tels que sa durée, le type du protocole, etc. A partir des valeurs de ces attributs, la *connexion* est considérée comme étant une connexion normale ou bien une attaque. Les attributs de chaque *connexion* sont soit de type discret ou bien sont de type continu. Nous considérons uniquement les seize attributs significatifs pour la phase de normalisation et qui sont : $A8$, $A9$, $A10$, $A11$, $A13$, $A16$, $A17$, $A18$, $A19$, $A23$, $A24$, $A32$, $A33$, $A1$, $A5$, et $A6$. Un attribut significatif veut dire un attribut qui pourra nous aider à classifier la connexion. L'algorithme de normalisation consiste à rendre la valeur numérique de chaque attribut Ai de la connexion au format normalisé, c'est-à-dire entre 0.0 et 1.0 selon l'équation (1).

$$X = \frac{X - MIN}{MAX - MIN} \quad (1)$$

Où : X est la valeur numérique de l'attribut Ai, MIN est la valeur minimale que pourra avoir l'attribut Ai, MAX est la valeur maximale que pourra avoir l'attribut Ai.

Après avoir étudié la base de données, les valeurs MIN et MAX de chaque attributs numériques Ai sont données comme suit : $A8$: fragment erroné [0,3], (MIN=0, MAX=3), $A9$: nombre de paquet urgent [0,14], $A10$: nombre d'indicateur hot [0,101], $A11$: nombre d'essais login raté [0,5], $A13$: nombre de condition de compromis [0,9], $A16$: nombre d'accès à la racine[0,7468], $A17$: nombre d'opérations de création de fichiers [0,100], $A18$: nombre de prompts Shell [0,5] , $A19$: nombre d'opérations sur les fichiers de contrôle d'accès [0,9], $A23$: nombre de connexions pour le même hôte [0,511], $A24$: nombre de connexions pour le même service [0,511], $A32$: nombre de connexions pour le même hôte [0,255], $A33$: nombre de connexions pour le même hôte utilisant le même service [0,255].

Toutefois, pour les attributs numériques $A1$, $A5$ et $A6$, nous avons constaté une valeur MAX très grande d'où la nécessité de calculer le log (10) de X puis le log (10) de la valeur MAX pour pouvoir normaliser la valeur de l'attribut en question. $A1$: durée de la connexion [0,58329]. ($MIN = 0$, $MAX =$58329),

$A5$: nombre de données (en octets) de la source vers la destination [0,1.3 milliard]. $A6$: nombre de données (en octets) de la destination vers la source [0,1.3 milliard].

2.2 Phase de recherche locale

Notre approche peut être subdivisée en deux grandes phases :

Une première phase initiale permettant de générer un ensemble de règles floues. Nous avons utilisé le concept de la logique floue dans la résolution du problème de la détection d'intrusion car la logique floue est un moyen efficace permettant l'introduction du concept de degré d'appartenance, qui détermine les *forces* avec lesquelles un objet appartient aux différentes classes. Cela repose sur le fait que la logique floue ne cherche pas un point de rupture qui décide de l'appartenance d'un objet à une classe, mais qu'elle raisonne plutôt sur la base d'un intervalle de valeurs. Pour plus de détails sur le concept de la logique floue, le lecteur pourra se référer à Zadeh [17].

Une deuxième phase de recherche locale permettant de chercher la bonne règle de classification. La recherche locale que nous proposons démarre d'une solution initiale aléatoire (une règle floue *si-alors*) et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une solution Sol correspond à des éléments adjacents à Sol dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à un certain nombre d'itérations fixé empiriquement.

Les grandes lignes de notre approche peuvent être données comme suit :

- 1) Créer d'une manière aléatoire une solution initiale Sol (une règle floue si-alors).
- 2) Evaluer la solution initiale Sol.
- 3) Générer une nouvelle règle floue *si-alors* Sol' pour l'itération suivante.
- 4) Remplacer la solution courante par la meilleure solution générée.
- 5) Mettre fin à l'algorithme si la condition d'arrêt est satisfaite, sinon aller à l'étape 2).

Codage d'une règle floue Une règle floue *si-alors* notée Rj est codée sous forme d'une chaîne de caractères. Cinq valeurs linguistiques sont utilisées et qui peuvent être associés aux attributs Ai. Les cinq valeurs sont : S : Small, MS : Medium Small, M : Medium, ML : Medium Large et L : Large comme l'indique la Figure 1.

Exemple :

- Soit la règle : Si $X1$ est moyen, $X2$ est moyen petit, $X3$ est grand et $X4$ est petit, alors la classe Cj avec $CF = CFj$. Où Xi sont les attributs de la connexion, Cj est la classe obtenue après classement de la règle et CFj son degré de confiance.
- Le codage correspondant : "M, MS, L, S"

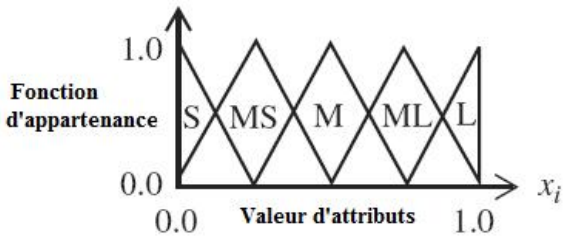


Figure 1. La fonction d'appartenance (Membership functions) des cinq valeurs linguistiques (S : small, MS : medium small, M : medium, ML : medium large, L : large)

Calcul de la fonction d'appartenance $\mu(X)$ Pour chaque attribut de la connexion, un ensemble flou contenant les cinq valeurs linguistiques est associé. La fonction d'appartenance (fuzzification) notée $\mu(X)$ est calculée par une projection sur le graphe de l'ensemble flou (Figure 2), La valeur de $\mu(X)$ est donnée par la formule (2).

$$\mu(X) = Max \ \{0, 1 - (|X - X0|/b)\} \quad (2)$$

Où ă : b c'est la base du triangle, b=0.5. $X0$= 0, 0.25, 0.5, 0.75, 1 correspondant à S, MS, M, ML, L. X est la valeur de l'attribut après la normalisation.

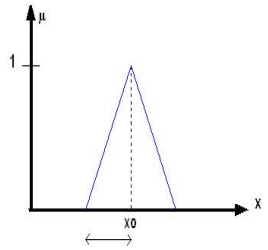


Figure 2. Méthode de fuzzification

Génération d'une solution initiale La solution initiale Rj représentant une règle floue "Si-alors" est générée aléatoirement. Pour chaque attribut Xi de la connexion Xp, une valeur linguistique (parmi les cinq valeurs de l'ensemble flou) est lui assigné, et ce pour la partie si de la règle floue.

Évaluation d'une règle floue *si-alors* La procédure d'évaluation est utilisée pour évaluer la règle floue "si-alors" Rj. Elle permet d'associer à la connexion

en entrée Xp une classe Cj avec un certain degré de confiance et une valeur du fitness. Nous utilisons ici, la méthode introduite par (Ishibuchi et Murata [8]. Les différentes étape d'évaluation d'une règle floue Rj sont données comme suită :

- 1) Calculer la compatibilité de connexions avec la règle générée Rj.
  Soit la règle floue "si-alors" notée $Rj = Aj_1 \quad Aj_2 \ldots Aj_n$, nous calculons la compatibilité de chaque connexion Xp avec la règle Rj par la formule (3).

$$\mu_{Rj}(Xp) = \mu_{Aj_1}(X1) \times \mu_{Aj_2}(X2) \times \ldots \mu_{Aj_n}(Xn), \quad p = 1, 2 \ldots m \qquad (3)$$

  Où $\mu_{Aj_i}()$ est la fonction d'appartenance de Aj_i, mă : est le nombre total de connexions, Xiă : sont les attributs de la donnée de la connexion Xp et n est le nombre d'attributs qui égal à 41.
- 2) Calculer la somme des qualités de compatibilité pour chaque catégorie des cinq classesă :
  Pour chaque classe h appartenant à l'ensemble des cinq classes DoS, $R2L$, $U2R$, $Probing$ et $Normal$, nous calculons la valeur de $\beta_{CLASS_h}(Rj)$ comme suit :

$$\beta_{CLASS_h}(Rj) = \sum_{Xp \in CLASS_h} \mu_{Rj}(Xp) \qquad h = 1, 2 \ldots C \qquad C = 5 \qquad (4)$$

- 3) Associer la classe h ayant la valeur maximale $\beta_{CLASS_h}(Rj)$ à la rège Rj.

$$\beta_{CLASS_{Cj}}(Rj) = \max \quad \{\beta_{CLASS_1}(Rj) \ldots \beta_{CLASS_C}(Rj)\} \qquad (5)$$

  La formule (5) nous permet de considérer la classe ayant la valeur maximale, et c'est la classe qui correspond à notre règle floue "si-alors" Rj générée. Une fois la classe Cj est déterminée, alors le degré de confiance CFj est spécifié dans la formule (6). Dans le cas où nous avons deux classes prenant la même valeur maximale, dans ce cas la classe n'est pas spécifiée et ă$Cj = nul$ ainsi que $CFj = 0$.

$$CFj = \frac{\beta_{CLASS_{Cj}}(Rj) - \bar{\beta}}{\sum_{h=1}^{C} \beta_{CLASS_h}(Rj)}$$

  Où

$$\bar{\beta} = \frac{\sum_{h \neq Cj} \beta_{CLASS_h}(Rj)}{C - 1} \qquad (6)$$

- 4) Calculer le fitness de la règle Rj. La fonction fitness [8, 1] que nous avons utilisée est donnée par la formule (7). Elle est égale à la somme des PPF (le facteur de puissance positive) pour l'ensemble des modèles considéré.

$$fitness(Rj) = \sum_{p \in CLASS_{Cj}} PPF_p^{Rj}$$

$$PPF^{Rj} = \begin{cases} 1 \text{ si } \mu_{Rj}(Xp) > 0 \\ 0 \text{ si } sinon \end{cases} \qquad (7)$$

Génération de solutions voisines Dans cette méthode, on part d'une configuration Sol représentant une règle floue si-alors. Ensuite, on fait subir à la configuration une modification élémentaire consistant en la modification aléatoire de la valeur d'un attribut. Un attribut peut prendre l'une des cinq valeurs linguistiques Aji. Suite à cette modification, on trouvera plusieurs solutions voisines possibles. Ces dernières sont évaluées selon la procédure d'évaluation et la solution ayant le fort fitness, notée Sol' sera considérée comme solution courante pour l'itération suivante de la recherche locale.

Critère d'arrêt Nous pouvons utiliser différents critères d'arrêt pour mettre fin à l'exécution de notre algorithme. Dans le présent travail, nous arrêtons la recherche après un certain nombre total d'itérations fixé d'une manière empirique.

2.3 L'algorithme de recherche locale proposé

Une version de l'algorithme de la recherche locale pour l'analyse du fichier d'Audit de Sécurité) (AFAS) est donnée dans l'algorithme 1.

Algorithm 1 : Un Algorithme de recherche locale Flou pour l'AFAS.

Require: une instance de AFAS contennat m connexions, max_itérations.
Ensure: un ensemble de règles floues R, classification de chaque connexions, leur degré de confiance ainsi que leur fitness.

1: **Pour** chaque connexions Xp de l'instance AFAS
2: **Faire**
3: Générer aléatoirement une solution initiale Sol ;
4: Appliquer la procédure d'évaluation à Sol
5: Attribuer une classe C à la règle Sol et calculer son degré de confiance CF
6: Evaluer le fitness de Sol
7: **Pour** I=1 a max_iterérations
8: **Faire**
9: Générer une solution voisine Sol' ;
10: Appliquer la procédure d'évaluation à Sol'
11: Attribuer une classe à Sol'
12: Evaluer le fitness de Sol'
13: **SI** le fitness (Sol') fitness(Sol) **alors** Sol=Sol'ă
14: **Finsi**
15: **Fait**
16: **return** la meilleure règle Sol trouvée ayant le fort fitness.
17: **return** la classification trouvée.
18: **Fait**

3 Expérimentation et résultats numériques

Pour implémenter notre application nous avons procédé par étapes. L'étape initiale consiste à subdiviser la donnée en entrée, qui est les 10% de la base de

donnée $KDD'99$ en cinq classes. Nous avons créé cinq matrices de donnéesă : quatre matrices pour chaque classe d'attaques plus la matrice de la classe *Normal* contenant les événements classés normaux. La deuxième étape est la création de l'interface graphique qui nous permet de choisir la matrice à analyser. La troisième étape est la partie traitement ou calcul faite sous MATLAB. Enfin, la dernière étape est l'étape d'affichage ou d'impression des résultats obtenus.

Nous avons donc cinq matrices à savoir la matrice contenant les éventements de type $U2R$, la matrice contenant les éventements de type $R2L$, la matrice contenant les éventements de type $Probing$, la matrice contenant les éventements de type DOS et enfin la matrice contenant les éventements de type $Normal$.

Après la création des cinq matrices, nous lançons la phase de normalisation des différents attributs de connexions de toutes les matrices et dont les valeurs sont rendues dans l'intervalle [0, 1]. En appliquant la règle de normalisation des attributs, on aura les cinq matrices normalisées de type $U2R$, $R2L$, $Probing$, $Normal$ et DOS. La phase qui suit la phase de normalisation est la génération des règles floues. Pour ce faire, nous avons utilisé la fonction rand (nombre aléatoire pour générer des nombres aléatoires qui doivent être parmi les cinq valeurs (1, 2, 3, 4, et 5) qui correspondent respectivement à (S : Petit, MS : Petit Moyen, M : Moyen, ML : Moyen Grand et L : Grand).

Remarque : Toutes les matrices Rand représentant les règles floues subissent un traitement itératif qui consiste à faire changer ces matrices aléatoirement selon le principe de notre recherche locale. Le nombre d'itérations de la recherche locale est fixé empiriquement à 200.

3.1 Résultats Numériques

Toutes les expériences ont été menées sur un ordinateur de CPU Intel Core de Duo 1.8GHz avec 2Go de Ram. Les tables de 1 à 5 donnent les résultats de classification trouvés par notre approche. Nous donnons ici les résultats concernant uniquement une dizaine de règles où la colonne 1 représente le numéro de la règle random générée, la colonne *classification* donne la classe trouvée par notre approche et attribuée à la règle floue. La colonne *degré de confiance* calcule la valeur du degré. Enfin la colonne *Fitness* donne la valeur de fitness trouvée pour chaque règle.

A travers les résultats que nous avons trouvés, on peut dire que notre méthode arrive à détecter les intrusions pour les quatre types d'attaques DOS, $R2L$, $U2R$ et $Probing$ et les fausses alertes sont minimes. Le taux de sucées est de 80% pour la classe DOS, 85% pour la classe $R2L$, 95% pour la classe $U2R$, et 80% pour la classe $Probing$. Toutefois, pour la classe $Normal$, nous avons remarqué que la méthode échoue et le taux de succès est de 10%.

Pour caractériser les performances de l'approche proposée, nous définissons tout d'abord le faux positif et le faux négatif et nous calculons par la suite les taux de détection.

– Faux positif : est le cas où une intrusion est signalée (détectée) mais où il n'y a pas attaque ; c'est typiquement une fausse alerte (l'IDS fait *mal* son travail).

Table 1. Résultats de classification trouvés par notre approche pour la classe DOS

Règles	Classification	degré de confiance	Fitness
R1	PROBING	1.0000	17
R2	NORMAL	0.3245	10
R3	DOS	1	5
R4	DOS	1	18
R5	DOS	1	18
R6	DOS	1	18
R7	DOS	1	11
R8	DOS	1	11
R9	DOS	1	5
R10	DOS	0.7132	2
R11	NORMAL	1	8
R12	NORMAL	1	8
R13	DOS	1	11

Table 2. Résultats de classification trouvés par notre approche pour la classe U2R

Règles	Classification	degré de confiance	Fitness
R1	DOS	0.7132	2
R2	U2R	0.8880	8
R3	R2L	0.3801	18
R4	U2R	0.7983	4
R5	DOS	1	5
R6	U2R	0.9745	5
R7	NON SPECIFIE	0	0
R8	DOS	1	1
R9	DOS	1	4
R10	DOS	1	11
R11	DOS	1	11
R12	NORMAL	0.9056	12
R13	DOS	1	4

Table 3. Résultats de classification trouvés par notre approche pour la classe R2L

Règles	Classification	degré de confiance	Fitness
R1	U2R	0.9469	8
R2	NON SPECIFIE	0	0
R3	PROBING	1	17
R4	R2L	0.9807	18
R5	NORMAL	0.9979	20
R6	PROBING	0.5291	11
R7	DOS	1	5
R8	R2L	0.5026	18
R9	U2R	0.7983	4
R10	NORMAL	1	8
R11	NORMAL	1	8
R12	DOS	1	11
R13	PROBING	1	17

Table 4. Résultats de classification trouvés par notre approche pour la classe PROBING

Règles	Classification	degré de confiance	Fitness
R1	NORMAL	1	8
R2	DOS	1	11
R3	PROBING	1	17
R4	PROBING	0.9966	17
R5	PROBING	1	17
R6	PROBING	1	17
R7	PROBING	1	17
R8	NORMAL	0.3245	10
R9	DOS	1	5
R10	U2R	0.9469	8
R11	NON SPECIFIE	0	0
R12	PROBING	1	17
R13	DOS	1	11

Table 5. Résultats de classification trouvés par notre approche pour la classe NORMAL

Règles	Classification	degré de confiance	Fitness
R1	U2R	0.9469	8
R2	R2L	1	1
R3	PROBING	1	17
R4	PROBING	1	17
R5	PROBING	1	17
R6	PROBING	1	17
R7	PROBING	1	17
R8	R2L	0.3801	18
R9	NORMAL	1	8
R10	PROBING	1	17
R11	PROBING	1	17
R12	PROBING	1	17
R13	NORMAL	1	8

– Faux négatif : est un cas d'attaque réelle non détectée ; dans ce cas là on considère que l'IDS ne fait pas son travail.
– Taux de vrais positifs TP : est la proportion de fausses alertes signalées par l'approche.
– Taux d'absence TA : est la proportion des cas d'attaques non signalées par l'approche.

Pour mesurer la performance de notre algorithme, nous avons calculé les valeurs de TP et TA pour les cinq classes d'attaques $U2R$, $R2L$, $Probing$, $Normal$ et DOS. A travers les courbes des taux TA et TP, on peut conclure que notre algorithme permet une discrimination acceptable entre attaques présentes dans la matrice analysée et attaques absentes de cette matrice comme le montre la Figure 3.

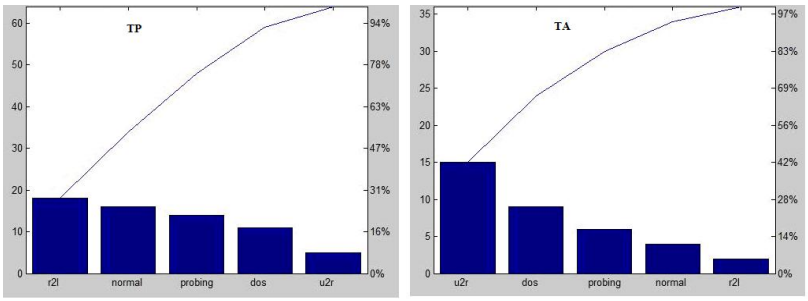


Figure 3. TP et TA pour les cinq classes

4 Conclusion

Dans ce papier, nous avons proposé et implémenté notre approche de recherche locale utilisant les concepts de la logique floue pour la résolution du problème de la détection d'intrusion. Les résultats obtenus ont montré l'efficacité de cette classification dans le domaine de la détection d'intrusion. Nous souhaitons étudier dans un travail futur la puissance des approches évolutives floues pour la détection d'intrusion. Il serait intéressant d'appliquer notre approche sur d'autres types d'attaques qui n'existent pas dans la batterie DARPA.

Références

1. M. Saniee Abadeha, J. Habibia, C. Lucasb, "Intrusion detection using a fuzzy genetics-based learning algorithm", *Journal of Network and Computer Applications*, 30 (2007) 414Ũ428.
2. N. B. Amor, S. Benferhat, and Z. Elouedi. "Naive Bayes vs Decision Trees in Intrusion Detection Systems". *In Proceedings of the ACM symposium on Applied computing*, pages 420Ũ424. ACM Press, 2004.

3. J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. "An architecture for intrusion detection using autonomous agents". *Technical Report 98/05*, COAST Laboratory - Purdue University, June 1998.
4. Dalila Boughaci, Habiba Drias, Ahmed Bendib, Youcef Bouznit, and Belaid Benhamou. "Distributed Intrusion Detection Framework Based on Mobile Agents". *In Proceedings of the International Conference on Dependability of Computer Systems*, pages 248Ũ255. IEEE Press, 2006.
5. H. Debar, M. Dacier, and A. Wespi. "Towards a taxonomy of intrusion-detection systems". *internal RZ 3030*, IBM Zurich Research Laboratory, Saumerstrasse 4, CH-8803 Ruschlikon, Switzerland, June 1998.
6. H. Debar, M. Becker, and D. Siboni. "A neural network component for an intrusion detection system". *In Proceedings of the IEEE Symposium of Research in Computer Security and Privacy*, pages 240-250, May 1992.
7. Kapil Kumar Gupta, Baikunth Nath, Kotagiri Ramamohanarao. "Layered Approach Using Conditional Random Fields for Intrusion Detection". *IEEE Trans. Dependable Sec. Comput.* 7(1) : 35-49 (2010)
8. Ishibuchi H, Murata T. "Techniques and applications of genetic algorithms-based methods for designing compact fuzzy classification systems". *Fuzzy theory systems techniques and applications*, V.3, section 40. New York : Academic Press ; 1999. p. 1081Ũ109.
9. H.S. Javitz, A. Valdes, T.F. Lunt, A. Tamaru, M. Tyson, and J. Lowrance. "Next generation intrusion detection expert system (NIDES)". *Technical Report A016-Rationales*, SRI, 1993.
10. S. Kumar and E.H. Spafford. "A pattern-matching model for misuse intrusion detection". *In Proceedings of the national computer security conference*, pages 11-21, 1994.
11. Ludovic Mé. Gassata, "A genetic algorithm as an alternative tool for security audit trails analysis". *In First international workshop on the Recent Advances in Intrusion Detection.* <http ://www.zurich.ibm.com/d̃ac/Prog_RAID98/Table_of_content.html. 1998.
12. W. Lee, S. Stolfo, and K. Mok. "Mining Audit Data to build Intrusion Detection Models". *In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 66Ũ72. AAAI Press, 1998.
13. T.F. Lunt and R. Jagannathan. "A prototype real-time intrusion-detection expert system". *In Proceedings of the IEEE Symposium on Security and Privacy*, pages 59-66, 1988.
14. L.Mé et V.Alanou. "Détection d'intrusions dans un système informatique : méthodes et outils". *TSI*, 15(4) :429-450, 1996.
15. H. Shah, J. Undercoffer, and A. Joshi. "Fuzzy Clustering for Intrusion Detection". *In Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, pages 1274Ũ1278. IEEE Press, Vol (2), 2003.
16. H.S. Vaccaro and G.E. Liepins. "Detection of anomalous computer session activity". *In Proceedings of the IEEE Symposium on Security and Privacy*, May 1989.
17. Zadeh, L. "PrefaceT̃, in R. J. Marks II (ed.), "Fuzzy logic technology and applications", *IEEE Publications*, 1994.

Service de tolérance aux fautes dans les réseaux Ad hoc à base de système multi agents

Esma Insaf Djebbar[1], Ghalem Belalem[2]

Département d'informatique, Faculté des sciences,
Université d'Oran
[1]d.insaf@yahoo.fr
[2]ghalem1dz@gmail.com

Résumé. Les réseaux ad hoc sont des réseaux distribués, auto-organisés ne nécessitant pas d'infrastructures. Dans un tel réseau, les infrastructures mobiles sont sujet à des déconnexions. Cette situation peut concerner une déconnexion volontaire ou involontaire des nœuds dus à la forte mobilité dans le réseau ad hoc. C'est dans ces problématiques que nous essayons à travers ce travail de contribuer à la résolution de ces problèmes dans un but d'assurer un service continu par la proposition d'un service pour la tolérance aux fautes à base des système multi agents qui permet de prédire un problème et la prise de décision par rapport aux nœuds critiques. Notre service se base essentiellement sur une réplication préventive et transparente des données et permet de répartir efficacement une information dans le réseau en sélectionnant certains objets du réseau pour être des réplicats de l'information.

Mots-clés: réseau Ad hoc, tolérance aux fautes, déconnexion, réplication, système multi agents.

1 Introduction

Les réseaux mobiles ad hoc sont définis comme une collection relativement dense d'entités mobiles interconnectées par une liaison sans fil, sans aucune administration ou support fixe. Une des spécificités fondamentales de ces réseaux c'est qu'ils doivent assurer automatiquement leur propre organisation interne sachant qu'aucune administration du réseau n'est fournie.

La topologie du réseau dans les réseaux ad hoc peut changer à tout moment à cause de la forte mobilité des nœuds, ce qui fait que la déconnexion des unités soit très fréquente. Les déconnexions sont volontaires ou involontaires. Les premières, décidées par l'utilisateur depuis son terminal mobile. Les secondes sont le résultat de coupures des connexions physiques du réseau, l'épuisement de la batterie, le partitionnement du réseau, ou la défaillance des nœuds.

La tolérance aux fautes est un sujet de recherche important pour les systèmes répartis. Ces systèmes disposent de deux propriétés importantes : la sûreté (un mauvais comportement n'apparaît jamais) et la vivacité (un bon comportement apparaît ultimement). Dans les réseaux ad hoc, Certaines défaillances ou déconnexions créent des situations de partitionnement dans les quelles le réseau est

divisé en partitions et la communication est réalisable seulement entre entité de la même partition.

L'objectif de ce travail est donc de construire un service de tolérance aux fautes dans les réseaux ad hoc qui intègre les fonctionnalités nécessaires pour une meilleure disponibilité des données. Notre contribution tient compte des caractéristiques des terminaux mobiles dans le but de réduire au maximum la perte d'information, et de minimiser la consommation de leurs ressources critiques qui est l'énergie.

Notre service est basé essentiellement sur la notion d'une réplication préventive et transparente des données avant l'occurrence d'une faute en intégrant les systèmes multi agents, cette intégration d'agents dans le réseau permettra de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents. Ce service est composé principalement de quatre sous services, à savoir : la gestion de groupe, la décision, la prédiction et réplication, et la gestion de cohérence.

La suite du papier est organisée comme suit. Dans les sections 2 et 3, nous décrivons les réseaux Ad hoc et leurs différentes déconnexions. Section 4, nous spécifions les problèmes additionnels résultant des contraintes imposées par l'environnement de réseau Ad-hoc, nous entamons par la suite dans la section 5 notre proposition du service de tolérance aux fautes en détaillant ces différents sous-services. Ensuite, nous présentons nos différentes expérimentations réalisées par notre simulateur conçu sous java. Enfin, nous concluons et donnons les perspectives de notre travail.

2 Réseaux Ad hoc

Un réseau Ad hoc peut être modélisé par un graphe $G_t=(V_t, E_t)$ où V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble les connections qui existent entre ces nœuds (voir la Figure 1). Si e = (u, v) $\in$ E_t, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t.

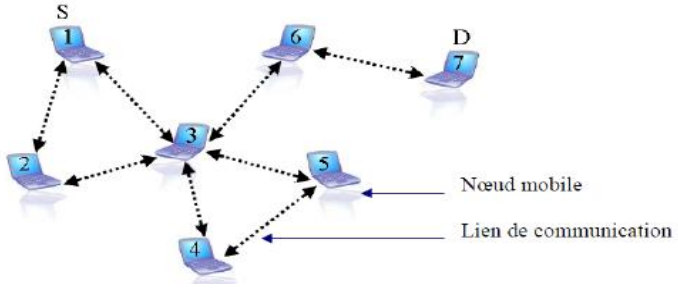


Fig. 1. La modélisation d'un réseau Ad hoc

La topologie du réseau peut changer à tout moment (voir la Figure 2), elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente.

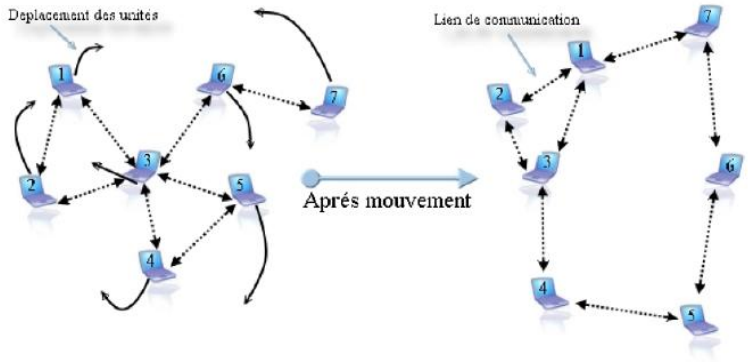


Fig. 2. Le changement de la topologie dans les réseaux Ad hoc

3 Déconnexion dans les réseaux Ad hoc

La topologie du réseau, dans les réseaux Ad hoc, peut changer à tout moment à cause de la forte mobilité des nœuds dans ces réseaux, ce qui fait que la déconnexion des unités soit très fréquente. Les déconnexions sont volontaires ou involontaires : les premières, décidées par l'utilisateur depuis son terminal mobile. Les secondes sont le résultat de coupures des connexions physiques du réseau, l'épuisement de la batterie, le partitionnement du réseau, ou la panne des nœuds, (voir Figure 3).

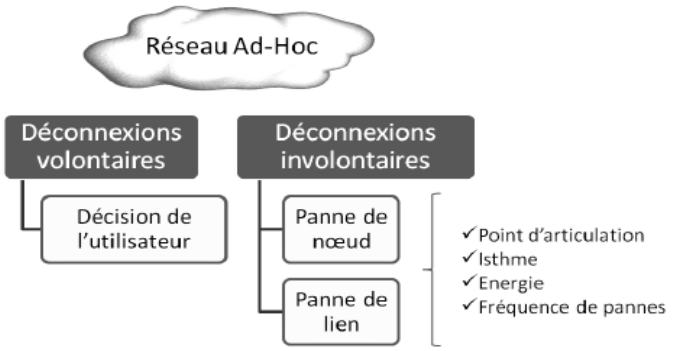


Fig. 3. Déconnexion dans les réseaux Ad hoc

Dans ce contexte, l'objectif de notre travail consiste à proposer un service de tolérance aux fautes à base des systèmes multi agents pour une meilleure disponibilité des données.

4 Tolérance aux fautes

4.1 Réplication pour la tolérance aux fautes

La réplication est une technique fondamentale utilisée dans les systèmes distribués. Elle consiste à stocker la même donnée ou service dans plusieurs nœuds. Les données ou les services sont souvent répliquées pour améliorer la disponibilité, la fiabilité, la tolérance aux fautes, et performance. La réplication peut également améliorer la disponibilité de données et de service quand le serveur tombe en panne. La mobilité ou la défaillance d'un nœud peut mener au partitionnement du réseau, où le réseau est fractionné en partitions disjointes, provoquant la possibilité de l'incohérence de données.

La réplication de données a été intensivement étudiée dans les systèmes distribués, particulièrement dans les réseaux filaires. Dans de tels systèmes, les nœuds qui tiennent la base de données sont plus fiables et moins pour déconnecter ou échouer. Il y a plus d'une réplique afin d'améliorer la latence de requête et le coût de mise à jour. Ils ne considèrent pas la disponibilité de données comme grand souci, puisque l'échec des liens et du nœud est peu fréquent et un nombre restreint de serveurs de reproduction peuvent fournir la disponibilité de données élevée. Les réseaux Ad-Hoc où les nœuds se déplacent librement et leurs batteries diminuent rapidement, qui causent des ruptures de lien et des échecs fréquents de nœuds. L'échec de quelques liens et nœuds considérés en tant que critique peut fractionner le réseau en plusieurs partitions. Cette situation réduit considérablement la disponibilité de données et provoque la non cohérence des données. En répliquant des données aux nœuds multiples, la disponibilité de données peut être améliorée. De plus, la réplication de données peut également réduire le retard de requête, puisque les nœuds mobiles obtiennent les données de quelques répliques voisines. En plus des problèmes de disponibilité et de performance qui ont été bien discutées dans les réseaux fixes, la réplication de données dans les réseaux Ad-hoc doit aborder les problèmes additionnels résultant des contraintes imposées par l'environnement de réseau Ad-hoc. Ces problèmes sont les suivants :

- **Problème de partitionnement du réseau :** En raison du partitionnement dans les réseaux Ad-Hoc, les chances d'accéder à une donnée deviennent de plus en plus faible puisque les utilisateurs mobiles peuvent ne pas être dans la même partition que le nœud qui détient la donnée. La réplication des données dans les partitions séparées avant l'occurrence du partitionnement de réseau peut améliorer la disponibilité des données. Pour faire ainsi, le protocole de réplique devrait déterminer le temps où le partitionnement de réseau pourrait se produire et répliquer des données à l'avance.

- **Problème de consommation d'énergie :** Les nœuds mobiles opèrent avec des batteries de basse puissance. Un serveur simple peut servir beaucoup de clients, ce qui cause l'épuisement rapide de sa batterie. Pour améliorer la disponibilité de données, le protocole de réplication devrait répliquer les données critiques sur les nœuds qui peuvent durer pendant une longue période. D'ailleurs, il devrait également répliquer des données de telle manière que la puissance d'énergie des serveurs soit réduite et soit équilibrée parmi tous les serveurs dans le réseau.

- **Problème d'évolutivité :** A mesure que la taille de réseau augmente, une requête envoyée par un nœud client peut traverser un long chemin pour atteindre le nœud serveur, augmentant le coût et la latence de requête. D'ailleurs, l'existence d'un grand nombre de requête clients provoque plus de controverse d'accès de canal, qui diminue considérablement la largeur de bande disponible et augmente le retard d'accès de canal. Le protocole de réplication devrait être conçu de sorte que son exécution ne soit pas considérablement affectée si le nombre de nœuds ou la taille de réseau augmente.

