

N° d'ordre : 43/2008-M/MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LE TECHNOLOGIE
HOUARI BOUMEDIENE
FACULTÉ DES MATHÉMATIQUES



MÉMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : Mathématiques

Spécialité : Recherche Opérationnelle

Option : Mathématiques de Gestion

Par : AIT MEHDI Meriem

Sujet

**CONTRIBUTION ALGORITHMIQUE
EN PROGRAMMATION
MULTIOBJECTIF DISCRÈTE**

Soutenu publiquement le *07 Juillet 2008*, devant le jury composé de :

Mr M.AIDER, Professeur, USTHB	Président
Mr M.ABBAS, Professeur, USTHB	Directeur du Mémoire
Mr D.CHAABANE, Maître de Conférences, USTHB	Examinateur
Mr R.OUAFI, Maître de Conférences, USTHB	Examinateur
Mr M.E.A.CHERGUI, Chargé de Cours, USTHB	Invité

*À la mémoire de mon grand-père Rabah,
à mes grands-mères,
à mes parents,
à mes trois sœurs,
et à toute ma famille.*

Remerciements

Je tiens d'abord à remercier Dieu de m'avoir guidée et menée jusqu'ici.

Je remercie ensuite Monsieur Abbas Moncef, Professeur à l'USTHB, pour avoir bien voulu être mon directeur de mémoire et pour la confiance et la liberté qu'il m'a accordées.

J'exprime toute ma gratitude à Monsieur Chergui Mohamed El Amine, Chargé de Cours à l'USTHB, pour son aide précieuse, les conseils et les connaissances dont il a su me faire profiter.

Je remercie Monsieur Aider Méziane, Professeur à l'USTHB, qui m'a fait l'honneur d'accepter de présider le jury de ce mémoire.

Que Messieurs Chaabane Djamel et Ouafi Rachid, Maîtres de Conférences à l'USTHB, trouvent ici l'expression de ma reconnaissance pour avoir accepté de juger ce travail en tant qu'examineurs.

Je ne pourrais oublier de remercier mon amie, Ouail Fatma Zohra, qui m'a accompagnée durant cette période.

Finalement, je voudrais remercier ma famille, en particulier, mes parents et grands-parents, pour leur soutien et appui inconditionnel.

Merci à tous.

Table des matières

Introduction	1
1 Introduction à l'optimisation multiobjectif discrète	3
1.1 Programmation linéaire unicritère discrète	4
1.1.1 Rappel de quelques définitions et résultats fondamentaux de la programmation linéaire	4
1.1.2 Formulation et complexité du problème	6
1.1.3 Méthode duale fractionnaire	7
1.2 Programmation linéaire multiobjectif discrète	9
1.2.1 Formulation du problème	9
1.2.2 Concepts de base	10
1.2.3 Détection graphique des solutions efficaces	13
1.2.4 Exemple illustratif	14
2 Quelques méthodes d'optimisation multiobjectif discrète	16
2.1 Méthode de Klein et Hannan [9]	17
2.2 Méthode de Gupta et Malhotra (1 ^{ère} procédure) [8]	20
2.3 Méthode de Gupta et Malhotra (2 ^{ème} procédure) [8]	25
2.4 Méthode de Abbas et Moumene [13]	26

2.5	Méthode de Sylva et Crema [17]	31
2.6	Méthode de Chergui <i>et al.</i> [3]	33
3	Contribution algorithmique et implémentation	35
3.1	Amélioration de la méthode de Abbas et Moumene [13]	36
3.1.1	Algorithme de la méthode améliorée	37
3.1.2	Résultats théoriques	38
3.1.3	Exemple numérique	40
3.2	Amélioration de la méthode de Chergui <i>et al.</i> [3]	44
3.2.1	Algorithme de la méthode améliorée	45
3.2.2	Résultats théoriques	46
3.2.3	Exemple numérique	50
3.3	Implémentation et résultats	55
4	Application : Réalisation d'un plan optimal d'insémination artificielle	64
4.1	Position du problème	65
4.2	Modélisation mathématique	65
4.3	Résolution du problème et résultats	68
4.3.1	Jeux de données	68
4.3.2	Analyse des résultats et commentaires	69
	Conclusion et perspectives	71
	Bibliographie	73

De nombreux problèmes rencontrés dans la pratique, requièrent l'optimisation simultanée de plusieurs objectifs qui sont en général conflictuels. Dans les problèmes de production, par exemple, on ne vise pas uniquement un bénéfice maximum mais également des coûts de production minimaux. Des problèmes classiques de l'optimisation combinatoire ont été également étudiés en considérant au moins deux objectifs, nous citons entre autres les problèmes : d'affectation, de plus court chemin, de sac à dos, de voyageur de commerce, de flot dans un réseau et d'ordonnancement. La notion d'optimalité disparaît pour les problèmes de ce type au profit de la notion d'efficacité. Une solution efficace (encore appelée Pareto optimale) est une solution à partir de laquelle, il est impossible d'augmenter la valeur d'un objectif sans diminuer celle d'au moins un autre.

Nous nous plaçons dans ce mémoire, dans le contexte de l'optimisation multiobjectif discrète où il s'agit de déterminer l'ensemble des solutions efficaces d'un problème de programmation linéaire multiobjectif en nombres entiers. Notre objectif est d'étudier quelques travaux menés dans ce sens, essayer d'apporter un regard critique sur chacun d'eux et d'en proposer des améliorations dans le cadre du possible.

Ce mémoire comporte quatre chapitres :

Le premier chapitre expose en deux parties le cadre de notre travail. Nous com-

mençons dans la première, par un bref rappel de quelques résultats importants en programmation linéaire, nous formulons ensuite le problème de programmation linéaire à variables discrètes en exhibant sa difficulté, pour finir, nous décrivons la méthode duale fractionnaire [7] permettant de résoudre ce type de problèmes. Dans la deuxième partie, nous définissons le problème de programmation linéaire multiobjectif en nombres entiers et présentons les concepts fondamentaux relatifs tels que la dominance, l'efficacité, les solutions supportées et non supportées. Nous donnons en dernier, une méthode graphique pour la détection des solutions efficaces.

Le chapitre deux est entièrement consacré à la description de quelques méthodes existantes d'optimisation multiobjectif discrète ([3], [8], [9], [13], [17]) en illustrant certaines par des exemples.

Notre contribution dans le domaine multiobjectif est abordée dans le chapitre trois. Nous proposons en effet des améliorations pour les méthodes de Abbas et Moumene [13] et de Chergui *et al.* [3] permettant de réduire de façon significative le temps d'exécution. Nous exposons d'abord les algorithmes décrivant les méthodes améliorées, puis nous donnons les principaux résultats justifiant les étapes de ces dernières. Pour finir, nous discutons les résultats expérimentaux obtenus à l'aide de programmes que nous avons développé et implémenté sous Matlab. Une comparaison des méthodes étudiées est également faite.

Pour sortir du cadre de la théorie, une étude d'un problème réel pouvant se modéliser par la programmation linéaire multiobjectif discrète est présentée dans le chapitre quatre. Il s'agit de réaliser un plan d'insémination artificielle pour une ferme d'élevage laitier dans la commune de Mazouna, wilaya de Relizane, satisfaisant au mieux les objectifs tracés par l'éleveur, à savoir, améliorer la quantité et la qualité de lait de ses futures vaches.

Enfin, nous terminons par une conclusion reprenant les différentes contributions apportées dans ce mémoire et donnant quelques perspectives de recherche.

Introduction à l'optimisation multiobjectif discrète

Dans de nombreuses situations pratiques, il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, par exemple, dans un problème de planification manufacturière, le nombre d'unités produites doit être un nombre entier. On parle alors de programmation linéaire discrète ou en nombres entiers. L'objet de ce chapitre introductif est de présenter quelques notions fondamentales concernant les problèmes de programmation linéaire en nombres entiers dans le cas où l'objectif à optimiser est unique et celui où plusieurs critères sont à considérer. Cet exposé est basé principalement sur les livres de M. Sakarovitch [15] et de R. Steuer [16].

Sommaire

1.1	Programmation linéaire unicritère discrète	4
1.1.1	Rappel de quelques définitions et résultats fondamentaux de la programmation linéaire	4
1.1.2	Formulation et complexité du problème	6
1.1.3	Méthode duale fractionnaire	7
1.2	Programmation linéaire multiobjectif discrète	9
1.2.1	Formulation du problème	9
1.2.2	Concepts de base	10
1.2.3	Détection graphique des solutions efficaces	13
1.2.4	Exemple illustratif	14

1.1 Programmation linéaire unicritère discrète

De nombreux problèmes d'optimisation relèvent de l'optimisation discrète. Dans ces problèmes, les variables de décision sont astreintes à prendre des valeurs entières et cette restriction les rend particulièrement difficiles.

1.1.1 Rappel de quelques définitions et résultats fondamentaux de la programmation linéaire

Définition 1.1. Un programme linéaire est un problème dans lequel les variables sont des réels qui doivent satisfaire un ensemble d'équations et/ou d'inéquations linéaires (dites contraintes) et la valeur d'une fonction linéaire de ces variables (appelée fonction objectif), doit être rendue maximum (ou minimum).

Sans perte de généralité, nous supposerons par la suite que nous considérons des problèmes de maximisation.

Définition 1.2. Soient une $m \times n$ -matrice A , un m -vecteur colonne b et un n -vecteur ligne c . Les programmes linéaires :

$$(PC) \begin{cases} \text{Max } z = cx \\ Ax \leq b \quad x \geq 0 \end{cases}$$

$$(PS) \begin{cases} \text{Max } z = cx \\ Ax = b \quad x \geq 0 \end{cases}$$

sont écrits sous forme canonique et sous forme standard respectivement.

Théorème 1.1. *Tout programme linéaire peut être écrit sous forme canonique ou sous forme standard.*

Définition 1.3. Le système linéaire (où A est une $m \times n$ -matrice) :

$$Ax = b \tag{1.1}$$

est dit de plein rang si le rang de A , c'est-à-dire le nombre de colonnes de A linéairement indépendantes, est égal à m . On considère le programme linéaire

$$(P) \begin{cases} \text{Max } z = cx \\ Ax = b \quad x \geq 0 \end{cases}$$

tel que le système $Ax = b$ soit de plein rang. Une base de ce programme linéaire est un ensemble $B \subset \{1, 2, \dots, n\}$ d'indices de colonnes tel que A^B soit carrée¹ non singulière.

Définition 1.4. À une base B du programme linéaire (P) , on associe la solution de base \bar{x} correspondant à B , solution de (1.1) définie par

$$\bar{x}_j = 0 \text{ pour } j \notin B$$

Une base B est dite réalisable si la solution de base associée est réalisable pour (P) , c'est-à-dire $\bar{x}_B = (A^B)^{-1}b \geq 0$.

Définition 1.5. Etant donnée une base réalisable B du programme linéaire (P) , le programme linéaire, équivalent à (P) :

$$(PC) \begin{cases} \widehat{c}^N x_N = z(\text{Max}) - \pi b \\ x_B + (A^B)^{-1} A^N x_N = (A^B)^{-1} b \quad x \geq 0 \end{cases}$$

où :

- $N = \{1, 2, \dots, n\} \setminus B$
- $\pi = c^B (A^B)^{-1}$ est dit vecteur multiplicateur relatif à la base B
- $\widehat{c} = c - \pi A$ est dit vecteur coût réduit relatif à la base B

est dit forme canonique de (P) par rapport à la base B .

Théorème 1.2. *Si le vecteur coût réduit \widehat{c} relatif à une base réalisable B est négatif ou nul, la solution de base correspondante est solution optimale de (P) . La base B est alors dite base optimale.*

¹ A^B désigne la $m \times |B|$ -matrice obtenue à partir de A en retenant les colonnes A^j pour $j \in B$.

Théorème 1.3 (théorème fondamental de la programmation linéaire). *Etant donné un programme linéaire :*

- *S'il admet une solution réalisable, il admet une solution réalisable de base.*
- *S'il admet une solution optimale, il admet une solution optimale de base.*
- *S'il admet une solution réalisable et si la valeur de la fonction objectif est bornée, il admet une solution optimale de base.*

Définition 1.6. Le domaine des solutions réalisables d'un programme linéaire est un convexe particulier, intersection d'un nombre fini de demi-espaces fermés, appelé polyèdre convexe.

Théorème 1.4. *Une solution de base réalisable du programme linéaire (P) correspond à un sommet du polyèdre convexe D des solutions réalisables. Une itération de l'algorithme du simplexe² correspond à un mouvement d'un sommet à un sommet adjacent de D.*

1.1.2 Formulation et complexité du problème

Si dans le programme linéaire (P), on contraint le vecteur x à être entier, on obtient le programme (Q), qui est un *programme linéaire en nombres entiers* (PLNE) :

$$(Q) \begin{cases} \text{Max } z = cx \\ Ax = b & x \geq 0 \\ x \in \mathbb{Z}^n \end{cases}$$

Ces contraintes supplémentaires (x_j entier pour $j = \overline{1, n}$) sont dites contraintes d'intégrité. En général, un tel problème n'a de solutions que si A et b sont aussi entiers.

Les principales motivations derrière la programmation linéaire en nombres entiers sont :

- le besoin de variables entières (un nombre de camions à acheter par exemple) ;

²Algorithme dédié à la résolution des programmes linéaires en variables continues, dû à G.B.Dantzig 1947 [4].

- la modélisation de contraintes et de conditions ingérables par la programmation linéaire en variables continues (c'est le cas par exemple du choix entre deux alternatives incompatibles ; on peut introduire la contrainte $x + y \leq 1$ avec x et y deux variables binaires valant 1 si l'alternative correspondante est choisie).

