



# Méthode Exacte Pour le Problème du Stable Multi-Objectif

Badjara Mohamed El Amine et Chergui Mohamed El-Amine

Laboratoire RECITS - USTHB - Bab Ezzouar - Alger

`mbadjara@usthb.dz` `mchergui@usthb.dz`

---

**Abstract:** An exact method per branch and bound is proposed for the generation of efficient vertex packing set in a graph that his vertices are valued with a weight's vector. She exploits the data structure of the problem to devid it, in an arborescence structure, to independent sub-problems of reduce size which can resolved with a general method for multi-objective integer linear problems.

**Keywords:** vertex packing in graph; integer linear program; branch and bound; multi-objective program; efficient solution.

---

**Résumé :** Une méthode exacte par séparation-évaluation est mise en œuvre pour la génération de l'ensemble des stables efficaces dans un graphe dont chaque sommet est valué par un vecteur poids. Elle exploite la structure des données du problème du stable pour le décomposer, dans une structure arborescente, en sous problèmes indépendants de tailles réduites, chacun pouvant être résolu par une méthode générale de résolution d'un problème multi-objectif linéaire discret.

**Mots clés :** Stable dans un graphe; programme linéaire en nombres entiers; séparation et évaluation; programme multi-objectif; solution efficace.

---

## 1 Introduction

Le problème du stable est un problème classique d'optimisation combinatoire connu pour être de type *NP-difficile* et il en est de même dans sa version multi-objectif, noté *MOISP*. Ce problème n'a pas reçu suffisamment l'attention des chercheurs à notre connaissance, et très peu de travaux existent seulement pour le problème général connu sous le nom "Set Packing Problem" dans le cas bi-objectif [2, 3]. Nous proposons une méthode exacte par séparation et évaluation pour trouver tous les stables efficaces de *MOISP*, qui exploite la structure de la matrice des contraintes du problème du stable avant de faire appel à une méthode générale dédiée à la résolution de problèmes d'optimisation multi-objectif linéaires en variables discrètes (*MOILP*). Nous montrons que notre méthode est plus performante que la méthode générale décrite par Chergui et al. dans [1] pour la résolution de *MOILP*.

## 2 Principe de la méthode

Commençant par donner le modèle général de *MOISP*. Soit  $G = (V, E, W)$  un graphe dont  $V$  est l'ensemble sommets,  $E$  l'ensemble des arêtes et  $W$  la matrice de poids des sommets ou chaque colonne  $j$  correspond à un vecteur de poids du  $j^{\text{ème}}$  sommet. Alors le modèle s'écrit :

$$P : \begin{cases} \text{Max} & Z(x) = Wx \\ & Ax \leq b \end{cases} ; x \in \{0, 1\}^n$$

où :  $A$  est la matrice d'incidence transposée du graphe  $G$ .

$b$  le vecteur du second membre dont tous ses composantes valent 1.

$n$  le nombre de sommets du graphe  $G$ .

Un vecteur  $Z(x) \in \mathbb{R}^p$  domine un autre vecteur  $Z(y) \in \mathbb{R}^p$  si  $Z(x) \geq Z(y)$  et  $Z(x) \neq Z(y)$ . Dans ce cas, la solution  $y$ , qui est un stable dans le graphe  $G$ , n'est pas efficace pour  $(P)$ . Le point idéal  $I$  de  $(P)$  est un point de l'espace des critères et a pour coordonnées  $(Z_1^*, \dots, Z_p^*)$ , où  $Z_k^*$  est le poids maximum d'un stable de  $G$  relativement au critère  $k$ ,  $k = (1, p)$ . Généralement, le point  $I$  n'est pas réalisable [4].

Le problème  $(P)$  a une structure particulière, la matrice des contraintes est creuse, chaque ligne contient exactement deux un (01), et le vecteur du second membre est égale à un (01) pour chacune de ses composantes.