4.2 Travaux connexes

Hauspie et al. [3, 4] ont proposé une nouvelle métrique pour détecter le partitionnement du réseau sans employer de GPS. La métrique est basée sur la recherche d'un ensemble de chemins disjoins entre un nœud client et un nœud serveur. Un ensemble de chemins disjoints est un ensemble de chemins qui n'ont aucun nœud commun excepté le nœud client et le nœud serveur. La décision pour répliquer un service ou une donnée est prise quand le raccordement entre un client et un serveur s'aggrave en termes de sociabilité, largeur de bande, retard, etc. Répliquant le service ou la donnée sur un nœud qui est plus près du nœud client peut augmenter la qualité du raccordement entre le client et les nœuds serveurs.

Jorgic et al. [5] ont proposé des algorithmes localisés pour détecter les nœuds et les liens critiques qui pourraient diviser le réseau. Un nœud *u* est dit *k*-critique si le sous graphe de ses voisins à *k* sauts, duquel on exclut *u* ainsi que les liens qui y mènent est non connexe. D'une manière similaire la détection de liens critiques, un lien *uv* est *k critique* si les ensembles des voisins à *k* sauts de *u* et de *v* (construits en supposant que le lien *uv* n'existe pas) sont disjoints. Si un lien est critique de façon globale, il sera *k*-critique quelque soit $k > 1$.

Les auteurs [6] ont proposé un schéma de réplication, appelé la réplication d'anneau en expansion (Expanding Ring Replication). Le serveur de données mesure la fréquence des demandes de chaque donnée. Si elle dépasse une valeur seuil, il réplique la donnée sur un ou plusieurs nœuds capables de son voisinage. La fonction de capacité considère des paramètres tels que l'espace mémoire disponible, la puissance de batterie restante, et la capacité de traitement.

5 Service de tolérance aux fautes

Le service proposé pour la tolérance aux fautes dans les réseaux Ad hoc est composé de quatre sous services, qui sont présentés dans la Figure 4. Le service de tolérance aux fautes est construit en une architecture à base de clusters qui sont identifiés par un nœud particulier appelé leader. Chaque leader est associé à un agent leader qui est responsable au lancement de trois agents : enregistrement, réplication et cohérence dans son groupe, les trois agents travaillent ensemble dans un esprit de coopération. Le service se base essentiellement sur la notion de réplication après prédiction de déconnexions ou de panne d'un objet dit critique dans le réseau.

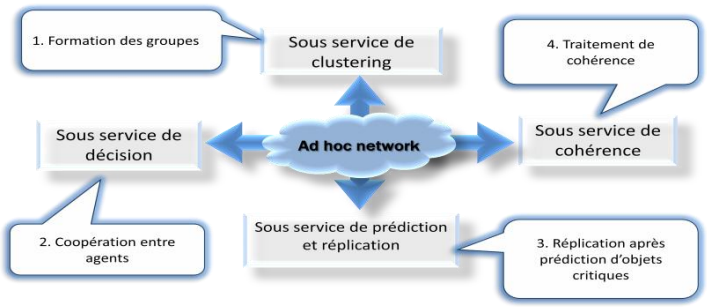


Fig. 4. Architecture du service de tolérance aux fautes.

5.1 Sous service de Clustering

Notre service de tolérance aux fautes est construit autour d'une architecture à base de clusters, car elle permet une meilleure gestion du réseau en réduisant le nombre des échanges entre les nœuds et le passage à l'échelle. Pour former les clusters, nous utilisons un algorithme entièrement distribué à base de nombre de voisins et le niveau d'énergie. L'algorithme de clusterisation illustre le pseudo code pour l'élection du leader (voir Algorithme 1).

Algorithme 1 : Clustering

```
1: DiffusionMessageElection(IDNœud, NbreVoisins, NiveauEnergie) /*
   Diffuser l'identifiant, le nombre de voisins et le niveau d'énergie au
   voisin à 1 saut */
2: ReceptionMessageElection(IDNœud, NbreVoisins, NiveauEenergie)
3: TrieListe(IDNœud, NbreVoisins, NiveauEnergie)
4: N ← PremierElementListe()
5: for all nœuds de TrieListe do
6:   if (N.NbreVoisins = Nœud.Nbre-voisins) then
7:     if (N.NiveauEnergie >= Nœud.Niveau-energie) then
8:       Nœud.QuiEstLeader ← N.IDNœud
9:       Nœud.EstLeader ← True
10:      EnvoieMessageLeader() /*Le leader informe ses voisions qu'il
         est leader*/
11:      ReceptionMessageLeader(IDNœud, NiveauEnergie, Réplique,
         VersionRéplique)
12:    else
13:      Nœud.EstLeader ← False
14:    end if
15:  end if
16: end for
```

5.2 Sous service de décision

Ce sous service permet de prendre certaines décisions en faisant coopérer un ensemble d'agents dotés d'un comportement intelligent. Cette coopération d'agents consiste à coordonner leurs buts et leurs plans d'action pour résoudre un problème.
Dans notre travail et après la phase de formation de clusters, nous avons proposé d'associer aux principaux services un ensemble d'agents qui coopèrent entre eux.

L'agent Leader est l'agent responsable à la gestion du groupe qui lance trois agents : enregistrement, réplication et cohérence. L'agent enregistrement chargé d'enregistrer les nœuds formant le groupe et la prise en charge des nouveaux nœuds et ceux qui quittent le groupe. L'agent réplication qui permet d'assurer le contrôle de nombre de répliques dans le groupe suite à la détection d'un objet dit critique dans son groupe. Pour l'agent cohérence qui est responsable au lancement de la propagation des mises à jour vers les nœuds. Dans le but de gérer notre réseau qui est constitué de plusieurs clusters, nous avons proposé d'intégrer un agent générique qui coordonne entre les agents leaders. Les principales motivations pour l'utilisation de l'agent générique sont :

- Eviter les conflits entre les agents leaders
- Réduire le processus de négociation entre les agents leaders
- Posséder une vue globale du réseau
- Réduire le nombre de messages

L'agent générique responsable de gérer tous les groupes, c'est un super Agent, il permet la récolte des informations des agents leaders, si un agent leader tombe en panne, alors l'agent générique peut le remplacer pour accomplir ses tâches. La Figure

5 montre le diagramme de classe UML des différentes entités constituant notre système.

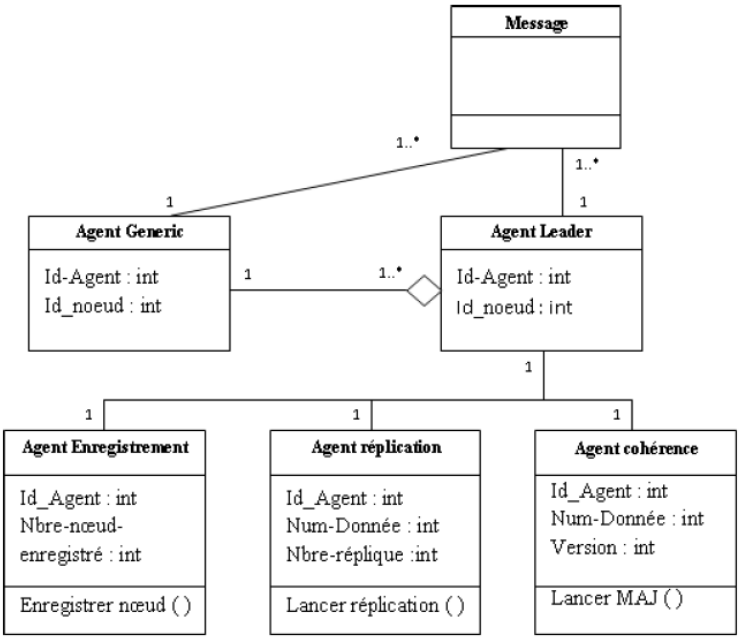


Fig. 5. Diagramme de classe du service de décision.

5.3 Sous service de prédiction et réplication

La fonction principale de ce sous-service est de prédire une éventuelle panne ou déconnection du réseau en établissant une liste d'objets critiques (sensibles). Chaque leader peut connaître l'état actuel de tous les nœuds de son cluster. Pour évaluer la criticité, le sous-service permet de détecter plusieurs types d'objets critiques : énergie, point d'articulation, isthme, fréquence de pannes.

5.4 Sous service de cohérence

Ce sous service s'occupe de la gestion de cohérence des répliques dans les réseaux ad hoc. La cohérence représente la capacité pour un système à refléter sur la copie d'une donnée les modifications intervenues sur d'autres copies de cette même donnée. Dans notre travail, nous nous intéressons à la cohérence forte entre les répliques. Pour la gestion de cohérence, nous avons proposé un protocole de cohérence des Quorums imbriqués. Chaque réplique est caractérisée par son numéro, sa taille et sa version et sa localisation. Une requête est lancée par un client et exécutée par le leader du groupe après la construction du quorum de lecture ou d'écriture. La formation du

quorum de lecture et d'écriture est détaillée dans l'algorithme de cohérence présenté dans l'Algorithme 2.

Algorithme 2 : Cohérence

```
1: EnvoieRequteLeader(NumReq, NumDonne, TypeReq)
2: if Requête=Lecture then
3:    Q_inter ← (NbreRéplique/2) /*Former un quorum de lecture dans
      le cluster*/
4:    Lancer la requête de lecture ( )
5: else
6:    for all Cluster do
7:       Q_inter ← (NbreRéplique/2 + 1) /*Former un quorum d'écriture
         pour chaque cluster*/
8:    end for
9:    Q_intra ← (NbreRépliqueClusters/2 + 1) /*Former un quorum
      d'écriture global*/
10:   Lancer la requête d'écriture ( )
11: end if
```

6 Simulation et résultats

Dans cette section, nous étudions la performance de notre service propose en utilisant Sim TF-SMA. Sim TF-SMA est un simulateur que nous avons développé sous Java.

6.1 Nombre de requêtes acceptées et perdues

Dans cette première simulation, nous avons mesuré le nombre de requêtes acceptées et le nombre de requêtes perdues. Cette simulation a été réalisée avec les paramètres de simulations suivants : 50 nœuds, 20 données, taille de la donnée 100 Mo, nombre de requêtes 100, la largeur de la bande passante 11Mb/s, surface de la simulation 700 m x 700 m, portée de 130 m, temps de simulation 60 secondes, temps de pause 0.5 seconde, vitesse de mobilité 5 m/s. Les résultats de cette simulation sont montrés dans les Figures 6 et 7.

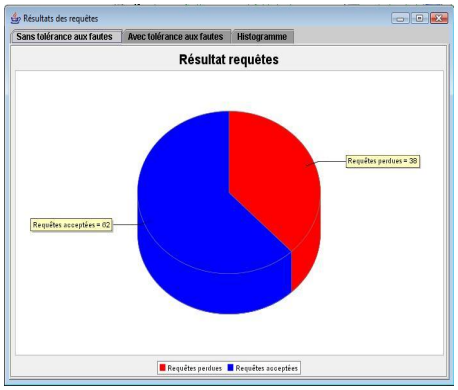


Fig. 6. Résultats des requêtes sans tolérance

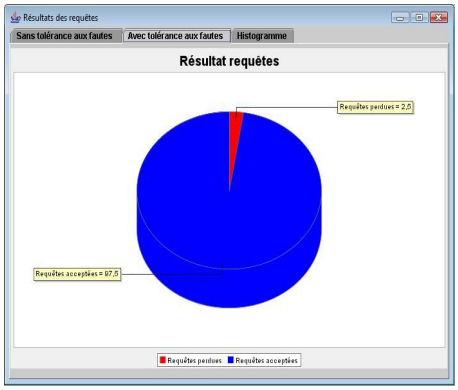


Fig. 7. Résultats des requêtes avec tolérance

Nous remarquons une diminution significative du nombre de requêtes perdues en appliquant notre service. Et cela est dû à la réplication préventive que nous avons appliquée pour ne pas perdre les données. Nous pouvons remarquer que avec notre approche a pu accepter 97 requêtes sur les 100 requêtes envoyées ce qui est équivalent à 97%, à l'inverse l'approche sans tolérance a accepté que 62 requêtes sur les 100 envoyées ce qui est équivalent à 62%.

6.2 Temps de réponse moyen des requêtes

Pour mesurer le temps de réponse moyen, nous avons simulé le réseau avec les paramètres suivants : 300 nœuds, 20 données, taille de la donnée
100 Mo, le nombre de requêtes 50, la largeur de la bande passante 11Mb/s, surface de la simulation 700 m x 700 m, portée de 200 m, temps de simulation 60 secondes, temps de pause 0.5 seconde, vitesse de mobilité 5 m/s. Les résultats de la simulation sont représentés dans la Figure 8.

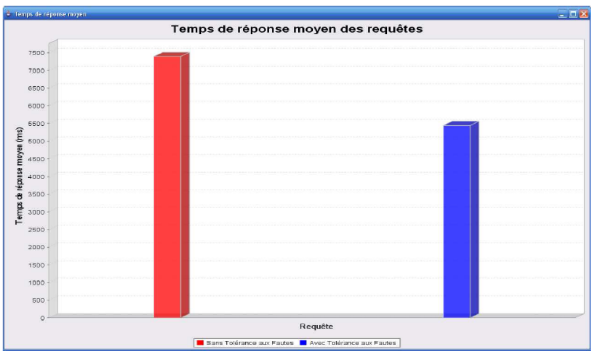


Fig. 8. Temps de réponse moyen des requêtes.

Nous remarquons d'après la Figure 8 que le temps de réponse moyen en appliquant le service de tolérance aux fautes est réduit par rapport à une approche sans tolérance aux fautes. Cela s'explique par le faite que la réplication réduit ce temps de réponse.

6.3 Consommation d'énergie

Un réseau est instable quand le temps de pause est faible (moins d'une seconde). Après plusieurs séries de simulations, nous avons aperçu que plus le réseau est stable plus notre service économisera l'énergie. Pour mesurer l'économie d'énergie, nous avons fait une simulation avec les paramètres suivants : 300 nœuds, portée 200 m, surface 700 m x 700 m, temps de simulation 20 s, vitesse de mobilité 5 m/s, nombre de données 20, nombre de requêtes 50.

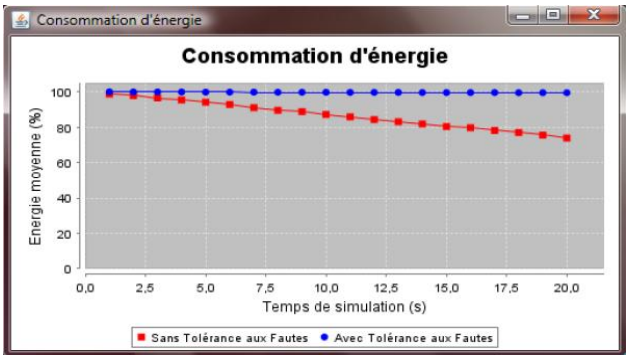


Fig. 9. Consommation d'énergie.

Nous remarquons à partir de la Figure 9 la consommation d'énergie des nœuds sans tolérance aux fautes est plus grande par rapport à une approche avec tolérance aux fautes, cela s'explique par l'utilisation du clustering qui minimise l'envoie des messages.

7 Conclusion et perspectives

Dans ce papier, nous avons présenté notre service de tolérance aux fautes dans les réseaux mobiles ad hoc. Ce service se base essentiellement sur la réplication par prédiction dans le but d'assurer une continuité de service de partage de données dans le réseau.

Le service de tolérance aux fautes proposé vise essentiellement à une réplication préventive et transparente de données avant l'occurrence de fautes en utilisant les systèmes multi agents, l'utilisation d'agents dans le réseau permettra de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-

ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents.

Le service de tolérances aux fautes appliqué se compose de quatre sous services à savoir (clustering, décision, réplication après prédiction, et la gestion de cohérence) permettant de mieux gérer le réseau et regroupant des fonctionnalités nécessaires pour une meilleure disponibilité des données. Notre contribution vise à réduire au maximum la perte d'information en tenant compte des caractéristiques des terminaux mobiles dans le but de minimiser la consommation de leurs ressources critiques qui est l'énergie.

Pour démontrer l'efficacité de notre service de tolérance aux fautes, plusieurs simulations ont été effectuées en faisant varier plusieurs critères d'évaluation. Les premiers résultats prouvent que le nombre de requêtes perdues dans le réseau diminuent de manière considérable puisque les données sont répliquées avant l'occurrence d'une faute dans un nœud (énergie faible, partition dans le réseau ou un nœud signalant plusieurs arrêts consécutifs). Le service appliqué a également l'avantage d'augmenter la durée de vie des nœuds dans le réseau puisque le nœud sera mis en pause avant qu'il atteint un arrêt définitif. D'autre part, le service a montré son efficacité dans la réduction du temps de réponse des requêtes puisque les données seront trouvées sur des nœuds plus proches et par conséquent la consommation d'énergie.

Dans la continuité et l'extension de notre travail, nous envisageons d'implémenter notre service de tolérance aux fautes dans un simulateur existant tel que NS2 ou GloMoSim, améliorer l'algorithme utilisé pour le sous service de gestion de groupes en prenant en compte de l'intention des nœuds, élargir le sous service de cohérence à base des modèles probabilistes, considérer d'autres types de données, par exemple les fragments de base de données mobiles et la prise en compte des différents protocoles de routage pour les réseaux Ad hoc.

Références

1. K. Bhargava, A. Helal and A. Heddaya. Replication Techniques in Distributed Systems. Kluwer Academic Publishers, 1996.
2. F. Coulouris and J. Dollimore. Distributed systems: concepts and design. Addison-Wesley Longman Publishing, 3rd edition, 1988.
3. D. Simplot, M, Hausphie and J.Carle. Replication decision algorithm based on link evaluation for services in manet. Technical report, Technical Report 2002-05, IRCICA/LIFL, University of Lille 1, 2002.
4. D. Simplot, M. Hausphie and J. Carle. Partition detection in mobile ad hoc networks. In A. Belghith, S. Tabbane, N. Ben Ali, and A. Gazdar, editors, Proc. 2nd IFIP Mediterranean Ad hoc Networking Workshop (MED-HOC-NET 2003), Mahdia, Tunisia, 2003.
5. M. Hauspie, M. Jorgic, I. Stojmenovic and D. Simplot-Ryl. Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks. In Proceedings of the 3rd Annual Mediterranean Ad hoc Networking Workshop Med-Hoc-Net, pages 360-371, Bodrum, Turkey, June 2004.
6. C. Almeroth, V. Thanedar and M. Belding-Royer. A lightweight content replication scheme for mobile ad hoc environments. In NETWORKING, pages 125-136, 2004.

7. N. Mitton, E.Fleury. Distributed Node Location in clustered multi-hop wirelessnetworks. PhD thesis, INSA de Lyon, Marsh 2006.
8. H. Yu, P. Martin, and H. Hassanein. Cluster-based replication for large-scale mobile ad-hoc. International conference: Wireless Networks, Communications and Mobile Computing, voulme 1, pages 552-557, 2005.
9. G. Belalem and Y. Slimani. A hybrid approach for consistency management in large scale systems. Networking and Services, International conference, 2006.
10. G. Belalem and Y. Slimani. A hybrid approach to replica management in data grids. IJWGS, 3(1): 2-18, 2007.
11. M. Hauspie and I. Simplot-Ryl. Enhancing nodes cooperation in ad hoc networks. In Proceedings of the 4th IEEE/IFIP Annual Conference on Wireless On demand Network Systems and Services WONS 2007.
12. A. Derhab. SQUIRREL: Self-Organizing Qos-routing for IntRa-flow Contention in Ad-Hoc Wireless Networks. IWSOS, pages 269-274, 2008.
13. N. Mitton, E. Fleury, I. Guérin Lassous, S. Tixeuil: Self-Stabilization in Self-Organized Multihop Wireless Networks. ICDCS Workshops, pages 909-915, 2005.
14. M. Hamdy, B. Konig-Ries, A service distribution protocol for mobile ad hoc networks. ICPS '08: Proceedings of the 5th international conference on Pervasive services, Sorrento, Italy (2008), pp.141-146.
15. P. Michiardi, R. Molva, Mobile Ad Hoc Networking. Wiley-IEEE Press, 2004, ch. Ad Hoc Network Security (2004), pp. 329–354.
16. C. Perkins, Ad Hoc Networking. Addison–Wesley, 2001.
17. B. Randell, A. Avizienis, J–C. Laprie, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1(1): 11–33, 2004.

Systèmes immunitaires artificiels pour la détection d'intrusions

Meriem ZEKRI[*], Labiba SOUICI-MESLATI[**]
[*] Laboratoire LABGED, Université Badji Mokhtar, BP 12, 23000, Annaba, Algérie.
zekri_meriem@yahoo.fr
[**] Laboratoire LRI, Université Badji Mokhtar, BP 12, 23000, Annaba, Algérie.
souici_labiba@yahoo.fr

Résumé. Le défi à relever, dans le domaine de la détection d'intrusion, est de pouvoir déterminer la différence entre un comportement normal et un comportement anormal d'un système. Ce défi n'est pas évident à relever à cause de la croissance et de la complexité des systèmes informatiques ainsi que celle des réseaux à protéger. La robustesse et l'efficacité des systèmes immunitaires naturels en ont fait une source d'inspiration idéale pour de nombreux chercheurs afin de combattre le fléau des intrusions sur les réseaux informatiques. En effet, l'immunologie artificielle tente d'utiliser des caractéristiques intéressantes des systèmes immunitaires naturels afin de réaliser des systèmes adaptatifs pour la résolution des problèmes complexes et évolutifs. Dans cet article, nous proposons deux systèmes immunitaires artificiels pour la détection d'intrusion. Le premier est basé sur la théorie du danger, qui comporte l'algorithme des cellules dendritiques et qui représente l'un des modèles les plus récents en immunologie artificielle. Le second est basé sur le modèle de la sélection négative, qui est l'un des premiers modèles immunitaires proposés pour la détection d'intrusion. Nous utilisons la base de données *KDD Cup'99* qui reste la plus appropriée pour nos expérimentations, au vu de son utilisation intensive par les chercheurs dans le domaine de la détection d'intrusion. Les résultats obtenus sont très intéressants.

Mots clés: Systèmes immunitaires artificiels, Détection d'intrusion, Détection d'anomalies, Théorie du danger, Algorithme des cellules dendritiques, Algorithme de la sélection négative.

1 Introduction

La détection d'intrusion est la détection de toute activité anormale dans un système informatique ou sur un réseau. Les anomalies représentent l'une des catégories les plus populaires d'intrusions, leur détection implique une discrimination, entre les données normales et anormales, qui est fondée sur la connaissance des données normales. La détection d'anomalie a un net avantage comparé à une approche plus traditionnelle comme la détection basée signature (détection des malveillances), c'est celui de détecter les nouvelles intrusions. Plusieurs systèmes ont été conçus pour

résoudre le problème de détection d'intrusion mais beaucoup d'entre eux peuvent être sujets à la génération de fausses alarmes.
Durant ces dernières années, un paradigme bio-inspiré assez récent a commencé à faire ses preuves dans plusieurs domaines, tels que la classification, la reconnaissance de formes et la fouille de données, c'est celui des systèmes immunitaires artificiels (AIS, Artificial Immune Systems) inspirés des systèmes immunitaires naturels (IS, Immune Systems) [1]. Son efficacité a encouragé les chercheurs à l'étudier et à s'inspirer de ses mécanismes pour mettre en œuvre des systèmes artificiels capables de détecter efficacement les intrusions [2].
Il existe plusieurs modèles inspirés des modèles théoriques du système immunitaire [1]. Nous nous intéressons particulièrement à celui de la théorie du danger, qui a connu un début tumultueux à causes de doutes multiples autour de son concept. Cependant, il y a quelques années, un groupe de chercheurs anglais l'a beaucoup étudié, ils ont même intitulé leur projet « The Danger project » [3]. La théorie du danger comporte essentiellement deux algorithmes qui sont l'algorithme des cellules dendritiques (DCA, Dendritic Cell Algorithm) et l'algorithme Tolk-like Receptor (TLR). L'algorithme DCA a été développé pour détecter des anomalies, en conséquence, il semble le plus approprié pour notre travail, en plus du fait que c'est un algorithme de la théorie du danger pour laquelle nous avons porté un grand intérêt depuis le début de notre étude des systèmes immunitaires artificiels car cette théorie représente un concept relativement nouveau en immunologie naturelle [1, 4]. En effet, tandis que la plupart des modèles immunitaires se basent sur la discrimination soi/non soi (self/non self) où tout corps étrangers est détecté puis éliminé, la théorie du danger, quant à elle, se base sur la détection du danger et non la détection de l'étrangeté. Des recherches récentes sur l'algorithme DCA [5, 6, 7] montrent qu'il présente non seulement des performances prometteuses sur le taux de détection, mais il peut aussi aider à réduire le nombre de fausses alarmes, comparativement à des systèmes similaires.
L'objectif de notre travail est de concevoir deux systèmes de classification des intrusions, le premier est basé sur l'algorithme DCA alors que le second est basé sur l'algorithme NSA (Negative Selection Algorithm), issu du modèle de la sélection négative, qui est l'un des premiers modèles immunitaires proposés pour la détection d'intrusion. Nous comparons les performances de ces deux approches immunitaires afin de déterminer laquelle est la plus appropriée pour le problème considéré, en utilisant l'ensemble de données *KDD cup'99*.
Notre article sera présenté comme suit. Dans la deuxième section nous présentons les systèmes immunitaires artificiels, suivis par les systèmes de détection d'intrusion dans la troisième section. La quatrième section est consacrée à l'utilisation des systèmes immunitaires artificiels pour la détection d'intrusion. Dans la cinquième section, nous présentons la description du système proposé qui sera suivie par une discussion relative aux expérimentations et aux résultats obtenus. Nous terminons cet article par une conclusion et des perspectives d'extensions futures.

2 Les systèmes immunitaires artificiels

Les systèmes immunitaires artificiels représentent une catégorie d'algorithmes inspirés par les principes et le fonctionnement du système immunitaire naturel. Ces

algorithmes exploitent typiquement les caractéristiques du système immunitaire pour ce qui est de l'apprentissage et de la mémorisation comme moyens de résolution de problèmes complexes [8]. Le domaine de l'immunologie artificielle a évolué de façon constante depuis 1985, avec un intérêt croissant vers le développement des modèles de calcul inspirés par plusieurs principes immunologiques. Certains modèles imitent les mécanismes abstraits du système immunitaire biologique pour mieux comprendre ses processus naturels et simuler son comportement dynamique en présence d'antigènes ou d'agents pathogènes, d'autres mettent l'accent sur la conception d'algorithmes de calcul, en utilisant des techniques de simplification (parfois obsolètes) de divers processus immunologiques [1].
Il y a différents type de modèles de systèmes immunitaires artificiels, ceux qui sont inspirés des cellules T comme la discrimination self/ non-self et l'algorithme de la sélection négative, il y a ceux qui sont inspirés des cellules B, tels que l'algorithme de la sélection clonale et les modèles des réseaux immunitaires, et il y a également des modèles récents tels que la théorie du danger qui n'est devenue populaire qu'au cours de la dernière décennie. Le tableau ci-dessous résume les différents concepts immunologiques utilisés pour la résolution de problèmes informatiques.

Tableau 1. Immunité basée sur des modèles de calcul et des concepts immunologiques spécifiques [1]

Concepts immunologiques et entités	Modèles basé immunité	Problèmes informatiques
Self/non Self : reconnaissance des cellules T.	Algorithme de la sélection négative	Erreurs, détection d'anomalie et de changements.
Réseaux idiotypiques, mémoire immunitaire, cellule B.	Théorie des réseaux immunitaires.	Apprentissage supervisé et non supervisé
Expansion clonale, maturation, cellule B	Algorithme de sélection clonale.	Recherche et optimisation
Immunité innée	Théorie du danger	Stratégie de défense

La théorie du danger est une théorie controversée dans le monde de l'immunologie, elle conteste les théories que l'on croyait au cœur de la fonction du système immunitaire humain. Le concept de la discrimination self/non-self effectuée par le système immunitaire a été la pierre angulaire de l'immunologie. Le principe central de l'immunologie, c'est que le système immunitaire répond à la présence d'entités étrangères (appelé non-Self) et ne répond pas à l'hôte (appelé Self). L'étude de la théorie du danger prend en considération deux aspects du modèle du danger. Les immunologistes examinent les signaux d'un danger potentiel et comment sont affectées les cellules du système immunitaire. En collaboration avec des immunologistes, des informaticiens ont cherché comment la constitution du modèle du danger pourrait être utilisée dans l'amélioration des AIS. Ceci est réalisé en vue d'améliorer les anomalies des systèmes de détection pour les ordinateurs sur réseaux. Il y a eu deux algorithmes développés inspirés de la théorie du danger, l'algorithme Tolk-like Receptor (Twycross en 2007) et l'algorithme des cellules dendritiques (Greensmith en 2006) [9].

3 Les systèmes de détection d'intrusion

En sécurité informatique, la détection d'intrusion est l'acte de détecter les actions qui essaient de compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource. La détection d'intrusion peut être effectuée manuellement ou automatiquement. Dans le processus de détection d'intrusion manuelle, un analyste humain procède à l'examen de fichiers logs à la recherche de tout signe suspect pouvant indiquer une intrusion [10]. Un système qui effectue une détection d'intrusion automatisée est appelé système de détection d'intrusion (IDS, Intrusion Detection System). La méthode de détection, le comportement sur la détection, la localisation de la source audit et la fréquence d'utilisation représentent les caractéristiques d'un IDS. La méthode de détection décrit les caractéristiques de l'analyseur, lorsque l'IDS utilise les informations sur le comportement normal du système, on le qualifie de « basé comportement ». Lorsque l'IDS utilise les informations sur les attaques, on le qualifie de « basé connaissances ». Le comportement sur la détection décrit la réponse de l'IDS aux attaques, lorsqu'il réagit aux attaques en prenant soit des actions correctives ou proactives, alors l'IDS est dit actif. Si l'IDS génère simplement des alarmes (incluant la pagination…etc.), il est dit passif. La localisation de la source audit établit une distinction entre les IDS basé sur le type des informations d'entrées qu'ils analysent. Ces informations d'entrées peuvent être les chemins d'audit, les journaux systèmes ou des paquets réseau. La fréquence d'utilisation est un concept orthogonal, certains IDS ont des capacités de monitoring continu en temps réel, tandis que d'autres doivent être exécutés périodiquement. Les trois premières caractéristiques sont regroupées dans la catégorie des caractéristiques fonctionnelles, parce qu'elles portent sur le fonctionnement interne du moteur de détection d'intrusion, à savoir ses informations d'entrée, son mécanisme de raisonnement et son interaction avec le système d'information. La quatrième caractéristique distingue la RTID (Real-Time Intrusion Detection) de scanners utilisés pour l'évaluation de la sécurité [10].

4 Les AIS pour les systèmes de détection d'intrusion

L'utilisation des AIS pour la détection d'intrusion est un concept intéressant pour deux raisons : le système immunitaire fournit une protection d'une manière distribuée contre les intrus en plus d'être adaptatif et les techniques actuelles utilisées dans la sécurité informatique ne sont pas en mesure de faire face à la nature dynamique et de plus en plus complexe des systèmes informatiques et leur sécurité. Afin de fournir un IDS viable, l'AIS doit construire un ensemble de détecteurs qui mesurent avec précision les antigènes correspondants. Dans les AIS actuels dédiés aux IDS, les connexions réseau et les détecteurs sont modélisés comme des chaînes. Les détecteurs sont créés de façon aléatoire, puis subissent une phase de maturation. Si les détecteurs ne correspondent à aucun de cela, ils sont éliminés, autrement ils deviennent matures. Ces détecteurs matures commencent à surveiller les nouvelles connexions au cours de leur vie. Si ces détecteurs matures ne correspondent à rien d'autre, dépassant un certain seuil, ils sont activés. Ceci est ensuite rapporté à un opérateur humain qui décide s'il ya une véritable anomalie. Une telle approche est connue sous le nom de la sélection négative, car seuls les détecteurs (anticorps) qui ne correspondent pas

survivent. Toutefois, cette approche attrayante montre des problèmes de mise à l'échelle quand elle est appliquée au trafic d'un réseau réel. De là, d'autres approches des AIS ont été appliquées à l'IDS, comme la sélection clonale, les réseaux immunitaires et la théorie du danger. C'est la théorie du danger qui a fourni, jusqu'à présent, les résultats les plus prometteurs, en particulier l'algorithme des cellules dendritiques (DCA, Dendritic Cell Algorithm) dont la principale capacité est de pouvoir gérer des données de grandes dimensions [11].

5 Description des systèmes proposés

Les systèmes que nous proposons, pour la détection d'intrusion en général et la détection d'anomalies en particulier, se basent sur deux algorithmes immunitaires artificiels qui sont : l'algorithme des cellules dendritiques et l'algorithme de la sélection négative avec C-detecteurs en utilisant seulement 10% de l'ensemble de données *KDD cup'99* au vu de sa grande dimension. Les 10% représente 494 021 enregistrements. Nous allons présenter, dans ce qui suit, les deux algorithmes ainsi qu'une description de la base de données et du processus de normalisation des données.