La difficulté de résolution de tels problèmes réside dans la construction de l'enveloppe convexe du domaine des solutions entières, c'est-à-dire le plus petit ensemble convexe contenant ce dernier. Aussi, ces problèmes sont connus pour être NP-Complets [14].

Les deux principales familles de méthodes actuellement connues pour résoudre ces programmes sont les méthodes arborescentes et les méthodes de troncatures (coupes). Nous présentons ci-dessous une méthode de coupe, proposée en 1958 par R.Gomory [7].

1.1.3 Méthode duale fractionnaire

Considérons le PLNE sous forme standard :

$$\left\{ \begin{array}{l} Max \ z = cx \\ Ax = b \\ x \in \mathbb{Z}^n \end{array} \right. \quad x \geq 0 \quad (1.2)$$

La relaxation continue de (1.2) est :

$$\left\{ \begin{array}{l} Max \ z = cx \\ Ax = b \end{array} \right. \quad x \geq 0$$

Ce problème est obtenu à partir de (1.2) en relachant (c'est-à-dire en oubliant) les contraintes d'intégrité.

L'idée principale des méthodes de coupes est d'ajouter des contraintes qui n'excluent aucun point entier réalisable. La méthode consistera à ajouter de telles contraintes linéaires une par une, jusqu'à ce que la solution optimale de la relaxation soit entière. Les contraintes ajoutées sont appelées troncatures ou coupes.

Description de la méthode

Commençons par résoudre la relaxation continue par l'algorithme du simplexe [4] et considérons le tableau obtenu à l'optimum ($\bar{A}x = \bar{b}$) : parmi les variables de base, choisissons une variable x_r , $r \in B$, fractionnaire (s'il n'y en a pas, on est à l'optimum). La contrainte correspondant à x_r (i^{ème} ligne du tableau optimal) s'écrit :

$$x_r + \sum_{j \in N} \bar{A}_i^j x_j = \bar{b}_i \quad (1.3)$$

Notations : étant donné un nombre réel y , on désigne par :

- $\lfloor y \rfloor$ le plus grand entier inférieur ou égal à y ;
- $\langle y \rangle = y - \lfloor y \rfloor$.

$\langle y \rangle$ est appelée la partie fractionnaire de y et $\lfloor y \rfloor$ sa partie entière.

Puisque toutes les variables sont positives ou nulles dans (1.3), on a :

$$\sum_{j \in N} \lfloor \bar{A}_i^j \rfloor x_j \leq \sum_{j \in N} \bar{A}_i^j x_j$$

Donc on a :

$$x_r + \sum_{j \in N} \lfloor \bar{A}_i^j \rfloor x_j \leq \bar{b}_i$$

La partie gauche de cette inéquation est entière. Le second membre (partie droite) peut donc être remplacé par $\lfloor \bar{b}_i \rfloor$:

$$x_r + \sum_{j \in N} \lfloor \bar{A}_i^j \rfloor x_j \leq \lfloor \bar{b}_i \rfloor \quad (1.4)$$

(1.3)–(1.4) donne :

$$\sum_{j \in N} (\bar{A}_i^j - \lfloor \bar{A}_i^j \rfloor) x_j \geq \bar{b}_i - \lfloor \bar{b}_i \rfloor$$

Finalement, nous obtenons la contrainte :

$$\sum_{j \in N} \langle \bar{A}_i^j \rangle x_j \geq \langle \bar{b}_i \rangle$$

C'est la coupe de Gomory.

Nous voulons ajouter cette coupe au problème initial. Pour garder un problème écrit sous forme canonique par rapport à une base, nous multiplions cette dernière inéquation par (-1) et ajoutons une variable d'écart x_s . On obtient :

$$-\sum_{j \in N} \langle \bar{A}_i^j \rangle x_j + x_s = -\langle \bar{b}_i \rangle$$

La nouvelle base formée de la base optimale précédente à laquelle on ajoute s vérifie les conditions d'optimalité (coûts réduits négatifs ou nuls) et n'est pas réalisable.

On peut appliquer l'algorithme dual du simplexe [10] pour résoudre le nouveau programme linéaire formé.

Par ce procédé, on ajoute ainsi, une par une des coupes jusqu'à obtention d'une solution de base entière qui est alors l'optimum de (1.2).

1.2 Programmation linéaire multiobjectif discrète

Comme son nom l'indique, un problème d'optimisation multiobjectif ou multicritère consiste à optimiser simultanément k fonctions objectifs ($k \geq 2$) souvent contradictoires (deux objectifs sont contradictoires lorsque la diminution de l'un entraîne une augmentation de l'autre). De ce fait, il est impératif de revoir la notion d'optimalité des solutions.

1.2.1 Formulation du problème

Un programme linéaire multiobjectif en nombres entiers (MOILP : **M**ultiple **O**bjective **I**nteger **L**inear **P**rogram) peut être formulé comme suit :

$$\left\{ \begin{array}{l} \text{Max } z_1 = c^1 x \\ \quad \vdots \\ \text{Max } z_k = c^k x \\ x \in S = \{x \in \mathbb{Z}^n \mid Ax = b, x \geq 0\} \end{array} \right.$$

Où $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ et $c^i \in \mathbb{R}^{1 \times n}$, $i = \overline{1, k}$.

Il peut être aussi écrit de la manière suivante :

$$\text{“Max” } \{z = Cx \mid x \in S\}$$

où C est une $k \times n$ -matrice composée des vecteurs lignes c^i .

On met Max entre guillemets pour dire qu'en général, on ne peut pas trouver une solution réalisable qui maximise les k objectifs simultanément.

1.2.2 Concepts de base

Soit (P) un programme linéaire multiobjectif en nombres entiers. S l'ensemble des solutions réalisables de (P) dans l'espace des décisions (l'espace \mathbb{Z}^n où se situe S).

Solutions réalisables dans l'espace des critères

L'ensemble Z donné par :

$$Z = \{z \in \mathbb{R}^k \mid z = Cx, x \in S\}$$

représente l'ensemble des points réalisables dans l'espace des critères (l'espace \mathbb{R}^k où se situe Z). En d'autres termes, Z est l'ensemble des images de tous les points de S . On impose une relation d'ordre partiel (une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres) sur cet ensemble de points, appelée *relation de dominance*.

Dominance

Soient deux vecteurs critères $z^1, z^2 \in Z$. On dit que z^1 domine z^2 si et seulement si $z_i^1 \geq z_i^2$ pour tout i avec au moins une inégalité stricte.

Efficacité

Une solution $\bar{x} \in S$ est *efficace* (ou Pareto optimale, ou encore non inférieure) si et seulement s'il n'existe pas de solution $x \in S$ telle que $c^i x \geq c^i \bar{x}$ pour tout i avec au moins une inégalité stricte.

Efficacité faible

Une solution $\bar{x} \in S$ est *faiblement efficace* si et seulement s'il n'existe pas de solution $x \in S$ telle que $Cx > C\bar{x}$.

Le vecteur critère associé à une solution (faiblement) efficace \bar{x} donné par $C\bar{x}$ est dit solution (faiblement) non dominée.

Solutions supportées et non supportées

Considérons le problème

$$(P') \begin{cases} \text{Max} & z_1 = c^1 x \\ & \vdots \\ \text{Max} & z_k = c^k x \\ & x \in D = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \end{cases}$$

relaxation continue du problème (P) .

Théorème 1.5 (Geoffrion [6]). *Étant donné le problème $(P_\lambda) : \text{Max} \{\lambda^t Cx \mid x \in D\}$ avec $\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0, i = \overline{1, k} \right\}$; alors \bar{x} est une solution efficace pour (P') si et seulement si \bar{x} est une solution optimale du problème paramétrique (P_λ) .*

D'après ce théorème, l'ensemble des solutions efficaces du problème (P') est bien caractérisé par la solution de (P_λ) . Différemment du cas continu, la difficulté principale rencontrée lorsqu'on traite les problèmes multiobjectifs à variables discrètes est l'existence de solutions efficaces qui ne sont pas optimales pour (P_λ) (en remplaçant D par S) et ce en raison de la non convexité du domaine des solutions réalisables S ,

ces solutions sont dites solutions efficaces *non supportées*. Celles qui sont optimales pour (P_λ) , sont appelées solutions efficaces *supportées*.

L'exemple numérique suivant (introduit par Bowman [1]) illustre ces deux notions :

$$(P^{Bow}) \begin{cases} \text{Max } z_1 = 6x_1 + 3x_2 + x_3 \\ \text{Max } z_2 = x_1 + 3x_2 + 6x_3 \\ x_1 + x_2 + x_3 \leq 1 \\ x_1, x_2, x_3 \in \{0, 1\} \end{cases}$$

Les solutions optimales du problème

$$(P_\lambda^{Bow}) \begin{cases} \text{Max } \lambda(6x_1 + 3x_2 + x_3) + (1 - \lambda)(x_1 + 3x_2 + 6x_3) \\ x_1 + x_2 + x_3 \leq 1 \\ x_1, x_2, x_3 \in \{0, 1\} \text{ avec } 0 < \lambda < 1 \end{cases}$$

sont des solutions efficaces. Elles sont données par $x^1 = (1, 0, 0)$ et $x^2 = (0, 0, 1)$ de vecteurs critères respectifs $(6, 1)$ et $(1, 6)$. Or il est facile de constater que la solution $x^3 = (0, 1, 0)$ dont le vecteur critère est $(3, 3)$ est également efficace mais n'est pas optimale pour le problème paramétrique (P_λ^{Bow}) .

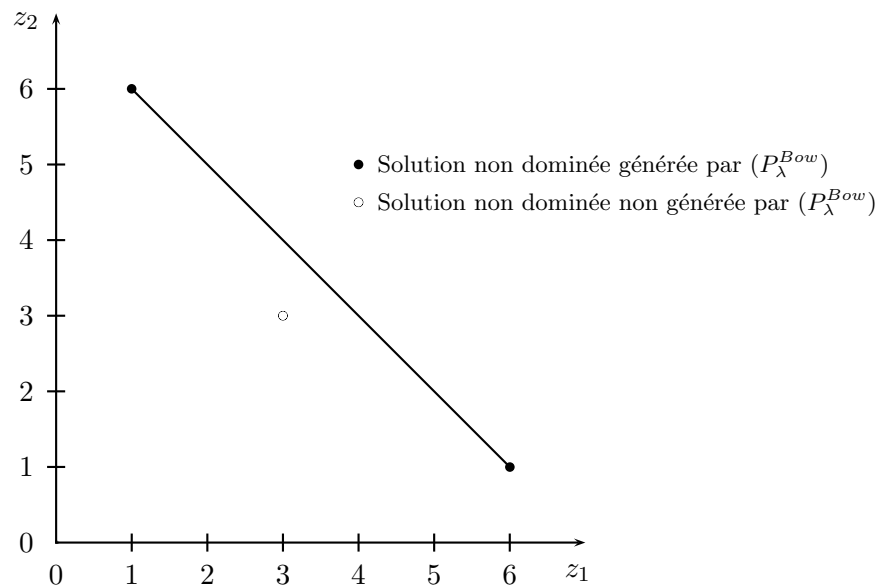


FIG. 1.1 – Exemple proposé par Bowman

1.2.3 Détection graphique des solutions efficaces

Pour définir graphiquement la notion d'efficacité, nous avons besoin d'introduire les concepts suivants :

Définition 1.7 (Cône). Soit $v \in V \subset \mathbb{R}^n, V \neq \emptyset$. Alors, V est un cône si et seulement si $\alpha v \in V$ pour tout scalaire $\alpha \geq 0$. L'origine $0 \in \mathbb{R}^n$ est contenu dans chaque cône.

Définition 1.8 (Vecteurs générateurs d'un cône). Soit un ensemble de vecteurs $\{v^1, \dots, v^k\}$ de \mathbb{R}^n et l'ensemble V tel que :

$$V = \{v \in \mathbb{R}^n \mid v = \sum_{i=1}^k \alpha_i v^i, \alpha_i \geq 0 \forall i\}.$$

V est l'ensemble de toutes les combinaisons linéaires à coefficients non négatifs des $v^i, i = \overline{1, k}$ et est le cône convexe généré par l'ensemble $\{v^1, \dots, v^k\}$.

Un générateur $v^r \in \{v^1, \dots, v^k\}$ est non essentiel si $\{v^1, \dots, v^k\} \setminus \{v^r\}$ peut générer V . En d'autres termes, c'est celui qui peut s'écrire comme combinaison linéaire à coefficients non négatifs des autres générateurs. Il est essentiel sinon.

Définition 1.9 (Cône polaire semi-positif). Soit $V \subset \mathbb{R}^n$ un cône convexe généré par $\{v^1, v^2, \dots, v^k\}$. Le cône convexe $V^>$ donné par :

$$V^> = \{y \in \mathbb{R}^n \mid y^t v^i \geq 0 \forall i \text{ et } y^t v^i > 0 \text{ pour au moins un } i \in \{1, \dots, k\}\} \cup \{0 \in \mathbb{R}^n\}$$

est le cône polaire semi-positif de V .

Définition 1.10 (Ensemble de dominance). Soit $\bar{x} \in S$ et $C^>$ le cône polaire semi-positif du cône généré par les gradients des k fonctions objectifs.