Le principe de la méthode réside dans l'exploitation de cette structure et l'utilisation du principe de la séparation et évaluation. Initialement, on supprime tous les sommets ayant leurs poids négatifs ou nuls avec au moins une composante strictement négative car ce type de sommets ne peut appartenir à aucun stable efficace. Viennent ensuite, les étapes suivantes :

**A. Opération de tri :** L'ordre de traitement des sommets du graphe  $G$  dans l'arborescence de recherche ce fait sur la base d'une opération de tri de ces derniers préalablement établie. Pour cela, plusieurs opérations de tri sont envisageables et nous présentons une opération possible basée sur la combinaison du tri de la domination des poids des sommets et le tri décroissant des sommets.

A chaque sommet  $j$  de  $G$  est associé un ensemble  $D(j)$  de sommets tels que  $Z(j)$  domine  $Z(t)$ ,  $\forall t \in D(j)$ . L'ordre de traitement des sommets  $j$  de  $G$  dans l'arborescence se fait selon l'ordre décroissant de la quantité  $\alpha(j)$  tel que :

$$\alpha(j) = |D(j)| + |\Gamma(j)|$$

où  $\Gamma(j)$  est l'ensemble des sommets adjacents au sommet  $j$ . D'autres opérations de tri ont été aussi testées, mais les meilleurs résultats sont obtenus moyennant l'opération décrite ci-dessus.

**B. Séparation :** Une fois que le tri des sommets est réalisé, la séparation est faite sur la base de la règle "retenir" ou "ne pas retenir" un sommet  $j$  dans des solutions stables efficaces ( $x_j = 1$  ou  $x_j = 0$ ). En outre, le fait de retenir un sommet aura un effet de propagation de contraintes sur ses voisins dans ce sens que; tous ses voisins seront automatiquement rejetés et beaucoup de contraintes du programme *MOISP* deviennent saturées, ce qui a pour premier effet bénéfique la diminution de la taille de *MOISP*. Suite à cet "effet domino", le deuxième effet bénéfique réside dans l'apparition possible de sommets isolés dans le sous graphe obtenu.

Le cas correspondant à "ne pas retenir" un sommet entraîne sa suppression du graphe courant et dans ce cas aussi, le sous graphe obtenu peut admettre des sommets isolés.

La règle générale de traitement d'un sommet  $j$  isolé est la suivante :

- Si tous les poids  $w_k^j$  du sommet isolé  $j$  sont positifs ou nuls alors,  $x_j = 1$ .
- S'il existe un poids  $w_k^j$  négatif, alors on aura deux cas à traiter : retenir le sommet  $j$  ( $x_j = 1$ ) ou rejeter le sommet  $j$  ( $x_j = 0$ ).

**C. Evaluation :** L'évaluation est faite en approximant par  $\tilde{I}_l$  le point idéal local  $I_l$  de chaque sous problème correspondant à un nœud  $l$  de l'arborescence de recherche. Initialement, l'approximation  $(\bar{Z}_k)$  du maximum de chaque critère  $k$  est faite par défaut sur le vecteur  $X = (x_j)_{j=1..n}$  avec  $x_j = 1$  si le poids  $w_k^j \geq 0$  et  $x_j = 0$  si le poids  $w_k^j < 0$ , c'est-à-dire  $\bar{Z}_k = \sum_{j \in \{1, \dots, n\}, w_k^j} w_k^j$ .

Lors de la séparation, si un sommet  $j$  est retenu dans une branche de l'arborescence, la mise à jour de  $(\bar{Z}_j)$  est faite par l'ajout des poids négatifs associés au sommet  $j$  à  $(\bar{Z}_j)$ . Si le sommet  $j$  n'est pas retenu, la mise à jour de  $(\bar{Z}_j)$  est faite par la soustraction des poids positifs associés au sommet  $i$  à  $(\bar{Z}_j)$ . De cette façon, on montre que  $\tilde{I}_l$  domine  $I_l$ .