5.1 Algorithme des cellules dendritiques

L'algorithme des cellules dendritiques (DCA) est un algorithme de corrélation qui peut effectuer la détection d'anomalies sur des ensembles de données classées, y compris en temps réel. Le processus de fusion du signal est inspiré par l'interaction entre les cellules dendritiques (DCs) et leur environnement. Le DCA a la capacité de combiner des signaux multiples pour évaluer le contexte courant de l'environnement. La corrélation entre le contexte et l'antigène est utilisée comme base de la détection d'anomalie dans cet algorithme [1]. Les antigènes sont nécessaires, ils représentent les données qui doivent être classées, avec la base de la classification qui ne découle pas de la structure de ces antigènes mais des proportions relatives des trois catégories de signaux d'entrées qui sont : PAMP, le signal de danger et le signal de sécurité [12]. La sémantique des trois catégories de signaux d'entrées est présentée comme suit :

- ***PAMP (Pathogenic associated molecular patterns) :*** indique la présence d'une anomalie définitive.
- ***Signal de danger (DS) :*** l'augmentation de la valeur de ce signal augmente la probabilité de présence d'une anomalie.
- ***Signal de sécurité (SS) :*** indique la présence d'une situation normale absolue.

Les signaux de sortie du processus DCA sont associés avec des poids prédéfinis pour produire trois signaux de sortie. Les trois signaux de sortie sont : le signal de co-stimulation[1] (Csm), le signal semi-mature (Semi) et le signal mature (Mat). Les poids prédéfinis utilisés sont présentés dans le tableau 2 et l'équation pour le calcul des signaux de sortie est la suivante :

$$O_j = \sum_{i=0}^{2} (W_{ij} \times S_i) \quad \forall j \qquad (1)$$

[1] Durant l'activation des lymphocytes, la costimulation est souvent cruciale pour le développement d'une réponse immunitaire efficace.

Tableau 2. Les poids recommandés pour l'équation (1).

	PAMP S_0	Signal de danger S_1	Signal de sécurité S_2
Csm O_0	2	1	2
Semi O_1	0	0	2
Mat O_2	2	1	-2

Le DCA introduit les seuils de migration assignés individuellement afin de déterminer la durée de vie d'une DC. Cela peut rendre l'algorithme suffisamment robuste et flexible pour détecter les antigènes trouvés durant certaines périodes. La DC individuelle additionne les signaux de sortie au fil du temps, résultant dans les Csm cumulatifs, les Semi cumulatifs et les Mat cumulatifs. Ce processus continue jusqu'à ce que la cellule atteigne la fin de sa durée de vie, qui survient quand le Csm cumulatif dépasse le seuil de migration, la DC cesse l'échantillonnage des signaux et des antigènes. A ce stade, les deux autres signaux cumulatifs sont évalués. Si le Semi cumulatif est supérieur à la valeur du Mat cumulatif, la cellule se différencie vers l'état semi-mature et est assignée à la valeur de contexte de « 0 » et vice versa [12].

Algorithme 1. Pseudo code de l'algorithme DCA

```
Les entrées : S = signaux d'entrées pré-classés + antigènes
Les sorties : E = antigènes + MCAV.
-Créer une population initiale de cellules dendritiques (DCs), D
Pour chaque entrée faire
|    -Créer un ensemble de 10 DCs sélectionnées aléatoirement de D, P
|  Pour les 10 DCs sélectionnées faire
|  |     - Obtenir l'antigène ;
|  |     - Stocker l'antigène ;
|  |     - Obtenir les signaux ;
|  |     - Calculer les signaux de sortie intérim ;
|  |     - Mettre à jour les signaux de sortie cumulatifs ;
|  |
|  |   Si cumulative CSM > seuil de migration alors
|  |   |  - Enlever la DC de la population ;
|  |   |  - Assigner le cell-context à la DC ;
|  |   |  - Toutes les DCs qui ont collecté l'antigène et qui ont un cell-context sortent
|  |   |       pour analyse ;
|  |   |   - Ajouter une nouvelle DC à la population ;
|  |   |  - Ajouter une nouvelle DC à la population
|  |  Sinon
|  |       -La DC retourne à la population ;
|  |  Fin
|  Fin
Fin
 Pour chaque entrée faire
|    -Calculer le nombre de DC mature et de DC semi-mature ;
|    Si nb DC semi-mature > nb DC mature alors
|    |      -Antigène = normal ;
|    |      -MCAV = 0
|    Sinon
|    |      -Antigène = anormal ;
|    |      -MCAV = 1 ;
|    Fin
Fin
```

Pour évaluer la nature potentiellement anormale d'un antigène, un coefficient est dérivé de valeurs totales sur la population, appelée MCAV (Mature Context Antigen Value) de cet antigène. Il s'agit de la proportion des présentations de contexte mature (valeur de contexte de 1) de cet antigène particulier, par rapport à la quantité totale d'antigènes présentés. Cela se traduit par une valeur comprise entre 0 et 1 pour lesquelles un seuil d'anomalie, appelé «Seuil MCAV », peut être appliqué. La valeur retenue pour ce seuil reflète des items normaux ou anormaux présentés dans l'ensemble initial de données. Une fois que cette valeur a été appliquée, les antigènes avec un MCAV qui dépasse ce seuil sont classés comme anormaux et vice versa.

5.2 Algorithme de la sélection négative avec C-detecteurs

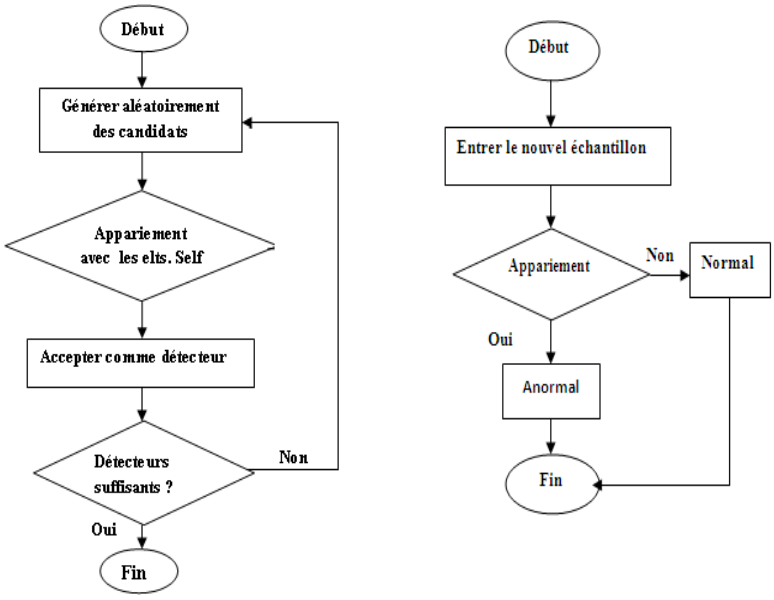
L'algorithme de sélection négative est considéré comme un processus de détection d'intrusion composé de trois phases principales :

1. La définition du self.
2. La génération de détecteurs.
3. La surveillance d'apparition des anomalies.

Il y a deux manières de mettre en œuvre l'algorithme de la sélection négative : avec V-detectors (nombre variable de détecteurs) et celui avec C-detectors (nombre constant de détecteurs) [8], que nous avons choisi dans notre travail.

Il y a deux processus essentiels dans l'algorithme NSA avec C-detecteurs, c'est le processus de génération des détecteurs et celui de la détection qui sont illustrés dans les figures suivantes :

Fig. 1. Processus de génération des détecteurs (à gauche) et processus de détection (à droite)



Algorithme 2. Pseudo code de l'algorithme de la sélection négative avec C-detecteurs.

Entrée : $\boldsymbol{S} \subseteq \boldsymbol{U}$ ≡ données étiquetées « normal ».
l, r *où* ***l :*** longueur de la chaine et ***r*** le seuil de correspondance (Matching threshold)
Sortie : ensemble de détecteurs $\boldsymbol{D} \subseteq \boldsymbol{U}$
Début
-Générer un ensemble *D* de détecteurs ; (de sorte qu'aucun ne correspond à un élément de *S*).
-Surveiller le nouvel échantillon $\boldsymbol{\delta} \in \boldsymbol{U}$; (vérifier continuellement avec les détecteurs en *D* vis-à-vis de $\boldsymbol{\delta}$
Si n'importe quel détecteur correspond à $\boldsymbol{\delta}$ **alors**
-Le classer comme anormal ;
Fin
Fin

5.3 L'ensemble de données et le processus de normalisation

En 1998, DARPA (Defense Advanced Research Projects Agency) a lancé le programme "DARPA Intrusion Detection Evaluation" qui a été élaboré et géré par le MIT Lincoln Labs. L'ensemble de données de *KDD cup'99* est dérivé de la DARPA 98, l'ensemble de données du laboratoire Lincoln pour l'application de techniques de data mining dans le domaine de la détection d'intrusion. L'ensemble de données considéré contient deux sources de données, qui sont les données du s*niffer* du réseau placé entre un routeur et une passerelle à l'extérieur et les données d'audit du système Solaris de l'hôte d'audit Solaris. *KDD cup'99* résume les deux sources de données dans les connexions (instances de données), chaque connexion dispose de 41 attributs. *KDD cup'99* est l'un des rares ensembles de données étiquetées disponible dans le domaine de la détection d'intrusion. Les instances de données sont étiquetées comme des connexions normales ou types d'attaques [13]. Comme la détection d'intrusion par les systèmes immunitaires artificiels suppose l'existence de deux classes, les étiquettes de chaque instance de données dans l'ensemble de données original sont remplacées soit par « normale » pour les connexions normales ou « anormale » pour les attaques. En raison de l'abondance des attributs, il est nécessaire de réduire la taille de l'ensemble de données en enlevant les attributs non pertinents. Pour cela, les gains d'information de chaque attribut sont calculés et les attributs avec les gains d'informations les plus bas sont retirés de l'ensemble de données. Le gain d'information d'un attribut indique la pertinence statistique de cet attribut par rapport à la classification [14]. Le gain d'information, appelé Gain (S, A) d'un attribut A par rapport à une collection d'exemples S, est défini dans l'équation suivante :

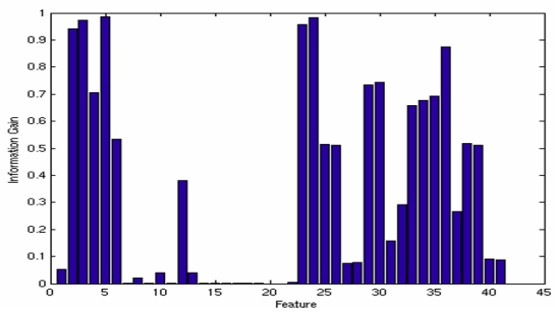
$$Gain\ (S, A) = Entropie\ (S) - \sum_{v\ \epsilon\ valeurs\ (A)} \left(\frac{|Sv|}{|S|}\ Entropie\ (S_v)\right) \qquad (2)$$

Où (A) correspond à l'ensemble des valeurs possible de l'attribut A, et S_v est le sous ensemble de S pour lequel l'attribut A à une valeur *v*, l'équation de l'entropie est la suivante (sachant que $\boldsymbol{p_i}$ est la proportion d'appartenance de S à la classe *i*):

$$Entropie\ (S) = \sum_{i=1}^{2} -p_i\ \log_2 p_i \qquad (3)$$

Après calcul des gains d'informations des 10% de la base de données *KDD cup'99*, nous obtenons l'histogramme ci-dessous.

Fig. 2. Les gains d'information de chaque attribut [14]



Il apparait qu'il n'y a que 10 attributs dont les gains d'informations sont les plus élevés qui ont été regroupés dans les trois catégories de signaux d'entrée. Il y a d'autres données d'entrées, en plus des signaux pré-classés dont le DCA a besoin, qui sont les antigènes, ils sont créés en combinant les trois attributs nominaux. Pour l'algorithme NSA, seuls ces 10 attributs seront utilisés [14].

6 Expérimentations et Résultats

Nos expérimentations consistent en l'implémentation de deux algorithmes des systèmes immunitaires artificiels, qui sont l'algorithme des cellules dendritiques (DCA) et l'algorithme de la sélection négative (NSA) avec C-detecteurs. Ces deux algorithmes n'ont pas le même type d'apprentissage, le DCA est un algorithme d'apprentissage non supervisé tandis que le NSA est un algorithme d'apprentissage supervisé. Les deux algorithmes ont été implémentés en langage Java sous NetBeans IDE. L'analyse "the receiver operating characteristics (ROC)" est effectuée afin d'évaluer les performances de la classification du DCA et du NSA. Les taux de vrais positifs (TP), de faux négatifs (FN), de faux positifs (FP) et de vrais négatifs (TN) de chaque expérimentation sont calculés, ainsi que le taux de détection (DT) et le taux de fausses alarmes (FAR). Pour toutes les expérimentations concernant l'algorithme DCA, la taille de la population des cellules dendritiques a été fixée à 100 et elle demeure constante à chaque itération du système. Le seuil de migration pour chaque cellule dendritique individuelle est choisi aléatoirement entre 100 et 300 afin d'assurer la survie de la cellule après plusieurs itérations du système. Nous avons appliqué quelques variantes lors de l'implémentation des deux algorithmes qui sont :

- **Expérience 1 :** DCA avec un chargement de données continu.
- **Expérience 2 :** DCA avec un chargement de données aléatoire.
- **Expérience 3 :** NSA avec un chargement de 1000 détecteurs aléatoires et avec différentes valeurs de ***r*** (2, 3, 4, 5, 6).
- **Expérience 4 :** NSA avec un chargement aléatoire d'un seul détecteur.

Chaque programme a été lancé plusieurs fois, 10 itérations pour le DCA et 100 itérations pour le NSA (seulement 10 itérations pour le DCA car son temps d'exécution est relativement plus grand que celui du NSA, 15 à 20 min pour une itération et il a besoin d'énormément d'espace mémoire, contre à peine 30 sec pour le NSA qui ne requiert que très peu d'espace mémoire). Nous avons donc utilisé l'analyse ROC [12] pour mesurer la performance réelle de nos classifieurs.

Tableau 3. Les résultats ROC des différentes expériences

Catégorie		**TP**	**TN**	**FP**	**FN**	**DT**	**FAR**
Exp. 1		0.7154	1.00	0.00	0.2846	0.7154	0.00
Exp. 2		0.6521	1.00	0.00	0.3179	0.6821	0.00
Exp. 3	*r = 2*	0.9211	0.4294	0.3705	0.0799	0.9211	0.4631
	r = 3	0.7548	0.5183	0.2361	0.2452	0.7548	0.3129
	r = 4	0.3455	0.6324	0.2005	0.6545	0.3455	0.2407
	r = 5	0.2845	0.7128	0.0085	0.7155	0.2845	0.0102
	r = 6	0.0814	0.1985	0.0007	0.9186	0.0814	0.0035
Exp. 4		0.7121	0.4987	0.2147	0.2879	0.1210	0.3009

Nous avons voulu tester si l'ordre des données pouvait affecter le bon fonctionnement du DCA. Les résultats des deux premières expériences, montrées dans le tableau 3, nous indiquent une légère baisse du taux de détection lorsque les données sont sélectionnées aléatoirement. Nous avons essayé de changer les poids de calcul des signaux de sortie, le résultat de ce changement a été catastrophique, aucune donnée n'a été correctement classifiée. Le DCA semble avoir fourni de bonnes performances du point de vue du taux de fausse alarme, qui est égal à 0, ce qui signifie que l'un des objectifs de la détection d'anomalie vient d'être atteint car il est important qu'il y ait le moins de fausses alertes possibles. Nous avons également noté que, lorsque l'ensemble de données est petit (1000 enregistrement par exemple), la classification du DCA est excellente et le taux de vrais positifs est relativement élevé (0.99 ou 1.00). Pour l'algorithme NSA, nous avons également effectué un chargement de détecteurs aléatoire et un chargement d'un seul détecteur, avec lequel s'est faite la correspondance avec tous nos exemples. L'utilisation de plus d'un détecteur sélectionné aléatoirement fournit de meilleurs résultats que l'utilisation d'un seul. Il y a également une autre variante de l'algorithme NSA qui est le changement de la valeur de ***r*** (règle de correspondance ***r bits contigus*****),** qui a énormément affecté le déroulement de la classification. Nous avons obtenu des résultats très différents les uns des autres, lorsque le ***r = 6***, il n'y a presque plus de classification correcte, et les résultats s'améliorent au fur et à mesure que ***r*** décroit. Par conséquent, la valeur de ce dernier semble très importante, plus le ***r*** est petit, meilleure est la classification. Ceci semble évident, vu qu'il ne fait l'appariement qu'entre deux attributs, ce qui reste insuffisant pour juger du bon fonctionnement du système. Il est à noter également que, plus la valeur de ***r*** est élevée, moins le classifieur fonctionne correctement. Il y a un autre problème que nous rencontrons avec l'algorithme NSA, particulièrement lorsque le choix d'un seul détecteur est effectué, c'est que les résultats sont vraiment aléatoires, au lancement de 10 itérations consécutives, nous pouvons obtenir un taux de vrais positifs allant de 0.08 jusqu'à 0.25 et dans certains cas 0.50, ce qui fait du NSA un algorithme peu stable et nous ne pouvons se fier à ses résultats.

Contrairement à l'algorithme DCA, qui n'a émis aucune fausse alarme, le NSA en a émis un nombre important, ce qui le rend peu fiable et peu adéquat pour la détection des anomalies. Ainsi, comparé au NSA, le DCA gère correctement les grands ensembles de données et ses résultats sont satisfaisants et prometteurs.

7 Conclusion et Perspectives

Nous avons exploité deux algorithmes immunitaires dans le domaine de la détection d'anomalies pour notre expérimentation avec l'ensemble de données *KDD cup'99*, les résultats obtenus concernant l'algorithme des cellules dendritiques (DCA) sont plutôt encourageants et prouvent que nous pouvons encore améliorer la mise en œuvre de cet algorithme afin d'obtenir de meilleurs résultats. A l'opposé, l'algorithme de sélection négative (NSA), n'a pas fourni de résultats probants, il émet un nombre important de fausses alarmes contrairement à l'algorithme DCA dont le taux de fausses alarmes avoisine le zéro. Nous constatons également que NSA rencontre des difficultés à gérer un ensemble de données de grande dimension, ce qui représente un sérieux inconvénient, au vu des tailles actuelles des bases de données des systèmes informatiques.

Des travaux similaires au nôtre ont été réalisés ces dernières années, en particulier ceux de Julie Greensmith [5, 6, 7] sur l'algorithme du DCA et son application à la détection d'anomalies, il y a eu également des comparaisons de l'algorithme DCA avec l'algorithme TLR [9] et les cartes auto-organisatrices de Kohonen (SOM) [12], ainsi que sur l'adéquation de NSA pour les problèmes de détection d'intrusions [15]. Les résultats de ces travaux sont tout aussi encourageants et placent l'algorithme DCA comme l'approche la plus approprié pour le problème de détection d'intrusions.

Les systèmes immunitaires artificiels (AIS) présentent des solutions prometteuses dans le domaine de la détection d'intrusion. Les recherches autour de ces systèmes représentent toujours le centre d'intérêt de différents chercheurs, afin de pouvoir exploiter tous les concepts et les mécanismes d'identification et de détection utilisés par le système immunitaire naturel. Dans ce que nous avons pu exploiter, les recherches futures qui peuvent être appliquées à l'algorithme DCA consistent à trouver le moyen de le rendre plus adaptatif et flexible. Nous pouvons également essayer de le tester avec des ensembles de données différents et aussi de réaliser des comparaisons rigoureuses avec d'autres méthodes des systèmes immunitaires artificiels afin de voir où il se situe par rapport aux performances des autres méthodes immunitaires.

Dans une optique plus générale, il serait intéressant de mener des comparaisons plus approfondies entre des classifieurs issus de l'immunologie artificielle et d'autres classifieurs, pouvant être bio-inspirés ou pas, en envisageant des applications intéressantes telles que la détection d'intrusions. Ces études comparatives pourront certainement nous mener à des conclusions très intéressantes dans le domaine attrayant de l'informatique bio-inspirée…

Références

1. Dasgupta D., Nino L. F., “Immunological Computation, theory and application”, Auerbach, 2009
2. Kim J., Bentley P., Aickelin U., Greensmith J., Tedesco G., Twycross J., “Immune System Approaches to Intrusion Detection - A Review”, Natural Computing, Vol. 6, No. 4, pp. 413-466, 2007.
3. http://ima.ac.uk/danger “The Danger Project”
4. Matzinger P., “Tolerance, danger and the extended family. Annual Reviews in Immunology, 12:991–1045, 1994.
5. Greensmith J., Aickelin U., et Twycross J., “Articulation and Clarification of the Dendritic Cell Algorithm”, In: Bersini, H., Carneiro, J. (eds.) ICARIS. LNCS, Vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
6. Greensmith, J., Aickelin U., “DCA for SYN Scan Detection”. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 49–56 (2007)
7. Greensmith J. “The Dendritic Cell Algorithm”, PhD Thesis, University of Nottingham, Nottingham, Britain, October 2007
8. Simon M. Garrett, “How Do We Evaluate Artificial Immune Systems”, Evolutionary Computation, Vol.13, No.2, pp. 145-178, (2005)
9. Aickelin U. and Greensmith J., “Sensing Danger: Innate Immunology for Intrusion Detection” University of Nottingham, Nottingham Britain 2007.
10. Debar H., “An Introduction to Intrusion-Detection Systems”, Proceedings of Connect-2000 Doha, Qatar, 2000
11. Aickelin U., Bentley P., Cayzer S, Kim J., McLeod J. “Danger Theory: The Link between AIS and IDS?” Digital Media Systems Laboratory, HP Laboratories Bristol, Nottingham Britain 2003
12. Greensmith J., Feyereisl J., Aickelin U. “The DCA: SOMe Comparison A comparative study between two biologically-inspired algorithms” Evolutionary Intelligence Vol. 1, No. 2, pp. 85-112, 2008
13. Tavallaee M., Bagheri E., Lu W., Ghorbani Ali A., “A Detailed Analysis of the KDD CUP 99 Data Set”, Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications Ottawa, Pages: 53-58, Ontario, Canada 2009.
14. Güneş Kayacık H., Nur Zincir-Heywood A., Heywood M. I., “Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets”, In: Third Annual Conference on Privacy, Security and Trust (PST) (2005).
15. Stibor T. “On the Appropriateness of Negative Selection for Anomaly Detection and Network Intrusion Detection”, PhD Thesis, Germany 2006

Génération automatique des modèles de services à partir des modèles de processus métiers : Approche dirigée par les ontologies

Mokhtar Soltani[1] et Sidi Mohammed Benslimane[2]

[1] Département des Sciences et de la Technologie, Faculté des Sciences et de la Technologie et Sciences de la matière, Université Ibn Khaldoun, Tiaret, Algérie

[2] Département d'Informatique, Faculté des Sciences de l'Ingénieur, Université Djillali Liabes, Sidi Bel Abbes, Algérie

{mokhtar.soltani@gmail.com, benslimane@Univ-sba.dz}

Résumé. L'intégration des paradigmes gestion des processus métiers (BPM), le calcul orienté service (SOC) et le développement dirigé par les modèle (MDD) pour améliorer le développement des solutions orientées services à partir des modèles métiers est devenu actuellement une exigence fondamentale pour beaucoup d'entreprises. La modélisation des processus métiers est également positionnée au centre des efforts de développement de logiciel, car la fabrication de ces modèles constitue explicitement la base pour la définition de services. Dans ce contexte, nous proposons une approche de formalisation et de construction d'une base de connaissances métier afin de supporter la génération automatique des modèles de services à partir des modèles de processus métiers. L'objectif étant d'améliorer et faciliter le processus de développement des systèmes d'information orientés services.

Mots clés: modélisation des processus métiers (BPM), calcul orienté service (SOC), architecture dirigé par les modèles (MDA), processus métier inter-organisationnel, interopérabilité.

1 Introduction

Aujourd'hui les entreprises sont organisées en réseaux, au sein desquels différents acteurs peuvent interagirent. La compétitivité de ces entreprises est profondément liée à la capacité de structurer, partager et échanger les connaissances et le savoir-faire avec l'ensemble des participants au réseau collaboratif. Ce besoin d'échanger des connaissances oblige les entreprises d'évoluer leurs systèmes d'informations et leurs applications afin de les rendre interopérables.

L'interopérabilité des applications d'entreprise permet d'assurer l'échange des fonctionnalités et des services d'une manière transparente à l'utilisateur. Chaque fonctionnalité, service, ou donnée possède son propre modèle. Un certain nombre de transformations entres ces modèles sont indispensable pour assurer l'interopérabilité

entre les différentes entités hétérogènes de l'entreprise [9]. Pour que ces transformations entres modèles soient une solution efficace permettant d'établir l'interopérabilité dans un environnement purement hétérogène ; il faut qu'elles doivent être guidées par un cadre de modélisation standard. L'approche MDA (Model-Driven Architecture) [9] fournit les bases pour soutenir l'interopérabilité dirigée par les modèles.

Pour développer un projet informatique, il est nécessaire d'en connaître la cible. Pou cela, on raisonne en générale en termes de besoin et de réponse à ce besoin en termes d'application. Le développement d'une application d'entreprise à grande échelle commence toujours par le niveau d'abstraction le plus élevé là où il ya la spécification et la représentation des métiers d'entreprise sous forme de modèles de processus métiers. Ces modèles doivent être projetés progressivement sur une architecture plus adaptée au besoin d'interopérabilité. Actuellement, le paradigme le plus adapté à la réalisation des applications interopérables est le paradigme orienté services. Puisque les services encapsulent les fonctionnalités des applications selon des processus métiers de l'entreprise, la compréhension des processus est un préalable nécessaire à la mise en œuvre d'une architecture orientée services (SOA).

L'association de la modélisation des processus métiers de l'entreprise (BPM), l'architecture dirigée par les modèles (MDA) et l'architecture orientée services (SOA) permet de nous offrir un cadre et une méthodologie puissante à la mise en œuvre des systèmes d'informations interopérables [6], [8], [10].

2 Etat de l'art

L'évolution des méthodes et des technologies dans le domaine du génie logiciel fait émerger de nouvelles approches pour le développement d'applications. Ainsi, ont fait leur apparition, le paradigme orientée objets, ensuite le paradigme orientée composants et de nos jours, le paradigme orientée services. Tous ces paradigmes reposent sur les principes de base du génie logiciel : l'abstraction, la séparation de préoccupations et la modularité. Chaque nouvelle approche est basée sur les anciens paradigmes tout en y ajoutant de nouveaux atouts.

L'idée de l'approche orientée services est de construire rapidement des applications par l'assemblage d'un ensemble de services. Le résultat de cet assemblage est appelé « application composite ». La mise en place effective de cette approche permet de faciliter l'intégration et l'interopérabilité des systèmes d'informations d'entreprise qui sont hétérogènes et n'ont pas été conçus pour être interopérables. Donc, l'approche orientée services propose un cadre pour résoudre cette problématique par le développement des applications interopérables, dynamiques, et complètement réparties.

2.1 Les services

L'approche à services est basée principalement sur le paradigme à composants dont l'idée de base était l'assemblage de blocs logiciels préfabriqués, appelés des

composants. Un composant est défini comme une entité logicielle composable, réutilisable, décrivant explicitement ses capacités et ses dépendances.

Le paradigme SOC (Service Oriented Computing) [6] propose que la construction d'une application soit réalisée par l'assemblage de services logiciels préexistants, testés et validés fournis par des fournisseurs divers ce qui implique que la productivité augmente par la réutilisation de ces services. Un autre avantage de cette approche est le faible couplage entre les différentes entités qui constituent une application la chose qui rendre l'application plus flexible contrairement a l'approche orientée composant qui est caractérisée par une rigidité forte, c'est-à-dire qu'une fois concevoir une application à base de composants il est difficile de modifier son architecture à l'exécution. Ce problème est dû notamment au fort couplage résultant de la définition explicite des liens entre les interfaces fournies et requises des composants constituant une application.

Dans l'approche orientée services, les entités de base sont appelées services. Un service est défini comme suit:

"*Services are autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled by using standard protocols*"

Un service est alors une entité logicielle autonome, auto-descriptive indépendante de la plateforme et qui fournit une fonctionnalité métier. Les principales caractéristiques d'un service sont:

- **La modularité :** un service est modulaire et réutilisable. Il est possible de construire des applications ou des services plus complexes à partir d'autres services atomiques,
- **L'autonomie :** un service est autonome, complet et cohérent dans la mesure où le service contient toute sa logique d'exécution ce qui lui permet de s'exécuter indépendamment des autres services,
- **La disponibilité :** un service est disponible est utilisable par des clients,
- **La description :** un service possède une description qui est lisible par des machines. La description sert à spécifier l'interface du service et sa localisation,
- **L'indépendance d'implémentation :** un service a une claire séparation entre sa description (ou interface) et son implémentation.
- **La publication :** la description du service est publiée dans un annuaire qui sera consultable par les clients du service.

La description du service est utilisée par les clients (consommateurs de services) pour rechercher, sélectionner et invoquer le service qui s'adapte le mieux à ses besoins. La description d'un service est indépendante de son implémentation.

2.2 L'Architecture Orientée Services (SOA)

Une architecture SOA (Service-Oriented Architecture) est composée de trois éléments principales : Le fournisseur de service, Le consommateur de service (Client de service), et l'annuaire de service (Registre de services). Le fournisseur d'un service représente une organisation (ou une personne) qui fournit une fonctionnalité sous la

forme d'un service. Ce fournisseur met à disposition des clients du service les informations nécessaires à l'utilisation du service. Ces informations sont regroupées dans un annuaire de services sous forme d'une description du service, la description peut contenir des informations sur la fonctionnalité offerte, le comportement du service, les propriétés non-fonctionnelles du service et les protocoles de communication qui doivent être utilisés lors de sa consommation.

Le fournisseur effectue la publication de la description du service dans un registre de services (ou annuaire des services). Le registre de services est un intermédiaire entre les fournisseurs et les consommateurs des services.

Le client réalise une recherche dans le registre pour obtenir des descriptions de services disponibles pouvant satisfaire ses besoins. Le client sélectionne parmi les services découverts le service le plus adapté à ses besoins. Une fois la phase de sélection termine, le client utilise les informations disponibles dans la description pour effectuer une communication avec le fournisseur afin de consommer la fonctionnalité du service.

Le point fort de l'utilisation du registre de services est l'optimisation du temps d'exécution au niveau du fournisseur de services c.-à-d. que les clients n'ont pas besoin de s'adresser directement aux fournisseurs avant de consommer le service.

2.3 L'architecture SOA dirigée par le processus métier

La vision processus joue un rôle important dans les théories des organisations comme dans le domaine système d'information où la modélisation des processus est considérée comme un élément clé de la représentation de la dynamique. La modélisation des processus métiers est un préalable nécessaire à la conception d'un système d'information organisationnel.

L'approche processus est imposée progressivement dans le management des entreprises pour en améliorer la productivité. Elle est s'est étendue à l'ensemble de l'organisation avec la théorie de la gestion par les activités ainsi que le nouveau besoin de la coopération entre les organisations à permet de les obliger de se concentrer leurs efforts sur la normalisation de touts les concepts intervenant dans l'approche afin d'assurer une interopérabilité entre ses différentes systèmes d'informations au niveau métier, logique et technique.

La norme [ISO 9000, 2000] définit un processus métier comme suit: " *Un ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrés en éléments de sortie*". Dans la gestion par les activités, un processus est défini comme: " *Une séquence d'activités différentes reliées par des relations client fournisseur qui s'enchaînent à partir d'un facteur de déclenchement commun*". Où une activité est " *un ensemble de tâches élémentaires réalisées par un individu ou un groupe, faisant appel à un savoir faire spécifique, homogènes du point de vue de leurs comportements de coût et de performance, permettant de fournir un output à un client interne ou externe à partir d'un panier d'input*".

Les définitions de processus métier reflètent implicitement les besoins fonctionnels. Cependant, il n'est pas suffisant de concevoir juste les activités métiers reliées par les flux de contrôle du processus. Pour représenter l'ensemble complet des exigences, une définition de processus doit indiquer explicitement toutes les entités

qui participent au processus. Ces exigences devraient être transformées, sans perte d'information, en des spécifications riches sémantiquement dont lesquels divers composants logiciels peuvent être dérivés.

Une architecture SOA dirigée par les processus (Process-Driven SOA) PDSOA [4], [5], [6], [9], [10] prolonge la SOA traditionnelle par une couche de spécification des processus métiers de haut niveau. En particulier, PDSOA fournit un paradigme de développement pour accomplir systématiquement et efficacement les fonctionnalités métiers en utilisant un moteur d'exécution des processus qui fait appel à des services SOA générés à partir les activités de processus en entré.

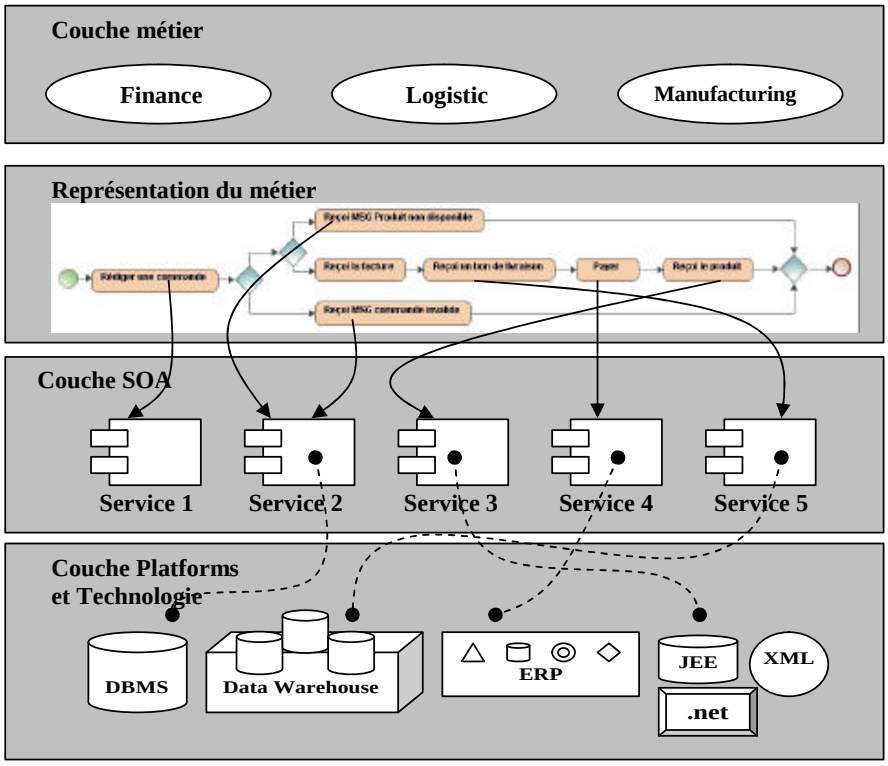


Fig. 1. L'architecture orientée service dirigée par les processus

La figure **Fig. 1.** représente la relation entre le niveau métier, les modèles de processus métiers, les services SOA implémentant ces processus, et les ressources technique.