L'ensemble de dominance en \bar{x} est donné par :

$$\begin{aligned} D_{\bar{x}} &= \{\bar{x}\} \oplus C^> \\ &= \{x \in \mathbb{R}^n \mid x = \bar{x} + y, y \in C^>\} \end{aligned}$$

L'ensemble de dominance en \bar{x} contient tous les points dont les vecteurs critères

dominant le vecteur critère de \bar{x} . Le théorème suivant montre l'importance de cet ensemble dans la détection des solutions efficaces :

Théorème 1.6. *Soit $D_{\bar{x}}$ l'ensemble de dominance en $\bar{x} \in S$. Alors, \bar{x} est efficace si et seulement si $D_{\bar{x}} \cap S = \{\bar{x}\}$.*

Si l'intersection de l'ensemble de dominance en \bar{x} avec l'ensemble des solutions réalisables S contient uniquement \bar{x} , alors \bar{x} est efficace.

1.2.4 Exemple illustratif

Soit à déterminer graphiquement l'ensemble des solutions efficaces du programme linéaire (P) à deux objectifs et deux variables bornées entières suivant :

$$(P) \begin{cases} \text{Max } z_1 = x_1 \\ \text{Max } z_2 = -x_1 + 3x_2 \\ (x_1, x_2) \in S = \{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 + 2x_2 \leq 7, 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3\} \end{cases}$$

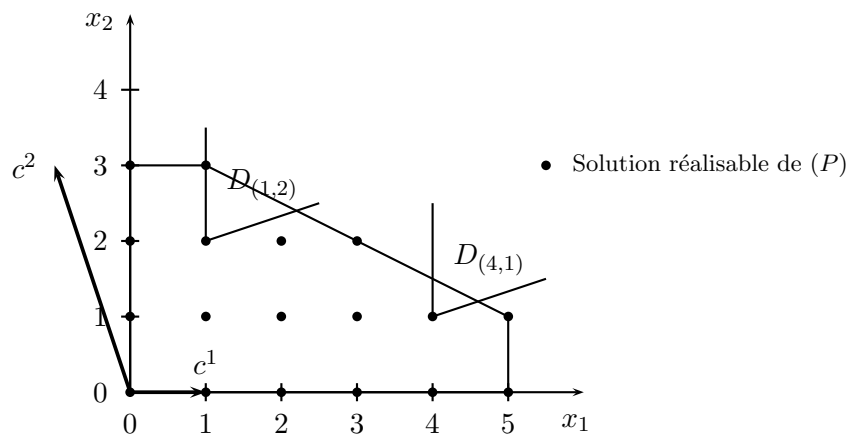


FIG. 1.2 – Détection graphique des solutions efficaces de (P)

Dans la figure 1.2, c^1 et c^2 sont les vecteurs gradients des fonctions objectifs z_1 et z_2 respectivement. Pour chaque point (x_1, x_2) de S , $D_{(x_1, x_2)}$ représente l'ensemble de dominance en ce point.

Comme le montre la figure 1.2, la solution réalisable $(1, 2)$ n'est pas efficace pour le problème (P) car $D_{(1,2)} \cap S = \{(1, 2), (1, 3)\} \neq \{(1, 2)\}$. Par contre, la solution $(4, 1)$

est efficace puisque $D_{(4,1)} \cap S = \{(1, 2)\}$. En suivant ce raisonnement, on obtient les solutions efficaces du problème (P) représentées par des carrés blancs et noirs dans la figure 1.3.

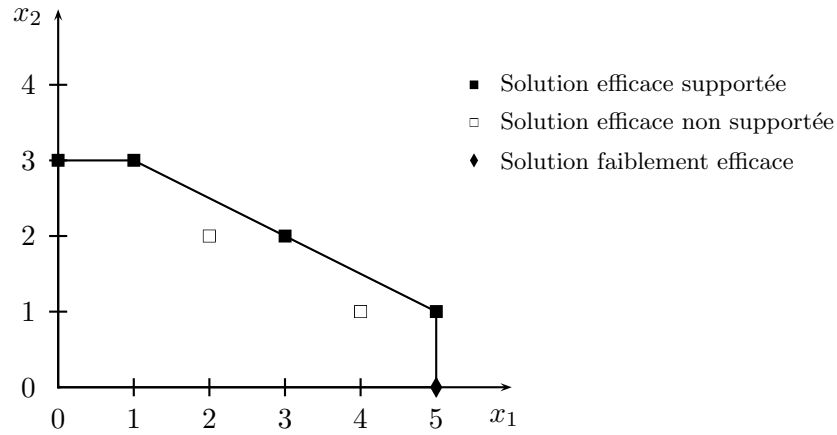


FIG. 1.3 – Représentation des différents types de solutions dans l'espace des décisions

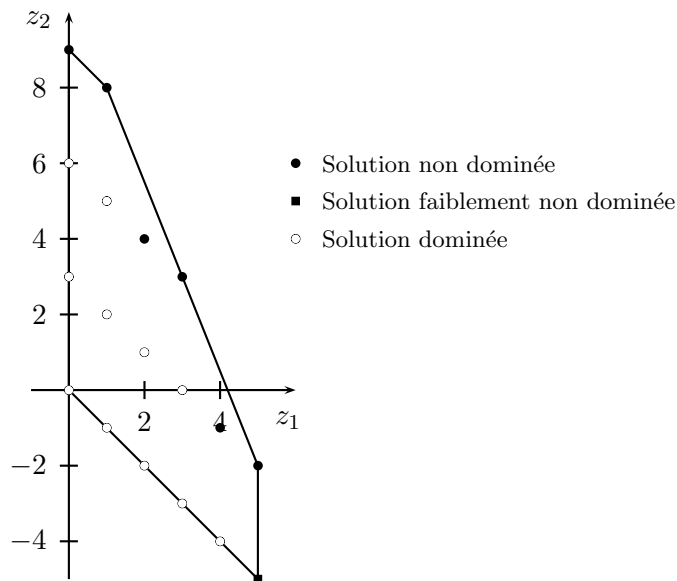


FIG. 1.4 – Représentation des différents types de solutions dans l'espace des critères

La figure 1.4 est une illustration de l'ensemble Z , image de l'ensemble S , où on distingue différents types de solutions (dominées, faiblement non dominées et non dominées). Il est clair que la solution $(5, -5)$ est faiblement non dominée puisqu'il n'existe pas de solution $z = (z_1, z_2) \in Z$ telle que $z_1 > 5$.

Quelques méthodes d'optimisation multiobjectif discrète

Résoudre un problème multiobjectif consiste à choisir parmi les solutions efficaces, une solution de meilleur compromis. Dès lors, une étape préalable importante concerne l'optimisation multiobjectif qui recherche les solutions efficaces. Dans ce chapitre, nous donnons une description détaillée de quelques méthodes existantes d'optimisation multiobjectif discrète en illustrant certaines par des exemples.

Sommaire

2.1	Méthode de Klein et Hannan [9]	17
2.2	Méthode de Gupta et Malhotra (1^{ère} procédure) [8]	20
2.3	Méthode de Gupta et Malhotra (2^{ème} procédure) [8]	25
2.4	Méthode de Abbas et Moumene [13]	26
2.5	Méthode de Sylva et Crema [17]	31
2.6	Méthode de Chergui <i>et al.</i> [3]	33

Considérons le programme linéaire multiobjectif en nombres entiers (P) suivant :

$$(P) \begin{cases} \text{Max} & z_1 = c^1 x \\ & \vdots \\ \text{Max} & z_k = c^k x \\ & x \in S = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\} \end{cases}$$

Où $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ et $c^i \in \mathbb{R}^{1 \times n}$, $i = \overline{1, k}$.

Dans les paragraphes 2.1, 2.3, 2.4 et 2.5, les paramètres coûts du problème (P) sont supposés entiers, c'est-à-dire $c^i \in \mathbb{Z}^{1 \times n}$, pour $i = \overline{1, k}$.

Notons par :

- D l'ensemble des solutions réalisables du problème relaxé de (P).
 $D = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$;
- $ESE(P)$ (resp. $SND(P)$) l'ensemble de toutes les solutions efficaces (resp. non dominées) de (P).

2.1 Méthode de Klein et Hannan [9]

Nous présentons ci-dessous une technique proposée par D. Klein et E. Hannan pour générer l'ensemble $ESE(P)$ ou une partie de ce dernier :

Étape 0 :

Choisir arbitrairement un critère l parmi les k critères du problème général (P) et résoudre le problème unicritère (P_0) suivant :

$$(P_0) \quad \text{Max} \{c^l x | x \in S\}.$$

Trouver toutes les solutions optimales de (P_0) et retenir celles dont les vecteurs critères sont non dominés formant l'ensemble $\Theta(P_0)$.

Étape j ($j \geq 1$) :

Soient $(x^p : p = \overline{1, r})$ les solutions efficaces trouvées aux étapes précédentes. On

résout le problème unicritère (P_j) défini comme suit :

$$(P_j) \left\{ \begin{array}{l} \text{Max } c^l x \\ x \in S \\ \bigcap_{p=1}^r \left(\bigcup_{i=1, i \neq l}^k c^i x \geq c^i x^p + f_i \right) \end{array} \right.$$

Où $f_i \geq 1$ et entier pour $i = \overline{1, k}$.

Le problème (P_j) est obtenu à partir de (P_0) en lui ajoutant des contraintes qui assurent que les solutions de (P_j) seront meilleures que toutes les solutions $(x^p : p = \overline{1, r})$ pour au moins un critère $i \neq l$.

Construire l'ensemble $\Theta(P_j)$ des solutions optimales de (P_j) dont les vecteurs critères sont non dominés.

La procédure s'arrête à l'étape (t) où le problème (P_t) est irréalisable.

Les auteurs de la méthode ont démontré que :

- $\Theta(P_j)$ constitue une partie de $ESE(P)$ pour tout j ;
- Si $\forall i, f_i = 1$, alors $\bigcup_{j=0}^{t-1} \Theta(P_j) = ESE(P)$.

Remarque

Il est important de noter que le nombre de sous-problèmes sous-jacents considérés à chaque étape j ($j \geq 1$) est donné par $(k - 1)^r$ où k est le nombre de critères du problème et r est le nombre de solutions efficaces trouvées aux étapes précédentes (c'est-à-dire de 0 à $(j - 1)$).

Exemple illustratif

Considérons le programme linéaire multiobjectif en nombres entiers (P) suivant :

$$(P) \begin{cases} \text{Max } z_1 = & x_2 \\ \text{Max } z_2 = & -x_1 + 2x_2 \\ \text{Max } z_3 = & 2x_1 + x_2 \\ & (x_1, x_2) \in S \end{cases}$$

Où $S = \{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 + x_2 \leq 3, x_2 \leq 2, x_1 - 2x_2 \leq 0, x_1 \geq 0, x_2 \geq 0\}$.

Prenons $l = 1$ et $f_i = 1$ pour tout i .

Étape (0)

La résolution de $(P_0) : \text{Max } \{z_1 = x_2 \mid (x_1, x_2) \in S\}$ donne le tableau optimal suivant :

B	x_1	x_4	x_B
x_3	1	-1	1
x_2	0	1	2
x_5	1	2	4
$-z_1$	0	-1	-2

(P_0) admet deux solutions optimales $x^1 = (0, 2)$ et $x^2 = (1, 2)$ dont les vecteurs critères sont respectivement $(2, 4, 2)$ et $(2, 3, 4)$. Ces derniers sont non dominés, donc $\Theta(P_0) = \{(0, 2), (1, 2)\}$.

Étape (1)

On considère le problème (P_1) suivant :

$$(P_1) \begin{cases} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ -x_1 + 2x_2 \geq 5 \quad \text{ou} \quad 2x_1 + x_2 \geq 3 \\ -x_1 + 2x_2 \geq 4 \quad \text{ou} \quad 2x_1 + x_2 \geq 5 \end{cases}$$

Pour trouver les solutions optimales de (P_1) , il faut résoudre les quatre sous-problèmes suivants :

$$(P_{11}) \left\{ \begin{array}{l} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ -x_1 + 2x_2 \geq 5 \\ -x_1 + 2x_2 \geq 4 \end{array} \right. \quad (P_{12}) \left\{ \begin{array}{l} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ -x_1 + 2x_2 \geq 5 \\ 2x_1 + x_2 \geq 5 \end{array} \right.$$

$$(P_{13}) \left\{ \begin{array}{l} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ 2x_1 + x_2 \geq 3 \\ -x_1 + 2x_2 \geq 4 \end{array} \right. \quad (P_{14}) \left\{ \begin{array}{l} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ 2x_1 + x_2 \geq 3 \\ 2x_1 + x_2 \geq 5 \end{array} \right.$$

Les trois premiers sous-problèmes sont irréalisables et le quatrième possède une unique solution optimale $x^3 = (2, 1)$ qui est en fait l'unique solution optimale de (P_1) . Le vecteur critère de celle-ci est $(1, 0, 5)$. Donc, $\Theta(P_1) = \{(2, 1)\}$.

Étape (2)

Considérons le problème (P_2) suivant :

$$(P_2) \left\{ \begin{array}{l} \text{Max } z_1 = x_2 \\ (x_1, x_2) \in S \\ -x_1 + 2x_2 \geq 5 \quad \text{ou} \quad 2x_1 + x_2 \geq 3 \\ -x_1 + 2x_2 \geq 4 \quad \text{ou} \quad 2x_1 + x_2 \geq 5 \\ -x_1 + 2x_2 \geq 1 \quad \text{ou} \quad 2x_1 + x_2 \geq 6 \end{array} \right.$$

Le problème (P_2) est irréalisable (les huit sous-problèmes sous-jacents ne possèdent pas de solutions). Terminer.

Finalement, $ESE(P) = \bigcup_{j=0}^1 \Theta(P_j) = \{(0, 2), (1, 2), (2, 1)\}$.