**D. Principe général :** Initialement, on supprime tous les sommets dont le vecteur poids est négatif ou nul, avec au moins une composante strictement négative, pour ensuite évaluer le premier nœud de l'arborescence. Le processus va se poursuivre par la séparation et l'évaluation de chaque nœud de l'arborescence, sachant qu'on a fixé au préalable le nombre de niveaux mesurant la profondeur de l'arborescence. L'effet domino a une incidence directe sur la réduction considérable de la taille du problème. Dans le pire des cas, pour réduire la matrice des contraintes à une matrice nulle, on aura  $2^n$  feuilles dans l'arborescence quand la densité du graphe devient nulle (dans ce cas le graphe est un ensemble de sommets isolés). Donc, la fixation de la

profondeur de l'arborescence a pour effet d'éviter l'exploration de  $2^n$  feuilles. Ainsi, il suffit d'exécuter une des méthodes connues dans la littérature pour la résolution des programmes MOILP sur chaque feuille qui correspond à un sous graphe de  $G$  de densité non nulle, donc un programme de taille réduite par rapport au programme initiale.

D'autre part, on a dit que la séparation a pour conséquence un effet domino sur la réduction de la matrice courante  $A$  des contraintes, correspondant à un sous-graphe  $G$ , cet effet est détaillé dans ce qui suit :

- Si un sommet  $j$  est retenu ( $x_j = 1$ ), sa colonne correspondante et toutes les lignes dans lesquelles le sommet  $j$  apparaît, sont supprimées de la matrice  $A$ . De plus, tous ses sommets adjacents dans  $G$  seront automatiquement supprimés et donc, toutes les colonnes qui leurs correspondent sont aussi supprimées de  $A$ .
- Si un sommet  $j$  est supprimé, on supprime sa colonne correspondante dans la matrice  $A$ . De plus, chaque ligne qui le contient dans  $A$  ne comportera plus qu'un seul élément non nul égal à un, correspondant à son sommet adjacent dans  $G$ .

On procédant ainsi, on peut avoir une colonne dans  $A$  avec un seul un (01), ce qui correspond à un sommet isolé  $j$  dans  $G$  obtenu après l'opération de réduction de la matrice  $A$ . Si tous ses poids sont positifs ou nuls avec au moins un poids strictement positif, alors on retient le sommet  $j$ . D'autre part, l'estimation  $\tilde{I}_l$  du point idéal  $I_l$  d'un sous problème correspondant à un nœud  $l$  de l'arborescence de recherche, a pour effet de sonder ce nœud si  $\tilde{I}_l$  est dominé par au moins une solution non dominée générée lors de la résolution de sous problèmes relatifs aux nœuds  $h$ ,  $h < l$ .

**E. Remarque :** Un cas reste à vérifier, si tous les poids  $w_i^j$  des sommets sont négatifs alors, chacun des sommets  $j$  de  $G$  dont le vecteur critère est non dominé, constitue un stable efficace. D'autre part, si tous les poids  $w_i^j$  des sommets sont égaux alors, le problème ( $P$ ) revient à déterminer tous les stables de cardinalité maximale et de fait, l'algorithme que nous proposons les génère tous.

### 3 Expérimentation numérique

La méthode des ensembles efficaces complets (EEC) décrite dans [1] et notre méthode par séparation et évaluation dédiée au problème du stable multi-objectif (BBMOISP) ont été mises en œuvre sous le langage de programmation MATLAB, en utilisant un PC Dual-Core, processeur 1.80 GHz et 1 Go de RAM. Les tests portent sur des instances générées aléatoirement avec  $n$  variables,  $n \in \{20, 25, 30, 35, 40, 50, 60\}$ ,  $m$  contraintes ( $m$  arêtes de  $G$ ),  $m \in \{100, 120, 150, 250, 350, 500, 1000, 1200\}$  et  $p$  fonctions à optimiser,  $p \in \{3, 7\}$ . Les coefficients des fonctions objectifs sont des coefficients entiers non corrélés répartis uniformément dans l'intervalle[0,9]. Pour chaque instance  $(n, m, p)$  fixée, vingt jeux de données sont générées et l'ensemble des stables efficaces a été généré pour chaque jeu de données. La colonne « MO » indique le nombre moyen et la colonne « MA » le nombre maximum de stables efficaces trouvés pour l'instance considérée. Sous les colonnes« EEC » et « BBMOISP », les colonnes « MIN », « MAX » et « MOY » concernent le temps CPU en secondes de chaque méthode. Notons d'abord, que les cases vides propres aux colonnes réservées à la méthode EEC indiquent que le temps CPU de cette dernière dépasse 2 heures et les calculs sont interrompus pour les instances correspondantes. D'autre part,