2.4 L'Architecture Dirigée par les Modèles (MDA)

MDA (Model-Driven Architecture) est un cadre de modélisation qui s'articule autour des modèles comme éléments de bases pour analyser, concevoir, développer, et maintenir des systèmes logiciels à travers une succession de transformation de modèles. L'approche MDA propose qu'à partir de la modélisation de haut niveau d'abstraction d'un système informatique, de générer l'implantation correspondante pour une plateforme spécifique dans le but de favoriser l'évolutivité, la réduction de coût du développement par la réutilisation des modèles et l'interopérabilité dirigée par les transformations des modèles.

2.4.1 Les modèles du MDA

Un modèle d'un système est la spécification formelle des fonctions, de la structure et/ou du comportement de ce système dans son environnement, dans le but de comprendre le système, échanger et partager les connaissances relatif au système, développer ou améliorer le système. Un modèle doit pouvoir être utilisé pour répondre à des questions sur le système modélisé. Il doit être suffisant et nécessaire pour permettre de répondre à certaines questions en lieu et place du système qu'il représente, exactement de la même façon que le système aurait répondu lui-même.

Dans le processus MDA, tout est considéré comme modèle, aussi bien les processus métiers, les systèmes, les programmes, les traces d'exécution, les plateformes, les transformations, les vérifications, les testes, et les composants.

MDA recommande l'élaboration de modèles: (i) d'exigence (*Computation Independent Model - CIM*) dans lesquels aucune considération informatique n'apparaît, (ii) d'analyse et de conception (*Platform Independent Model - PIM*), (iii) de code (*Platform Specific Model - PSM*), (iv) de plateforme (*Platform Description Model - PDM ou PM*).

CIM (Computation Independent Model) : C'est le modèle métier ou le modèle du domaine d'application. Il sert à représenter ce que le système devra exactement faire sans rentrer dans les détails d'implémentation. Il est utile, non seulement comme aide pour comprendre le fonctionnement du futur système, mais également comme source de vocabulaire partagé utiliser pour dériver d'autres modèles.

PIM (Platform Independent Model) : Une fois les analystes métiers définis clairement le modèle d'exigences, la phase de conception peut commencer. Dans cette phase, les concepteurs de logiciel dérivent un modèle PIM à partir des connaissances encapsulées dans le modèle CIM. Le PIM est un modèle informatique qui représente une vue partielle d'un CIM. Il représente l'architecture logique du système sous forme de modèles de conception. Il représente le fonctionnement des services logiciels mais sans montrer les détails de son déploiement sur la plateforme.

PSM (Platform Specific Model) : Le modèle PSM est dépendant de la plate-forme technique sur laquelle la future application va être exécutée. Il est considéré comme une interface entre les modèles de conceptions et la plateforme technique. Il sert notamment à la génération de code exécutable vers la ou les plates-formes techniques.

PDM (Platform Description Model) ou PM (Platform Model) : Il décrit la plateforme cible sur laquelle le système va être exécuté. Un PDM contient des informations utilisées pour la transformation du modèle PIM vers les modèles PSM.

Le passage de PIM à PSM fait intervenir des mécanismes de transformation de modèles et un modèle de description de la plateforme (PDM). Cette démarche s'organise selon un cycle de développement « en Y » propre au MDD (Model Driven Development).

L'objectif du MDA est d'automatiser le maximum que possible les opérations sur les modèles. Ces modèles doivent être exprimés en un format exploitable par machine. La structure dans laquelle les modèles sont exprimés doit être clairement définie sous forme de méta-modèle.

3. Une transformation automatique d'un processus métier collaboratif en une architecture SOA

Avec l'évolution de l'architecture orientée services (SOA) les développeurs de logiciels ont focalisé sur le développement des services réutilisables pour augmenter la productivité. Ces services atomiques qui offrent des petites fonctionnalités sont assemblés en services composés afin d'automatiser des parties réutilisable d'un processus métiers.

La compréhension des processus métiers inter-organisationnels est un préalable nécessaire à la mise en œuvre des systèmes d'informations interopérables à base d'une architecture SOA. Pour cela on à besoin de définir une méthode qui nous permettre de dériver les composants d'une architecture SOA à partir d'un modèle de processus métier afin d'assurer la construction semi-automatique des applications interopérables. Plusieurs questions peuvent être adressées afin de réaliser cet objectif :

Quels sont les éléments qui doivent être modélisés afin de définir et d'analyser les processus métiers collaboratifs ?

Quelle est la notation graphique la plus appropriée pour modéliser les éléments identifiés ?

Comment dériver les processus privés de chaque participant dans la collaboration à partir d'un seul processus publique commun ?

Comment les éléments modélisés peuvent être transformés en composants d'une architecture SOA ?

Plusieurs éléments peuvent être structurés dans différentes catégories de modèles afin de fournir un cadre pour modéliser les processus collaboratif. Trois catégories de modèles sont considérées comme une base nécessaire à la définition des interactions inter-organisationnelles :

Le modèle de rôle qui fournit une vue d'ensemble des différents rôles impliqués dans la coopération au niveau d'organisation.

Le modèle de processus qui décrit le flux d'activité effectué par les différents rôles et indiqué les interfaces de processus entre les entreprises participantes au réseau collaboratif.

Et finalement, le modèle de l'information qui représente les objets d'information appropriés sert comme base pour définir le contenu des messages.

Le modèle de rôle décrit les responsabilités des personnes et des unités organisationnelles qui sont directement impliqués et visibles aux collaborateurs externes. Ainsi, il établit une compréhension commune de la façon dont les participants de la collaboration interagissent. Ceci est considéré comme un préalable pour modéliser les processus inter-organisationnels. Les principaux éléments impliqués dans ce modèle sont des rôles et des unités organisationnelles ou des positions. Une description de rôle définit les responsabilités et les fonctions. Les rôles peuvent être assignés à un niveau d'une unité organisationnelle ou à une seule personne qui exécute une activité. Les organismes et les personnes peuvent exécuter des rôles multiples.

Le modèle de processus publique est l'élément le plus important pour modéliser les processus inter-organisationnels. Il définit l'ordre des activités qui sont exécutées par les différents rôles : Ce modèle se concentre sur le processus public qui contient les principales activités acceptés par les participants afin d'effectuer une coopération inter-organisationnelle. L'interface de processus entre les participants de la collaboration doit être détaillée afin de fournir l'entrée nécessaire pour la conception de services et permettre la transformation des éléments de modèles en une architecture SOA.

Le modèle d'information structure les objets d'information nécessaires qui doivent être traités durant l'exécution du processus collaboratif. Ce type de modèle peut être exprimé en un diagramme de classe UML dont chaque classe représente un objet de l'information qui est encore décrit par des attributs.

Les trois types de modèles expliqués ci-dessus (modèle de rôle, modèle de processus, modèles d'information) assurent l'interopérabilité à un niveau de processus.

Comme réponse à la deuxième question, les développeurs métier supposent que la spécification des modèles de processus métiers exige que la notation utilisée dans la modélisation doive être compréhensible à la foie par l'homme et par la machine. Pour cela l'initiative BPMI et l'OMG (Object Management Group) ont proposés la spécification BPMN (Business Process Modeling Notation). BPMN a pour but d'offrir une notation explicite, facile et accessible à tous les utilisateurs métiers. BPMN propose aussi des passerelles pour l'exécution automatique des processus sous forme des instructions BPEL (Business Process Execution Langage) exécutées dans un moteurs BPMS (Business Process Management System).

Comme réponse à la troisième question, nous supposons que la mise en application des modèles de processus collaboratifs permet de gérer la collaboration inter-entreprise avec leurs participants pour améliorer leurs performances et compétitivité. Les modèles collaboratifs peuvent être réalisés par les collaborations B2B (Business-to-Business) qui nécessitent une interopérabilité orientée processus entre des entreprises hétérogènes et autonomes.

L'établissement de l'interopérabilité inter-entreprise exige la spécification détaillée de différentes interactions entre les différents participants. Ces interactions sont exprimées sous forme de modèles de processus métiers collaboratifs. Dans les collaborations B2B l'intégration et l'interopérabilité des processus et des systèmes d'entreprises sont nécessaires pour soutenir l'exécution des processus collaboratifs. Chaque entreprise doit définir le processus d'interface qui représente le rôle qu'elle exécute dans la collaboration à partir d'un modèle de processus collaboratif qui décrit

la vue globale des interactions d'entreprise afin d'implémenter ce processus d'interface dans un système de gestion de processus métiers.

L'interopérabilité inter-entreprise doit être réalisée dans un niveau métier et dans un niveau technologique. Au niveau métier, les entreprises concentrent sur la conception des processus collaboratifs pour définir le comportement de la collaboration inter-entreprise. Le processus métier collaboratif permet de définir la vue globale des interactions entre les entreprises pour réaliser des objectifs métiers communs. Au niveau technologique, les entreprises se concentrent sur l'intégration et l'interopérabilité de leurs systèmes B2B pour exécuter des processus collaboratifs. Ceci implique une génération des spécifications B2B, c.-à-d. des interfaces des systèmes des participants et des spécifications des processus métiers sont exigées par chaque entreprise pour exécuter le rôle indiqué dans un processus collaboratif et pour le mettre en application dans un système de gestion des processus métiers.

La conception et la gestion des processus collaboratifs dans les deux niveaux impliquent de nouveaux défis, principalement la réalisation de plusieurs exigences :

- Autonomie : les entreprises sont considérées comme étant d'entités autonomes, cachant leurs décisions internes, leurs activités et leurs processus. Les systèmes d'information qui gèrent les collaborations B2B à chaque entreprise, doivent être indépendants.
- Gestion décentralisée des processus collaboratifs contrôlés conjointement par les entreprises.
- Interactions de Pair à Pair: les systèmes d'information des entreprises agissent l'un avec l'autre d'une manière directe sans médiation d'un tiers.
- Négociation : qu'elle est exigée dans la gestion des processus collaboratif.
- L'alignement entre la solution métiers et la solution technologique afin de garantir que la solution technologique fournit un support au comportement des processus collaboratifs.

Pour accomplir les exigences ci-dessus, on a proposé une méthode basée sur MDA pour la conception d'une architecture orientée service à partir un processus collaboratifs.

Une collaboration B2B exige la définition d'une solution métiers aussi bien qu'une solution technologique, en raison qu'elle comporte l'intégration d'entreprise au niveau métier et au niveau technologique. En outre, deux vues à dans les solutions métiers et technologiques doivent être considérées: la vue de collaboration, qui se rapporte aux exigences globales et publiques a convenu par les participants; et la vue du participants, qui se rapporte aux besoins particuliers dont lesquels un participants doit répondre pour pouvoir collaborer avec d'autres participants.

Au niveau métiers, la vue de collaboration est représentée par les processus collaboratifs qui définissent le comportement de la collaboration inter-entreprise. Un processus métier collaboratif définit, d'un point de vue globale, l'échange de message entre les participants.

Une fois que les participants sont acceptés sur la vue de la collaboration, ils définissent leurs besoins métiers dans leur vue de participants. Le rôle qu'un participant exécute dans un processus de collaboration est décrit dans un processus métier d'interface (également appelé processus abstrait, ou interface comportementale). Un processus d'interface définit le comportement public externe et visible d'un participant en termes d'activités qui soutiennent la réception et l'envoi des

messages avec les autres participants à la collaboration, c.-à-d. les activités qui communiquent avec d'autres processus métiers externes. Ce comportement public peut être dérivé à partir d'un processus collaboratif. Finalement, les participants définissent leurs processus métier d'intégration (également appelés privés, exécutable, ou processus d'orchestration) à partir du processus d'interface. Un processus d'intégration ajoute la logique métier interne nécessaire à soutenir le rôle qu'un participant exécute dans un processus collaboratif. La logique métier interne inclut les activités pour produire et traiter l'information échangée aussi bien que les transformations de données et les invocations des systèmes internes.

Bien que les processus de collaboration et d'interface définissent comment les participants coordonneront leurs actions, ces processus ne sont pas exécutables. Au niveau technologique, les participants doivent produire des interfaces de leurs systèmes B2B et les spécifications exécutables des processus d'intégration en employant un langage de processus B2B standard. Puis ces spécifications peuvent être transformées en une architecture SOA qui implémente le processus d'intégration.

Pour développer les collaborations B2B, on a proposé une approche systématique pour transformer les modèles conceptuels des processus collaboratifs en modèles et spécifications concrètes des processus métiers exécutables puis transformer les éléments modélisés en un ensemble de servies implémentant ces processus. L'approche proposée implique les étapes suivantes :

a) Analyse et conception des processus collaboratifs en se basant sur un point de vue métiers pour représenter la vue de la collaboration B2B, c.-à-d. la définition des besoins métiers et des objectifs métiers communs de la collaboration B2B.
b) Annotation de ce processus métier inter-organisationnel par une ontologie. Le but est de représenter formellement les connaissances métiers encapsulées dans le diagramme de processus afin de faciliter la dérivation des processus d'interface pour chaque participant à la collaboration ainsi que la génération automatique d'une architecture orientée services qui exécute le processus métier globale.
c) Dérivation des processus d'interface à partir des processus collaboratifs afin de définir la vue publique de chaque participant. Un moteur de transformation interroge l'ontologie pour réaliser cette étape.
d) Conception des processus d'intégration en incorporant la logique privée requise pour soutenir l'échange de message avec les autres participants afin de définir la vue privée de chaque participant. Une opération de mise à jour de l'ontologie par les concepts relatifs à la logique interne de chaque partenaire est nécessaire afin d'exécuter l'étape suivante.
e) Génération de la solution technologique des modèles de processus métiers de la solution métiers, c.-à-d. les servies requis pour exécuter les processus collaboratif: interfaces des systèmes et des spécifications de processus de services exécutables des participants basés sur une norme B2B.
f) Raccorder les servies générés avec les ressources logiciels du système.

Cette méthode est basée sur MDA pour permettre la génération automatique des modèles de processus d'interface des différents participants à partir des modèles de processus collaboratifs. Le langage BPMN est utilisé pour représenter les modèles de processus collaboratif et les processus d'interface.

La méthode proposée se concentre sur des transformations horizontales entre les modèles de processus métiers collaboratifs en des modèles de processus d'interface puis une conception des processus d'intégration et finalement la génération d'un ensemble de services implémentant les processus d'intégration. Les services générés peuvent être relié avec les différentes ressources logicielles existantes (cf. **Fig. 2.**).

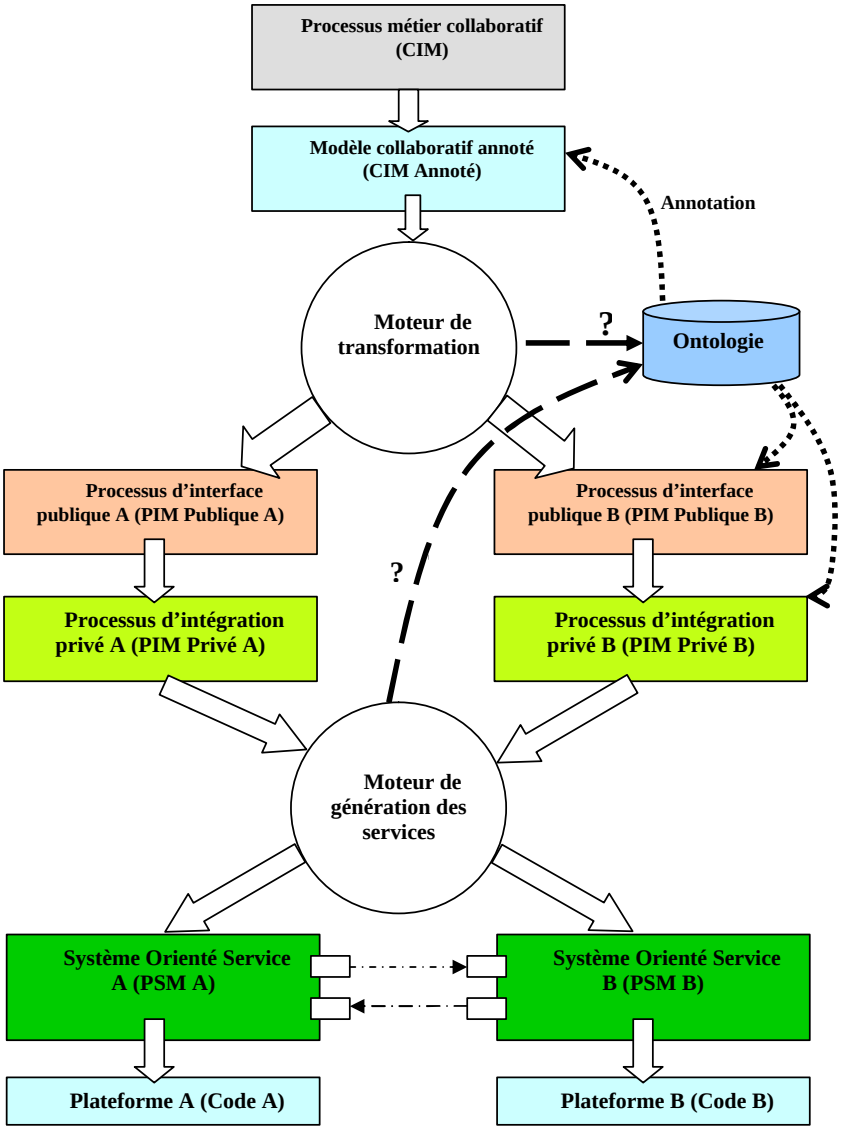


Fig. 2. Génération automatique des services SOA

4. Conclusion et perspectives

Dans cet article nous avons présenté un cadre architectural pour soutenir et faciliter le processus de développement des systèmes d'informations interopérable et plus particulièrement la génération automatique d'une architecture SOA à partir d'un processus métier collaboratif ou inter-organisationnel. La méthode proposée est basée sur l'architecture dirigée par les modèles pour transformer les modèles de processus collaboratif en modèles de processus d'interface spécifique à chaque participant à la collaboration. En appliquant cette méthode, des processus d'interface sont garantis pour être interopérable car ils sont définis suivant un processus collaboratif global. Ce processus de développement est dirigé par une ontologie qui encapsule l'ensemble des connaissances nécessaires à la génération de l'architecture SOA.

Dans la prochaine future nous allons nous focaliser sur la spécification de la structure de l'ontologie, la formalisation des algorithmes de transformation du processus inter-organisationnel en processus d'interface, la formalisation du générateur de services, et finalement l'implémentation et l'évaluation de notre approche.

Références

1. Chiara Di Francescomarino, Chiara Ghidini, Marco Rospocher, Luciano Serafini, and Paolo Tonella. Semantically-aided business process modeling. Italy, 2009.
2. Michele Missikoff, Maurizio Proietti, Fabrizio Smith. A business Process Knowledge Base for composite Services Development. Italy, 2010.
3. Antonio De Nicola, Michele Missikoff, Maurizio Proietti, Fabrizio Smith. A Logic-Based Method for Business Process Knowledge Base Management. Italy, 2009.
4. Christine Legner, Tobias Vogel, Jan Löhe, Christian Mayerl. Transforming Inter-Organisational Business Process into Service-Oriented Architectures Method and Application in the Automotive Industry, 2006.
5. Andrea Delgado, Ignacio Garcia-Rodriguez de Guzman, Francisco Ruiz, Mario Piattini. Tool support for Service Oriented development from Business Processes, 2010.
6. Christian Emig, Karsten Krutz, Stefan Link, Christof Momm, Sebastian Abeck. Model-Driven Development of SOA Services. Germany, 2007.
7. Juan M. Vara, M Valeria De Castro, Marcos Didonet Del Fabro, Esperanza Marcos. Using Weaving Models to automate Model-Driven Web Engineering proposals, 2008.
8. Ivanna M. Lazarte, Omar Chiotti and Pablo D. Villarreal. Transforming Collaborative Process Models into Interface Process Models by Applying an MDA Approach. AIS Transactions on Enterprise Systems, 2009.
9. Brian Elvesæter, Axel Hahn, Arne-Jorgen Berre, Tor Neple. Towards an Interoperability Framework for Model-Driven Development of Software Systems, 2005.
10. Frédérick Bénaben, Jihed TOUZI, Vatcharaphun Rajsiri, Sébastien Truptil, Jean-Pierre Lorré, and Hervé Pingaud. Mediation Information System Design in a collaborative SOA Context through a MDD Approch. Proceedings of MDISIS, 2008.

Ordonnancement

Complexity Analysis of Scheduling Linear Deteriorating Jobs in a Single-Machine for Minimum Sum of Completion Times

Abdesselem Kali [1,1] and Ali Derbala [1,2]

[1] LAMDA-RO laboratory, University Saad Dahlab of Blida. BP 270, Route de Soumaa, Blida -Algeria.

[1] aeskali@gmail.com
[2] aliderbala@univ-blida.dz

Abstract: We consider the single-machine scheduling problem which has the natural character that the processing-time of the job is an increasing linear function dynamically determined by the starting time of its processing. This phenomenon is known as linear deteriorating jobs. The scheduling objective is the total completion time minimization. The computationally of the problem is still intractable and open question. We prove that the problem is NP-Complete in the ordinary sense.

Keywords: Deterministic Single-Machine Scheduling; Linear Deteriorating Jobs; Total Completion Time Minimization; Complexity.

1 Introduction

In this paper, a single-machine is used to process a set of n independent jobs. All jobs are simultaneously available with no ready times or deadlines at time 0 and can be processed without interruption. Each job i is characterized by a normal processing-time $a_i > 0$ interpreted as the length of time required to complete the job if it is scheduled first, and a parameter $b_i > 0$ depending on the job starting time S_i, interpreted as the growth rate of its processing-time. The actual processing-time p_i increases linearly with its starting times $S_i \geq 0$ and is given by $p_i(S_i) = a_i + b_i S_i$. Similar to the classical scheduling problems, a_i is assumed to be positive integers. The processing rates b_i are not integers in many practical cases, b_i is allowed to be a positive rational number. Let $y \geq 0$ denote a given threshold, the flow time problem is to decide whether there is a feasible schedule with $\sum_{i=1}^{n} C_i \leq y$. Adapting the three-field notation $\alpha|\beta|\gamma$ introduced by Graham et al. [1] used to describe the machine characteristics, job characteristics, and performance measure of interest, the above problem is denote as $1|p_i(S_i) = a_i + b_i S_i| \sum C_i$. The problem was formulated

for the first time in [2] and its time complexity is yet an open question even if $a_i = 1$ for $1 \leq i \leq n$ [3, 4].

This paper solves the complexity of the problem $1|p_i(S_i) = a_i + b_i S_i| \sum C_i$. We show that the problem is NP-complete in ordinary sense by a reduction from Subset Product Problem.

2 Computational Complexity

We proof that the problem $1|p_i(S_i) = a_i + b_i S_i| \sum C_i$, is NP-hard. The associated decision problem is the following:

SDC: Given a single-machine, n jobs, $a_i \in \mathbb{N}^*(i = 1, 2, \dots, n)$, $b_i \in \mathbb{Q}_+^*$ $(i = 1, 2, \dots, n)$ and $y \in \mathbb{R}_+$. Is there a schedule such that $\sum_{i=1}^{n} C_i \leq y$?

Our proof of NP-completeness proceeds by a reduction in polynomial time from "Subset Product" which was shown to be NP-complete in the ordinary sense (see [5] and the correction of the complexity status in Johnson [6]) into problem **SDC**.

The Subset Product Problem **SP** can be stated as follows:

SP: Given $m+1$ positive integers $x_1, x_2, \dots, x_m, B$ such that $\prod_{i=1}^{m} x_i = B^2$. Is there a subset Y of the set $X = \{x_1, x_2, \dots, x_m\}$ such that $\prod_{i \in Y} x_i = \prod_{i \in X \setminus Y} x_i = \sqrt{\prod_{i \in X} x_i} \equiv B$?

In the instances of **SP**, we can omit the element $x_i = 1$ because it will not affect the product of any subset. Therefore, without loss of generality, we can assume that $x_i \geq 2$ for every $x_i \in X$.

Our first task in proving the NP-Completeness of **SDC** is to show it lies in NP. But this is obviously true. To check if a guessed schedule has $\sum_{i=1}^{n} C_i \leq y$ is very easy and can clearly be accomplished in a number of operations bounded by a polynomial in *n*. So **SDC** lies in NP.

Our next aim is to show that in polynomial time we can turn any **SP** into an equivalent **SDC** such the former has a yes answer if and only if the latter does.

In order to show that **SP** can be reduced to **SDC,** we construct the corresponding instance of **SDC,** as follows:

- We sort respectively Y and $X \setminus Y$ in the non-decreasing order and let $(\beta_1, \beta_2, \dots, \beta_l)$ and $(\beta_{l+1}, \beta_{l+2}, \dots, \beta_m)$ denote respectively the sequences obtained for Y and $X \setminus Y$. These transformations can be done in $O(\log l(m-l))$ where $l = |Y|$.
- Given a single machine and $n = m + 1$ jobs available for processing at time zero. The parameters of l jobs constructed on the basis of the elements from the subset Y are given as:
  $a_i = \omega_i \prod_{k=1}^{i} \beta_k$ and $b_i = \beta_i - 1$ $(i = 1, 2, \dots, l)$, where the sequence (ω_i) is given as: $\omega_1 = 1 - \frac{1}{\beta_1}$ and $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$; $i = 2, \dots, l$.
- The parameters of $(m-l)$ jobs constructed on the basis of the elements from the set $X \setminus Y$, are given as:
  $a_i = \omega'_i \prod_{k=l+1}^{i} \beta_k$ and $b_i = \beta_i - 1$ $(i = l+1, \dots, m)$, where the sequence (ω'_i) is set as $\omega'_{l+1} = 1 - \frac{1}{\beta_{l+1}}$ and $\omega'_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$; $i = l+2, \dots, m$.

- And the parameters of the $(m+1)$ job, called further the extra job, are given as $a_e = \prod_{k=1}^{l-1} \beta_k$ and $b_e = \frac{1}{\prod_{k=1}^{l} \beta_k}$.
- Finally, we define a threshold $y = (3+\lambda)B + \left(1 - \frac{1}{\beta_l}\right)\lambda - 1 - \frac{1}{\beta_l}$ where $\lambda = \sum_{i=l+1}^{m} \prod_{k=l+1}^{i} \beta_k$.

The above construction can be accomplished in polynomial time, i.e., in $O(m)$.

Under the assumption that the normal processing time a_i of job i is a positive integer, jobs with indices respectively in Y and $X\backslash Y$ must be scheduled in the non-decreasing order of the deterioration rates b_i. More precisely, we have the following result:

Preliminary 1. For $i = 2, \ldots, l$ (respectively, for $i = l+2, \ldots, m$) a_i is an positive integer if, and only if, $b_i \geq b_{i-1}$.

Proof:

Since, for $i = 2, \ldots, l$, $a_i = \omega_i \prod_{k=1}^{i} \beta_k$ where $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$ and $b_i = \beta_i - 1$, it is easy matter to see that:

$$a_i = (b_i - b_{i-1}) \prod_{k=1}^{i-2} \beta_k.$$

With the fact that $\prod_{k=1}^{i-2} \beta_k > 0$, we conclude that $a_i \geq 0$ if, and only if, $b_i \geq b_{i-1}$. For $i = l+2, \ldots, m$ the proof is similar to that for $i = 2, \ldots, l$ and hence is omitted. □

Remark. In view of the condition that, $b_{l+1}, b_{l+2}, \ldots, b_m$ is in non-decreasing order, the sum $\lambda = \sum_{i=l+1}^{m} \prod_{k=l+1}^{i} (1+b_k) = \sum_{i=l+1}^{m} \prod_{k=l+1}^{i} \beta_k$ is minimized [7].

In the instance of **SDC**, if $\prod_{i \in Y} \beta_i \geq \prod_{i \in X\backslash Y} \beta_i$ and accordingly to preliminary 1, the jobs with indices in Y are processed first in the non-decreasing order of b_i, then the extra job, and then the remaining jobs in the non-decreasing order of b_i as well. Else, the jobs constructed on the basis of the elements from the set $X\backslash Y$ are performed first. Without loss of generality, we assume in what follows that $\prod_{i \in Y} \beta_i \geq \prod_{i \in X\backslash Y} \beta_i$ and hence the growth rate of the extra job is smallest as possible.

Notice that for the instance of **SDC**, the completion time of the job with index respectively in Y and $X\backslash Y$ can be, by induction with respect to i, expressed as follows:

$$C_i = \sum_{j=1}^{i} a_j \prod_{k=j+1}^{i} (1+b_k), \text{ for } 1 \leq i \leq l,$$

$$C_i = \sum_{j=l+1}^{i} a_j \prod_{k=j+1}^{i} (1+b_k) + C_e \prod_{k=l+1}^{i} (1+b_k), \text{ for } l+1 \leq i \leq n.$$

Since the job processed before the extra job is l, the completion time of the extra job can be written as:

$$C_e = a_e + (1+b_e) C_l.$$

Using the above established expressions, the completion time of each job in the constructed schedule of **SDC** is given in the next preliminary, easily proven by simple replacing a_i and b_i by their definitions in Y and $X\backslash Y$ respectively:

Preliminary 2. For the jobs constructed on the basis of the elements from the subset Y, the completion time is $C_i = \left(\prod_{k=1}^{i} \beta_k\right)\left(1 - \frac{1}{\beta_i}\right)$.
For the extra job, we obtain $C_e = \left(\prod_{k=1}^{l} \beta_k\right) + 1 - \frac{1}{\beta_l}$.
And for the jobs constructed on the basis of the elements from the set $X \backslash Y$, we obtain $C_i = \left(1 - \frac{1}{\beta_i}\right)\left(\prod_{k=l+1}^{i} \beta_k\right) + \left(1 - \frac{1}{\beta_l}\right)\left(\prod_{k=l+1}^{i} \beta_k\right) + \left(\prod_{k=1}^{i} \beta_k\right)$.

Proof:
Recall that the parameters of l jobs with indices in Y are the following:
for $i = 1, \dots, l$, $a_i = \omega_i \prod_{k=1}^{i} \beta_k$,
where $b_i = \beta_i - 1$, $\omega_1 = 1 - \frac{1}{\beta_1}$ and $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$ for $i = 2, \dots, l$.
And using the equality $C_i = \sum_{j=1}^{i} a_j \prod_{k=j+1}^{i}(1 + b_k)$, we get:

$$\begin{aligned} C_i &= \sum_{j=1}^{i}\left(\omega_j \prod_{k=1}^{j} \beta_k\right) \prod_{k=j+1}^{i} \beta_k \\ &= \sum_{j=1}^{i} \omega_j \prod_{k=1}^{i} \beta_k \\ &= \left(\prod_{k=1}^{i} \beta_k\right) \sum_{j=1}^{i} \omega_j . \end{aligned}$$

It is simple to show that $\sum_{j=1}^{i} \omega_j = 1 - \frac{1}{\beta_i}$, thus:

$$C_i = \left(1 - \frac{1}{\beta_i}\right)\left(\prod_{k=1}^{i} \beta_k\right).$$

For the extra job, we have:

$$C_e = a_e + (1 + b_e) C_l .$$

Since $a_e = \prod_{k=1}^{l-1} \beta_k$ and $b_e = \frac{1}{\prod_{k=1}^{l} \beta_k}$, then:

$$\begin{aligned} C_e &= \left(\prod_{k=1}^{l-1} \beta_k\right) + \left(1 + \frac{1}{\prod_{k=1}^{l} \beta_k}\right)\left(1 - \frac{1}{\beta_l}\right)\left(\prod_{k=1}^{l} \beta_k\right) \\ &= \left(\prod_{k=1}^{l} \beta_k\right) + 1 - \frac{1}{\beta_l} . \end{aligned}$$

Remember that the parameters of remaining $(m - l)$ with indices in $X \backslash Y$, are set as: $a_i = \omega'_i \prod_{k=l+1}^{i} \beta_k$ and $b_i = \beta_i - 1$ $(i = l + 1, \dots, m)$ where $\omega'_{l+1} = 1 - \frac{1}{\beta_{l+1}}$ and $\omega'_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$; $i = l + 2, \dots, m$.
then

$$\begin{aligned} C_i &= \sum_{j=l+1}^{i} a_j \prod_{k=j+1}^{i}(1 + b_k) + C_e \prod_{k=l+1}^{i}(1 + b_k) \\ &= \sum_{j=l+1}^{i} \omega'_j \left(\prod_{k=l+1}^{j} \beta_k\right)\left(\prod_{k=j+1}^{i} \beta_k\right) + \left[\left(\prod_{k=1}^{l} \beta_k\right) + 1 - \frac{1}{\beta_l}\right]\left(\prod_{k=l+1}^{i} \beta_k\right) \\ &= \left(\prod_{k=l+1}^{i} \beta_k\right) \sum_{j=l+1}^{i} \omega'_j + \left(\prod_{k=1}^{i} \beta_k\right) + \left(\prod_{k=l+1}^{i} \beta_k\right) 1 - \frac{1}{\beta_l}\left(\prod_{k=l+1}^{i} \beta_k\right) \end{aligned}$$

Recognizing that $\sum_{j=1}^{i} \omega'_j = 1 - \frac{1}{\beta_i}$, hence:

$$C_i = \left(1 - \frac{1}{\beta_i}\right)\left(\prod_{k=l+1}^{i} \beta_k\right) + \left(1 - \frac{1}{\beta_l}\right)\left(\prod_{k=l+1}^{i} \beta_k\right) + \left(\prod_{k=1}^{i} \beta_k\right). \square$$

Using the results established in preliminary 2, the next preliminary gives the expression of the total completion time for the constructed schedule.