2.2 Méthode de Gupta et Malhotra (1^{ère} procédure) [8]

Dans ce paragraphe, nous avons besoin d'introduire la définition suivante :

Définition 2.1. Soient le programme linéaire $(Q) : \text{Max } \{z = cx | x \in D\}$, B une base de (Q) , $\bar{x} = (\bar{x}_i)_{i=1, n}$ la solution de base correspondant à B et lig l'application

définie par :

$$\begin{aligned} \text{lig} : \{1, \dots, n\} &\longrightarrow \{1, \dots, m\} \\ i &\longmapsto \text{lig}(i) = \text{ligne correspondant à la variable de base } x_i \end{aligned}$$

Une arête E_j incidente à la solution \bar{x} est définie comme étant l'ensemble :

$$E_j = \left\{ x = (x_i) \in D \left| \begin{array}{l} x_i = \bar{x}_i - \theta_j \widehat{A}_{\text{lig}(i)}^j, \quad i \in B \\ x_j = \theta_j \\ x_l = 0, \quad l \in N \setminus \{j\} \end{array} \right. \right\}$$

$$\text{où } \widehat{A}^j = (A^B)^{-1}A^j, \quad N = \{1, 2, \dots, n\} \setminus B \text{ et } 0 \leq \theta_j \leq \min_{i \in B} \left\{ \frac{\bar{x}_i}{\widehat{A}_{\text{lig}(i)}^j}, \widehat{A}_{\text{lig}(i)}^j > 0 \right\}.$$

Remarque

Pour déterminer les solutions entières sur l'arête E_j incidente à une solution réalisable entière \bar{x} , il faut rajouter la condition : θ_j est un entier positif et $\theta_j \widehat{A}_{\text{lig}(i)}^j$ sont des entiers pour tout $i \in B$.

Notations

À une solution de base x^p , optimale pour le problème unicritère suivant :

$$\text{Max} \{c_p^1 x \mid A_p x = b_p, \quad x \geq 0\}$$

on associe :

B_p (resp. N_p) : ensemble des indices des variables de base (resp. hors base) de x^p ;

\widehat{c}_p^i : vecteur coût réduit relatif au critère i , $i = \overline{1, k}$, donné par :

$$\widehat{c}_p^i = c_p^i - (c_p^i)^{B_p} (A_p^{B_p})^{-1} A_p$$

Γ_p : l'ensemble défini comme suit :

$$\Gamma_p = \{j \in N_p \mid (\widehat{c}_p^1)^j < 0 \text{ et } (\widehat{c}_p^i)^j > 0 \text{ pour au moins un } i \in \{2, \dots, k\}\}$$

où $(\widehat{c}_p^i)^j$ est la $j^{\text{ème}}$ composante du vecteur \widehat{c}_p^i , $i = \overline{1, k}$.

La méthode de Gupta et Malhotra construit progressivement l'ensemble $SND(P)$ en se basant sur la technique des coupes. Elle procède comme suit :

Étape 1 :

Résoudre le problème (P_1) : $Max \{z_1 = c^1 x | x \in S\}$ en utilisant la méthode duale fractionnaire de Gomory. Au lieu de (P_1) , on peut construire d'une manière similaire l'un des problèmes (P_i) , $i = \overline{2, k}$ et le résoudre.

1.1 Si la solution optimale de (P_1) est unique, soit $x^1 = (x_i^1)$, déterminer le vecteur critère (z_1^1, \dots, z_k^1) correspondant à x^1 et former l'ensemble Z_0 contenant ce k -uplet.

1.2 Si le problème (P_1) admet plusieurs solutions optimales, déterminer toutes ces solutions ainsi que le vecteur critère correspondant à chacune d'elles. Eliminer les vecteurs critères dominés par comparaison deux à deux et garder seulement ceux non dominés formant l'ensemble Z_0 . Soient (z_1^1, \dots, z_k^1) le premier k -uplet de Z_0 dans l'ordre lexicographique (c'est-à-dire celui qui a la plus grande valeur de z_2 , si toutes les secondes composantes sont identiques, on choisit celui qui a la plus grande valeur de z_3 et ainsi de suite) et $x^1 = (x_i^1)$ la solution correspondante.

Étape 2 :

Choisir $j_1 \in \Gamma_1$ et calculer le nombre $\theta = \min_{i \in B_1} \left\{ \frac{x_i^1}{(\widehat{A}_1)_{lig(i)}^{j_1}}, (\widehat{A}_1)_{lig(i)}^{j_1} > 0 \right\}$ correspondant à la solution x^1 .

2.1 Si $\theta < 1$, alors aucune solution entière ne peut être obtenue sur l'arête E_{j_1} .

2.2 Si $\theta \geq 1$, déterminer toutes les solutions entières se trouvant sur l'arête E_{j_1} et former l'ensemble Z_1 des vecteurs critères correspondant à ces solutions.

Eliminer l'arête E_{j_1} par la coupe $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$ et appliquer la méthode duale du simplexe et les coupes de Gomory si nécessaire pour obtenir une solution réalisable entière x^2 dans la région tronquée. Ajouter le vecteur critère de celle-ci à Z_1 s'il n'est pas dominé par l'un des k -uplets retenus précédemment.

Dans le cas où pour tout $j \in \Gamma_1$, $\theta < 1$, choisir n'importe quel indice $j_1 \in \Gamma_1$ et appliquer la coupe $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$.

Étape générale p ($p \geq 3$) :

Choisir $j_{p-1} \in \Gamma_{p-1}$. Explorer l'arête $E_{j_{p-1}}$ pour d'éventuelles solutions entières et déterminer les vecteurs critères correspondants (en cas d'existence). Former l'ensemble Z'_{p-1} à partir de Z_{p-2} en lui ajoutant ces nouveaux k -uplets. Éliminer tous les k -uplets dominés de Z'_{p-1} pour obtenir l'ensemble Z_{p-1} . Éliminer ensuite l'arête $E_{j_{p-1}}$ par la coupe $\sum_{j \in N_{p-1} \setminus \{j_{p-1}\}} x_j \geq 1$ et déterminer une solution optimale entière x^p dans la région tronquée.

Étape finale ($n + 1$) :

La procédure prend fin à l'étape ($n + 1$) dans laquelle :

(1) $\Gamma_n = \emptyset$ et $(\widehat{c}_n^1)^j < 0$, $\forall j \in N_n$

ou (2) $\Gamma_n \neq \emptyset$ mais pour tout $j \in \Gamma_n$, toutes les solutions entières se trouvant sur l'arête E_j sont inefficaces.

Finalement, l'ensemble $SND(P)$ est la réunion des ensembles Z_0 et Z_{n-1} .

D'après Gupta et Malhotra, cette méthode permet de générer toutes les solutions efficaces d'un programme linéaire multiobjectif en nombres entiers. Malheureusement ce n'est pas toujours le cas, ceci est dû au second critère d'arrêt posé par les auteurs. Moulai [12] et Chaabane [2] ont présenté un contre exemple illustrant l'erreur de ce dernier et ont apporté des rectifications et des améliorations à celle-ci.

Nous donnons ci après un autre contre exemple pour la méthode.

Considérons le programme linéaire multiobjectif en nombres entiers (P) suivant :

$$(P) \left\{ \begin{array}{l} \text{Max } z_1 = x_1 + 3x_2 \\ \text{Max } z_2 = x_1 - x_2 \\ x_1 + 2x_2 \leq 7 \\ x_1 \leq 5 \\ -x_1 + x_2 \leq 2 \\ x_i \geq 0 \text{ entier, } i = \overline{1, 2} \end{array} \right.$$

Étape (1)

La résolution de (P_1) donne le tableau optimal suivant :

B	x_3	x_5	x_B
x_1	$\frac{1}{3}$	$-\frac{2}{3}$	1
x_4	$-\frac{1}{3}$	$\frac{2}{3}$	4
x_2	$\frac{1}{3}$	$\frac{1}{3}$	3
$-z_1$	$-\frac{4}{3}$	$-\frac{1}{3}$	-10
$-z_2$	0	1	2

La solution de (P_1), $x^1 = (1, 3)$ est unique et le premier couple non dominé est $(z_1^1, z_2^1) = (10, -2)$. $Z_0 = \{(10, -2)\}$.

$$N_1 = \{3, 5\}, \Gamma_1 = \{5\}.$$

Étape (2)

$j_1 = 5$, $\theta = 6 > 1$. Les solutions entières se trouvant sur l'arête E_5 sont $(3, 2)$ et $(5, 1)$. Les vecteurs critères correspondant à ces solutions sont respectivement $(9, 1)$ et $(8, 4)$. $Z_1 = \{(9, 1), (8, 4)\}$.

L'arête E_5 sera donc tronquée par $x_3 \geq 1 \Rightarrow -x_3 + x_6 = -1$. En ajoutant cette contrainte au tableau précédent et en appliquant la méthode duale du simplexe et des coupes de Gomory, on obtient le tableau optimal suivant :

B	x_6	x_7	x_B
x_1	1	-2	2
x_4	-1	2	3
x_2	0	1	2
x_3	-1	0	1
x_5	1	-3	2
$-z_1$	-1	-1	-8
$-z_2$	-1	3	0

$x^2 = (2, 2)$ et $(z_1^2, z_2^2) = (8, 0)$ qui est dominé par rapport aux couples retenus précédemment. $N_2 = \{6, 7\}$, $\Gamma_2 = \{7\}$.

Étape (3)

$j_2 = 7$, l'arête E_7 contient un seul point entier, le point $(4, 1)$. Ceci donne le couple dominé $(7, 3)$. $\Gamma_2 \neq \emptyset$ et toutes les solutions entières se trouvant sur l'arête E_7 sont inefficaces ; d'après Gupta et Malhotra, la procédure s'arrête en donnant l'ensemble des solutions efficaces $ESE(P) = \{(1, 3), (3, 2), (5, 1)\}$ et celui des solutions non dominées $SND(P) = \{(10, -2), (9, 1), (8, 4)\}$. Il se trouve que le point $(5, 0)$ dont le vecteur critère est $(5, 5)$ est efficace pour le problème (P) et pourtant, il n'a pas été généré par la méthode.

Conclusion

Le second critère d'arrêt, à savoir, $\Gamma_n \neq \emptyset$ mais pour tout $j \in \Gamma_n$, toutes les solutions entières se trouvant sur l'arête E_j sont inefficaces, n'est pas correct. Si ce dernier n'est pas pris en considération, la méthode permet effectivement de générer toutes les solutions efficaces ainsi que toutes les solutions non dominées du problème.

2.3 Méthode de Gupta et Malhotra (2^{ème} procédure) [8]

Cette méthode est une version améliorée de celle de Klein et Hannan [9]. Elle permet de réduire considérablement le nombre de contraintes additionnelles à chaque étape.

Étape 0 :

Résoudre le problème (P_0) : $Max \{z_1 = c^1 x | x \in S\}$.

Trouver toutes les solutions optimales de (P_0) et retenir celles dont les vecteurs critères sont non dominés. Notons par $\Theta(P_0)$ l'ensemble de toutes les solutions efficaces trouvées à l'étape (0).

Étape j :

Choisir une solution $x^* \in \Theta(P_{j-1})$, ensemble de toutes les solutions potentiellement efficaces obtenues à l'étape $(j - 1)$ telle que $z_1(x^*) = \max_{x \in \Theta(P_{j-1})} \{c^1 x\}$ et résoudre le problème (P_j) suivant :

$$(P_j) \left\{ \begin{array}{l} Max \quad z_1 = c^1 x \\ x \in S \\ c^1 x \leq c^1 x^* - 1 \\ c^i x \geq c^i x^* + 1 \quad \text{pour au moins un } i \in \{2, \dots, k\} \end{array} \right.$$

Former l'ensemble $\Theta(P_j)$ de toutes les solutions potentiellement efficaces obtenues à l'étape (j) .

Étape finale n :

La procédure prend fin à l'étape (n) dans laquelle toutes les solutions du problème (P_n) sont inefficaces pour (P) ou le problème (P_n) est irréalisable.

Les méthodes qui vont suivre permettent de générer toutes les solutions non dominées et pas nécessairement toutes les solutions efficaces de (P) , c'est-à-dire au moins une solution efficace pour chaque point non dominé.

2.4 Méthode de Abbas et Moumene [13]

La méthode de Abbas et Moumene se présente comme suit :

Initialisation :

La première étape de cette méthode consiste à éliminer les vecteurs critères non essentiels. Pour ce faire, résoudre pour chaque $l \in \{1, \dots, k\}$, le programme linéaire (Rd_l) suivant :

$$(Rd_l) \left\{ \begin{array}{l} \text{Min } \varphi_l = \sum_{j=1}^n v_j \\ \sum_{i=1, i \neq l}^k c_j^i y_i + v_j = c_j^l \quad j = \overline{1, n} \\ y_i \geq 0 \quad i = \overline{1, k}, i \neq l \\ v_j \geq 0 \quad j = \overline{1, n} \end{array} \right.$$

Si $\varphi_l = 0$, supprimer le $l^{\text{ème}}$ critère du problème (P) .

Réordonner les k' vecteurs critères restants (essentiels).

Créer une liste F des problèmes non encore résolus et l'initialiser à (P_0) , le programme linéaire unicritère à variables entières suivant :

$$(P_0) \quad \text{Max} \left\{ \sum_{i=1}^{k'} \lambda_i c^i x \mid x \in S \right\}.$$

Où $\lambda_i > 0$ pour $i = \overline{1, k'}$.

Étape générale :

Si la liste F est vide, terminer, l'ensemble des solutions non dominées $SND(P)$ est entièrement déterminé. Sinon, résoudre le problème (P_r) de plus fort indice r dans F .