Instances	MO	MA	EEC			BBMOISP		
			MIN	MAX	MOY	MIN	MAX	MOY
(20,100,3)	6,9	11	2,85	6,89	5,03	0,29	1,45	0,65
(25,100,3)	6,8	15	11,5	78,7	47,7	0,71	11,6	4,09
(30,120,3)	8,5	22	163	829	453	1,91	101	38,8
(20,150,3)	6,6	11	3,82	6,96	5,29	0,33	0,72	0,58
(25,150,3)	9,6	15	22,8	65,1	42,2	1,28	7,38	2,86
(25,250,3)	7	12	29,7	62,4	43,9	1,37	2,1	1,74
(30,120,7)	64,8	100	523	1076	842	11	198	74,3
(35,150,3)	13,3	23	1188	4140	2196	69,4	795	273
(35,350,3)	9,3	15	772	1451	978	21,2	57,2	30,5
(35,500,3)	8,6	16	755	1143	869	10,35	20,7	14,9
(40,150,3)	22,7	38	-	-	-	8,81	2533	796
(40,350,3)	13,8	26	-	-	-	39,3	120	59,3
(50,250,3)	27,3	39	-	-	-	55,8	2618	989
(50,650,3)	18,5	28	-	-	-	36	65,8	47,1
(50,1000,3)	12,1	20	-	-	-	18,25	32,9	24,2
(60,1200,3)	17,6	31	-	-	-	72,1	151	100

TABLE 1 – Expériences numériques

les résultats obtenus montrent clairement une amélioration nette de la méthode EEC adaptée au problème MOISP. Un simple calcul permet d'affirmer que le temps CPU de la méthode BBMOISP est de l'ordre dix (10) fois plus petit, en moyenne, que celui de la méthode EEC et donc, la nouvelle méthode BBMOISP surclasse la méthode générale EEC.

## 4 Conclusion

Une méthode exacte, notée BBMOISP, basée sur le principe par séparation et évaluation est mise au point pour le problème du stable multi-objectif, MOISP. Elle permet la génération de tous les stables efficaces par décomposition du problème initial en sous problèmes disjoints de tailles réduites, chacun pouvant être résolu par une méthode générale relatée dans la littérature et dédiée au problème MOILP. Les résultats de l'expérimentation, même partiels, confirment l'adage "diviser pour régner" et montrent que la méthode est avantageuse par rapport à la méthode générale pour résoudre le problème MOILP décrite dans [1].

Dans le but d'accélérer la vitesse de convergence de la méthode à même de permettre le traitement d'instances de plus grandes dimensions, une investigation minutieuse à la recherche de la "meilleure" méthode pour la résolution du problème MOILP s'impose.

## Références

- [1] CHERGUI M.E-A., MOULAÏ M. and OUAÏL F.Z. : Solving the multiple objective integer linear programming problem. *Modelling, Computation and Optimization in Information Systems and Management Sciences Communications in Computer and Information Science* **vol.**(14), 69-76 (2008).
- [2] Xavier Delorme, Xavier Gandibleux and Fabien Degoutin : *Evolutionary, constructive and hybrid procedures for the bi-objective set packing problem* , *European Journal of Operational Research*, **vol.** (204), **Issue** (2), 206-217 (2010).
- [3] Xavier Delorme, Xavier Gandibleux and Fabien Degoutin : *GRASP for set packing problems* , *European Journal of Operational Research*, **vol.** (153), **Issue** (3), 564-580 (2004).
- [4] Steuer Ralph E. *Multiple Criteria Optimization : Theory, Computation and Application* - New York : *John Wiley and Sons*, 1986.