Preliminary 3. The total completion time for the constructed schedule is given as:
$\sum C_i = 2\left(\prod_{k=1}^{l}\beta_k\right) + \left(\prod_{k=l+1}^{m}\beta_k\right) - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right) + \left(\prod_{k=1}^{l}\beta_k\right)\right]\sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right)$.

Proof:
The sum of completion times for constructed schedule can be expressed as follows:

$$\sum C_i = \sum_{i=1}^{l} C_i + C_e + \sum_{i=l+1}^{m} C_i.$$

Applying preliminary 2, we thus obtain:
$\sum C_i =$
$\sum_{i=1}^{l}\left(1-\frac{1}{\beta_i}\right)\left(\prod_{k=1}^{i}\beta_k\right) + \left(\prod_{k=1}^{l}\beta_k\right) + 1 - \frac{1}{\beta_l} + \sum_{i=l+1}^{m}\left[\left(1-\frac{1}{\beta_i}\right)\left(\prod_{k=l+1}^{i}\beta_k\right) + \left(1-\frac{1}{\beta_l}\right)\left(\prod_{k=l+1}^{i}\beta_k\right) + \left(\prod_{k=1}^{i}\beta_k\right)\right]$.
Since $\sum_{i=1}^{l}\left(\prod_{k=1}^{i}\beta_k\right)\left(1-\frac{1}{\beta_i}\right) = \left(\prod_{k=1}^{l}\beta_k\right) - 1$ and $\sum_{i=l+1}^{m}\left(1-\frac{1}{\beta_i}\right)\left(\prod_{k=l+1}^{i}\beta_k\right) = \left(\prod_{k=l+1}^{m}\beta_k\right) - 1$, it follows:
$\sum C_i = \left(\prod_{k=1}^{l}\beta_k\right) - 1 + \left(\prod_{k=1}^{l}\beta_k\right) + 1 - \frac{1}{\beta_l} + \left(\prod_{k=l+1}^{m}\beta_k\right) - 1 + \sum_{i=l+1}^{m}\left[\left(1-\frac{1}{\beta_l}\right)\left(\prod_{k=l+1}^{i}\beta_k\right) + \left(\prod_{k=1}^{i}\beta_k\right)\right]$
That is :
$\sum C_i =$
$2\left(\prod_{k=1}^{l}\beta_k\right) + \left(\prod_{k=l+1}^{m}\beta_k\right) - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right) + \left(\prod_{k=1}^{l}\beta_k\right)\right]\sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right)$. □

Now we shall show that answering **SP** is equivalent to answering **SDC**. Before we can do so, we show that the following two statements hold:

Lemma 1. If the given instance of **SP** has a solution then the constructed instance of **SDC** also has one.

Proof:
Let a subset *Y* such that $\prod_{i\in Y}\beta_i = \prod_{i\in X\setminus Y}\beta_i = \sqrt{\prod_{i\in X}\beta_i} = B$.
The schedule **SDC** is defined as follows: first we execute the jobs with indices in the subset *Y* in the non-decreasing order of b_i, then the extra job, and then the remaining jobs in the non-decreasing order of b_i as well, i.e., jobs with indices in the set $X\setminus Y$.
The sum of completion times has then the following value:

$$\begin{aligned}\sum C_i &= 2\left(\prod_{k=1}^{l}\beta_k\right) + \left(\prod_{k=l+1}^{m}\beta_k\right) - 1 - \frac{1}{\beta_l} + \\ &\left[\left(1-\frac{1}{\beta_l}\right) + \left(\prod_{k=1}^{l}\beta_k\right)\right]\sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right) \\ &= 3B - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right) + B\right]\lambda,\end{aligned}$$

where $\lambda = \sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right)$.
It follows that:

$$\begin{aligned}\sum C_i &= (3+\lambda)B + \left(1-\frac{1}{\beta_l}\right)\lambda - 1 - \frac{1}{\beta_l}. \\ &= y.\end{aligned}$$

Therefore, for the schedule considered we have $\sum C_i \le y$, as required.

It has been shown that the sum of completion times for **SDC** is not greater than y. Thus, we proved that if **SP** has a solution **SDC** also has one. □

Lemma 2. If the given instance of **SP** has no solution then the constructed instance of **SDC** has no solution either.

Proof:
If the instance of **SP** has no solution then the product of elements in both subsets Y and $X\backslash Y$ is no equal.
Then in view of the condition $\prod_{i\in Y}\beta_i \geq \prod_{i\in X\backslash Y}\beta_i$ made in the construction of **SDC**, we get:

$$\prod_{i\in Y}\beta_i > \prod_{i\in X\backslash Y}\beta_i .$$

Let α denote the product $\prod_{i\in Y}\beta_i$, it follows that:

$$\alpha > B,$$

and then:

$$\prod_{i\in X\backslash Y}\beta_i = \frac{B^2}{\alpha}.$$

Notice that α and $\frac{B^2}{\alpha}$ are positive integers.
We shall show now that, the sum of completion times is greater than y, i.e., $\sum C_i > y$.
We have from preliminary 3:

$$\sum C_i = 2\left(\prod_{k=1}^{l}\beta_k\right) + \left(\prod_{k=l+1}^{m}\beta_k\right) - 1 - \frac{1}{\beta_l} +$$
$$\left[\left(1-\frac{1}{\beta_l}\right) + \left(\prod_{k=1}^{l}\beta_k\right)\right]\sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right),$$

then:

$$\sum C_i = 2\alpha + \frac{B^2}{\alpha} - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right)+\alpha\right]\lambda > 3B - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right)+B\right]\lambda = y,$$

where $\lambda = \sum_{i=l+1}^{m}\left(\prod_{k=l+1}^{i}\beta_k\right)$.
The last inequality is true, since the function $f(x) = (2+\lambda)x + \frac{1}{x} - (3+\lambda)$ is strictly positive over the interval $]1\ ;\ +\infty[$:
Indeed, the inequality:

$$2\alpha + \frac{B^2}{\alpha} - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right)+\alpha\right]\lambda > 3B - 1 - \frac{1}{\beta_l} + \left[\left(1-\frac{1}{\beta_l}\right)+B\right]\lambda$$

can be written as:

$$(2+\lambda)\frac{\alpha}{B} + \frac{B}{\alpha} - (3+\lambda) > 0.$$

With a change of variable $x = \frac{\alpha}{B}$ (where $\alpha > B$), we obtain the expression and the domain of definition of the above function.
Thus, it has been shown, that if there is no solution for **SP**, then there is not one for **SDC** either. □

Lemmas 1 and 2 immediately lead to the following theorem:

Theorem. *The problem* $1|p_i(S_i) = a_i + b_i S_i|\sum C_i$ *is NP-complete in the ordinary sense.*

Conclusion

The NP-completeness of the single machine scheduling problem with deteriorating jobs for sum of completion times minimization has been shown NP-Complete in the ordinary sense. In this problem, the processing time of a job is a non-decreasing linear function dependent on the starting time of the job processing. For further research, it is suggested to investigate the model $1|p_i(S_i) = 1 + b_i S_i| \sum C_i$ and for the same objective, the non-increasing linear function model .

References

1. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics, N° 5, 287–326 (1976).
2. Mosheiov G.: V-shaped policies for scheduling deteriorating jobs, Oper. Res. 39 979–991 (1991).
3. Cheng T.C.E., Ding Q., Lin B.M.T.: A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research. N° 152 1-13 (2004).
4. Gawiejnowicz S.: Time-dependent Scheduling. Springer-Verlag Inc, New York, (2008).
5. Garey M.R., Johnson D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman: New York, (1979).
6. Johnson D.S.: The NP-completeness column: an ongoing guide, J. Algorithms N° 4 393-405 (1981).
7. Mosheiov G.: Scheduling jobs under simple linear deterioration. Computers and Operations Research. 21 (6), 653–659 (1994).

Resolution of the parallel machines scheduling problem with preemption and transportation delays

Amina Haned [1] *, Mourad Boudhar [2], and Ameur Soukhal [3]

[1] Université Dely Brahim, Faculté des sciences économiques et de gestion,
2 rue Ahmed Waked, Dely Brahim, Alger, Algérie
[2] Université USTHB, Faculté de Mathématiques,
BP 32 Bab-Ezzouar, El-Alia 16111, Alger, Algérie
[3] Université François Rabelais de Tours, Laboratoire d'Informatique,
64 avenue Jean Portalis, 37200 Tours, France
{amina_haned,mboudhar}@yahoo.fr
ameur.soukhal@univ-tours.fr

Abstract. In this study we consider the problem of scheduling a set of independent and preemptive jobs on identical parallel machines. We consider the case where one preempted job is processed on two different machines, after preemption the job is transported from one machine to another, the transportation of this job is performed by a conveyor and takes a time called transportation delay. The aim is to find a solution that satisfies all the constraints and minimize the total completion time (makespan). Some results concerning the complexity of this problem are given, also we propose a metaheuristic and heuristics to solve the problem. Experimental tests are performed to evaluate our algorithms.

Keywords: Scheduling, parallel machines, transportation delays, preemption, Ant Colony Optimization.

1 Introduction

The parallel machines scheduling problem is one of the most known problems in scheduling theory. It consists to assign jobs to the machines in order to optimize one or more objectives under several constraints. In this study, we consider the problem of scheduling a set of n independent and preemptive jobs $J = \{J_1, J_2, \ldots, J_j, \ldots, J_n\}$ on a set of m identical parallel machines $M = \{M_1, M_2, \ldots, M_i, \ldots, M_m\}$. If a job J_j processed on a machine M_i is preempted to complete its processing on another machine $M_{i'}$, then the transportation time required is $delay_{ii'}$. The transportation of the preempted jobs is performed by conveyor system. This handling system performs transportation between two machines M_i and $M_{i'}$. We assume that the conveyor is available at any moment and there are not constraints on its capacity. The aim is to find a solution that

* Corresponding author.

satisfies all the constraints and minimize the total completion time called the makespan denoted by C_{max}.

The scheduling on parallel machines is largely studied. The classical problem $Pm||C_{max}$, is ordinary NP-hard and $P||C_{max}$ is strongly NP-hard. Mc Naughton [5] has studied the problem $P|pmtn|C_{max}$ when the preemption of jobs is allowed. He has considered various objective functions for independent jobs. The problem of the makespan minimization is solved in $O(n)$ such that $C_{max} = \max\{\frac{1}{m}\sum p_j, \max\{p_j\}\}$ is an optimal value. The algorithm consists to select the jobs one by one in any order and fill up the machines $M_1, \ldots, M_m$ within the time interval $[0, C_{max}]$. If on a current machine M_i, C_{max} is reached and the job J_j is not yet completed then this job will be preempted and complets its processing at the beginning of the next machine M_{i+1}. For the scheduling problems with setup times (the time needed to prepare the machines or the jobs before starting processing), a classification has been given by Allahverdi et al. in [1] and applications have been cited. When the semi-finished jobs are transported from one machine to another for further processing, the transportation delays are considered. Several studies have been performed as Lee and Chen [12] have studied the flow-shop problem. Jansen, Mastrolilli, Soli-Oba [11] have treated the job-shop scheduling problem. For the case with parallel machines, Rayward-Smith [13], Fishkin et al. [8], Boudhar and Haned [2] have studied the problem where preemption and transportation of preempted jobs are considered.

The meaning of the transportation delays in this study is the required time to transport one preempted job from one machine to another. It has a different meaning in the case of the problem studied by Giroudeau et al. [3] and [4], where they consider the problem with a set of precedence constraints and a communication delay between two different jobs submitted to a precedence constraint and processed by two different machines.

The considered problem gets closer to the pick up and delivery problem with time windows $PDPTW$ which is a generalization of the Vehicle Routing Problem VRP. Over the VRP problem is a generalization of the traveling salesman problem TSP. Since the TSP is NP-hard [16], the $PDPTW$ is NP-hard too. Many researches have focused on solving PDP problem. In [15], a classification of these problems has been given. A state-of-the art has been done by Snezana Mitrovic-Mini [16].

The rest of this paper is organized as fellow: in the next section, we present some results concerning the complexity of the scheduling problem defined above. After that, resolution method are proposed, numerical tests are performed and comparative analysis between heuristics is given. Finally a conclusion summarizes the main results.

2 Complexity of the scheduling problem

The problem defined previously is denoted by $Pm|pmtn(delay_{ii'})|C_{max}$ when the number of the machines m is fixed and by $P|pmtn(delay_{ii'})|C_{max}$ in the other case. This problem has been studied by Rayward-Smith [13], he shown

that the problem $P|pmtn(delay_{ii'} \geq 2)|C_{max}$ is NP-hard. Fishkin et al. [8] have studied the problem $P|pmtn(delay_{ii'} = d)|C_{max}$ with identical transportation delays $delay_{ii'} = d$, they have given some results concerning the complexity and have proposed a polynomial time approximation scheme ($PTAS$). In [2] some sub-problems are discussed, mathematical model and heuristics are proposed for the problem $P|pmtn(delay_{ii'} = d)|C_{max}$. Recently in [14], a dynamic programming algorithm and a fully polynomial time approximation scheme have been proposed for the problem with two and three machines $P2|pmtn(delay_{ii'})|C_{max}$, $P3|pmtn(delay_{ii'})|C_{max}$ respectively.

2.1 The problem $Pm|pmtn(delay_{ii'} = d), p_j = p|C_{max}$

Consider the particular case of the problem with fixed transportation delays ($delay_{ii'} = d, \quad \forall i \neq i'$) and equal size jobs ($p_j = p, \quad \forall j$): $Pm|pmtn(delay_{ii'} = d), p_j = p|C_{max}$. This problem is solved in $O(n)$ by algorithm $A1$ proposed in [14]. The algorithm $A1$ considers two cases, when the jobs number n is a multiple of the machines number m, then the schedule is obtained by processing the jobs one by one without preemption, each machine process n/m jobs. When n is not a multiple of m, we apply the Mc Naughton's algorithm by taking into account the transportation delays.
By adding the conveyor constraint with capacity $c \geq 1$, we can give the following results:
Result 1. The problem $Pm|pmtn(delay_{ii'} = d), p_j = p|C_{max}$ with a conveyor of capacity $c \geq 1$ is polynomial.
Proof Since the transportation delays are fixed, any path through all machines has a duration $(m-1)d$. The conveyor visits machines to pick up and deliver one preempted job at a time. So we apply the algorithm $A1$ with any sequence of the machines.

2.2 The problem $Pm|pmtn(delay_{ii'}), p_j = p|C_{max}$

Now, we assume that only the processing times of the jobs are identical. According to Proposition 2 [14], the problem is solved in $O(m! \times n)$ by algorithm $A1$. In fact, we have $m!$ permutations for storing machines. We have to choose the best permutation. Some modifications are made to the algorithm $A1$. Below the new version of this algorithm.
Algorithm $\mathcal{A}'1$
begin

if n is a multiple of m **then**
each machine processes $\frac{n}{m}$ jobs. $C_{max} = \frac{n}{m}p$
else if $\max_{i \neq i'} \{delay_{ii'}\} \leq (\frac{n}{m} - 1)p$ **then**
$C_{max} = \frac{n}{m}p$, use the Mc Naughton's algorithm to find the solution
else if$(\frac{n}{m} - 1)p < \min_{i \neq i'} \{delay_{ii'}\} < (\lceil \frac{n}{m} \rceil - 1)p$ **then**
use the Mc Naughton's algorithm and make a shift.

else $C_{max} = \lceil \frac{n}{m} \rceil p$, the preemption is not interesting.
end if

end if

end if

end.

We also propose the following heuristic H_TR which gives the sequence of machines. The main idea is to look for the best path that starts with the machine M_i. We obtain m paths of length D_i. We choose the one satisfying $D_t = \min\limits_{1 \leq i \leq m} \{D_i\}$. Befor de description of this heuristic, we give some notations: Let M be the set of machines, $|M| = m$, S is the set of machines in the path. We define L the length of the path.
Heuristic H_TR
Begin
for i:=1 to m **do**

initialization $S := \{\}$; $S := S \cup \{i\}$; $M := M/\{i\}$; $L := 0$;
repeat
- find the closest machine to the machine M_i, let M_k be this machine ;
- $S := S \cup \{k\}$; $M := M/\{k\}$;
- $L := L + delay_{ik}$; $i := k$;

until $M = \{\}$;
$D[i] := L$;
endfor;
find $\min\limits_{1 \leq i \leq m} \{D[i]\}$;

End.
This heuristic provides the shortest path in $O(m^2)$. The machines are token in the inverse order of the path.

3 Resolution method

In this section, we present the resolution methods to solve the general problem $Pm|pmtn(delay_{ii'})|C_{max}$ with arbitrary transportation delays and arbitrary processing times. Our strategy has two steps: in the first step we determine the sequence of machines, in the second step the assignment of jobs to the machines. The first step can be reduced to the hamiltonian path problem. There exist several resolution method for this problem, among them metaheuristics. In the next, we adopt the Ants Colony Optimization (ACO) technique to arrange the machines. In the second step, Jobs are assigned according to the heuristics described in the next.

3.1 Ant colony optimization

The ant colony algorithms are a family of mataheuristics inspired by insect behavior. They were introduced by Marco Dorigo and his colleagues in the 90s. The first algorithm, called Ant System (AS) [17], was designed to solve the traveling salesman problem TSP. The algorithm's parameters are: the pheromone trail matrix Γ. Initially all elements of the matrix Γ are equal to τ_0. The maximum number of iterations NI_{max} can decide whether to continue the implementation of the algorithm, the numbers of ants. Two positive parameters α and β, their values determine the relation between pheromone information and heuristic information. Dorigo et al. conclude that α should be around 1 and β is between 2 and 5 for the tested traveling salesman problem [18]. The pheromone trail decay coefficient ρ varies in $[0, 1]$.The algorithm has the following steps:

1. Initialize the pheromone trail of all arcs to τ_0;
   Each ant k puts his starting city (node) in its memory list to store the already visited nodes. This list is denoted by $tabou_k$
2. Each ant k selects the next city to visit according to a transition probability:
   $P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{w \notin tabou_k} [\tau_{iw}]^\alpha [\eta_{iw}]^\beta}, if \;\; j \notin tabou_k; \\ 0 \qquad\qquad\qquad\qquad otherwise. \end{cases}$ [18]
   Recall that τ_{ij} is the amount of pheromone trail on the arc (i, j) and η_{ij} is the visibility that is inversely proportional to the distance between two cities (nodes) i and j, d_{ij}. α, β control the issue of the relative intensity of the pheromone trail and visibility. If the city is chosen, it will be added to the memory list of the ant k. This list contains, in order, the cities visited by the ant k.
3. When all ants finish the construction of their tours, they deposit on each arc (i, j) of the path a quantity of pheromone denoted $\Delta\tau_{ij}$ calculated by the formula: $\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L_k}, if \;\; the \;\; arc \;\; (i,j) \;\; is \;\; in \;\; the \;\; tour \;\; of \;\; the \;\; ant \;\; k; \\ 0, \quad otherwise. \end{cases}$
   L_k is the length of the tour done by the ant k
4. At the end of each iteration, the pheromone trails of the ants are updated by: $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)$, $\rho\tau_{ij}(t)$ is the quantity of pheromone evaporation.

For our problem we use the ant colony algorithm to find the sequence of machines, the distance matrix is replaced by the transportation delays matrix. The algorithm provide us the best path, we deduce the sequence of machines by taking the inverse order of the path.

3.2 Scheduling heuristics

In the following, we describe the heuristics to assign jobs to the machines. The first heuristic consists to assign jobs, ordered by Longest Processing Time (LPT) rule, to one selected machine until the lower bound. If we can not assign entirely the job J_j, then it will be preempted. We verify if the transportation delay is

respected, then the job J_j will be split into two parts and processed at the last position on the machine M_i' and the first position on the machine M_i. In the other case, we search a job that can be entirely processed before the calculated makespan. If this job does not exist,we search another job for which the transportation delay is respected. In the case where does not exist a job which satisfies one of the two previous conditions, we have to make a shift. The following algorithm gives the different steps of this heuristic denoted Ass-job.

Algorithm Ass-job;

Begin

- Arrange the jobs by the LPT rule, the obtained list is denoted NS;
- Calculate the lower bound $LB = \max\left\{\max_{1\leq j\leq n}\{p_j\}; \frac{1}{m}\sum_{j=1}^{n} p_j\right\}$;
- Initialization: the starting time $t = 0$;

  While $(NS \neq \emptyset)$ **do**

  (1) **If**$(t + p_j < LB)$**then**

  - process the job J_j on the machine M_i at time t;
  - $NS := NS - \{J_j\}$; $t := t + p_j$; $j := j + 1$;// the next job of the list

  (2) **Else If**$(t + p_j = LB)$**then**

  - process the job J_j on the machine M_i at time t and the next job on the next machine at $t = 0$;
  - $NS := NS - \{J_j\}$; $t := 0$; $j := j + 1$; $i := i + 1$;// the next machine

  Else

  (3) **If** $(p_j + delay_{ii+1} \leq LB)$ **then**

  - the job J_j is split in two parts, it is processed at the last position on the machine M_{i-1} for $(LB - t)$ units of time and at the first position on the machine M_i;
  - $NS := NS - \{J_j\}$;

  Else

  - find the job J_{ind} such that $(t + p_{ind} \leq LB)$ and go to (1) or (2);
  - if this job does not exist, find the job J_{ind} such that $(p_{ind} + delay_{ii+1} \leq LB)$ and go to (3);
    otherwise, the job J_j will be preempted and make a shift to satisfy the transportation delay.

  endwhile

end.

This heuristic provides a solution in $O(nlogn)$ time and it minimizes the number of preempted jobs.

Example: We consider the problem $Pm|pmtn, (delay_{ii'})|C_{max}$, with $n = 10$ jobs and $m = 5$ machines. Processing times and transportation delays are given in the following tables.

Table 1. Processing times

J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
p_j	5	6	1	6	5	4	5	6	6	2

Table 2. Transportation delays

	M_1	M_2	M_3	M_4	M_5
M_1	0	24	1	29	5
M_2	29	0	10	3	14
M_3	5	9	0	16	9
M_4	17	3	18	0	18
M_5	18	21	3	12	0

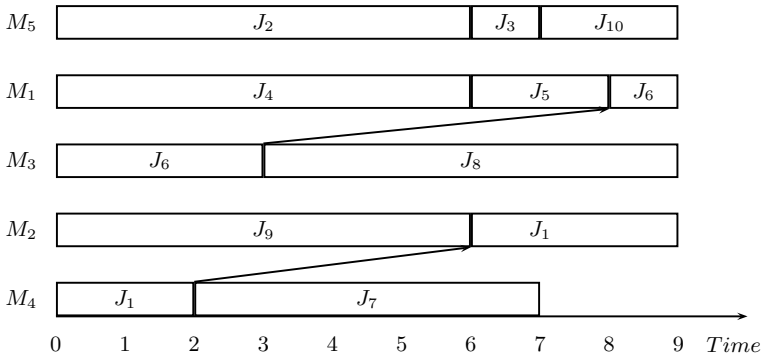


Figure 1. Solution given by $Ass-job$ heuristic

The sequence of machines found by the application of the ants algorithm is: $M_4 \overset{3}{\rightarrow} M_2 \overset{10}{\rightarrow} M_3 \overset{5}{\rightarrow} M_1 \overset{5}{\rightarrow} M_5$. So, the transportation is from the machine M_4 to the machine M_2 and from machine M_2 to the machine M_3 and so on. If there is a preempted job, it starts its processing on machine M_4 and finishes at the last position on machine M_2.
The solution given by the heuristic $Ass-job$ is represented by the Gantt chart of the figure 1. The preempted jobs are J_1 and J_6, we can see that the job J_1 is transported from the machine M_4 to the machine M_2. The J_6 is transported from the machine M_3 to the machine M_1.

In the following, we define a new heuristic named H_List based on the list of jobs. The first list, obtained by heuristic $Ass-job$, has at most $(m-1)$ preempted jobs where m is the number of machines. The new lists are obtained by permuting one preempted job of the first list with a no preempted job chosen at random manner. We obtain at most $(m-1)$ lists. After that we assign the jobs of the lists one by one to a selected machine till the lower bound LB. If one job

is not completed before LB, it will be preempted and transported. At the end we have at most $(m-1)$ solutions, we choose the best one. This heuristic has two procedures, the $Permut-list$ procedure used to create the different lists and $Ass-list$ procedure used to assign the jobs.

Procedure $Permut-list$
Begin

Determine the number of the preempted jobs (denoted nb);
For l:=1 to nb **do**
- Select randomly a no preempted job, let ind be this job;
- Permute the job ind with the l^{th} preempted job;
- Save the new list;

endfor;

End.
Procedure $Ass-list$
Begin

For l:=1 to nb **do**
Initialization : $t=0$; $i=1$; $j=1$; $C_{max}=LB$;
For each job i of the list **do**
If $(t+p_i < C_{max})$ **then**
Assign the job i to the machine j at t;
$t := t+p_i$;
else If $(t+p_i = C_{max})$ **then**
Assign the job i to the machine j at t, the next job will be assigned to the next machine;
else the job i will be preempted and transported,
endif;
endif;
endfor;
endfor;
Choose the best solution;

End.

Another heuristic $H2\_mod$ based on the first available machines algorithm that consists to assign jobs arranged by LPT rule to the first available machine. The heuristic gives the best solution between the solution given by the heuristic $H2V2$ with sequence of machines obtained by the heuristic H_TR and the solution given by the first available machines algorithm.

4 Numerical experiments

To evaluate the proposed heuristics, several instances are randomly generated according to the uniform law and by using different combinations of jobs and machines. For each instance the number n of jobs is in $\{10, 20, 50, 100, 200\}$,

the number m of machines is in $\{5, 10, 15, 20\}$. Processing times of the jobs are in the interval $[1, 30]$, the transportation delays will be taken in the intervals $[1, 50], [1, 100]$. To find the sequence of machines, we apply the ant colony algorithm with the following parameters:

- The initial pheromone trail $\tau_0 = 0.5$.
- The maximum number of iterations $NI_{max} = 10$.
- The numbers of ants equals the number of machines.
- The two parameters α and β are equal to 1, and 5 respectively.
- The pheromone trail decay coefficient ρ is fixed to 0.5.

Our algorithms are coded in Delphi (Version 7.0) and have run them on a Pentium IV 2.0 GHz Personal Computer with 2 Go RAM under professional Windows XP. For each value of m and n, 100 instances are generated. We calculate the average deviation of the found solutions respect to lower bound, also we calculate the average execution time (in seconds). We consider a comparative analysis between the proposed heuristic $Ass-job$ and heuristic $H2V2$ proposed in [2].

The obtained results are given in the following tables. The first and the second columns indicate the number of machines and the number of jobs respectively. The third column shows the average execution time of the metaheuristic combined with the heuristic $Ass-job$ and the fourth column gives the average deviation from the lower bound. The next columns give the average execution time and the average deviation of the heuristic $H2V2$, $H2\_mod$ and H_List respectively. In table table 3, we can see the results of instances with delays in $[1, 50]$ and in table 4 these with delay in $[1, 100]$.

Table 3. Delays in $[1, 50]$

m	n	$Ass-job$		$H2V2$		$H2\_mod$		H_List	
		$T_{Ass-job}$	$R_{Ass-job}\%$	T_{H2V2}	$R_{H2V2}\%$	T_{H2_mod}	$R_{H2\_mod}\%$	T_{H_List}	$R_{H\_List}\%$
5	10	<0.001	17.09	<0.001	34.74	<0.001	6.68	<0.001	3.70
	20	<0.001	0.52	<0.001	1.57	<0.001	2.31	0.0011	<0.01
	50	0.0015	<0.01	<0.001	<0.01	<0.001	0.22	0.002	<0.01
10	20	0.007	19.53	<0.001	44.06	<0.001	5.5	0.0087	4.09
	50	0.008	<0.01	<0.001	<0.01	<0.001	1.51	0.0157	<0.01
	100	0.0081	<0.01	<0.001	<0.01	<0.001	0.307	0.0171	<0.01
15	50	0.032	<0.01	<0.001	4.49	<0.001	3.21	0.041	<0.01
	100	0.0327	<0.01	<0.001	<0.01	<0.001	0.69	0.047	<0.01
	200	0.033	<0.01	0.0013	<0.01	0.0018	0.066	0.062	<0.01
20	50	0.096	1.48	<0.001	21.70	<0.001	5.48	0.11	0.28
	100	0.0965	<0.01	<0.001	0.0126	<0.001	1.64	0.012	<0.01
	200	0.0966	<0.01	<0.001	<0.01	0.0015	0.27	0.138	<0.01

These tests lead us to the following conclusions:
The execution time of the heuristic $Ass-job$ and H_List is greater than the

Table 4. Delays in $[1, 100]$

m	n	$Ass-job$		$H2V2$		$H2\_mod$		H_List	
		$T_{Ass-job}$	$R_{Ass-job}\%$	T_{H2V2}	$R_{H2V2}\%$	T_{H2_mod}	$R_{H2\_mod}\%$	T_{H_List}	$R_{H\_List}\%$
5	10	0.0012	71.45	<0.001	84.81	<0.001	6.84	0.0015	36.31
	20	<0.001	12.54	<0.001	20.49	<0.001	2.58	0.0012	3.17
	50	<0.001	<0.01	<0.001	<0.01	<0.001	0.26	0.0067	<0.01
10	20	0.0075	76.97	<0.001	95.50	0.0011	6.97	0.011	33.062
	50	0.0102	0.78	<0.001	2.96	0.0012	1.6	0.025	<0.01
	100	0.0088	<0.01	<0.001	<0.01	<0.001	0.31	0.037	<0.01
15	50	0.032	21.28	<0.001	35.20	<0.001	3.20	0.0576	1.96
	100	0.033	<0.01	0.0015	0.45	0.0021	0.85	0.089	0.0895
	200	0.033	<0.01	0.0024	<0.01	0.0023	0.052	0.122	<0.01
20	50	0.098	49.60	0.0014	72.11	<0.001	5.73	0.147	13.59
	100	0.098	<0.01	<0.001	3.019	0.00125	1.48	0.178	<0.01
	200	0.098	<0.01	0.00157	<0.01	0.0023	0.26	0.208	<0.01

execution time of the heuristic $H2V2$ and $H2\_mod$, since the two heuristics $Ass-job$ and H_List use the metaheuristic in their first step. Also we denote that the execution times is acceptable, it less than 1 second in almost all instances with delays in $[1, 50]$ and $[1, 100]$. We can see that the average execution time increases with the number of machines and jobs.
The average deviation decreases for the large size instances. When the delays are in $[1, 50]$, for the instances with 10 jobs the average deviation is about 17.09% and 34.74% for the heuristic $Ass-job$ and $H2V2$ respectively. Concerning the two heuristics $H2\_mod$ and H_List, their average deviation is better, it about 6.68% and 3.70% respectively. For the instances with 20 and 50 jobs, the average deviation becomes smaller. This can be explained by the fact that the large number of jobs increases the gap between parts of the preempted job then the transportation delays (in $[1, 50]$) are respected and the solution is near to the lower bound. This result coincides with the theorem 1 [8].
The worst cases are obtained for the instances with transportation delays in $[1, 100]$, the solutions are far from the lower bound because the delays are large and a shift is made in some cases. For these instances, we can see that the average deviation of the heuristic $H2\_mod$ is acceptable comparing with the other heuristics. It is around 7% for the instances with 5 machines, 10 jobs and 10 machines, 20 jobs. This deviation decreases for the large size instances.

5 Conclusion

In this study, we consider the scheduling problem on parallel machines with preemption and transportation delays. Our objective is to find a solution satisfying all constraints and minimizes the makespan. We have presented some related works and some results concerning the complexity of the problem. We have proposed an ant colony algorithm to find the best sequence of the machines.

The application of the ant colony algorithm will be more interesting when the number of machines is very large. Also, another heuristic to find the sequence of machines has been proposed. To assign jobs, we have developed some heuristics. A numerical experiments have been performed for several instances with different sizes. The results lead us that the combined metaheuristic and heuristic produces the good solutions for the large size instances.

For the future work, we can improve the proposed heuristics and propose another resolution method by using the known exact method to solve the hamitonian path problem which gives the best sequence of machines. Also we propose to study the problem by including another constraints as the release date of jobs and consider other objective.