Si (P_r) n'admet pas de solution, faire $F \leftarrow F \setminus \{P_r\}$.

Sinon, soit x^r la solution optimale de (P_r) , celle-ci est efficace pour (P) . Former k' problèmes identiques au problème (P_r) et les noter $(P_{r+1}), \dots, (P_{r+k'})$. Ajouter alors

à (P_{r+1}) , la contrainte $c^1x \geq c^1x^r + 1$ et à (P_{r+i_0}) , $i_0 = \overline{2, k'}$, le système

$$\begin{cases} c^{i_0}x \geq c^{i_0}x^r + 1 \\ c^{i_1}x \leq c^{i_1}x^r, \quad i_1 = \overline{1, (i_0 - 1)} \end{cases}$$

Actualiser la liste F ($F \leftarrow F \cup \{P_{r+1}, \dots, P_{r+k'}\} \setminus \{P_r\}$).

Contrairement à ce qu'il a été affirmé par les auteurs de cette méthode, une solution optimale d'un problème (P_r) n'est pas nécessairement efficace pour le problème initial (P) comme nous allons le voir à travers l'exemple ci-dessous.

Soit à résoudre le problème (P) suivant :

$$(P) \begin{cases} \text{Max } z_1 = x_1 \\ \text{Max } z_2 = x_2 \\ \text{Max } z_3 = x_3 \\ (x_1, x_2, x_3) \in S \end{cases}$$

Où $S = \{(x_1, x_2, x_3) \in \mathbb{Z}^3 \mid x_1 + 2x_2 \leq 6, x_2 + 2x_3 \leq 6, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0\}$.

Dans cet exemple, les solutions réalisables et les vecteurs critères correspondants sont confondus.

Prenons $\lambda_1 = 1$, $\lambda_2 = 3$ et $\lambda_3 = 2$.

Soit (P_0) le problème : $\text{Max } \{z = x_1 + 3x_2 + 2x_3 \mid (x_1, x_2, x_3) \in S\}$ et $F \leftarrow \{P_0\}$.

Étape (1)

On sélectionne le problème (P_0) . La résolution de ce dernier donne le point optimal $(2, 2, 2)$.

On génère trois sous-problèmes (P_1) , (P_2) et (P_3) en ajoutant respectivement au problème (P_0) , les systèmes de contraintes suivants :

$$\begin{cases} x_1 \geq 3 \end{cases}, \quad \begin{cases} x_2 \geq 3 \\ x_1 \leq 2 \end{cases} \quad \text{et} \quad \begin{cases} x_3 \geq 3 \\ x_1 \leq 2 \\ x_2 \leq 2 \end{cases}$$

$F \leftarrow \{P_1, P_2, P_3\}$.

Étape (2)

On sélectionne le problème (P_3) . La résolution de ce dernier donne le point optimal $(2, 0, 3)$.

On génère trois sous-problèmes (P_4) , (P_5) et (P_6) en ajoutant respectivement au problème (P_3) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 3 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 1 \\ x_1 \leq 2 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 4 \\ x_1 \leq 2 \\ x_2 \leq 0 \end{array} \right.$$

$$F \leftarrow \{P_1, P_2, P_4, P_5, P_6\}.$$

Étape (3)

On sélectionne le problème (P_6) . Ce dernier est irréalisable. $F \leftarrow \{P_1, P_2, P_4, P_5\}$.

Étape (4)

On sélectionne le problème (P_5) . Ce dernier est irréalisable. $F \leftarrow \{P_1, P_2, P_4\}$.

Étape (5)

On sélectionne le problème (P_4) . Ce dernier est irréalisable. $F \leftarrow \{P_1, P_2\}$.

Étape (6)

On sélectionne le problème (P_2) . La résolution de ce dernier donne le point optimal $(0, 3, 1)$.

On génère trois sous-problèmes (P_3) , (P_4) et (P_5) en ajoutant respectivement au problème (P_2) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 1 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 4 \\ x_1 \leq 0 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 2 \\ x_1 \leq 0 \\ x_2 \leq 3 \end{array} \right.$$

$$F \leftarrow \{P_1, P_3, P_4, P_5\}.$$

Étape (7)

On sélectionne le problème (P_5) . Ce dernier est irréalisable. $F \leftarrow \{P_1, P_3, P_4\}$.

Étape (8)

On sélectionne le problème (P_4) . Ce dernier est irréalisable. $F \leftarrow \{P_1, P_3\}$.

Étape (9)

On sélectionne le problème (P_3) . Ce dernier est irréalisable. $F \leftarrow \{P_1\}$.

Étape (10)

On sélectionne le problème (P_1) . La résolution de ce dernier donne le point optimal $(6, 0, 3)$.

On génère trois sous-problèmes (P_2) , (P_3) et (P_4) en ajoutant respectivement au problème (P_1) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 7 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 1 \\ x_1 \leq 6 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 4 \\ x_1 \leq 6 \\ x_2 \leq 0 \end{array} \right.$$

$$F \leftarrow \{P_2, P_3, P_4\}.$$

Étape (11)

On sélectionne le problème (P_4) . Ce dernier est irréalisable. $F \leftarrow \{P_2, P_3\}$.

Étape (12)

On sélectionne le problème (P_3) . La résolution de ce dernier donne le point optimal $(4, 1, 2)$.

On génère trois sous-problèmes (P_4) , (P_5) et (P_6) en ajoutant respectivement au problème (P_3) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 5 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 2 \\ x_1 \leq 4 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 3 \\ x_1 \leq 4 \\ x_2 \leq 1 \end{array} \right.$$

$$F \leftarrow \{P_2, P_4, P_5, P_6\}.$$

Étape (13)

On sélectionne le problème (P_6). Ce dernier est irréalisable. $F \leftarrow \{P_2, P_4, P_5\}$.

Étape (14)

On sélectionne le problème (P_5). Ce dernier est irréalisable. $F \leftarrow \{P_2, P_4\}$.

Étape (15)

On sélectionne le problème (P_4). Ce dernier est irréalisable. $F \leftarrow \{P_2\}$.

Étape (16)

On sélectionne le problème (P_2). Ce dernier est irréalisable. $F \leftarrow \emptyset$. Terminer.

Finalement, les solutions trouvées par la méthode sont : $(2, 2, 2)$, $(2, 0, 3)$, $(0, 3, 1)$, $(6, 0, 3)$ et $(4, 1, 2)$. Il est clair que la solution $(2, 0, 3) \notin SND(P)$ puisqu'elle est dominée par la solution $(6, 0, 3)$. Il sera donc plus judicieux de comparer entre toutes les solutions générées afin de trouver l'ensemble de toutes les solutions non dominées du problème.

2.5 Méthode de Sylva et Crema [17]

Cette méthode est une variante de celle de Klein et Hannan [9], maximisant à chaque étape non pas un des critères du problème mais une combinaison positive de toutes les fonctions objectifs. Ainsi, la solution optimale de chaque programme linéaire en nombres entiers résolu est efficace pour le problème multiobjectif.

Les résultats suivants permettent de justifier la méthode :

Proposition 2.1. Soient x^1, \dots, x^r des solutions efficaces pour le problème (P) et $\Delta_l = \{x \in \mathbb{Z}^n | Cx \leq Cx^l\}$ où C est une $k \times n$ -matrice composée des vecteurs lignes c^i , $i = \overline{1, k}$. Soit x^* une solution efficace pour le problème (P_r) suivant :

$$(P_r) : \text{“Max”} \left\{ Cx \mid x \in S - \bigcup_{l=1}^r \Delta_l \right\}.$$

Alors x^* est une solution efficace pour (P) .

De plus, si le problème (P_r) est irréalisable, alors $\{Cx^l\}_{l=1}^r$ est l'ensemble de tous les vecteurs critères non dominés de (P) .

Corollaire 2.1. Soient x^1, \dots, x^r des solutions efficaces pour le problème (P) et $\Delta_l = \{x \in \mathbb{Z}^n | Cx \leq Cx^l\}$. Si x^* est une solution optimale pour le problème

$$(P_r^\lambda) : \text{“Max”} \left\{ \lambda^t Cx \mid x \in S - \bigcup_{l=1}^r \Delta_l \right\}$$

pour certaines valeurs de $\lambda \in \mathbb{R}^k$, $\lambda > 0$, alors elle est efficace pour (P) .

Description de la méthode

Étape 0 :

Après avoir choisi le paramètre λ , on résout le problème $(P_0) : \text{Max} \{ \lambda^t Cx \mid x \in S \}$. Si ce problème est irréalisable, il en est de même pour (P) . Sinon, une solution optimale x^1 est trouvée; celle ci est efficace pour le problème (P) .

Étape r ($r \geq 1$) :

Résoudre le problème :

$$(P_r) \left\{ \begin{array}{l} \text{Max} \quad \lambda^t Cx \\ x \in S \\ (Cx)_i \geq ((Cx^l)_i + 1)y_i^l - M_i(1 - y_i^l) \quad \forall l = \overline{1, r}; i = \overline{1, k} \\ \sum_{i=1}^k y_i^l \geq 1 \quad \forall l = \overline{1, r} \\ y_i^l \in \{0, 1\} \quad \forall l = \overline{1, r}; i = \overline{1, k} \end{array} \right.$$

où $-M_i$ est un minorant pour toute valeur réalisable de la $i^{\text{ème}}$ fonction objectif. Dans le cas où les éléments de C sont tous positifs, M_i peut être fixé à 0 pour $i = \overline{1, k}$.

Si (P_r) est irréalisable, terminer. $\{Cx^l\}_{l=1}^r$ est l'ensemble de tous les vecteurs critères non dominés de (P) .

Sinon, une nouvelle solution efficace pour (P) , notée x^{r+1} , est trouvée.

Pour des problèmes de grande taille, l'énumération de toutes les solutions non dominées n'est plus envisageable. Néanmoins, un sous-ensemble de ces dernières peut être obtenu en remplaçant le problème (P_r) par :

$$\left\{ \begin{array}{l} \text{Max} \quad \lambda^t Cx \\ x \in S \\ (Cx)_i \geq ((Cx^l)_i + f_i)y_i^l - M_i(1 - y_i^l) \quad \forall l = \overline{1, r}; i = \overline{1, k} \\ \sum_{i=1}^k y_i^l \geq 1 \quad \forall l = \overline{1, r} \\ y_i^l \in \{0, 1\} \quad \forall l = \overline{1, r}; i = \overline{1, k} \end{array} \right.$$

Où f_i est un entier supérieur à 1, représentant l'amélioration minimale dans le $i^{\text{ème}}$ critère pour $i = \overline{1, k}$.

2.6 Méthode de Chergui *et al.* [3]

Cette méthode génère l'ensemble des solutions non dominées d'un programme linéaire multiobjectif en nombres entiers en résolvant une suite finie de programmes linéaires unicritères en variables discrètes. À l'étape r de la méthode, le problème (P_r) est créé à partir de $(P_l)_{0 \leq l < r}$ en lui ajoutant une contrainte appelée *pseudocoupe* qui s'écrit sous la forme $x_{j_l} \geq 1$ où j_l est un indice de variable hors base relativement à x^l , solution optimale de (P_l) . Cet indice est choisi de sorte que la solution de (P_r) sera meilleure que x^l pour au moins un critère.

Notations

Soit (P_l) le programme linéaire unicritère obtenu à la $l^{\text{ème}}$ étape de la méthode :

$$(P_l) \left\{ \begin{array}{l} \text{Max} \quad z_1(x) \\ x \in D_l \\ x \text{ entier} \end{array} \right.$$

Initialement $D_0 = D$.

Après résolution de (P_l) , on définit les notions suivantes :

- x^l est une solution optimale du problème (P_l) ;
- B_l (resp. N_l) est l'ensemble des indices des variables de base (resp. hors base) de x^l ;
- $H_l = \{j \in N_l \mid \exists i \in \{1, \dots, k\}; (\tilde{c}_l^i)^j > 0\}$ où $(\tilde{c}_l^i)^j$ est la $j^{\text{ème}}$ composante du vecteur coût réduit \tilde{c}_l^i relatif à la fonction objectif z_i ;
- $D_r = \{x \in D_l \mid x_{j_l} \geq 1\}$, $j_l \in H_l$, $r > l$.

Cet ensemble désigne la région tronquée du domaine D_l obtenue par adjonction de la pseudocoupe $x_{j_l} \geq 1$ et éventuellement par des coupes successives de Gomory.

La méthode de Chergui *et al.* peut être décrite comme suit :

Étape 0 (initialisation) :

Résoudre le problème (P_0) .

Trouver toutes les solutions optimales de (P_0) et retenir celles dont les vecteurs critères sont non dominés. Adjoindre ces derniers à $SND(P)$.

Étape r ($r \geq 1$) :

Considérer une solution optimale entière x^l d'un programme linéaire (P_l) , $0 \leq l < r$. Choisir un indice $j_l \in H_l$ non encore étudié. Construire l'ensemble D_r et résoudre le programme linéaire (P_r) sur cet ensemble. Si ce problème admet une solution optimale, déterminer le vecteur critère correspondant et mettre à jour l'ensemble $SND(P)$.

Critère d'arrêt :

La procédure s'arrête dès que $H_l = \emptyset$ ou bien (P_l) n'admet pas de solutions réalisables pour toute étape l telle que $0 \leq l \leq r$.

Contribution algorithmique et implémentation

Nous nous intéressons dans ce chapitre à deux méthodes d'optimisation multiobjectif discrète parmi celles décrites précédemment. Il s'agit de la méthode de Abbas et Moumene [13] et celle de Chergui et al. [3]. Nous présentons tout d'abord une revue rapide de ces dernières, puis nous exposons en détail quelques améliorations proposées pour chacune d'elles illustrées par des exemples et pour finir, des résultats expérimentaux sont donnés et commentés.

Sommaire

3.1 Amélioration de la méthode de Abbas et Moumene [13]	36
3.1.1 Algorithme de la méthode améliorée	37
3.1.2 Résultats théoriques	38
3.1.3 Exemple numérique	40
3.2 Amélioration de la méthode de Chergui et al. [3]	44
3.2.1 Algorithme de la méthode améliorée	45
3.2.2 Résultats théoriques	46
3.2.3 Exemple numérique	50
3.3 Implémentation et résultats	55

Soient c^1, \dots, c^k , des vecteurs de $\mathbb{R}^{1 \times n}$ et S l'ensemble des vecteurs $x \in \mathbb{Z}^n$ satisfaisant les contraintes $Ax = b$ et $x \geq 0$ où $A \in \mathbb{Z}^{m \times n}$ et $b \in \mathbb{Z}^m$. Cet ensemble est supposé compact et non vide.

Considérons le programme linéaire multiobjectif en nombres entiers (P) suivant :

$$(P) : \text{“Max” } \{Cx \mid x \in S\}$$

où C est une $k \times n$ -matrice composée des vecteurs lignes c^i .

3.1 Amélioration de la méthode de Abbas et Moumene [13]

Nous avons vu dans le chapitre précédent que la méthode de Abbas et Moumene [13] génère toutes les solutions non dominées et pas nécessairement toutes les solutions efficaces de (P) en procédant comme suit :

Initialement, la liste (F) des problèmes non encore résolus est réduite au problème (P_0) et l'ensemble des solutions non dominées de (P), noté $SND(P)$, à l'ensemble vide. Rappelons que (P_0) est le programme linéaire unicritère à variables entières suivant :

$$(P_0) : \text{Max} \left\{ \sum_{i=1}^{k'} \lambda_i c^i x \mid x \in S \right\}$$

où k' est le nombre de vecteurs critères essentiels du problème et $\lambda_i > 0$ pour $i = \overline{1, k'}$. À chaque étape de la méthode, un problème (P_r) de plus fort indice est résolu et est retiré de (F). Si ce dernier possède une solution, l'ensemble $SND(P)$ est mis à jour et les problèmes $(P_{r+1}), \dots, (P_{r+k'})$ sont créés à partir de (P_r) en lui ajoutant des contraintes sur les valeurs des objectifs, ces k' problèmes sont alors ajoutés à (F).

La procédure prend fin lorsque la liste (F) sera vide.

La méthode qui va être exposée dans ce paragraphe est une version améliorée de celle décrite plus haut où à chaque fois qu'un problème (P_r) admet une solution, on restreint le nombre de sous-problèmes à créer au nombre d'objectifs pouvant encore augmenter. Nous montrons en effet, qu'un sous-problème obtenu à partir de (P_r) en lui ajoutant une contrainte traduisant l'augmentation d'un critère ayant toutes les composantes du vecteur coût réduit associé négatives ou nulles, ne possède aucune

solution réalisable et il n'y a pas d'intérêt à le créer.

3.1.1 Algorithme de la méthode améliorée

Ici, tous les critères du problème sont considérés et le problème (P_0) devient :

$$(P_0) : \text{Max} \left\{ \sum_{i=1}^k \lambda_i c^i x \mid x \in S \right\} \text{ où } \lambda_i > 0 \text{ pour } i = \overline{1, k}.$$

En plus, nous supposons que $c^i \in \mathbb{Z}^{1 \times n}$ pour $i = \overline{1, k}$.

Algorithme 1 Recherche des solutions non dominées d'un problème multiobjectif

Initialisation : $SND(P) \leftarrow \emptyset$, $F \leftarrow \{P_0\}$.

Tant que $F \neq \emptyset$ **faire**

Résoudre le problème (P_r) de plus fort indice r dans F en utilisant la méthode duale fractionnaire de Gomory.

Si (P_r) est irréalisable **alors**

$F \leftarrow F \setminus \{P_r\}$.

Sinon

soit x^r la solution optimale de (P_r) , déterminer le vecteur critère correspondant et mettre à jour l'ensemble $SND(P)$;

construire l'ensemble $I_r = \{i \in \{1, \dots, k\} \mid \exists j \in N_r, (\tilde{c}_r^i)^j > 0\}$ où

N_r est l'ensemble des indices des variables hors base de x^r et $(\tilde{c}_r^i)^j$ est la $j^{\text{ème}}$ composante du vecteur coût réduit \tilde{c}_r^i relatif au critère i .

Si $I_r \neq \emptyset$ **alors**

former t_r problèmes identiques au problème (P_r) où t_r est le cardinal de I_r et les noter $(P_{r+1}), \dots, (P_{r+t_r})$;

considérer l'ensemble J_r initialement vide et poser $l \leftarrow 1$.

Pour $i_0 \in I_r$ **faire**

Ajouter au problème (P_{r+l}) , la contrainte $c^{i_0} x \geq c^{i_0} x^r + 1$.

Pour $i_1 \in J_r$ **faire**

Ajouter au problème (P_{r+l}) , la contrainte $c^{i_1} x \leq c^{i_1} x^r$.

Fin pour

$J_r \leftarrow J_r \cup \{i_0\}$, $l \leftarrow l + 1$.

Fin pour

$F \leftarrow F \cup \{P_{r+1}, \dots, P_{r+t_r}\} \setminus \{P_r\}$.

Fin si

Fin si

Fin tant que

Retourner l'ensemble des solutions non dominées $SND(P)$.

3.1.2 Résultats théoriques

Nous donnons dans ce paragraphe, les principaux résultats justifiant les étapes de l'algorithme décrit plus haut.

Soient x^r la solution optimale du problème (P_r) , S_r son domaine d'admissibilité et I_r l'ensemble défini par : $I_r = \{i \in \{1, \dots, k\} \mid \exists j \in N_r, (\tilde{c}_r^i)^j > 0\}$ où k est le nombre de critères du problème (P) , N_r est l'ensemble des indices des variables hors base de x^r et $(\tilde{c}_r^i)^j$ est la $j^{\text{ème}}$ composante du vecteur coût réduit \tilde{c}_r^i relatif au critère i .

Proposition 3.1. *Considérons les ensembles*

$$A = S_r \setminus \{x \in S_r \mid c^i x \leq c^i x^r, i = \overline{1, k}\};$$

$$B = \bigcup_{i=1}^k B_i \text{ où } \begin{cases} B_1 = \{x \in S_r \mid c^1 x > c^1 x^r\} \\ B_i = \{x \in S_r \mid c^i x > c^i x^r, c^p x \leq c^p x^r, p = \overline{1, (i-1)}\}, i = \overline{2, k} \end{cases}$$

On a alors l'égalité suivante : $A = B$.

De plus, on a $B_i \cap B_j = \emptyset, \forall i \in \{1, \dots, k\}, \forall j \in \{1, \dots, k\}, i \neq j$.

Démonstration. Commençons par montrer que $A = B$.

Soit $V_i = \{x \in S_r \mid c^i x > c^i x^r\}, i = \overline{1, k}$.

\Rightarrow) Supposons qu'il existe $\bar{x} \in A$ tel que $\bar{x} \notin B$.

1. D'une part, $\bar{x} \in A$ signifie qu'il existe un ensemble $K \subseteq \{1, \dots, k\}$ tel que $K \neq \emptyset$ et pour tout $i \in K$, on a $\bar{x} \in V_i$.

2. D'autre part, $\bar{x} \notin B$ signifie que $\forall i \in \{1, \dots, k\}, \bar{x} \notin B_i$.

Or,

- $1 \notin K$, car sinon, $\bar{x} \in V_1$ et par conséquent, $\bar{x} \in B_1$, ce qui contredirait l'hypothèse que $\bar{x} \notin B$.
- $2 \notin K$, car sinon, $\bar{x} \in V_2$ et par conséquent, $\bar{x} \in V_1$ puisque $\bar{x} \notin B_2$, ce qui contredirait le fait que $1 \notin K$.
- $3 \notin K$, car sinon, $\bar{x} \in V_3$ et par conséquent, $\exists i_0 \in \{1, 2\}$ tel que $\bar{x} \in V_{i_0}$ puisque $\bar{x} \notin B_3$, ce qui contredirait le fait que $i_0 \notin K$.
- $k \notin K$, car sinon, $\bar{x} \in V_k$ et par conséquent, $\exists i_0 \in \{1, \dots, (k-1)\}$ tel que

$\bar{x} \in V_{i_0}$ puisque $\bar{x} \notin B_k$, ce qui contredirait le fait que $i_0 \notin K$.

Contradiction avec $K \neq \emptyset$.

\Leftrightarrow) Supposons qu'il existe $\bar{x} \in B$ tel que $\bar{x} \notin A$.

1. D'une part, $\bar{x} \notin A$ signifie que $c^i \bar{x} \leq c^i x^r$, $i = \overline{1, k}$.

2. D'autre part, $\bar{x} \in B$ signifie qu'il existe un i_0 tel que $\bar{x} \in B_{i_0}$, donc $c^{i_0} \bar{x} > c^{i_0} x^r$ et $c^p \bar{x} \leq c^p x^r$ pour $p = \overline{1, (i_0 - 1)}$. Contradiction.

Conclusion : $A = B$.

Démontrons maintenant que $\forall i, j \in \{1, \dots, k\}$, $i \neq j$, on a $B_i \cap B_j = \emptyset$.

Supposons qu'ils existent $i, j \in \{1, \dots, k\}$, $i \neq j$, tels que $B_i \cap B_j \neq \emptyset$.

Soit $x \in B_i \cap B_j$, ce qui signifie que

$$x \in S_r \text{ et } \begin{cases} c^i x > c^i x^r, & c^p x \leq c^p x^r, & p = \overline{1, (i - 1)} \\ c^j x > c^j x^r, & c^l x \leq c^l x^r, & l = \overline{1, (j - 1)} \end{cases}$$

1^{er} cas :

Si $i > j$, alors pour $p = j$, on a, $c^j x \leq c^j x^r$. Or $c^j x > c^j x^r$, contradiction.

2^{ème} cas :

Si $i < j$, alors pour $l = i$, on a, $c^i x \leq c^i x^r$. Or $c^i x > c^i x^r$, contradiction.

Donc, $\forall i, j \in \{1, \dots, k\}$, $i \neq j$, $B_i \cap B_j = \emptyset$. □

Proposition 3.2. Soit $i_0 \in \{1, \dots, k\} \setminus I_r$.

a) Le domaine S_r^1 définit par : $S_r^1 = \{x \in S_r \mid c^{i_0} x \geq c^{i_0} x^r + 1\}$ est vide.

b) Soit $S_r^2 = \{x \in S_r \mid c^{i_0} x \leq c^{i_0} x^r\}$. On a alors $S_r^2 = S_r$.

Démonstration. Soit $i_0 \in \{1, \dots, k\} \setminus I_r$.

a) Supposons que $S_r^1 \neq \emptyset$. Alors, $\exists x \in S_r^1$ tel que $x \in S_r$ et $c^{i_0} x \geq c^{i_0} x^r + 1$. Ainsi, x doit vérifier le système suivant :

$$\begin{cases} c^{i_0} x = c^{i_0} x^r + \sum_{j \in N_r} (\widehat{c}_r^{i_0})^j x_j & (1) \\ c^{i_0} x \geq c^{i_0} x^r + 1 & (2) \end{cases}$$

De (1) et (2), on obtient : $\sum_{j \in N_r} (\tilde{c}_r^{i_0})^j x_j \geq 1$, ce qui est absurde compte tenu du fait que $(\tilde{c}_r^{i_0})^j \leq 0, \forall j \in N_r$.

b) Soit $x \in S_r$, alors $c^{i_0}x = c^{i_0}x^r + \sum_{j \in N_r} (\tilde{c}_r^{i_0})^j x_j$ et comme $(\tilde{c}_r^{i_0})^j \leq 0, \forall j \in N_r$, alors $c^{i_0}x \leq c^{i_0}x^r$, donc $x \in S_r^2$ et par conséquent, $S_r \subset S_r^2$, d'où l'égalité des deux ensembles. \square

Ce résultat nous permet de considérer un nombre réduit d'ensembles B_i définis dans la proposition 3.1 compte tenu du fait que ceux non considérés sont vides. Ainsi, le nombre de sous-problèmes à traiter sera inférieur à celui rapporté dans [13].

Proposition 3.3. *L'algorithme ci-dessus se termine après un nombre fini d'étapes.*

Démonstration. L'ensemble des solutions réalisables S est par hypothèse fermé et borné. De plus, à chaque étape de la méthode, on détermine x^r , solution optimale du problème (P_r) lorsqu'elle existe. L'ensemble S_r privé de toutes les solutions x vérifiant $Cx \leq Cx^r$ est alors séparé en sous-ensembles disjoints $S_l, l > r$ vérifiant $\bigcup_{l | l > r} S_l \subset S_r$ compte tenu de la proposition 3.1. Ainsi, l'algorithme converge vers des sous-ensembles vides et prend fin lorsqu'ils sont tous atteints. \square

3.1.3 Exemple numérique

Reprenons l'exemple qui a servi comme illustration de la méthode de Abbas et Moumene dans le chapitre précédent.

$$(P) \begin{cases} \text{Max } z_1 = x_1 \\ \text{Max } z_2 = x_2 \\ \text{Max } z_3 = x_3 \\ (x_1, x_2, x_3) \in S \end{cases}$$

Où $S = \{(x_1, x_2, x_3) \in \mathbb{Z}^3 \mid x_1 + 2x_2 \leq 6, x_2 + 2x_3 \leq 6, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0\}$.