References

1. A. Allahverdi, C. T. Ng, T. C. E. Cheng, M. Y. Kovalyov, A survey of scheduling problems with setup times or costs, European Journal of Operational Research, 187 : 985-1032, 2008.
2. M. Boudhar, A. Haned, "Preemptive scheduling in the presence of transportation times", Computers & Operations Research vol.36, n8: 2387–2393, 2009.
3. R. Giroudeau, J-C Konig, F. K. Moulai, J. Palaysi, Complexity and approximation for precedence constrained scheduling problems with large communication delays,Theoretical Computer Science, 401: 107119, 2008.
4. P. Chrtienne, E. G. Coffman,Jr, J. K. Lenstra and Z. Liu, Scheduling theory and its applications, John Wiley & sons, 1995.
5. Mc Naughton R, Scheduling with deadline and loss functions, Management Science, Vol. 6: 1-12, 1959.
6. P. Brucker, Scheduling algorithms, 5^{th} edition Springer, 2007.
7. S. Dunstall, A. Wirth, Heuristic methode for the identical parallel machine flowtime problem with set-up times, Computer & Operations Research, 32 : 2479-2491, 2005.
8. A. Fishkin, K. Jansen, S. Sevastyanov, R. Sitters, Preemptive scheduling of independent jobs on identical parallel machines subject to migration delays, Lecture Notes of Computer Science, No. 3669 : 580–591, 2005.
9. MR. Garey, DS. Johnson, Computers and intractability: a guide to the theory of NP-Completeness, W.H. Freeman and Company: San Francisco, 1979.
10. H. Hoogeveen, G. J. Woeginger, Avery difficult scheduling problrm with communication delay, Operations Research Letters, 29 : 241- 245, 2001.
11. K. Jansen, M. Mastrolilli, R. Solis-Oba, Approximation algorithms for flexible job shop problems, International journal of foundations of computer science, vol. 16, No. 2 : 361- 379, 2005.
12. C. Lee, Z. Chen, Machine scheduling with transportation considerations, Journal of scheduling, 4 : 3- 24, 2001.
13. V. J. Rayward-Smith, The complexity of preemptive scheduling given interprocessor communication delays, Information Processing Letters, 25: 123-125, 1987.
14. A. Haned, M. Boudhar, A. Soukhal and N. Huynh Tuong, Premptive Sheduling with jobs transportaion, in the proceding of PMS'10, The 12^{th} International Conference devoted to Project Management and Scheduling, 2010.
15. G. Berbeglia, J. F. Cordeau, I. Gribkovskaia , G. Laporte , Static pickup and delivery problems: a classification scheme and survey, Top 15: 1-31, 2007.

16. M.M. Snezana, Pickup and Delivery Problem with Time Windows: A Survey, SFU CMPT TR 12, May 1998.
17. M. Dorigo, G. Dicaro, L. M. Gambardella : Ant algorithm for discret optimization, Artificial life, 5:137-172, 1999.
18. S. Morin, Algorithme de fourmis avec apprentissage et comportementspécialisés pour l'ordonnancement de voitures, mémoire de maîtrise en informatique, université du Québec, 2005.

Contribution à l'Ordonnancement Réactif Modélisation Booléenne

Yamina DEDDOUCHE[1], Baghdad ATMANI[1], Nassima AISSANI

[1] Equipe de recherche Simulation, Intégration et Fouille de données "SIF"
Laboratoire d'informatique d'Oran "LIO"
Département d'informatique, Faculté des sciences, Université d'Oran
BP 1524, EL M'Naouer, Es Senia, 31000 Oran, Algérie
yamina.deddouche@gmail.com, atmani.baghdad@gmail.com,
aissani.nassima@univ-oran.dz

Résumé. Dans cet article nous nous intéressons à l'ordonnancement réactif dans les ateliers de production. L'ordonnancement réactif n'est basé sur aucun ordonnancement préventif, il s'agit plutôt de faire des allocations des ressources aux tâches en temps-réel. Lorsqu'une ressource se libère, une tache parmi les taches qui se trouvent dans sa file d'attente est choisie pour être exécutée, ce choix se fait selon des règles de priorité. Il s'agit alors des approches souvent basées sur l'intelligence artificielle pour développer une politique d'action à chaque fois qu'un problème d'allocation se pose. Ce papier présente notre contribution qui concerne la modélisation booléenne du processus d'ordonnancement réactif. Le but, après une modélisation booléenne, est double: d'une part proposer une méthode cellulaire d'extraction des règles de priorité guidée par fouille de données, et, d'autre part réduire la complexité de stockage, ainsi que le temps de calcul. Seule la modélisation booléenne de l'ordonnancement réactif est décrite dans ce papier.

Mots-clés: système de pilotage, système de production, ordonnancement réactif, apprentissage automatique, machine cellulaire, modélisation booléenne.

1 Introduction

Les tendances actuelles dans le domaine de la production industrielle indiquent que l'un des rôles les plus importants du pilotage des systèmes de production consiste à réagir efficacement aux perturbations afin de maintenir les objectifs de performance assignés. Les perturbations sont des événements dont l'occurrence n'est pas planifiée, et sont susceptibles de gêner le déroulement d'opérations de production et dans certains cas, de remettre en cause l'objectif même de production [03]. Ces perturbations peuvent être internes survenant dans l'atelier (pannes de ressources, absence de personnel, etc.) ou externes provenant de son environnement (retard d'approvisionnement, arrivée imprévue d'un ordre de fabrication, etc.).

L'ordonnancement est un problème d'optimisation d'un certain objectif, tel que la durée totale de production en allouant des ressources aux tâches. Lorsque cet ordonnancement doit prendre en considération des événements incertains pour proposer des solutions, on fait alors face à un ordonnancement dynamique. L'ordonnancement dynamique est constitué principalement de trois types: l'ordonnancement proactif, l'ordonnancement réactif et l'ordonnancement hybride [01]. Dans cet article nous nous intéressons à l'ordonnancement réactif. L'ordonnancement réactif n'est basé sur aucun ordonnancement préventif, il s'agit plutôt de faire des allocations des ressources aux tâches en temps-réel. Ce papier est composé de 07 parties, la première est dédiée à une introduction la deuxième à une définition de l'ordonnancement, La partie suivante présente un état de l'art sur les travaux réalisés dans le domaine d'ordonnancement réactif, ensuite une partie qui expose la méthode de [50][51] pour la génération des règles d'ordonnancement par fouille de données, la sixième partie présente notre modélisation adoptée pour résoudre ce type de problème, enfin, la dernière partie donne une conclusion et quelques perspectives.

2 Définition de l'Ordonnancement

Le problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles et des contraintes portant sur l'utilisation et la disponibilité des ressources requises [18]. D'après la définition de l'ordonnancement citée ci-dessus, nous en déduisons qu'un ordonnancement est constitué principalement de quatre éléments suivants : Les tâches, les ressources, les contraintes et les objectifs ou les critères d'optimisation. Notions que nous définissons dans ce qui suit :

2.1 Les tâches [19]

Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début et une date de fin, dont la réalisation est caractérisée par une durée et par l'intensité avec laquelle elle consomme certains moyens *k*, ou ressources [54], [21].

2.2 Les ressources [19]

Une ressource *k* est un moyen technique ou humain requis pour la réalisation d'une tâche et disponible en quantité limitée, sa capacité est supposée constante. Une ressource peut être renouvelable, c'est à dire qu'elle peut être utilisée et qu'une fois la tâche terminée, elle est à nouveau disponible. Mais elle peut aussi ne pas l'être, on parle alors de ressource consommable. Si une ressource ne peut exécuter qu'une seule tâche à la fois elle est dite disjonctive (ou non partageable). Dans le cas où une ressource pourrait être utilisée dans le traitement de plusieurs tâches simultanément, comme dans le cas où plusieurs ressources sont utilisées pour la même tâche, on parle de ressource cumulative (ou partageable).

2.3 Les Contraintes

Les contraintes expriment des restrictions sur les valeurs que peuvent prendre certaines variables [19], [07]. On distingue deux types de contraintes, les contraintes temporelles et les contraintes de ressources:

1. Les contraintes temporelles intégrant;
   (a) les contraintes de temps alloué, issues généralement d'impératifs de gestion et relatives aux dates limites des tâches (délais de livraison, disponibilité des approvisionnements) ou à la durée totale d'un projet,
   (b) les contraintes d'antériorité qui décrivent le positionnement relatif de certaines tâches par rapport à d'autres,
2. Les contraintes de ressources, traduisent la disponibilité des ressources et le fait qu'elles soient en quantité limitée.

2.4 Les Objectifs

Lors de la résolution d'un problème d'ordonnancement, on vise à optimiser un certain critère, à le minimiser ou le maximiser, exemple minimiser les délais d'attente, maximiser la rentabilité des ressources [17].

Dans ce qui suit, nous donnerons un aperçu sur les différentes typologies d'ordonnancement des systèmes de production, en nous intéressant plus particulièrement à l'ordonnancement réactif. Un état de l'art sur des travaux qui sont intéressés à l'ordonnancement réactif sera présenté dans la quatrième section.

3 Différentes approches d'ordonnancement

Selon la littérature, il existe deux types d'ordonnancement : ordonnancement prévisionnel (hors- ligne) et celui dynamique (en- ligne)(pour plus de details consulter Aissani 2010) :

3.1 Ordonnancement prévisionnel

Un ordonnancement hors-ligne «off-line» est effectué avant le lancement du système. Ceci implique que tous les paramètres des tâches soient connus a priori, et notamment les dates d'activation. Cet ordonnancement permet une meilleure prédiction de la satisfaction ou non des contraintes temporelles. De même, la puissance de calcul disponible hors-ligne permet de calculer un ordonnancement optimal.

3.2 Ordonnancement Dynamique

Un ordonnancement dynamique « on-line » permet de résoudre le problème d'allocation au fur et à mesure que le système de production est opérationnel. L'ordonnancement dynamique prend en considération les événements imprévus, les données incertaines et l'incomplétude de l'information. Une classification d'ordonnancement dynamique a été proposée dans [15], et reprise dans [07], [01] et [32], elle constitue une référence souvent utilisée dans le domaine de l'ordonnance-

ment. Cette classification distingue trois types d'approches : les approches proactives, les approches réactives et les approches hybrides [31].

L'approche réactive ou autrement dis l'ordonnancement réactif produit un ensemble de décisions, en réaction à des événements incertains [32]. L'avantage de cette réaction est la flexibilité, et l'adaptabilité à l'environnement que procure le calcul en ligne. Par contre, ce calcul devant être exécuté par le système en temps réel, doit être le plus simple et le plus rapide possible, ce qui permet notre modélisation booléenne en proposant une approche de résolution qui soit réactive et adaptative aux évènements internes et/ou externes du système. L'ordonnancement réactif ainsi qu'un état de l'art sur les travaux réalisés dans ce domaine, sera présenté de manière plus détaillée dans la section suivante.

4 Ordonnancement réactif: état de l'art

La problématique de l'ordonnancement réactif n'a attiré que récemment l'attention des chercheurs. Dans un ordonnancement réactif, il est plutôt question d'établir une politique d'allocation [41]. Le principe est que lorsqu'une machine devient disponible et qu'il existe plusieurs opérations en attente d'exécution sur cette machine, on doit choisir l'opération à effectuer selon une règle dite de priorité [34]. Parmi les règles de priorités les plus citées dans la littérature : FIFO[1], SPT[2], LPT[3], etc.

Les règles de priorités ont fait l'objet d'un grand nombre de travaux d'ordonnancement réactif parce qu'elles sont relativement simples à mettre en œuvre. Les techniques de choix des règles de priorité sont présentées dans les sous sections suivantes inspirées de [01] :

4.1 Règles de priorité et Simulation

Dans le premier cas, l'idée consiste à simuler l'application d'un ensemble de règles au moment d'une prise de décision, en se projetant éventuellement sur le futur proche, puis de sélectionner celle fournissant la meilleure performance, comme les travaux de [26] ,[25],[27] et récemment les travaux de [40], [12] et [30]. Mirdamadi, lui aussi a proposé l'utilisation d'un outil de simulation enligne pendant la phase d'exploitation pour identifier des événements critiques par rapport au planning prévu et l'anticipation des perturbations [33]. Dans le même contexte un travail réalisé par [55] ou les règles de priorités sont utilisées comme solution pour la gestion de la file d'attente par une simulation sur un modèle job shop, une classification a été faite selon différents objectifs pour déterminer si une règle de priorité peut être très performante sous certaines conditions de fonctionnement et très mauvaise sous d'autres conditions. Une autre étude [02] qui permettra de recenser les classements des règles par ordre d'efficacité pour les critères les plus capitaux à l'évaluation de la performance d'un FMS.

[1] Le premier arrive est le premier servis (First Come First Serve)
[2] La plus petite d'abord (Shortest Processing Time)
[3] La plus longue d'abord (Longest Processing Time

4.2 Sélection des règles de priorités en utilisant une base de connaissances

Dans le deuxième cas, des techniques issues de l'Intelligence Artificielle sont utilisées [29]. Le but ici est de construire un modèle de connaissances en hors-ligne, c'est-à-dire dans une phase d'apprentissage qui pourra être utilisé par la suite en enligne dans une phase d'exploitation pour déterminer à tout instant la règle de priorité à utiliser. Dans [39], Un processus de Cinq phases a été proposé pour la construction d'un modèle à base de connaissances: la définition des paramètres de contrôle, génération des exemples d'apprentissage, et la définition des conditions d'activation des règles. Une autre méthode à base de connaissances est présentée dans [11], leur étude est basée sur la détermination des paramètres pertinents pour la sélection des règles, en utilisant une approche neuronale.

Toujours dans le domaine d'intelligence artificielle les systèmes experts en particulier sont utilisés pour améliorer les résultats [16]. Le système expert utilise une base de connaissances composée de règles d'ordonnancement, obtenues à partir de l'expérience accumulée par certains individus clés [52]. Plusieurs projets ont été menés pour développer de tels systèmes : ESPRIT 418 [35], ESPRIT 809 [53] et plusieurs prototypes ont été développés : OPAL/ORIGAN [06], OPIS [48]. On peut aussi classer dans cette catégorie les différents travaux dont le rôle du système expert est la sélection automatique des meilleures règles de priorité en fonction des paramètres de l'atelier considéré, par exemple dans [08] « ordonnancement job shop », [37] « sélection dynamique », [24] « règles floues » ou [49] « sélection dynamique par Analytic Hierarchy Process » [44].

D'autres auteurs ont proposé de mettre en place les deux types d'apprentissage à savoir l'apprentissage déductif [28], comme les systèmes expert basés sur la connaissance experte, traduite en règles qui appliquées par un moteur d'inférence permettent un raisonnement déductif [23], et l'apprentissage inductif qui extrait la connaissance à partir des données simulées [10]. [45] ont proposé une modélisation par un Processus Décisionnel Markovien (MDP) au problème d'ordonnancement avec une modélisation stochastique des incertitudes de l'environnement de production.

4.3 Sélection des règles de priorités en utilisant des techniques distribuées

D'autres approches se sont basées sur les systèmes distribués de prise de décision en utilisant les systèmes multi agents, les systèmes bioniques, ou encore les systèmes holoniques [37, 39, 43,47, 48]. Ces approches ont essayé de développer des systèmes intelligents pour le contrôle réactif de processus industriels de plus en plus critiqués comme insuffisamment flexibles [47]. Dans [43], le système est constitué d'un ensemble d'agents autonomes dont chacun essaie d'optimiser localement sa fonction objective, éventuellement différente dans chaque agent, en répondant aux sollicitations des autres agents. Dans [31], les auteurs proposent un mécanisme d'attracteur itératif pour faciliter le processus d'affectation des tâches et manipuler la négociation entre agents [29].

Des approches artificielles récentes, dites systèmes immunitaires artificiels, se sont inspirées des mécanismes que l'immunité biologique met en œuvre pour réagir aux agressions et maintenir un état de bon fonctionnement dans les organismes vivants. Darmoul s'est inspiré de ces mécanismes pour la conception d'un système d'ordonnancement réactif et adaptatif [14].

Des chercheurs ont pensé à hybrider entre différentes approches, les algorithmes génétiques à d'autres méthodes, tel que la fouilles de Données, les Règles de Priorité [54]. Ainsi que l'ordonnancement réactif a été hybridé avec celui proactif comme était proposé par [13] pour l'amélioration de la production d'une ligne de traitement de surface.

5 Extraction des règles de priorités par fouille de données

La fouille de données « Data Mining » est l'analyse des observations de larges jeux de données dans le but d'identifier des relations non soupçonnées et de résumer la connaissance incluse au sein de ces données sous de nouvelles formes à la fois compréhensibles et utiles [22] C'est un domaine émergent de recherche qui s'appuie sur l'apprentissage automatique pour acquérir ces connaissances. La fouille de données a eu un impact important sur de nombreuses industries en général, en particulier, sa fonction dans le domaine d'ordonnancement n'a reçu que récemment l'attention des chercheurs.

L'incomplétude et le nombre immense des données générées par le système de production pour la fonction d'ordonnancement, augmente la complexité et provoque une certaine difficulté de capturer et représenter le tout par un modèle mathématique. En effet les connaissances découvertes par fouille de données peuvent être généralisées et utilisées dans le futur pour produire des nouveaux scénarios d'ordonnancement.

L'extraction des règles de priorités par fouille de données a montré une grande performance, en connaissant les relations cachées dans l'ensemble des attributs du système, l'utilisateur peut concevoir des règles de priorités effectives [50][51].

Olafsson [50] a montré comment utiliser la fouille de données pour générer des connaissances implicites sous forme de nouvelles règles qui peuvent être utilisées pour automatiser l'ordonnancement. Le processus d'apprentissage associé comporte deux phases principales:

1 Phase de préparation des données qui inclue l'agrégation, la construction et la sélection des attributs afin de construire un échantillon d'apprentissage approprié ;
2 Phase d'induction du modèle par arbre de décision et interprétation.

Exemple numérique:

Pour illustrer l'approche proposée dans [50], un exemple composé de cinq jobs notés j_1,j_2, j_3, j_4 et j_5 sont traités sur une seule machine, chaque job a une date de sortie désignée par un temps d'achèvement r_j, un temps de début s_j, une durée d'exécution p_j et une date de fin d'exécution c_j . Au départ il est supposé que rien n'est connu sur la façon d'ordonnancer ces jobs mais il est souhaité de concevoir un système automa-

tique qui ordonnance des nouveaux jobs selon la même logique. Le but est l'induction de quelques règles qui peuvent être utilisées pour ordonnancer les jobs.

La règle utilisée pour la transformation des données en un exemple est LPT, c'est à dire le job qui s'exécute en premier est celui qui est libéré et qui a la plus longue durée. Mais cette règle est supposée inconnue. La table 1 présente l'échantillon d'apprentissages construite.

Table 1. Échantillon d'apprentissage.

Job 1	P_1	r_1	Job 1	P_2	r_2	Classe
J1	15	10	*J2*	5	30	Oui
J1	15	10	*J3*	20	18	Oui
J1	15	10	*J4*	7	0	Oui
J1	15	10	*J5*	17	0	Non
J2	5	30	*J3*	20	18	Non
J2	5	30	*J4*	7	0	Non
J2	5	30	*J5*	17	0	Non
J3	20	18	*J4*	7	0	Oui
J3	20	18	*J5*	17	0	Non
J4	7	0	*J5*	17	0	Non

Le variable endogène « attribut classe » peut prendre deux valeurs possibles: « oui » qui indique que le job j_j s'exécute en premier et « non » pour le contraire. Les valeurs r_j,p_j et encore un autre attribut est ajouté pour améliorer la performance des règles extraites, noté p_d qui représente la différence entre les durées d'exécution des deux jobs comparés. p_j

Sur cet ensemble d'apprentissage un algorithme C4.5 de Quinlan1993 est utilisé pour générer un arbre de décision présenté dans la Figure 1. Un nœud feuille qui se termine par "oui" indique que le job associé doit s'exécuter en premier, l'arbre de décision ou les règles de décision correspondantes peuvent être utilisées directement pour ordonnancer n'importe quels deux nouveaux jobs. Un exemple de règle déduite à partir de cet arbre est:

Si un job1 s'achève et il est plus long qu'un autre job 2 alors le job1 doit être exécuté en premier.

L'exemple présenté dans cette section fournit un aperçu important sur l'application directe de la fouille de données pour découvrir des nouvelles règles de priorité. Notons que l'avantage de cette approche est la prise des décisions intelligentes en se basant sur ce modèle, qui est à la fois significatif, et explicatif.

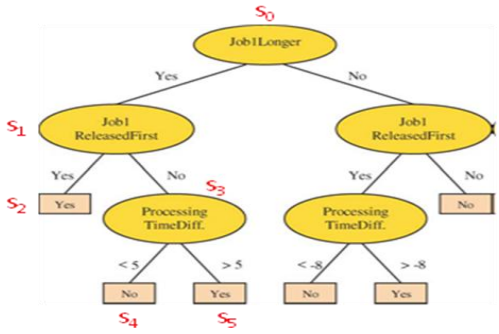


Fig. 1. Arbre de décision pour l'ordonnancement illustré par la table 1.

Cet exemple illustre la démarche suivie par Olafsson pour ordonnancer deux jobs avec un peu de paramètres « attributs ». En effet, l'inconvénient majeur de cette approche est qu'elle n'est applicable que sur des simples échantillons, et il est souhaité d'améliorer ses performances pour des cas plus complexes telles que les ateliers flow shop ou job shop, qui nécessite un temps de calcul élevé, et pour remédier à cela nous proposons une modélisation booléenne qui permettra l'extension de cette approche, La modélisation booléenne est basée sur la machine cellulaire CASI qui sera présenté dans la section suivante.

6 Modélisation Booléenne

6.1 La machine cellulaire CASI [04]

CASI (Induction Symbolique par Automate Cellulaire) est un automate cellulaire qui simule le fonctionnement du cycle de base d'un moteur d'inférence en utilisant deux couches finies d'automates finis. La première couche, CELFACT, pour la base des faits et, la deuxième couche, CELRULE, pour la base de règles [04, 05]. Chaque cellule à l'instant t+1 ne dépend que de l'état des ses voisines et du sien à l'instant t. Dans chaque couche, le contenu d'une cellule détermine si et comment elle participe à chaque étape d'inférence : à chaque étape, une cellule peut être active (1) ou passive (0), c'est-à-dire participe ou non à l'inférence.

[04, 05, 09] ont supposé qu'il y a ***l*** cellules dans la couche CELFACT, et ***r*** cellules dans la couche CELRULE. Toute cellule i de la première couche CELFACT est considérée comme fait établi si sa valeur est 1, sinon, elle est considérée comme fait à établir. Toute cellule j de la deuxième couche CELRULE est considérée comme une règle candidate si sa valeur est 1, sinon, elle est considérée comme une règle qui ne doit pas participer à l'inférence.

Les états des cellules se composent de trois parties : EF, IF et SF, respectivement ER, IR et SR, sont l'entrée, l'état interne et la sortie d'une cellule de CELFACT, respectivement d'une cellule de CELRULE. L'état interne, IF d'une cellule de CELFACT indique le rôle du fait : dans le cas d'un graphe d'induction IF = 0 corres-

pond à un fait du type sommet (s_i), IF = 1 correspond à un fait du type attribut=valeur (x_1= valeur) Pour une cellule de CELRULE, l'état interne IR peut être utilisé comme coefficient de probabilité que nous n'aborderons pas dans cet article [03]. Pour notre contribution nous nous proposons une modélisation booléenne de la méthode de [50] par la machine cellulaire CASI « Cellular Automata for Symbolic Induction » [04].

6.2 Exemple d'illustration de l'induction des règles booléennes inductives

Pour illustrer l'architecture et le principe de fonctionnement de CASI, nous considérons l'arbre présenté par la figure 1, à partir des partitions s_0, s_1, s_2, s_3, s_4 et s_5, nous déduisons des règles sous la forme : *Ri : Si Prémisse i Alors Conclusion i*
Après une représentation cellulaire de ces règles:

-- Les items de prémisse i et Conclusion i vont constituer les faits : FAITS

-- Les R i vont constituer les règles : REGLES

La figure 2 montre comment la base de connaissance extraite à partir de cet arbre, cette base est représentée par les couches CELFACT et CELRULE. Initialement, toutes les entrées des cellules dans la couche CELFACT sont passives (EF = 0), exceptée celles qui représentent la base des faits initiale (EF(1) = 1).

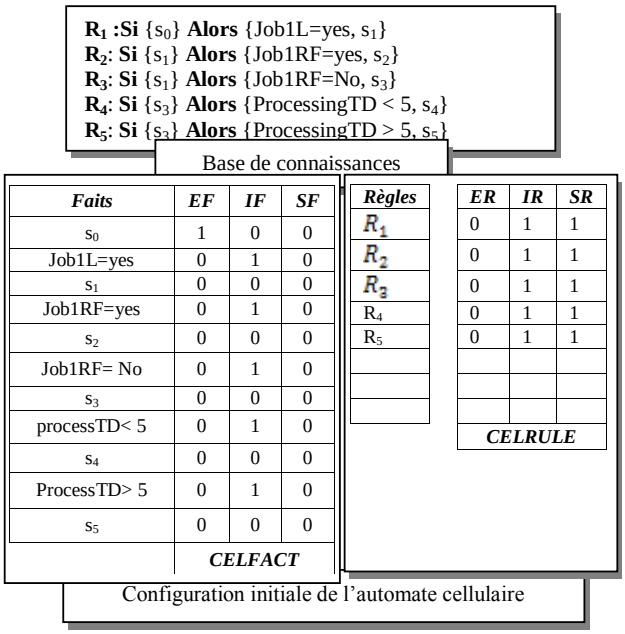


Fig. 2. Modélisation Booléenne de la base de connaissances.

Le voisinage est introduit par la notion de matrice d'incidence. Dans la figure 3 sont respectivement représentées les matrices d'incidence d'entrée R_E et de sortie R_S de l'automate cellulaire. La relation d'entrée, notée iR_Ej , est formulée comme suit : $\forall i = 1,...I, \ \forall j = 1,...r,$ si (le Fait $i \in$ à la Prémisse de la règle j) alors $R_E(i,j) \leftarrow 1$. De même la relation de sortie, notée iR_Sj , est formulée comme suit : $\forall i = 1,...I, \ \forall j = 1,...r,$ si (le Fait $i \in$ à la Conclusion de la règle j) alors $R_S(i,j) \leftarrow 1$.

	R1	***R2***	***R3***	***R4***	***R5***
s_0	0	0	0	0	0
Job1L=yes	1	0	0	0	0
s_1	1	0	0	0	0
Job1RF=yes	0	1	0	0	0
s_2	0	1	0	0	0
Job1RF= No	0	0	1	0	0
s_3	0	0	1	0	0
processTD< 5	0	0	0	1	0
s_4	0	0	0	1	0
ProcessTD> 5	0	0	0	0	1
s_5	0	0	0	0	1
Matrice d'Entrée Re					

	R1	***R2***	***R3***	***R4***	***R5***
s_0	1	0	0	0	0
Job1L=yes	0	0	0	0	0
s_1	0	1	1	0	0
Job1RF=yes	0	0	0	0	0
s_2	0	0	0	0	0
Job1RF= No	0	0	0	0	0
s_3	0	0	0	1	1
processTD< 5	0	0	0	0	0
s_4	0	0	0	0	0
ProcessTD> 5	0	0	0	0	0
s_5	0	0	0	0	0
Matrice de Sortie Rs					

Fig. 3. Les matrices d'incidences d'entrée R_E et de Sortie R_S pour la BC de la figure 2.

La dynamique du moteur d'inférence CIE de la machine cellulaire utilise deux fonctions de transitions δ_{Fact} et δ_{Rule} , où δ_{Fact} correspond à la phase d'évaluation, de sélection et de filtrage, et δ_{Rule} correspond à la phase d'exécution.

La fonction de transition δ_{Fact} :

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta fact} (EF, IF, \mathbf{EF, ER} + \mathbf{(R_E^T \cdot EF)}, IR, SR)$$

La fonction de transition δ_{Rule} :

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{rule}} (\mathbf{EF} + \mathbf{(R_S \bullet ER)}, IF, SF, ER, IR, \mathbf{\overline{ER}})$$

Où la matrice R_E^T désigne la transposé de la matrice R_E .

Le moteur d'inférence exploite la base de connaissances et les couches CELFACT et CELRULE pour établir un fait en chaînage avant, Le cycle est répété jusqu'à ce que le fait soit ajouté à la base des faits, ou s'arrête lorsqu'aucune règle n'est applicable.

7 Conclusion

Notre étude se voulait être assez novatrice, dans la mesure où nous avons été incités à utiliser des techniques prouvées de la machine cellulaire CASI [04, 05, 09] combinées à une fouille de données [50]. De ce fait deux motivations concurrentes nous ont amenés à adopté le principe de la modélisation booléenne, pour la génération, la représentation, l'optimisation et l'utilisation des règles de priorités. En effet, nous avons

non seulement souhaité avoir une base de règles optimale, qui nécessite des temps de traitements assez réduits, mais aussi, nous avons également souhaité améliorer cette base, mais aussi l'ordonnancement réactif.

Les avantages de cette méthode basée sur l'automate cellulaire peuvent être récapitulés comme suit :

- L'acquisition de l'information ainsi que son contrôle sont simples, sous forme de matrices binaires exigeant un prétraitement minimal. ;
- La facilité de l'implémentation des fonctions de transition δ_{Fact} et δ_{Rule} qui sont de basse complexité, efficaces et robustes pour des valeurs extrêmes et bien adapté aux situations avec beaucoup d'attributs ;
- Les matrices d'incidences, R_E et R_S, facilitent la transformation de règles dans des expressions équivalentes booléennes, qui nous permet d'utiliser l'algèbre de Boole élémentaire pour examiner d'autres simplifications.

Références

1. Aissani. N, (2010), « Pilotage adaptatif et réactif des systèmes de production à flux : application à l'industrie pétrolière », Thèse de Doctorat, université d'Oran, Université de Valenciennes (Co-tutelle).
2. Ahmed HASSAM, Talib Hicham BETAOUAF, Abdellah BENGHALEM et Zaki SARI.: La sélection des règles de priorité en function des critères de performances d'un FMS (2007).
3. Aline Cauvin: Analyse , Modélisation et Amélioration de la réactivité de décision dans les organizations industrielles, thèse de doctorat(2005).
4. Atmani B. & B. Beldjilali.: Knowledge Discovery in Database : Induction Graph and Cellular Automaton, Computing and Informatics Journal, Vol.26, N°2 (2007) 171-197.
5. Benamina M, Atmani B.:WCSS: un système cellulaire d'extraction et de gestion des connaissances, 3ème Atelier sur les systèmes décisionnels, 10-11 oct. 2008, Mohammadia, Maroc, ISBN 978-9981-1-3000-1, pp. 223-234, (2008).
6. Bensana. E.: Utilisation des techniques d'intelligence artificielle pour l'ordonnancement d'atelier. PhD thesis, Thèse de l'ENSAE, Toulouse (1987).
7. Billaut J.C., Moukrim A. & Sanlaville E. Flexibilité et robustesse en ordonnancement, Chapitre 1, 6 et 7, Hèrmes Sciences Publications (2005).
8. Boucon D.:Ordonnancement d'ateliers : aide au choix de règles de priorité, thèse de l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (1991).
9. Brahami, M., B. Atmani.: Vers une cartographie des connaissances guide par la fouille des données : 1ère étape modélisation booléenne. 2ème Conférence Francophone GECSO'09, Revue électronique ISDM, ISSN: 1265-499x, ISDM N° :36, (2009a).
10. Chebel-Morello B., Michaut D.,Baptiste P.: A knowledge discovery process for a flexible manufacturing system" Proc of the Emerging Technologies anf Factoty Automation, volume1 october 15 – 18 , Antibes, pp 652 – 659, (2006).
11. Chen, C.C., aet Yih, Y.: Identifying attributes for knowledge-based development in dynamic scheduling environments". International Journal of Production Research, Vol 34, No 6, pp 1739- 1755, (1996).
12. Chong, A.I. Sivakuma & R. Gay.: Simulation-based scheduling for dynamic discrete manufacturing. Dans Winter Simulation Conference, pages 1465_1473, New Orleans, USA, (2003).
13. Chové, Etienne.: Pilotage d'une unité de traitement de surface : couplage entre approches prédictives et réactives, (2006).

14. Darmoul. S, Henri Pierreval, Sonia Hajri Gabouj . :Quel potentiel des systèmes immunitaires artificiels pour le pilotage réactif et adaptatif de systèmes de production ?, CPI - Rabat, Maroc, (2007).
15. Davenport, A.J., and Beck, J.C.: A survey of techniques for scheduling with uncertainty. http://www.eil.utoronto.ca/chris/chris.papers.html. (2000).
16. El-Djillali TALBI. : Sélection et réglage de paramètres pour l'optimisation de logiciels d'ordonnancement industriel, thèse de doctorat (2004).
17. Erwan tranvouez. : IAD et ordonnancement, une approche cooperative du réordonnacement par systèmes multi-agents, thèse de doctorat. (2001).
18. Esquirol, P., & Lopez, P. : L'ordonnancement. Paris : Economica. (1999).
19. Esquirol, P., & Lopez, P. : Concepts et méthodes de base en ordonnancement de la production. In P. Esquirol & P. Lopez (Eds.), Ordonnancement de la production (pp. 25- 54). Paris : Hermes. (2001).
20. Fatma Tangour Toumi.: Ordonnancement Dynamique dans les Industries Agroalimentaires, thèse de doctorat, (2007).
21. François Marmier.: Contribution à l'ordonnancement des activités de maintenance sous contrainte de compétence : une approche dynamique, proactive et multi-critè », thèse de doctorat 2007, tel-00212750, version 1 - 23 Jan 2008, (2007).
22. Fayyad, U, Shapiro, G. P, Smyth, P. : The KDD process for extraction useful knowledge from volumes data, Communication of the ACM; (1996).
23. Geneste L., Grabot B. : Implicit versus explicit knowledge representation for job shop scheduling", International Journal of expert systems, research and applications, vol.10 1997, n°1, p 37-52. (1997).
24. Grabot. R, Talbi E. D., Geneste L. : Meta-heuristics for optimal set-up of an industrial scheduling software, CESA'2003, 9-11 juillet 2003, Lille, France. (1997).
25. Jeong, K.C., and Kim, Y.D.: A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules, International Journal of Production Research, 36(9), 2609-2626. (1998).
26. Kim, M., and Kim, Y. : Simulation based real-time scheduling in a flexible manufacturing system, Journal of Manufacturing Management Systems, 13(2), 85-93. (1994).
27. Kutanoglu, E. and Sabuncuoglu, I. : An analysis of heuristics in a dynamic job shop with weighted tardiness objectives, International Journal of Production Research, 37(1), 165-187. (1999).
28. Latouzey. : Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs, thèse de doctorat 2001, N° d'ordre : 1848. (2001).
29. LA Hoang Trung. : Utilization d'ordres partiels pour la caractérisation de solutions robustes en ordonnancemtnt, thèse de doctorat. (2005).
30. Lopez, P., and Roubellat, F. : Production scheduling, ISTE/Wiley, London. (2008).
31. Lim M.K.& Z. Zhang.: Iterative multi-agent bidding and coordination based on genetic algorithm. Dans 3 Complex Systems, and E-Businesses, pages 682_689, Erfurt, Germany. (2002).
32. Matthieu Dupuy. : Contributions à l'analyse des systèmes industriels et aux problèmes d'ordonnancement à machines parallèles flexibles : application aux laboratoires de contrôle qualité en industrie pharmaceutique », thèse de doctorat. (2005).
33. Mirdamadi . : Pilotage d'un atelier de production en temps reel à base de simulation de flus à évènements discrets, EDSys 2007 – 8e congrès des doctorants, (2007).
34. Mirdamadi . : Modélisation du processus de pilotage d'un atelier en temps réel, à l'aide de la simulation en ligne couplée à l'exécution, thèse de doctorat. (2009).
35. Milin, A.M. : Amélioration des solutions d'ordonnancement pour le pilotage d'atelier. Thèse de doctorat, Université de Bordeaux I, France, (1987).