$\lambda_1 = 1, \lambda_2 = 3$ et $\lambda_3 = 2$.

Soit (P_0) le problème : $\text{Max } \{z = x_1 + 3x_2 + 2x_3 \mid (x_1, x_2, x_3) \in S\}$. $F \leftarrow \{P_0\}$ et $SND(P) \leftarrow \emptyset$.

Étape (1)

On sélectionne le problème (P_0) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_4	x_5	x_7	x_B
x_2	0	-1	1	2
x_3	0	1	$-\frac{1}{2}$	2
x_1	1	2	-2	2
x_6	0	1	$-\frac{3}{2}$	1
$-z$	-1	-1	0	-12
$-z_1$	-1	-2	2	-2
$-z_2$	0	1	-1	-2
$-z_3$	0	-1	$\frac{1}{2}$	-2

$x^0 = (2, 2, 2)$, le vecteur critère correspondant est $(2, 2, 2)$. $SND(P) \leftarrow \{(2, 2, 2)\}$.

$I_0 = \{1, 2, 3\}$. On génère donc, trois sous-problèmes (P_1) , (P_2) et (P_3) en ajoutant respectivement au problème (P_0) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 3 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 3 \\ x_1 \leq 2 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 3 \\ x_1 \leq 2 \\ x_2 \leq 2 \end{array} \right.$$

$$F \leftarrow \{P_1, P_2, P_3\}.$$

Étape (2)

On sélectionne le problème (P_3) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_5	x_8	x_9	x_B
x_2	1	2	0	0
x_3	0	-1	0	3
x_1	0	0	1	2
x_6	-2	-3	0	4
x_7	-2	-2	0	2
x_4	-2	-4	-1	4
x_{10}	-1	-2	0	2
$-z$	-3	-4	-1	-8
$-z_1$	0	0	-1	-2
$-z_2$	-1	-2	0	0
$-z_3$	0	1	0	-3

$x^3 = (2, 0, 3)$, le vecteur critère correspondant est $(2, 0, 3)$.

$SND(P) \leftarrow \{(2, 2, 2), (2, 0, 3)\}$. $I_3 = \{3\}$.

On génère un sous-problème (P_4) en ajoutant au problème (P_3), la contrainte $x_3 \geq 4$.

$F \leftarrow \{P_1, P_2, P_4\}$.

Étape (3)

On sélectionne le problème (P_4). Ce dernier est irréalisable. $F \leftarrow \{P_1, P_2\}$.

Étape (4)

On sélectionne le problème (P_2). La résolution de ce dernier donne le tableau optimal suivant :

B	x_4	x_7	x_8	x_B
x_2	0	0	-1	3
x_3	0	$\frac{1}{2}$	1	1
x_1	1	0	2	0
x_6	0	$-\frac{1}{2}$	1	0
x_5	0	-1	-1	1
x_9	-1	0	-2	2
$-z$	-1	-1	-1	-11
$-z_1$	-1	0	-2	0
$-z_2$	0	0	1	-3
$-z_3$	0	$-\frac{1}{2}$	-1	-1

$x^2 = (0, 3, 1)$, le vecteur critère correspondant est $(0, 3, 1)$.

$SND(P) \leftarrow \{(2, 2, 2), (2, 0, 3), (0, 3, 1)\}$. $I_2 = \{2\}$.

On génère un sous-problème (P_3) en ajoutant au problème (P_2), la contrainte $x_2 \geq 4$.

$F \leftarrow \{P_1, P_3\}$.

Étape (5)

On sélectionne le problème (P_3). Ce dernier est irréalisable. $F \leftarrow \{P_1\}$.

Étape (6)

On sélectionne le problème (P_1) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_4	x_5	x_{10}	x_B
x_2	0	-1	2	0
x_3	0	1	-1	3
x_1	1	2	-4	6
x_6	0	1	-3	4
x_7	0	0	-2	2
x_8	1	2	-4	3
x_9	0	1	-2	1
$-z$	-1	-1	0	-12
$-z_1$	-1	-2	4	-6
$-z_2$	0	1	-2	0
$-z_3$	0	-1	1	-3

$x^1 = (6, 0, 3)$, le vecteur critère correspondant est $(6, 0, 3)$. Celui-ci domine le point $(2, 0, 3)$ qui va donc être retiré de $SND(P)$.

$SND(P) \leftarrow \{(2, 2, 2), (0, 3, 1), (6, 0, 3)\}$. $I_1 = \{1, 2, 3\}$.

On génère trois sous-problèmes (P_2) , (P_3) et (P_4) en ajoutant respectivement au problème (P_1) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 7 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 \geq 1 \\ x_1 \leq 6 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_3 \geq 4 \\ x_1 \leq 6 \\ x_2 \leq 0 \end{array} \right.$$

$F \leftarrow \{P_2, P_3, P_4\}$.

Étape (7)

On sélectionne le problème (P_4) . Ce dernier est irréalisable. $F \leftarrow \{P_2, P_3\}$.

Étape (8)

On sélectionne le problème (P_3) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_4	x_{10}	x_{11}	x_B
x_2	0	0	-1	1
x_3	0	1	1	2
x_1	1	0	2	4
x_6	0	-1	1	3
x_7	0	-2	0	2
x_8	1	0	2	1
x_9	0	0	1	0
x_5	0	-2	-1	1
x_{12}	-1	0	-2	2
$-z$	-1	-2	-1	-11
$-z_1$	-1	0	-2	-4
$-z_2$	0	0	1	-1
$-z_3$	0	-1	-1	-2

$x^3 = (4, 1, 2)$, le vecteur critère correspondant est $(4, 1, 2)$.

$SND(P) \leftarrow \{(2, 2, 2), (0, 3, 1), (6, 0, 3), (4, 1, 2)\}$. $I_3 = \{2\}$.

On génère un sous-problème (P_4) en ajoutant au problème (P_3), la contrainte $x_2 \geq 2$.

$F \leftarrow \{P_2, P_4\}$.

Étape (9)

On sélectionne le problème (P_4). Ce dernier est irréalisable. $F \leftarrow \{P_2\}$.

Étape (10)

On sélectionne le problème (P_2). Ce dernier est irréalisable. $F \leftarrow \emptyset$. Terminer.

Finalement, les solutions non dominées du problème (P) sont : $(2, 2, 2)$, $(0, 3, 1)$, $(6, 0, 3)$ et $(4, 1, 2)$.

3.2 Amélioration de la méthode de Chergui *et al.* [3]

Pour commencer, rappelons brièvement comment la méthode de Chergui *et al.* [3] génère-t-elle toutes les solutions non dominées du programme linéaire multiobjectif en nombres entiers (P).

Initialement, le problème (P_0) : $Max \{c^1 x | x \in S\}$ est résolu et la(les) première(s)

solution(s) non dominée(s) est(sont) trouvée(s). À l'étape r de la méthode, un nouveau problème, noté (P_r) , est créé à partir de $(P_l)_{0 \leq l < r}$ en lui ajoutant la contrainte $x_{j_l} \geq 1$ où j_l est un indice de variable hors base relativement à x^l , solution optimale de (P_l) . Cet indice est choisi de sorte à améliorer au moins l'un des $(k - 1)$ critères restants. Le problème (P_r) est alors résolu pour trouver une éventuelle solution non dominée.

Cette méthode présente un inconvénient majeur, les domaines d'admissibilité des problèmes (P_r) issus du problème $(P_l)_{0 \leq l < r}$ ne sont pas disjoints. Nous proposons dans ce qui suit une technique permettant de contourner cette difficulté. Aussi, dans la perspective de déterminer toutes les solutions efficaces de (P) , nous allons également considérer, pour chaque solution x^l , les indices de variables hors base j_l vérifiant : $(\tilde{c}_i^{j_l})^{j_l} = 0, \forall i = \overline{1, k}$ où $(\tilde{c}_i^{j_l})^{j_l}$ est la $j_l^{\text{ème}}$ composante du vecteur coût réduit $\tilde{c}_i^{j_l}$ relatif au critère i .

3.2.1 Algorithme de la méthode améliorée

L'algorithme qui va suivre décrit une méthode améliorée de celle de Chergui *et al.* [3] qui tient compte du problème énoncé plus haut et détermine toutes les solutions efficaces du programme linéaire multiobjectif en nombres entiers (P) .

Désignons par :

- $ESE(P)$, l'ensemble de toutes les solutions efficaces de (P) ;
- (P_0) , le programme linéaire unicritère à variables entières suivant :

$$(P_0) \quad Max \{ \lambda^t C x \mid x \in S \}$$

où $\lambda \in \mathbb{R}^k, \lambda > 0$;

- F , une liste contenant les programmes linéaires non encore résolus.

Algorithme 2 Recherche des solutions efficaces d'un problème multiobjectif

Initialisation : $ESE(P) \leftarrow \emptyset$, $F \leftarrow \{P_0\}$, $r \leftarrow 1$.

Tant que $F \neq \emptyset$ **faire**

Choisir un problème (P_l) de F et le résoudre en utilisant la méthode duale fractionnaire de Gomory.

Si (P_l) est irréalisable **alors**

$F \leftarrow F \setminus \{P_l\}$.

Sinon

soit x^l la solution optimale de (P_l) , mettre à jour l'ensemble $ESE(P)$;

construire l'ensemble H_l suivant :

$$H_l = \{j \in N_l \mid \exists i \in \{1, \dots, k\}; (\tilde{c}_i^j)^j > 0\} \cup \{j \in N_l \mid \forall i \in \{1, \dots, k\}; (\tilde{c}_i^j)^j = 0\}$$

où N_l est l'ensemble des indices des variables hors base de x^l et $(\tilde{c}_i^j)^j$ est la $j^{\text{ème}}$ composante du vecteur coût réduit \tilde{c}_i^j relatif au critère i ;

considérer l'ensemble J_l initialement vide.

Tant que $H_l \neq \emptyset$ **faire**

Choisir un indice $j_l \in H_l$;

créer un nouveau problème (P_r) à partir de (P_l) en lui ajoutant les contraintes

$$x_{j_l} \geq 1 \text{ et } \sum_{j \in J_l} x_j \leq 0 \text{ si } J_l \neq \emptyset;$$

$$F \leftarrow F \cup \{P_r\}, H_l \leftarrow H_l \setminus \{j_l\}, J_l \leftarrow J_l \cup \{j_l\}, r \leftarrow r + 1.$$

Fin tant que

$F \leftarrow F \setminus \{P_l\}$;

Fin si

Fin tant que

Retourner l'ensemble des solutions efficaces $ESE(P)$.

3.2.2 Résultats théoriques

Dans ce paragraphe, la justification des étapes de l'algorithme ci-dessus est présentée.

Associés à x^l , solution optimale du problème (P_l) , les ensembles suivants :

$$H_l = \{j \in N_l \mid \exists i \in \{1, \dots, k\}; (\tilde{c}_i^j)^j > 0\} \cup \{j \in N_l \mid \forall i \in \{1, \dots, k\}; (\tilde{c}_i^j)^j = 0\}$$

où N_l est l'ensemble des indices des variables hors base de x^l et $(\tilde{c}_i^j)^j$ est la $j^{\text{ème}}$ composante du vecteur coût réduit \tilde{c}_i^j relatif au critère i ;

$$\bar{H}_l = N_l \setminus H_l;$$

$$S_r = \{x \in S_l \mid x_{j_l} \geq 1\}, j_l \in H_l, r > l;$$

$$T_r = \{x \in S_l \mid x_{j_l} \geq 1\}, j_l \in \bar{H}_l, r > l.$$

Où S_l est l'ensemble des solutions réalisables du problème (P_l) .

Proposition 3.4 ([3]). *L'ensemble S_l privé de x^l est obtenu par réunion des ensembles $S_r \subset S_l$ et $T_r \subset S_l$, $0 \leq l < r$.*

Démonstration. Notons Q_l l'ensemble $(\bigcup_{r \mid 0 \leq l < r} S_r) \cup (\bigcup_{r \mid 0 \leq l < r} T_r)$.

Soit $x \in S_l \setminus Q_l$ une solution réalisable de (P_l) , alors $x \in S_l$ et $\forall j_l \in (H_l \cup \overline{H}_l)$, $x_{j_l} = 0$ et par suite $x \in S_l$ et $\forall j_l \in N_l$, $x_{j_l} = 0$. Ainsi, x a le même ensemble des indices des variables hors base que x^l et donc, le même ensemble des indices des variables de base que x^l . Par conséquent, $x = x^l$, ce qui prouve que $S_l \setminus Q_l = \{x^l\}$ ou de façon analogue, $S_l \setminus \{x^l\} = Q_l$. \square

Ce résultat montre qu'aucune solution appartenant au domaine S_l des solutions réalisables du problème (P_l) privé de x^l ne peut être ignorée lorsque nous considérons la réunion des ensembles $S_r \subset S_l$ et $T_r \subset S_l$, $0 \leq l < r$.

Proposition 3.5 ([3]). *Les seuls sous-ensembles de S_l susceptibles de contenir des solutions efficaces pour le problème multiobjectif (P) , sont les sous-ensembles $S_r \subset S_l$, $0 \leq l < r$.*

Démonstration. Soit x^l une solution optimale pour le problème (P_l) , alors dans le tableau du simplexe donnant cette solution, tout choix d'une colonne pivot $j_l \in \overline{H}_l$ a pour conséquence l'obtention d'une solution x telle que $x_{j_l} \geq 1$.