36. Monostori L., Szelke E., Kadar B.: Management of changes and disturbances in manufacturing systems », Annual Reviews in Control, vol. 22, 1998, p. 85-97. (1998).
37. Pierreval, H., & Mebarki, N. : Dynamic selection of dispatching rules for manufacturing system scheduling. International Journal of Production Research, 35, 1575-1591. (1997).
38. Pierreval, : Proposition de typologie des activités de décision en temps réel agissant sur les flux des systèmes de production », actes de la seconde conference Francophone de modélisation et de simulation : modélisation et simulation des flux physiques et informationnels, octobre 1999, Annecy, pp. 331-336. (1999).
39. Priore, P., Garcia, D.D., and Quesada, I.F. : Manufacturing systems scheduling through machine learning, Neural Computation, NC'98, Vienna, Austria, 914 -917. (1998).
40. Priore, P., De la Fuente, D., Gomez, A., and Puente, J.: Review of machine learning in dynamic scheduling of flexible manufacturing systems. Artificial Intelligence for Engineering Design Analysis and Manufacturing, 15(3), 251-263. (2001).
41. Pinedo . : Scheduling theory, algorithms and systems , Prentice-Hall, NJ. (1995).
42. Pujo P., Kieffer J.P.: Sous la direction de), Méthodes du pilotage des systèmes de production, Paris, Hermès. (2002b).
43. Pujo, P., and Kieffer, J.-P. : Méthodes de pilotage des syst`emes de production. Lavoisier, Hermes, (2002).
44. Saaty. T.L . : The analytic hierarchy process, McGrawHill, (1980).
45. Schneider. J .G, Boyan. J. A, Moore. A. W.: Value Function based Production scheduling", Proc. ICML The Fifteenth International Conference on Machine Learning. July 24-27, in Madison, Wisconsin. (1998).
46. Shen, W. D. Xue, and D. H.: Norrie, "An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems," in Proc. PAAM'98, London, UK, pp. 533-548. (1998, 1999, 2001).
47. Shen, W., Wang, L., and Hao, Q. : Agent-Based Distributed Manufacturing ProcessPlanning and Scheduling: A State-of-the-Art Survey, IEEE transactions on systems, man, and cybernetics-part c: Applications and Reviews, 36(4), 563-577. (2006).
48. Smith, S.F., Ow, P.S., Potvin, J.Y., Muscettola, N., and Matthys, D. : OPIS: An Opportunistic Factory Scheduling System. Proceedings of the Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-90), Knoxville, USA, May. (1990).
49. Subramaniam, V., Lee, G. K., Hong, G. S., Wong, Y. S., and Ramesh, T.: Dynamic selection of dispatching rules for job shop scheduling. Production planning & control, 11 (1), pp. 73-81. (2000).
50. Sigurdur Olafsson, Xiaonan Li: Discovering dispatching rules using data mining. Journal of Scheduling, 8(6):515–527. (2005).
51. Sigurdur Olafsson, Xiaonan Li. : Learning effective new single machine dispatching rules from optimal scheduling data. Int. J. Production Economics 128 (2010) 118–126; (2010).
52. Toal D., Coffey, T., and Smith, P.: Expert systems and simulation in scheduling, Proc. IMC11, Belfast. (1994).
53. Van Der Pluym, B. : Knowledge-based decision making for job-shop scheduling. International Journal of Integrated Manufacturing, 6 (3), pp. 354-363. (1990).
54. Youssef Harrat. : Algorithmes génétiques et fouille de données pour un ordonnancement réactif dans un atelier de type job-shop. (2003).
55. Zoheir Karaouzene, Zaki Sari. : Ordonnancement et gestion des files d'attente par les règles de priorité dans un système de production, 7e Conférence Internationale de MOdélisation et SIMulation - MOSIM'08 - du 31 mars au 2 avril 2008 – Paris- France « Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination.». (2008).

Single Machine Scheduling Problems : ILP formulations using dominance conditions

S. Ourari[1] and C. Briand[2], and B. Bouzouia[1]

[1]Centre de Développement des Technologies Avances, Alger, Algrie
[2]Laboratoire d'Architecture et d'Analyse des Systmes-CNRS; Toulouse, France
sourari@cdta.dz

Abstract. This paper considers the problem of scheduling n jobs on a single machine. A fixed processing time and an execution interval are associated with each job. Preemption is not allowed. On the basis of analytical and numerical dominance conditions, original integer linear programming formulations are proposed, considering the feasibility problem, the minimization of the maximum lateness problem ($1|r_i|L_{\max}$) and the minimization of the number of late jobs problem ($1|r_j|\sum U_j$).

1 Introduction

A single machine scheduling problem (SMSP) consists of a set V of n jobs to be sequenced on a single disjunctive resource. The interval $[r_j, d_j]$ defines the execution window of each job j, where r_j is the release date of j and d_j, its due-date. The processing time p_j of j is known and preemption is not allowed. A job sequence σ is said feasible if, for any job $j \in V$, $s_j \geq r_j$ and $s_j + p_j \leq d_j$, s_j being the earliest starting time of Job j in σ.

Seeking whether it exists a feasible sequence (i.e., all jobs meet their due dates) is NP-complete [10] and consequently, finding a sequence that minimizes the maximum lateness (problem refers to as $1|r_i|L_{\max}$) is NP-hard. For this problem, the well-known branch-and-bound procedure of Carlier [6] can be used for solving problems having more than one thousand jobs. Considering the problem of finding a job sequence that minimizes the number of late jobs, problem referred to as $1|r_j|\sum U_j$ in the litterature, where U_j is set to 1 if job j is late, *i.e.*, $U_j \leftarrow (s_j + p_j > d_j)$ is also NP-hard [9]. For this problem, efficient branch-and-bound procedures are reported in [1, 8, 4] that solve problem instances with up to 200 jobs. Even if the Carlier's procedure provides good performance for the $1|r_i|L_{\max}$) problem, and had-hoc methods exists for solving the $1|r_j|\sum U_j$ problem, it is still interesting to design other exact approaches that can be used in a more generic framework such as integer linear programming (ILP).

This paper presents an original ILP formulation for the SMSP that is based on analytical and numerical dominance conditions. It first presents a dominance theorem that is the foundation of the work. Then considering the feasibility

problem, ILP models are progressively issued, starting from particular SMSP cases up to the general case. In last, we show how to use the original ILP model for modeling both the $1|r_i|L_{\max}$ problem and the $1|r_j|\sum U_j$ problem.

2 A general dominance theorem for the SMSP

In the sequel, some analytical dominance conditions are used for the SMSP. They have been originally proposed in the early eighties by Erschler et al. [7] within a theorem. This theorem uses the notions of a *top* and a *pyramid*, which are defined below. It defines a set S_{dom} of dominant job sequences for the SMSP with regard to the feasibility problem. Let us recall that a job sequence σ_1 dominates another job sequence σ_2 if σ_2 feasible $\Rightarrow \sigma_1$ is feasible. By extension, a set of job sequences S_{dom} is said dominant if, for any job sequence $\sigma_2 \notin S_{\text{dom}}$, it exists $\sigma_1 \in S_{\text{dom}}$ such that σ_2 feasible $\Rightarrow \sigma_1$ is feasible. When searching a feasible job sequence, only the set of dominant sequences need to be explored since, if there does not exist any feasible sequence in the dominant set, then it can be asserted that the original problem does not admit any feasible solution. This leads to a significant reduction of the search space.

Definition 1 *A job $t \in V$ is a top if there does not exist any other job $i \in V$ such that $r_i > r_t \wedge d_i < d_t$ (i.e., the execution window of a top job does not strictly include any other execution window).*

The top jobs are indexed in ascending order with respect to their release dates or, in case of a tie, in ascending order with respect to their due dates. When both their release dates and due dates are equal, they can be indexed in an arbitrary order. Thus, if t_a and t_b are two top jobs then $a < b$ if and only if $(r_{t_a} \leq r_{t_b}) \wedge (d_{t_a} \leq d_{t_b})$. Let m be the total number of top jobs.

Definition 2 *A pyramid P_k related to a top job t_k is the set of jobs $i \in V$ such that $r_i < r_{t_k} \wedge d_i > d_{t_k}$ (i.e., the set of jobs for which their execution window strictly includes the execution window of the top job).*

Considering the previous definition, let us observe that a non-top job can belong to several pyramids. Let $u(j)$ ($v(j)$ respectively) denotes the index of the first pyramid (the last pyramid resp.) to which Job j can be assigned.

The following theorem, proved in [7], can now be stated.

Theorem 1 *The set of sequences in the form $\sigma = \alpha_1 \prec t_1 \prec \beta_1 \prec \cdots \prec \alpha_k \prec t_k \prec \beta_k \prec \cdots \prec \alpha_m \prec t_m \prec \beta_m$ is dominant for finding a feasible sequence, where:*

- *t_k is the top job of the pyramid P_k, $\forall k = 1 \ldots m$;*
- *α_k and β_k are two job subsequences located at the left and the right of top job t_k respectively, such that, for any k, the jobs of α_k are ordered by increasing release dates and the jobs of β_k are ordered by increasing due dates ;*
- *any non-top job j is assigned to one sequence α_k or β_k, with $u(j) \leq k \leq v(j)$.*

In the previous theorem, it must be pointed out that the numerical values of the processing times p_j as well as those of r_j and d_j are not used. Only the relative order of the release and due dates is considered, hence the genericity of the result. In order to illustrate the theorem, let us consider an instance of seven jobs, so that the relative order among the release dates r_j and the due dates d_j of the jobs is $r_6 < r_1 < r_3 < r_2 < r_4 < d_2 < d_3 < d_4 < (r_5 = r_7) < d_6 < d_5 < d_1 < d_7$. This instance has four top jobs : $t_1 = 2$, $t_2 = 4$, $t_3 = 5$ and $t_4 = 7$, which characterize four pyramids : $P_1 = \{1, 3, 6\}$, $P_2 = \{1, 6\}$, $P_3 = \{1\}$ and $P_4 = \emptyset$ (in accordance with the definition, a top job does not belong to the pyramid it characterizes). According to Theorem 1, whatever the real values of r_i and d_i (provided they are compatible with the previous interval structure), the dominant sequences are in the form $\alpha_1 \prec 2 \prec \beta_1 \prec \alpha_2 \prec 4 \prec \beta_2 \prec \alpha_3 \prec 5 \prec \beta_3 \prec 7$, where jobs belonging to subsequence α_k (β_k respectively) are sequenced with respect to the increasing order of their r_j (d_j respectively). Thus, 24 dominant sequences (out of $7! = 5040$) are characterized.

3 FINDING A FEASIBLE JOB SEQUENCE

3.1 THE MONO-PYRAMIDAL SMSP

In this section, the focus is on the mono-pyramidal SMSP: it is assumed that the problem has a single top job $t \in V$ (hence a single pyramid P_t). With respect to Theorem 1, the set S_{dom} of dominant job sequences is in the form $\alpha \prec t \prec \beta$, where jobs belonging to subsequence α are sequenced with respect to the increasing order of their r_j, and jobs belonging to β, with respect to the increasing order of their d_j. Hence, 2^n sequences need to be explored for checking the feasibility (instead of $(n+1)!$). Below, another numerical dominance theorem is stated that allows to compare the 2^n sequences each other. It is based on a necessary and sufficient condition of feasibility.

As stated in [5] regarding the general SMSP, let us first recall that a job sequence σ is feasible iff:

$$p_i + \sum_{i \prec k \prec j} p_k + p_j \leq d_j - r_i \ , \forall (i,j) \in \sigma \text{ such that } i \prec j \tag{1}$$

The previous necessary and sufficient condition of feasibility can be drastically simplified for the mono-pyramidal SMSP. For matter of notation simplification, given a dominant sequence in the form $\alpha \prec t \prec \beta$, the following notations are set:

- $R_a = r_a + p_a + \sum_{k \in \alpha, a \prec k}(p_k), \forall a \in \alpha$;
- $D_b = d_b - p_b - \sum_{k \in \beta, k \prec b}(p_k), \forall b \in \beta$;
- $R^{\max} = \max\left[r_t, \max_{a \in \alpha}(R_a)\right]$;
- $D^{\min} = \min\left[d_t, \min_{b \in \beta}(D_b)\right]$.

Clearly, $R^{\max}$ corresponds to the earliest starting time of Top t, further refers to as est_t. Symmetrically, $D^{\min}$ corresponds to the latest completion time of Top t, further refers to as lft_t.

Now the following theorem can be stated.

Theorem 2 *P being a pyramid of top t, any dominant sequence in the form $\alpha \prec t \prec \beta$ is admissible if and only if $D^{\min} - R^{\max} - p_t \geq 0$.*

The fact that Theorem 2 permits to numerically compare sequences each other is helpful since it immediately involves the following dominance condition.

Corollary 1 *P being a pyramid, job t being its top job, and σ_1 and σ_2 being two dominant sequences in the form $\alpha_1 \prec t \prec \beta_1$ and $\alpha_2 \prec t \prec \beta_2$ respectively, if $D_1^{\min} - R_1^{\max} \geq D_2^{\min} - R_2^{\max}$, then σ_1 dominates σ_2 with regard to the feasibility problem.*

Consequently, the problem of searching a feasible job sequence in a mono-pyramidal SMSP can be reformulated as an optimization problem where the objective function is to maximize $D^{\min} - R^{\max}$. An integer linear program (ILP1) is given below for modeling such a problem.

$$\max \; z = D - R$$

$$\text{s.t.} \quad \begin{cases} R \geq r_t & (3.1) \\ D \leq d_t & (3.2) \\ R \geq r_i + \sum_{\{j \in V \setminus \{t\} | r_j \geq r_i\}} p_j x_j^+ \;, \; \forall i \in V \setminus \{t\} & (3.3) \\ D \leq d_i - \sum_{\{j \in V \setminus \{t\} | d_j \leq d_i\}} p_j x_j^- \;, \; \forall i \in V \setminus \{t\} & (3.4) \\ x_i^- + x_i^+ = 1 \;, \; \forall i \in V \setminus \{t\} & (3.5) \\ x_i^- \;, \; x_i^+ \in \{0,1\} \;, \; \forall i \in V \setminus \{t\} & \\ D \;, \; R \in \mathbb{Z} & \end{cases}$$

In the above ILP1 formulation, every job $j \in V \setminus \{t\}$ has to be sequenced either at the left side (i.e., $x_j^+ = 1$ and $j \in \alpha$) or at the right side (i.e., $x_j^- = 1$ and $j \in \beta$) of Top t, but not both (i.e., $x_i^- + x_i^+ = 1$, constraint 3.5). The constraints (3.1) and (3.3) define the earliest starting time of the top t while constraints (3.2) and (3.5) define the latest completion time of the top t. If $z = D - R$ is maximized, while respecting the constraints, then, from Corollary 1, the obtained sequence dominates all the others. Moreover, if $z^* \geq p_t$ then, from Theorem 2, the sequence is feasible. If $z^* < p_t$ then it can be asserted that it does not exist any feasible sequence for the considered problem.

3.2 The independent multi-pyramidal SMSP

Before considering the general SMSP, let us focus on the particular SMSP characterized by m top jobs, hence m pyramids, such that any non-top job only

belongs to a single pyramid (i.e., the pyramids are independent). With respect to Theorem 1, any dominant job sequence is in the form $\sigma = \alpha_1 \prec t_1 \prec \beta_1 \prec \cdots \prec \alpha_m \prec t_m \prec \beta_m$, where t_k is the top job of the pyramid P_k, $\forall k = 1 \dots m$ and the jobs of α_k and β_k only belong to pyramid P_k.

Therefore, the problem of finding the most dominant job sequence for the independent multi-pyramidal SMSP can be decomposed into m interdependent mono-pyramidal subproblems. For each of them, the most dominant job sequence $\alpha_k \prec t_k \prec \beta_k$ has to be found. In other words, if R_k refers to as the earliest starting time of Top t_k and D_k to as the latest completion time of Top t_k, we have to maximize $D_k - R_k - p_{t_k}$.

These mono-pyramidal subproblems are interdependent because the sequences $\alpha_k \prec t_k \prec \beta_k$ should not overlap. This requirement can easily be achieved by setting, for any job $j \in P_k$, the constraints: $r_j = \max(r_j, eft_{k-1})$ and $d_j = \min(d_j, lst_{k+1})$, where eft_{k-1} (lst_{k+1} resp.) refers to as the earliest completion time of Subsequence β_{k-1} (the latest starting time of Subsequence α_{k+1} resp.).

Consequently, the problem of searching a feasible job sequence in a multi-pyramidal independent SMSP can be formulated by the following ILP (ILP2):

$$\max \; z = \min_{k=1,\dots,m}(D_k - R_k - p_{t_k})$$

$$\text{s.t.} \quad \begin{cases} R_k \geq r_{t_k} & , \; \forall k \in [1\ m] & (4.1) \\ D_k \leq d_{t_k} & , \; \forall k \in [1\ m] & (4.2) \\ R_k \geq r_i + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x_j^+ & , \; \forall i \in P_k & (4.3) \\ D_k \leq d_i - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x_j^- & , \; \forall i \in P_k & (4.4) \\ R_k \geq R_{k-1} + \sum_{\{j \in P_{k-1}\}} p_j x_j^- + p_{t_{k-1}} & & \\ \qquad + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x_j^+ & , \; \forall k \in [2\ m] & (4.5) \\ D_k \leq D_{k+1} - \sum_{\{j \in P_{k+1}\}} p_j x_j^+ - p_{t_{k+1}} & & \\ \qquad - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x_j^- & , \; \forall k \in [1\ (m-1)] & (4.6) \\ x_i^- + x_i^+ = 1 & , \; \forall i \in P_k & (4.7) \\ x_i^- \ , \ x_i^+ \in \{0,1\} & , \; \forall i \in P_k & \\ D_k \ , \ R_k \in \mathbb{Z} & , \; \forall k \in [1\ m] & \end{cases}$$

By definition:

$$R_k = \max(r_{t_k}, eft_{k-1} + \sum_{\{j \in \alpha_k\}} p_j, \max_{i \in \alpha_k}(r_i + p_i + \sum_{\{j \in \alpha_k | i \prec j\}} p_j)) \qquad (2)$$

where eft_{k-1} is the earliest completion time of the job subsequence β_{k-1}. As the variable R_{k-1} corresponds to the earliest starting time of Job t_{k-1}, it comes that $eft_{k-1} = R_{k-1} + p_{t_{k-1}} + \sum_{j \in \beta_{k-1}} p_j$. Therefore, the constraints (4.1), (4.3) and (4.5), according to Equation (2), allow to determine the value of R_k.

Symmetrically, by definition:

$$D_k = \min(d_{t_k}, lst_{k+1} - \sum_{\{j \in \beta_k\}} p_j, \min_{i \in \beta_k}(d_i - p_i - \sum_{\{j \in \beta_k | j \prec i\}} p_j)) \qquad (3)$$

where lst_{k+1} is the latest starting time of the job subsequence α_{k+1}. As the variable D_{k+1} corresponds to the latest finishing time of Job t_{k+1}, it comes that $lst_{k+1} = D_{k+1} - p_{t_{k+1}} - \sum_{j \in \alpha_{k+1}} p_j$. Therefore, the constraints (4.2), (4.4) and (4.6), according to Equation (3), give to D_k its value.

In the previous formulation, similarly with ILP1, the binary variable x_i^+ (x_i^- resp.) is set to 1 if Job i, which only belongs to pyramid P_k, is sequenced in α_k (in β_k resp.), provided that it cannot be sequenced both in α_k and β_k (i.e., $x_i^- + x_i^+ = 1$). The constraints (4.1)-(4.4) are identical with the constraints (3.1)-(3.4) of ILP1. The constraints (4.5) and (4.6) impose that the subsequences $\alpha_k \prec t_k \prec \beta_k$ do not overlap. Under these constraints, one can determine for each pyramid P_k, the margin $R_k - D_k + p_{t_k}$. Hence, the optimal value $z*$ defines the maximum margin that allows the most feasible sequence. If $z^* \geq 0$ then it can be asserted that it exists a feasible sequence for the considered problem.

3.3 The general SMSP

Now, the general SMSP is considered. Here, a non-top job j can belong to several pyramids, that is the only difference with the multi-pyramidal independent SMSP. Let us note $u(j)$ ($v(j)$ resp.) the index of the first pyramid (the last pyramid resp.) to which the job j belongs. With respect to Theorem 1, any dominant sequence is in the form $\sigma = \alpha_1 \prec t_1 \prec \beta_1 \prec \cdots \prec \alpha_m \prec t_m \prec \beta_m$, where t_k is the top job of the pyramid P_k, $\forall k = 1 \ldots m$, and any non-top job j can be sequenced either in α_k or β_k, for any k such that $u(j) \leq k \leq v(j)$.

It is easy to see, once the assignments of the non-top jobs to their pyramid done, that the problem turns into a multi-pyramidal independent problem (that can be solved using the ILP2 formulation). Consequently, a general formulation can be obtained by simply adding binary variables to the ILP2 formulation, so that each non-top job can only be assigned to a single pyramid P_k. The following general formulation (ILP3) is then obtained:

$$\max \; z = \min_{k=1,\ldots,m}(D_k - R_k - p_{t_k})$$

$$\text{s.t.} \quad \begin{cases}
R_k \geq r_{t_k} & , \; \forall k \in [1\ m] & (5.1)\\
D_k \leq d_{t_k} & , \; \forall k \in [1\ m] & (5.2)\\
R_k \geq r_i + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x^+_{kj} & , \; \forall i | k = u(i) & (5.3)\\
D_k \leq d_i - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x^-_{kj} & , \; \forall i | k = v(i) & (5.4)\\
R_k \geq R_{k-1} + \sum_{\{j \in P_{k-1}\}} p_j x^-_{(k-1)j} + p_{t_{k-1}} & & \\
\qquad + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x^+_{kj} & , \; \forall k \in [2\ m] & (5.5)\\
D_k \leq D_{k+1} - \sum_{\{j \in P_{k+1}\}} p_j x^+_{(k+1)j} - p_{t_{k+1}} & & \\
\qquad - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x^-_{kj} & , \; \forall k \in [1\ (m-1)] & (5.6)\\
\sum_{k=u(i)}^{v(i)} (x^-_{ki}) + x^+_{u(i),i} = 1 & , \; \forall i & (5.7)\\
x^-_{ki} \;,\; x^+_{u(i),i} \in \{0,1\} & , \; \forall k \in [1\ m], \forall i \in P_k & \\
D_k \;,\; R_k \in \mathbb{Z} & , \; \forall k \in [1\ m] &
\end{cases}$$

In ILP3, similarly with ILP2 with a bit difference, the binary variable $x^+_{u(i),i}$ is set to 1 if Job i, belonging to Pyramid $P_{u(i),i}$ is sequenced in $\alpha_{u(i),i}$ and x^-_{ki} is set to 1 if Job i, belonging to Pyramid P_k (i.e. $u(i) \leq k \leq v(i)$), is sequenced in β_k, provided that it cannot be sequenced both on the left and on the right of a pyramid and it is assigned to one and only one pyramid (i.e., the constraint (5.7) is satisfied for i). The constraints (5.1)-(5.6) remain unchanged, when compared to ILP2.

Now let us give few comments concerning the number of constraints and variables that are involved. There is one constraint of type (5.1) and (5.2) for each top job. There is one constraint of type (5.3) and (5.4) for each non-top job and there are $(n-m)$ non-top jobs. There are $(m-1)$ constraints of type (5.5) and $(m-1)$ constraints of type (5.6). There is one constraint of type (5.7) for each non-top job. Therefore, the total number of constraint is $(m+3n-2)$. The number of binary variables is $((n-m) + \sum_{k=1}^{m}(n_k))$, where n_k is the number of non-top jobs that can be assigned to P_k. There are $(2m)$ integer variables but we notice that their value can be directly deduced from those of the binary variables.

4 THE $(1|r_i|L_{\max})$ PROBLEM

We recall that Theorem 1, dominant regarding the feasibility problem, is also dominant regarding to the maximum lateness minimization problem [2].

Theorem 3 *The optimal ILP1 solution minimizes the maximum lateness and* $L^*_{\max} = -z^*$.

Considering the $1|r_i|L_{\max}$ problem, Theorem 3 given in this section and demonstrated in [3] shows that there is a strict equivalence between the maximum lateness minimization and the maximum of the objective function z given in ILP3 ; z expressing the maximal margin that offers the feasible sequence. The optimal solution that determine the most feasible sequence is then also optimal with regard to the maximum lateness minimization. ILP3 constitute an original integer linear programming formulation for solving the general $1|r_i|L_{\max}$.

5 THE $1|r_j|\sum U_j$ PROBLEM

In this section, the $\sum U_j$ criterion is considered. Searching optimal solution for $1|r_j|\sum U_j$ problem amounts to determine a feasible sequence for the largest selection of jobs $E \subseteq V$. Let E^* be this selection. The jobs of E^* are on time while others are late. The late jobs can be scheduled after the jobs of E^* in any order. So they do not need to be considered when searching a feasible job sequence for on-time jobs. Consequently, Theorem 1 can be applied only to the jobs belonging to E^*. There are $\sum_{k=1\ldots n} C_n^k$ possible different selections of jobs. Regarding the $\sum U_j$ criterion, the following corollary is proved.

Corollary 1 *The union of all the dominant sequences that Theorem 1 characterizes for each possible selection of jobs is dominant for the* $\sum U_j$ *criterion.*

As already pointed out, the number of possible job selections is quite large. Nevertheless, as explained in [11], it is not necessary to enumerate all the possible job selections to get the dominant sequences. Indeed, they can be characterized using one or more *master-pyramid sequences*. The notion of a master-pyramid sequence is somewhat close to the notion of a master sequence that Dauzères-Pérès and Sevaux proposed in [8]. It allows to easily verify if a job sequence belongs to the set of dominant sequences that Theorem 1 characterizes.

For building up a master-pyramid-sequence associated with a job selection $E \subseteq V$, the m_E tops and pyramids have first to be determined. Then, knowing that the set of dominant sequences is in the form $\alpha_1(E) \prec t_1(E) \prec \beta_1(E) \prec \cdots \prec \alpha_k(E) \prec t_k(E) \prec \beta_k(E) \prec \cdots \prec \alpha_{m_E}(E) \prec t_{m_E}(E) \prec \beta_{m_E}(E)$, it is assumed that any non-top job j is sequenced both in $\alpha_k(E)$ and $\beta_k(E)$ (these subsequences being ordered as described in Theorem 1), $\forall k$ such that $u(j) \leq k \leq v(j)$.

For illustration, let us consider a problem instance with 7 jobs such that the relative order among the release and due dates of the jobs is $r_6 < r_1 < r_3 < r_2 < r_4 < d_2 < d_3 < d_4 < (r_5 = r_7) < d_6 < d_5 < d_1 < d_7$. For this example, the-master-pyramid-sequence associated with the selection $E = V$ is (tops are in bold):

$$\sigma_{\Delta}(V) = (6, 1, 3, \mathbf{2}, 3, 6, 1, 6, 1, \mathbf{4}, 6, 1, 1, \mathbf{5}, 1, \mathbf{7})$$

Any job sequence of n jobs *compatible* with $\sigma_\Delta(V)$ belongs to the set of dominant sequences. A sequence s is said compatible with the master-pyramid sequence $\sigma_\Delta(V)$ if the order of the jobs in s does not contradict the possible orders defined by $\sigma_\Delta(V)$, this will be denoted as $s \Subset \sigma_\Delta(V)$. Under the hypothesis that all tops are on-time, it is obvious that $\sigma_\Delta(V)$ also characterizes the set of dominant sequences (according to the $\sum U_j$ criterion) of any job selection E such that $\{t_1, \ldots, t_m\} \subseteq E$. Indeed, the master-pyramid sequence $\sigma_\Delta(E)$ associated with such a selection is necessarily compatible with the master-pyramid sequence $\sigma_\Delta(V)$, *i.e.*, if s is a job sequence such that $s \Subset \sigma_\Delta(E)$ then $s \Subset \sigma_\Delta(V)$.

Nevertheless, $\sigma_\Delta(V)$ does not necessarily characterize all the job sequences being dominant for the $\sum U_j$ criterion. This assertion can easily be illustrated considering a problem V with 4 jobs having the total order: $r_d < r_b < r_c < r_a < d_a < d_b < d_c < d_d$. Job a is the top of the corresponding structure and the master-pyramid sequence $\sigma_\Delta(V)$ is $(d, b, c, \mathbf{a}, b, c, d)$. Now, let us imagine that a is not selected (it is late and its interval can be ignored). In this case, there are two tops b and c and the master-pyramid sequence $\sigma_\Delta(V \setminus \{a\})$ is $(d, \mathbf{b}, d, \mathbf{c}, d)$. As it can be observed, $\sigma_\Delta(V \setminus \{a\})$ is not compatible with $\sigma_\Delta(V)$ since, in the former, Job d cannot be sequenced between b and c, while it is possible in the latter.

This simple example shows that the complete characterization of the set of dominant sequences requires to enumerate all the non-compatible master-pyramid sequences, their number being possibly exponential in the worst case.

In this section, we take an interest in finding a feasible job sequence in the form $\alpha_1 \prec t_1 \prec \beta_1 \cdots \prec \alpha_m \prec t_m \prec \beta_m$ that minimizes the $\sum U_j$ criterion, knowing that a non-top j is not necessarily sequenced, that is the major difference with Section 3. As discussed in Section 5, the sequences in this form are dominant for the $\sum U_j$ criterion only under the hypothesis that the top jobs are scheduled on time. As it does not necessarily exist any optimal sequence satisfying this assumption, finding an optimal sequence in this desired form only gives an upper bound to the general $1|r_j|\sum U_j$ problem.

The MIP formulation of Section 3 can easily be adapted for solving the previous problem:

$$\min \; z = \sum_{\{j \in V \setminus \{t_1 \ldots t_m\}\}} (1 - x^+_{u(j),j} - \sum_{k=u(j)}^{v(j)} (x^-_{jk})) + \sum_{k=1}^{m} y_{t_k}$$

$$\text{s.t.} \begin{cases}
R_k \geq r_{t_k} & , \; \forall k \in [1\ m] & (6.1) \\
R_k \geq r_i + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x^+_{kj} & , \; \forall k \in [1\ m], \forall i \in P_k & (6.2) \\
R_k \geq R_{k-1} + \sum_{\{j \in P_{k-1}\}} p_j x^-_{(k-1)j} + p_{t_{k-1}} & & \\
\qquad + \sum_{\{j \in P_k\}} p_j x^+_{kj} & , \; \forall k \in [2\ m] & (6.3) \\
D_k \leq d_{t_k} & , \; \forall k \in [1\ m] & (6.4) \\
D_k \leq d_i - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x^-_{kj} & , \; \forall k \in [1\ m], \forall i \in P_k & (6.5) \\
D_k \leq D_{k+1} - \sum_{\{j \in P_{k+1}\}} p_j x^+_{(k+1)j} - p_{t_{k+1}} & & \\
\qquad - \sum_{\{j \in P_k\}} p_j x^-_{kj} & , \; \forall k \in [1\ (m-1)] & (6.6) \\
\sum_{k=u(i)}^{v(i)} (x^-_{ki}) + x^+_{ki} \leq 1 & , \; \forall i \in P_k & (6.7) \\
D_k - R_k \geq p_{t_k}(1 - y_{t_k}) & , \; \forall k \in [1\ m] & (6.8) \\
y_{t_k} \; , \; x^-_{ki} \; , \; x^+_{u(i),i} \in \{0,1\} & , \; \forall k \in [1\ m], \forall i \in P_k & \\
D_k \; , \; R_k \in \mathbb{Z} & , \; \forall k \in [1\ m] &
\end{cases}$$

Let us comment this MIP. First, constraints (6.1)-(6.6) are identical to constraints (5.1)-(5.6) since integer variables R_k and D_k are determined in the same way. Allowing a non-top job to be late is easy by relaxing constraint (5.7), replacing it by constraint (6.7). As the feasibility of the obtained sequence is required, the constraint $D_k - R_k \geq p_{t_k}$ is set, $\forall k = 1, \ldots, m$. Nevertheless, we observe that this constraint is a bit too strong since, when two consecutive tops t_k and t_{k+1} are such that $d_{t_{k+1}} - r_{t_k} < p_{t_{k+1}} + p_{t_k}$, the MIP is unfeasible (*i.e.*, there does not exist any feasible sequence that keeps both t_k and t_{k+1} on time). For avoiding this infeasibility, the binary variable y_{t_k} is introduced (see constraint (6.8)): y_{t_k} equals 1 if the processing time of t_k is ignored, 0 otherwise. Therefore, the $\sum U_j$ criterion can easily be expressed using the binary variables y_{t_k}, $x^+_{u(i),i}$ and x^-_{ki} since, if $y_{t_k} = 1$, Top t_k is late and, if $\sum_{k=u(j)}^{v(j)} (x^-_{jk}) + x^+_{u(j)k}) = 0$, non-top job j is late.

From now the focus is on a particular SMSP where any pyramid P_k, $\forall k = 1, \ldots, m$ is said *perfect*, *i.e.*, $\forall (i,j) \in P_k \times P_k$, $(r_i \geq r_j) \Leftrightarrow (d_i \leq d_j)$, *i.e.*, the execution intervals of the jobs belonging to P_k are included each inside the other. By extension, when all the pyramids are perfect, the corresponding SMSP will be said perfect. For this special case, the following theorem is proved:

Theorem 4 *Given a perfect SMSP V, the master-pyramid sequence $\sigma_\Delta(V)$ characterizes the complete set of sequences being dominant for the $\sum U_j$ criterion.*

According to Theorem 4, when all the pyramids are perfect, the set of sequences in the form $\alpha_1 \prec t_1 \prec \beta_1 \cdots \prec \alpha_m \prec t_m \prec \beta_m$ is dominant for the $\sum U_j$ criterion. The problem can be optimally solved by the following MIP:

$$\min \; z = \sum_{\{j \in V \setminus \{t_1 \ldots t_m\}\}} (1 - x^+_{jk} - \sum_{k=u(j)}^{v(j)} (x^-_{jk})) + \sum_{k=1}^{m} y_{t_k}$$

s.t.
$$\begin{cases}
R_k \geq r_{t_k} + \mathbf{y_{t_k}}(\mathbf{r_{n_k}} - \mathbf{r_{t_k}}) & , \; \forall k \in [1\ m] & (7.1) \\
R_k \geq r_i + (\mathbf{1} - \mathbf{x}^+_{u(i),i})(\mathbf{r_{n_k}} - \mathbf{r}_i) & & \\
\qquad + \sum_{\{j \in P_k | r_j \geq r_i\}} p_j x^+_{kj} & , \; \forall k \in [1\ m], \forall i \in P_k & (7.2) \\
R_k \geq R_{k-1} + \sum_{\{j \in P_{k-1}\}} p_j x^-_{(k-1)j} + p_{t_{k-1}} & & \\
\qquad + \sum_{\{j \in P_k\}} p_j x^+_{kj} & , \; \forall k \in [2\ m] & (7.3) \\
D_k \leq d_{t_k} + \mathbf{y_{t_k}}(\mathbf{d_{n_k}} - \mathbf{d_{t_k}}) & , \; \forall k \in [1\ m] & (7.4) \\
D_k \leq d_i + (\mathbf{1} - \mathbf{x}^-_{v(i),i})(\mathbf{d_{n_k}} - \mathbf{d}_i) & & \\
\qquad - \sum_{\{j \in P_k | d_j \leq d_i\}} p_j x^-_{kj} & , \; \forall k \in [1\ m], \forall i \in P_k & (7.5) \\
D_k \leq D_{k+1} - \sum_{\{j \in P_{k+1}\}} p_j x^+_{(k+1)j} - p_{t_{k+1}} & & \\
\qquad - \sum_{\{j \in P_k\}} p_j x^-_{kj} & , \; \forall k \in [1\ (m-1)] & (7.6) \\
\sum_{k=u(i)}^{v(i)} (x^-_{ki}) + x^+_{ki} \leq 1 & , \; \forall i \in P_k & (7.7) \\
D_k - R_k \geq p_{t_k}(1 - y_{t_k}) & , \; \forall k \in [1\ m] & (7.8) \\
y_{t_k} \;,\; x^-_{ki} \;,\; x^+_{u(i),i} \in \{0,1\} & , \; \forall k \in [1\ m], \forall i \in P_k & \\
D_k \;,\; R_k \in \mathbb{Z} & , \; \forall k \in [1\ m] &
\end{cases}$$

with:

- $r_{n_k} = \min_{\{j \in P_k\}} r_j$;
- $d_{n_k} = \max_{\{j \in P_k\}} d_j$;

This MIP differs from the above one only by the addition of the terms in bold that allow to deactivate the constraints of type (7.1), (7.2), (7.4) or (7.5) when some jobs are late. For instance, if t_k is late, *i.e.*, $y_{t_k} = 1$, the term $y_{t_k}(r_{n_k} - r_{t_k})$ ($y_{t_k}(d_{n_k} - d_{t_k})$ resp.) deactivates the constraint (7.1) (the constraint (7.4) resp.). Indeed, we know that the inequality $R_k \geq r_{n_k}$ (resp. $D_k \leq d_{n_k}$) is obviously always verified. Similarly, in the case where $i \notin \alpha_k$ ($i \notin \beta_k$ resp.), the term $(1 - x^+_{u(i),i})(r_{n_k} - r_i)$ ($(1 - x^-_{v(i),i})(d_{n_k} - d_i)$ resp.) deactivates the constraint (7.2) (the constraint (7.5) resp.). Note that the deactivation of constraints allows to ensure that only the constraints that concern the on-time jobs are taken into account.

In a minimization problem, a lower bound is obtained by relaxing some constraints and optimally solving the relaxed problem. Moreover, for any problem, we know that it is always possible to decrease the r_j values (or increase the d_j values) of some jobs in order to make the pyramids perfect, *i.e.*, such that $\forall (i,j) \in P_k \times P_k$, $(r_i \geq r_j) \Leftrightarrow (d_i \leq d_j)$, $\forall k = 1, \ldots, m$. Doing so, a relaxed problem can be optimally solved by the above ILP. Finding an optimal sequence for the relaxed problem gives a lower bound to the general $1|r_j|\sum U_j$ problem..

6 Conclusion

Designing MIP models for solving efficiently basic SMSPs is of interest since MIP approaches are often adaptable for dealing with new constraints or new objective. As a proof of this statement, this paper shows how an original MIP model, proposed for searching the most feasible sequence is used for solving the $1|r_j|L_{\max}$ problem, and is also adapted for dealing with the more complex $1|r_j|\sum U_j$ problem. Since the analytical dominance condition used for designing the MIP formulation of the former problem is valid for the $\sum U_j$ criterion only under some restrictions (tops are not late), only an upper bound can be computed. In the particular case where the considered SMSP is *perfect* (see Section 5), the paper gives a MIP model that allows to directly find the optimal $\sum U_j$ value. Since it is always possible to relax the release dates or the due dates of any SMSP in order to make it perfect, this MIP also allows to formulate a lower bounds.

References

1. Baptiste P., Peridy L and Pinson E., "A Branch and Bound to Mininimze the Number of Late Jobs on a Single Machine with Release Time Constraints", European Journal of Operational Research, 144 (1), pp1-11 (2003).
2. Briand C., La H.T., Erschler J. "Une approche pourl'ordonnancement robuste de tâches sur une machine", *4ème Conférence Francophone de MOdélisation et SIMulation* (MOSIM'03), pp 205-211, Toulouse, France, 2003 (*in french*).
3. Briand C., Ourari S. "An efficient ILP formulation for the single machine scheduling problem", RAIRO-Operation Research, Vol.44 No 1, pp61-71 (2010).
4. R. M'Hallah, R.L. Bulfin "Minimizing the weighted number of tardy jobs on a single machine with release dates", European Journal of Operational Research, 176, pp727-744 (2007).
5. Erschler J., Roubellat, F., Vernhes J.-P, "Characterizing the set of feasible sequences for n jobs to be carried out on a single machine", *European Journal of Operational Research*, Vol. 4, pp189-194, 1980.
6. Carlier, J., "The one-machine sequencing problem", 1982, European Journal of Operational Research, 11, 42-47.
7. Erschler, J., Fontan, G., Merce, C., Roubellat, F.,"A New Dominance Concept in Scheduling n Jobs on a Single Machine with Ready Times and Due Dates", Operations Research, Vol. 31, pp114-127 (1983).
8. Dauzère-Pérès S., Sevaux M., "An exact method to minimize the number of tardy jobs in single machine scheduling", Journal of Scheduling, Vol. 7, No 6, pp405-420 (2004).
9. Michael R. Garey, David S. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness", W. H. Freeman and Company (1979).
10. Lenstra J.K., Rinnooy Han A.H.G , Brucker P., "Complexity of machine scheduling problems", Annals of Discrete Mathematics, vol. 1, pp343-362 (1977).
11. Ourari S., Briand C. "Minimizing the number of tardy jobs in single machine scheduling using MIP" (MISTA09), Dublin, Irland, (2009).

Posters

Blood cell images recognition using clustering algorithms: a survey

Nadjia Khatir, Safia Nait Bahloul
Department of Computer science
Es-Senia University
Oran 31000 Algeria
khatirnadjia@gmail.com, nait1@yahoo.fr

Abstract. In Algeria it is estimated that there are more than 109 new cases of Chronic Myeloid Leukemia (CML) are being diagnosed every year and the overall incidence rate is 0.34(According to the statistics taken from the Algerian Journal of Hematology 2010) and its incidence worldwide varies from 0.7 in Sweden and China to 1.7 in Switzerland and the United States. The shape and the number of Leukocytes in the blood image is the important information for the diagnosis of Leukemia. However since the Leukocytes anomalous were surrounded by numerous red cells and platelets, it is difficult to detect them clearly. Therefore, clustering recognition method automatically detecting these elements is needed to help doctors analyze the blood cell images.

Keywords: Blood cell; Medical Image; clustering algorithm; Content-based Image recognition

Un Modèle Générique pour les Travaux Pratiques à Distance

Mohamed Ramdane, Rachid Ahmed-Ouamer

Laboratoire de Recherche en Informatique LARI, Département d'Informatique
Université Mouloud Mammeri de Tizi-Ouzou, 15000 Tizi-Ouzou, Algérie
ramdane.moh27@yahoo.fr, rachid.ahmedouamer@yahoo.fr

Abstract. A l'instar des travaux pratiques classiques, les travaux pratiques à distance (télé-TP) sont indispensables aux environnements de téléformation, notamment dans les disciplines scientifiques et techniques. En effet, les services actuels de formation à distance reposent essentiellement sur des télé-cours, des télé-TD ou des télé-projets sans une réelle activité pratique. La mise en place de télé-TP se heurte à des difficultés techniques et organisationnelles. Les récents travaux de recherche en télé-TP tentent de donner un nouvel élan à cette nouvelle discipline en proposant des solutions et des architectures génériques favorisant la réutilisation et l'interopérabilité. Dans ce papier est proposé un modèle générique pour les télé-TP conformément aux standards de l'e-learning.

Etude des critères de distribution de l'algorithme OPTICS

Souhila Ghanem[1], T ahar Kechadi[2], A.Kamel Tari[1]

[1]Faculté des Sciences et des Sciences de l'Ingénieur Département d'Informatique,
Laboratoire des Mathématiques Appliqués Université de Béjaia
souhila.ghanem@gmail.com, atari@mail.cerist.dz

[2] School of Computer Science and Informatics
University College Dublin, Belfield, Dublin 04, Ireland
Tahar.kechadi@ucd.ie

Résumé.

Les collections de données deviennent de plus en plus volumineuses et dans la majorité des cas ne résident pas dans un emplacement centralisé, ce qui complique l'application des techniques de Data Mining sur des données distribuées et souvent hétérogènes. La plupart des algorithmes distribués se basent sur l'agrégation des modèles produits de manière locale et notre approche rentre dans ce cadre. Nous présentons une nouvelle approche distribuée qui prend en compte certaines caractéristiques de l'algorithme OPTICS. Les données seront traitées localement sur chaque site en exécutant l'algorithme OPTICS séquentiel pour produire des clusters à partir des données locales, ensuite nous construisons les clusters globaux de manière hiérarchique, pour cela les représentants de chaque cluster sont calculés et envoyés aux sites d'agrégats. Les représentants d'un cluster sont les points de sa bordure. Les clusters sont régénérés afin de tester les recouvrements pour enfin construire les clusters globaux. Notre but est de concevoir une approche distribuée sans tenir compte de la version séquentielle car cette dernière génère beaucoup de communications. Notre approche tente de minimiser les communications, maximiser le parallélisme et d'équilibrer la charge sur les différents sites. Cette technique est évaluée et comparée à la version séquentielle en utilisant des jeux de données artificiels.

Assigning a Neuronal Approach for the Treatment of multisource image

RIFFI Mohamed Amine[1], FIZAZI Izabatene Hadria[2]
[1]Laboratory Signal-Image-Word (SIMPA), Dept.Informatique, Faculty of Science, University of Science and Technology of Oran -USTO, Algérie. BP 1505, Oran El Mnaouer 31000.

{ r.amine78@gmail.com ; hadriafizazi@yahoo.fr}

Abstract

Satellite images have aroused much interest in various fields, because of the overall coverage of the earth's surface and the repetitiveness of the data. To operate properly and effectively the amount of information contained in the satellite images, several techniques have been developed. Supervised classification is one of the techniques used to exploit the maximum information contained in these images, to represent them in an understandable and interpretable. In our article we used a neural approach based on neural network multilayer perceptron (MLP) for classification of multisource satellite imagery. This approach is applied to a satellite image showing the region of Oran (Western Algeria), taken by Landsat TM5 in 1993. "This study area is characterized by its diversity.

Keywords: Neural network, Multisource images, Supervised Classification, Remote Sensing, Multi Layer Perceptron.

-

Une architecture basée sur les composants fractals pour l'intégration des applications d'entreprises

Soumia Bendekkoum , Mahmoud Boufaida

Laboratoire LIRE , Université Mentouri Constantine, Algérie
{bendekkoums, boufaida_mahmoud}@yahoo.fr

Résumé. L'évolution des modèles d'organisation d'entreprises est aujourd'hui guidée par l'intensification de la concurrence et la performance des communications entre les applications. Dans ce contexte, les entreprises ont pris conscience de l'importance de l'orientation service, notamment en termes de standardisation, de facilité de communication et de composition des services dans différentes plateformes. Cependant, les approches existantes et les outils de spécification des modèles d'architecture orientée service ne permettent pas aux applications fournisseurs de services de s'adapter aisément aux changements continus de l'environnement et des besoins des clients. Par ailleurs, les applications clients ne s'adaptent pas aux changements de la logique métier des applications fournisseurs dynamiques. Plusieurs travaux ont introduit les principes de la programmation orientée aspect (POA) pour adapter les services web, en utilisant le concept principal de la POA (*tisseur*) et pour ajouter de nouvelles fonctionnalités à la logique métier de service. Néanmoins, ces travaux présentent des limitations vis-à-vis de l'ajout des préoccupations fonctionnelles. Dans ce projet, nous proposons une architecture d'intégration d'applications basée sur les composants fractals, dans laquelle nous bénéficions des caractéristiques importantes de la reconfiguration dynamique du fractal pour modifier le comportement d'un service simple (ou encore une composition de services), en utilisant les principaux concepts du modèle des composants fractals (*contenu*, *interface interne*, *interface externe*). Nous présentons également un exemple illustratif montrant la mise en place de notre architecture proposée, ainsi que le cycle d'un service adapté. Nous visons à faciliter la programmation et le déploiement du modèle orienté services, considéré comme un point nécessaire pour les organisations, en utilisant les modèles hiérarchiques de spécification des composants fractals.

Mots clés: Intégration d'applications, orientation services, Architecture-orientée services (SOA), modèle de composants fractals, services web.

Medical information extraction from clinical reports: a Boolean approach

Fatiha Barigou[1], Bouziane Beldjilali[2], Baghdad Bouziane[3]

Equipe Simulation, Intégration et Fouille de données
Université d'Oran
BP 1524, El M'Naouer, 31 000 Oran, Algérie
[1] fatbarigou@gmail.com, [2] bouzianebeldjilali@yahoo.fr [3] atmani.baghdad@gmail.com

Abstract. Much of the information that exists in patient clinical reports is difficult to access, as it is often in unstructured text form. To make easy access and search, our research aims to develop a system for extracting information from unstructured medical text. In a previous work, a rule-based approach is applied to a clinical reports corpus of infectious diseases to extract structured data in the form of named entities and properties. We propose in this paper, the use of a Boolean inference engine based on cellular automata to do extraction. Our motivation to adopt this Boolean modeling approach is twofold: first optimize storage, second reduce the response time while searching the entity class. The system proposed for this extraction task consists of two modules. The first one is responsible for building the Boolean knowledge base following the principle of the cellular automaton CASI. The second module uses the Boolean inference engine of CASI to classify named entities. To extract named entities from medical reports written in free natural language, our contribution adopts the following approach: (i) manual construction of named entities classification rules; (ii) Boolean Modeling of constructed rules; (iii) Linguistic Analysis of clinical reports for extracting nominal phrases with their morphosyntactic and semantic properties; (iv) Boolean Inference for classifying nominal phrases into different classes (e.g. person, date, symptom, disease...).

Keywords: Clinical reports, information extraction, rule based approach, Boolean inference engine.

Méthode adaptée pour la résolution d'un problème de programmation quadratique convexe à variables mixtes

Abdelhek Laouar et Mohand-Ouamer Bibi

Laboratoire LAMOS, Université de Béjaia, 06000 Béjaia, Algérie.

Résumé Dans cet article, on propose une méthode adaptée de résolution d'un problème quadratique convexe à variables mixtes. Cette méthode généralise la méthode directe de support utilisant la métrique du simplexe. Les expérimentations numériques comparant notre méthode avec les méthodes classiques sur des problèmes tests générés aléatoirement montrent que la méthode proposée donne une amélioration certaine par rapport à celle du support. Elle s'avère aussi compétitive avec les autres méthodes d'activation des contraintes, et celles de points intérieurs pour les problèmes de taille moyenne.

Mots clés :programmation quadratique convexe, méthodes de points intérieurs, méthode de support, critère de suboptimalité, méthode adaptée de support, variables mixtes.

A New Segmentation Algorithm for Off-Line Handwriting Arabic Character

Kef Maâmar[1], Chergui Leila[2], and Chikhi Salim[3]

[1] Department of Computer Sciences, University Hadj Lakhdar, Algeria
[2] Department of Computer Sciences, University Larbi Ben Mhidi, Algeria
[3]Department of Computer Sciences, University Mentouri, Algeria

Abstract. The segmentation and recognition of Arabic handwritten text has been an area of great interest in the past few years. In many handwritten word recognition systems, segmentation is a very delicate stage, especially for Arabic handwriting script because of the fact that words in Arabic may have sub-words which are separated by spaces. This paper describes a new segmentation algorithm for handwritten Arabic characters using increasing windows to suggest the position of a possible segmentation point. The nominated point is confirmed or rejected by a neuronal classifier which evaluates the proposition. In our approach we choose to use invariant Hu moments as feature vectors exploited by a Multilayer Perceptron to learn and classify cursive characters. We performed several experiments using IFN/ENIT database to test our system. It has achieved a high segmentation rate of 87.55%.

Keywords: Arabic handwriting, Segmentation, Character recognition, Hu moments, Multilayer Perceptron, Classification.

Auto-organisation dans les réseaux Ad hoc

Ali Kies[1], Zoulikha Mekkakia Maaza[2], Redouane Belbachir [3],
Département d'informatique,
Université des Sciences et de la Technologies d'Oran
USTO, BP 1505 El M'Naouar, Oran, Algeria
[1]kies_ali@yahoo.fr, [2]mekkakia@univ-usto.dz,
[3]belbachir_red@yahoo.fr

Résumé Les réseaux ad hoc sont des réseaux spontanés sans fil. Ils réunissent un grand nombre d'objets communicants sans fil, sans infrastructure et tous ces objets sont mobiles. Par conséquent, la conception des protocoles de routage représente un problème complexe. Une des solutions à ce problème, passe par la mise en place de topologies adaptées au routage.

Mots-clés : Ad hoc, Routage, Topologie virtuelle, CDS, Auto Organisation.

Generalized hypertree decomposition for solving constraint satisfaction problems

Habbas Zineb[1] and Amroun Kamal[2]

[1] Université de Metz, Laboratoire d'Informatique Théorique et Appliquée, France,
zineb@univ-metz.fr
[2] Université de Béjaïa, Département d'Informatique, Algérie,
k_amroun25@yahoo.fr,

Constraint satisfaction is NP-Complete in general. However, there are various classes of constraint satisfaction problems (CSPs) that can be solved in polynomial time. Some of them can be identified by exploiting their structural properties. It is well known that acyclic CSPs can be solved efficiently. Different methods exist for converting CSP instances to their solution equivalent instances with acyclic structure. Among these methods *Generalized Hypertree Decomposition* and *fractional hypertree decomposition* have been proved to be the most general methods.

We propose the method called HD2_DBT for Dual Backtracking algorithm guided by an order induced by the computed generalized Hypertree Decomposition of the instance. The main idea of this approach is that for each node n_i of the hypertree, we compute only one join tuple t_i (which is a solution of the subproblem at the current node n_i) which is compatible with the tuple t_j already computed for the subproblem at it's parent node n_j.
Also, in this method, once we find that a tuple (solution) for the subproblem at the current node n_i is compatible with the solution of the subproblem at it's father node, we filter the constraints relations in each son node r of the current node : We remove from all the constraints relations in each son node r all the tuples that will not participate to a solution. Different heuristics and implementations are presented showing its practical interest.

UML Extensions for Security Requirements Analysis of IS

Salim CHEHIDA
Department of Informatics, University of Mostaganem
Doctoral school STIC of Mostaganem
Mostaganem, Algeria
salimchehida@yahoo.fr

***Abstract*—** With the fulgurating growth of the world of telecommunications, pulled by Internet and stimulated by the penetration of transmission technologies, the problems of processes and data security have currently become of paramount importance. The transactions made through the network can be intercepted, more especially since adequate legislation has not yet been fully enforced on the Internet. The functional specification of the information systems is not sufficient, the design and the realization of these systems must take into account, in addition to the functional needs, the various requirements of security. Taking into account the various constraints of security (Availability, Authentication, Integrity, Secrecy, Non-Repudiation, etc.) in the modeling process constitutes one of the principal challenges for the designer of these systems. UML is the standard language for the modeling of the multiple views of an information system by using the various mechanisms of extension. *UMLsec* is an extension of UML proposed by J.Jürjens (Munich University of Technology) that includes, at the conceptual level, profiles for secure systems development. After the definition of UML extension mechanisms and *UMLsec* profiles, this paper proposes new extensions for the modeling of the security requirements of information systems. The *secure context model* and the *security cases model* for the specification of the security needs, the *critical scenarios model* consist in describing the interactions or the actions which involve a risk and the *secure interactions of objects model* for the specification of the security constraints on the messages exchanged by objects. In the analysis model, we defined security properties on the data. At last, for the modeling architecture, the *protected hardware configuration model* allows to express the implementation constraints at the physical level with the integration of the prevention tools in order to fulfill the security requirements.

Keywords: *IS, Computer Security, Modeling, UML, and UMLsec.*

Application du Réseau de Neurones Multicouche à la classification d'une image Hyperspectrale

FIZAZI Izabaten Hadria, BELKADI Abdellah
Laboratoire SIMPA (SIgnal IMage PArole),
Département Informatique,
Faculté des Sciences,
Université des Sciences et Technologie d'Oran,
BP. 1505 EL M'Naouer Oran 31000, Algérie
hadriafizazi@yahoo.fr , abdellahbelkadi@yahoo.fr

Résumé :

La télédétection a montré ses potentiels dans l'acquisition de données et l'extraction d'informations nécessaires pour la classification. En fait, le besoin aux informations précises et réalistes est nécessaire. Récemment, l'imagerie hyperspectrale a été employée dans différentes applications, elle peut nous aider efficacement dans la classification des objets. On s'intéresse aux images hyperspectrales parce qu'elles véhiculent des quantités de données importantes dont nous avons besoin.

Le traitement d'images hyperspectrales est la généralisation de l'analyse des images couleurs, à trois composantes rouge, vert et bleu, aux images multivariées à plusieurs dizaines ou plusieurs centaines de composantes. Dans un sens général, les images hyperspectrales ne sont pas uniquement acquises dans le domaine des longueurs d'ondes mais correspondent à une description d'un pixel par un vecteur. Le vecteur associé à chaque pixel est appelé spectre.

La classification est l'attribution des pixels d'une image à des classes spécifique. Cette attribution a besoin d'un certain degré d'abstraction pour pouvoir extraire des généralités à partir des exemples dont on dispose. Dans ce cadre on a utilisé le réseau de neurone Perceptron multicouche où l'apprentissage se fait par l'algorithme de Rétropropagation du Gradient. Cet algorithme a besoin d'introduire à priori le nombre et la connectivité des unités cachées et déterminer les poids initiaux des connexions.

Afin d'obtenir une représentation de l'image avec un nombre restreint de canaux. On a fait une réduction de dimension spectrale de la zone d'étude. A partir de cette image de plus faible dimension, nous avons effectué une classification supervisée pour grouper les pixels en classes spectralement homogènes.

D'après la comparaison des résultats obtenus de la méthode utilisée avec les données de vérité d'un terrain dont on en a peu d'informations et l'évaluation du taux d'erreur de classification réalisée par différentes structures de réseaux de neurones pour différentes ensembles d'échantillons, on constate que le réseau de neurone Perceptron multicouche nous donne une classification des images hyperspectrales mieux que des images à trois composants RVB, et c'est grâce à l'apprentissage supervisé.

Mots clés : Télédétection, Classification, Image Hyperspectrale, Réseau de neurones, Rétropropagation.

Conception d'un système de diagnostic des turbines à gaz par raisonnement à partir de cas

F.Anguel, M.Sellami

Université Badji Mokhtar. BP 12, 23000 Annaba, Algérie

fanguel@yahoo.fr sellami@lri-annaba.net

Abstract. La maintenance corrective des équipements est une préoccupation importante sur les sites industriels. Une des tâches les plus délicates de cette maintenance est le diagnostic des pannes. La difficulté majeure survient de l'indisponibilité des techniciens expérimentés «experts du domaine » pour prendre en charge toutes les activités de maintenance. C'est pourquoi les entreprises ont pris conscience de l'importance de leur capital immatériel détenu par leurs employés.

Dans ce travail nous présentons une démarche de capitalisation des connaissances de diagnostic des turbines à gaz associée à un mécanisme de raisonnement à partir de cas (RàPC). A partir de l'étude du processus de maintenance des turbines à gaz, des concepts de sûreté de fonctionnement et de la pratique des experts de maintenance des turbines à gaz, nous avons établi deux modèles : le modèle de domaine développé à travers une ontologie du domaine de maintenance en se servant de l'éditeur d'ontologie Protégé et un modèle de raisonnement associé à la base de cas. Un cas dans notre base de cas possède une représentation orienté objet structuré en deux parties : partie problème et partie solution. La partie problème est à son tour composée de deux champs « localisation » et « identification » par contre la partie solution comporte le champ « détection ». Chaque champ est décrit par un ensemble de descripteurs. Pour le raisonnement suite à une demande d'intervention on procède à l'élaboration du cas et cela par le remplissage d'un formulaire par l'opérateur de maintenance, à partir de ce formulaire on extrait les descripteurs pertinents du cas. Une fois le cas cible est élaboré on procède à la remémoration des cas sources en calculant le degré de similarité du cas cible avec les différents cas sources. Nous procédons par la suite à la réutilisation des solutions des problèmes sources retenues comme similaires. Les solutions peuvent être modifiées en changeant les paramètres et nous parlons ici d'une phase d'adaptation. Dans certains cas la solution est proposée sans changement. L'opérateur de la maintenance exécute ainsi le mode opératoire proposé. Dans le cas ou le résultat est satisfaisant ce nouveau cas est mémorisé dans la base de cas, ce qui réfère à la phase « mémorisation» du cycle de RàPC

Actuellement nous développons la base de cas, puis nous allons ajouter d'autres fonctionnalités à ce système notamment l'assistance des experts distants dans le processus de maintenance.

Fuzzy knowledge-intensive case based classification applied in the automatic cardiac arrhythmias diagnosis

A. Khelassi*,**, MA Chikh*,***

*Medical Engineering Lab, Abou Bakr Belkaied University, Tlemcen, Algeria

**Department of informatics, Abou Bakr Belkaied University, Tlemcen, Algeria

*** Department of electronics, Abou Bakr Belkaied University, Tlemcen, Algeria

Khelassi.a@gmail.com, ma_chikh@mail.univ-tlemcen.dz

Abstract. Case Based Reasoning CBR is an intelligent approach inspired from many disciplines. It draws the human reasoning model. It consists to use the prior expertise to resolve a new problem. In this work we have developed an original fuzzy knowledge intensive case based reasoning system dedicated for the automatic cardiac arrhythmias diagnosis. This application combines between many intelligent approaches and algorithms for satisfying the biomedical needs which are the accuracy and the performance. Trough the system criteria and some empirical experiments we can concludes that the classification system achieves such average accuracies and performance better than most of the current state-of-the-art approaches.

Keywords: Knowledge Intensive Case Based Reasoning, fuzzy sets, multi agents system, cardiac arrhythmia diagnosis.

Amélioration d'un algorithme évolutionnaire quantique pour la classification non supervisée des données

Ramdane Chafika[1], Mohamed khireddine Kholladi[2]
[1] Département d'informatique, Université de Skikda,
Laboratoire MISC, Université de Constantine,
E-mail: ramdanechafika@yahoo.fr
[2]Département d'informatique,Laboratoire MISC, Université de Constantine,
E-mail: kholladi@yahoo.fr

Résumé. La classification non supervisée des données est une tâche essentielle pour plusieurs domaines comme la fouille des données et la reconnaissance des formes. Elle vise à découvrir des groupes cohésifs dans des grands jeux de données. Pour résoudre ce problème, une approche évolutionnaire quantique (QEAC) a été proposée [1]. Deux caractéristiques principales caractérisent QEAC: la représentation quantique de l'espace de recherche et la dynamique évolutionnaire quantique. QEAC consiste à optimiser une mesure de qualité de groupes pour trouver une partition du jeu de données. Durant le processus d'optimisation, des opérations quantiques sont appliquées sur la représentation quantique. La supériorité de QEAC par apport à d'autres algorithmes de type génétique et évolutionnaire quantique a été montrée dans [1], mais cette approche QEAC est parfois pénalisée par la lenteur de la convergence due au coinçage dans un minimum local. Dans ce papier, nous apportons des modifications et des améliorations à QEAC afin d'élargir l'exploration de l'espace de recherche. Deux nouvelles opérations ont été ajoutées à l'approche améliorée IQEAC et deux autres opérations ont été modifiées. Les deux algorithmes QEAC et son amélioration IQEAC incluent les opérations : la mesure, l'interférence constructive et destructive, la régénération, la migration globale et locale, mais à la différence de QEAC, IQEAC inclut deux nouvelles opérations: le croisement et une opération d'éloignement au minimums locaux. Les paramètres de l'interférence constructive et la migration globale ont été également modifiées. Les résultats sur des jeux de données réels et synthétiques montrent que les modifications apportées à l'approche améliorent la qualité et la vitesse de la convergence.

1. Ramdane, C., Meshoul, S.,Batouche, M. and Kholladi, M-K.:A quantum evolutionary algorithm for data clustering, Int. J. Data Mining, Modelling and Management, Vol. 2, No. 4, pp.369–387, (2010).

Une approche hybride pour l'optimisation de l'apprentissage adaptatif

Riad Bourbia, Hamid Seridi, Mourad Hadjeris, Ali Seridi et Noureddine Gouasmi

LabSTIC (**Lab**oratoire des **S**ciences et **T**echnologies de l'**I**nformation et de la **C**ommunication), Université 08 Mai 45- BP 401 Guelma 24000 - ALGERIE

{Bourbia.riad, SeridHamid, M_Hadjeris, a_seridi}@yahoo.fr
n_gouasmi@hotmail.com

Résumé. L'un des principaux enjeux de l'apprentissage en ligne est l'autonomie de l'apprenant. L'elearning adaptatif va permettre d'améliorer l'utilisation des plates-formes en proposant des cours adaptés aux résultats, comportements, goûts... des apprenants, sans que ceux-ci en aient conscience. Dans le présent document, nous avons proposé une approche hybride basée sur l'utilisation de l'algorithme de colonies de fourmis pour recommander des parcours d'apprentissages qui correspondent le mieux aux profils des apprenants. Ces parcours sont constitués dynamiquement à partir d'objets pédagogiques élémentaires. L'algorithme de filtrage est utilisé afin d'organiser les apprenants dans des groupes en fonction des similarités afin d'accélérer le processus de recommandation. La mise à l'épreuve du système se fait par le biais des simulations qui seront faites d'abord pour obtenir un calibrage global des paramètres numériques. La méthode s'avère à la fois adaptative et robuste. Pour l'instant, l'expérience qui se déroule au département d'informatique de l'université de Guelma, n'est qu'a son début, mais en croisent les doigts pour qu'elle puisse évoluer correctement.

Mot clés: Elearning, Objet d'apprentissage, Apprentissage adaptatif, Colonies de fourmis, Filtrage collaboratif.

Construction d'entité consciente

Toualbia Ilyes[1] , Seridi Hamid[2,3],

[1] Departement de gestion, Université 08 Mai 45 Guelma, B.p. 401, Algérie,
[2] Departement d'informatique, Université 08 Mai 45 Guelma, B.p. 401, Algérie,
[3]CResTIC, EA 3804, Université de Reims, B.P 1035, 51687, Reims, Cedex, France,
{toualbiai, seridihamid}@yahoo.fr

Résumé. Le but de ce papier est de présenter l'état d'avancement de nos travaux de recherche. Ces derniers visent à utiliser le web en tant que ressource inépuisable d'informations et données pour recueillir selon des critères bien définis une base d'apprentissage. Cette dernière sera utilisée pour construire une entité « consciente » qui peut reconnaitre des objets dans l'environnement qui l'entoure.

Mots clés: Conscience artificielle, intelligence artificielle, apprentissage automatique, reconnaissance des formes, recherche d'informations.