De plus, le vecteur critère de x^l domine forcément celui de x compte tenu du fait que $j_l \in \overline{H}_l$ signifie que $(\hat{c}_i^l)^{j_l} \leq 0$ pour $i = \overline{1, k}$, avec au moins une inégalité stricte. Ainsi, toute opération de pivotage autour d'une colonne $j_l \in \overline{H}_l$, lorsqu'elle est possible, entraîne inévitablement une solution $x \in T_r$ qui n'est pas efficace pour le problème (P) . \square

Le résultat qui suit, permet de définir une partition pour l'ensemble

$$S_l \setminus \left\{ \left(\bigcup_{r \mid 0 \leq l < r} T_r \right) \cup \{x^l\} \right\}.$$

Proposition 3.6. Soit $H_l = \{j_1, j_2, \dots, j_t\} \subseteq N_l$.

On définit les ensembles

$$A^{j_i} = \{x \in S_l \mid x_{j_i} \geq 1\} \text{ pour } i = \overline{1, t};$$

$$B^{j_1} = A^{j_1};$$

$$B^{j_i} = \left\{ x \in S_l \mid x_{j_i} \geq 1, \sum_{p=1}^{i-1} x_{j_p} < 1 \right\} \text{ pour } i = \overline{2, t}.$$

On a alors l'égalité suivante : $\bigcup_{i=1}^t A^{j_i} = \bigcup_{i=1}^t B^{j_i}$.

De plus, on a $B_{j_i} \cap B_{j_s} = \emptyset, \forall j_i, j_s \in H_l, i \neq s$.

Démonstration. Commençons par montrer que $\bigcup_{i=1}^t A^{j_i} = \bigcup_{i=1}^t B^{j_i}$.

\Rightarrow) Supposons qu'il existe $\bar{x} \in \bigcup_{i=1}^t A^{j_i}$ tel que $\bar{x} \notin \bigcup_{i=1}^t B^{j_i}$.

1. D'une part, $\bar{x} \in \bigcup_{i=1}^t A^{j_i}$ signifie qu'il existe un $s \in \{1, \dots, t\}$ tel que $\bar{x} \in A^{j_s}$ et par conséquent, $\bar{x}_{j_s} \geq 1 \dots (1)$.

2. D'autre part, $\bar{x} \notin \bigcup_{i=1}^t B^{j_i}$ signifie que $\forall i \in \{1, \dots, t\}, \bar{x} \notin B^{j_i}$.

On aura,

- $\bar{x} \notin B^{j_1}$, par conséquent $\bar{x}_{j_1} < 1$. Comme $\bar{x}_{j_1} \in \mathbb{N}$, alors $\bar{x}_{j_1} = 0$.
- $\bar{x} \notin B^{j_2}$, par conséquent $\bar{x}_{j_2} < 1$ puisque $\bar{x}_{j_1} < 1$. Comme $\bar{x}_{j_2} \in \mathbb{N}$, alors $\bar{x}_{j_2} = 0$.
- $\bar{x} \notin B^{j_3}$, par conséquent $\bar{x}_{j_3} < 1$ puisque $\sum_{p=1}^2 \bar{x}_{j_p} = 0 < 1$. Comme $\bar{x}_{j_3} \in \mathbb{N}$, alors $\bar{x}_{j_3} = 0$.

En suivant ce raisonnement, on trouve $\bar{x}_{j_s} < 1$. Contradiction avec (1).

\Leftarrow) Supposons qu'il existe $\bar{x} \in \bigcup_{i=1}^t B^{j_i}$ tel que $\bar{x} \notin \bigcup_{i=1}^t A^{j_i}$.

1. D'une part, $\bar{x} \in \bigcup_{i=1}^t B^{j_i}$ signifie qu'il existe un $s \in \{1, \dots, t\}$ tel que $\bar{x} \in B^{j_s}$ et par conséquent, $\bar{x}_{j_s} \geq 1$ et $\sum_{p=1}^{s-1} \bar{x}_{j_p} < 1 \dots (2)$.

2. D'autre part, $\bar{x} \notin \bigcup_{i=1}^t A^{j_i}$ signifie que $\forall i \in \{1, \dots, t\}$, $\bar{x} \notin A^{j_i}$, en particulier, pour $i = s$, ce qui donne $\bar{x}_{j_s} < 1$. Contradiction avec (2).

Conclusion : $\bigcup_{i=1}^t A^{j_i} = \bigcup_{i=1}^t B^{j_i}$.

Démontrons maintenant que $\forall j_i, j_s \in H_l$, $i \neq s$, on a $B_{j_i} \cap B_{j_s} = \emptyset$.

Supposons qu'ils existent $j_i, j_s \in H_l$, $i \neq s$, tels que $B_{j_i} \cap B_{j_s} \neq \emptyset$.

Soit $x \in B_{j_i} \cap B_{j_s}$, ce qui signifie que

$$x \in S_l \text{ et } \begin{cases} x_{j_i} \geq 1, \sum_{p=1}^{i-1} x_{j_p} < 1 \\ x_{j_s} \geq 1, \sum_{h=1}^{s-1} x_{j_h} < 1 \end{cases}$$

Comme $x \in S_l$, alors x possède des composantes entières, il s'ensuit que

$$\begin{cases} x_{j_p} = 0 \quad \forall p = \overline{1, (i-1)} \\ x_{j_h} = 0 \quad \forall h = \overline{1, (s-1)} \end{cases}$$

1^{er} cas :

Si $i > s$, alors pour $p = s$, on a, $x_{j_s} = 0$. Or $x_{j_s} \geq 1$, contradiction.

2^{ème} cas :

Si $i < s$, alors pour $h = i$, on a, $x_{j_i} = 0$. Or $x_{j_i} \geq 1$, contradiction.

Donc, $\forall j_i, j_s \in H_l$, $i \neq s$, on a $B_{j_i} \cap B_{j_s} = \emptyset$. □

Proposition 3.7 ([3]). *L'algorithme décrit plus haut converge en un nombre fini d'étapes.*

Démonstration. L'ensemble S des solutions réalisables de (P) étant fermé et borné, il contient un nombre fini de solutions entières. À chaque étape de la méthode, on détermine x^l , solution optimale du problème (P_l) lorsqu'elle existe. Cette solution est éliminée lorsqu'on étudie (P_r) sur l'ensemble S_r , $0 \leq l < r$, pour tous les indices $j_l \in H_l$, compte tenu de la proposition 3.4. □

3.2.3 Exemple numérique

Considérons le programme linéaire multiobjectif en nombres entiers (P) suivant :

$$(P) \begin{cases} \text{Max } z_1 = 2x_1 - 2x_2 + x_3 \\ \text{Max } z_2 = -3x_1 + 2x_2 + x_3 \\ (x_1, x_2, x_3) \in S \end{cases}$$

$$\text{Où } S = \left\{ (x_1, x_2, x_3) \in \mathbb{Z}^3 \mid \begin{array}{l} -2x_1 + 3x_2 - x_3 \leq 4, \quad 3x_1 + 3x_2 + x_3 \leq 6, \\ x_1 + x_2 + 3x_3 \leq 6, \quad x_i \geq 0, \quad i = \overline{1, 3} \end{array} \right\}.$$

Prenons $\lambda = (1, 1)^t$.

Soit (P_0) le problème : $\text{Max } \{z = -x_1 + 2x_3 \mid (x_1, x_2, x_3) \in S\}$.

$F \leftarrow \{P_0\}$ et $ESE(P) \leftarrow \emptyset$.

Étape (1)

On sélectionne le problème (P_0). La résolution de ce dernier donne le tableau optimal suivant :

B	x_1	x_2	x_6	x_B
x_4	$-\frac{5}{3}$	$\frac{10}{3}$	$\frac{1}{3}$	6
x_5	$\frac{8}{3}$	$\frac{8}{3}$	$-\frac{1}{3}$	4
x_3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	2
$-z$	$-\frac{5}{3}$	$-\frac{2}{3}$	$-\frac{2}{3}$	-4
$-z_1$	$\frac{5}{3}$	$-\frac{7}{3}$	$-\frac{1}{3}$	-2
$-z_2$	$-\frac{10}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	-2

$x^0 = (0, 0, 2)$, le vecteur critère correspondant est $(2, 2)$. $ESE(P) \leftarrow \{(0, 0, 2)\}$.

$N_0 = \{1, 2, 6\}$, $H_0 = \{1, 2\}$.

On génère deux sous-problèmes (P_1) et (P_2) en ajoutant respectivement au problème (P_0), les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 1 \\ x_2 \geq 1 \\ x_1 \leq 0 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_2 \geq 1 \\ x_1 \leq 0 \end{array} \right.$$

$F \leftarrow \{P_1, P_2\}$.

Étape (2)

On sélectionne le problème (P_1). La résolution de ce dernier donne le tableau optimal suivant :

B	x_7	x_8	x_9	x_B
x_4	-5	4	-3	7
x_6	0	-2	-1	2
x_3	0	1	0	1
x_1	-1	0	0	1
x_2	1	-1	1	0
x_5	0	2	-3	2
$-z$	-1	-2	0	-1
$-z_1$	4	-3	2	-3
$-z_2$	-5	1	-2	2

$x^1 = (1, 0, 1)$, le vecteur critère correspondant est $(3, -2)$.

$ESE(P) \leftarrow \{(0, 0, 2), (1, 0, 1)\}$, $N_1 = \{7, 8, 9\}$, $H_1 = \{7, 8, 9\}$.

On génère trois sous-problèmes (P_3), (P_4) et (P_5) en ajoutant respectivement au problème (P_1), les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_7 \geq 1 \\ x_7 \leq 0 \end{array} \right. , \quad \left\{ \begin{array}{l} x_8 \geq 1 \\ x_7 \leq 0 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_9 \geq 1 \\ x_7 + x_8 \leq 0 \end{array} \right.$$

$F \leftarrow \{P_2, P_3, P_4, P_5\}$.

Étape (3)

On sélectionne le problème (P_3). La résolution de ce dernier donne le tableau optimal suivant :

B	x_2	x_9	x_{10}	x_B
x_4	4	1	-1	8
x_6	-2	-3	-2	4
x_3	1	1	1	0
x_1	0	0	-1	2
x_8	-1	-1	-1	1
x_5	2	-1	2	0
x_7	0	0	-1	1
$-z$	-2	-2	-3	2
$-z_1$	-3	-1	1	-4
$-z_2$	1	-1	-4	6

$x^3 = (2, 0, 0)$, le vecteur critère correspondant est $(4, -6)$.

$ESE(P) \leftarrow \{(0, 0, 2), (1, 0, 1), (2, 0, 0)\}$, $N_3 = \{2, 9, 10\}$, $H_3 = \{2, 10\}$.

On génère deux sous-problèmes (P_6) et (P_7) en ajoutant respectivement au problème (P_3) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_2 \geq 1 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_{10} \geq 1 \\ x_2 \leq 0 \end{array} \right.$$

$F \leftarrow \{P_2, P_4, P_5, P_6, P_7\}$.

Étape (4)

On sélectionne le problème (P_6) . Ce dernier est irréalisable. $F \leftarrow \{P_2, P_4, P_5, P_7\}$.

Étape (5)

On sélectionne le problème (P_7) . Ce dernier est irréalisable. $F \leftarrow \{P_2, P_4, P_5\}$.

Étape (6)

On sélectionne le problème (P_4) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_7	x_9	x_{10}	x_B
x_4	-5	-3	4	3
x_6	0	-1	-2	4
x_3	0	0	1	0
x_1	-1	0	0	1
x_2	1	1	-1	1
x_5	0	-3	2	0
x_8	0	0	-1	1
x_{11}	1	0	0	0
$-z$	-1	0	-2	1
$-z_1$	4	2	-3	0
$-z_2$	-5	-2	1	1

$x^4 = (1, 1, 0)$, le vecteur critère correspondant est $(0, -1)$ qui est dominé par celui de x^0 .

$N_4 = \{7, 9, 10\}$, $H_4 = \{7, 9, 10\}$.

On génère trois sous-problèmes (P_8) , (P_9) et (P_{10}) en ajoutant respectivement au problème (P_4) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_7 \geq 1 \end{array} \right. , \quad \left\{ \begin{array}{l} x_9 \geq 1 \\ x_7 \leq 0 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_{10} \geq 1 \\ x_7 + x_9 \leq 0 \end{array} \right.$$

$$F \leftarrow \{P_2, P_5, P_8, P_9, P_{10}\}.$$

Étape (7)

On sélectionne le problème (P_8) . Ce dernier est irréalisable. $F \leftarrow \{P_2, P_5, P_9, P_{10}\}$.

Étape (8)

On sélectionne le problème (P_9) . La résolution de ce dernier donne le tableau optimal suivant :

B	x_7	x_{10}	x_{12}	x_B
x_4	-5	4	-3	6
x_6	0	-2	-1	5
x_3	0	1	0	0
x_1	-1	0	0	1
x_2	1	-1	1	0
x_5	0	2	-3	3
x_8	0	-1	0	1
x_{11}	1	0	0	0
x_9	0	0	-1	1
x_{13}	1	0	0	0
$-z$	-1	-2	0	1
$-z_1$	4	-3	2	-2
$-z_2$	-5	1	-2	3

$x^9 = (1, 0, 0)$, le vecteur critère correspondant est $(2, -3)$ qui est dominé par celui de x^0 et de x^1 .

$$N_9 = \{7, 10, 12\}, H_9 = \{7, 10, 12\}.$$

On génère trois sous-problèmes (P_{11}) , (P_{12}) et (P_{13}) en ajoutant respectivement au problème (P_9) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_7 \geq 1 \end{array} \right. , \quad \left\{ \begin{array}{l} x_{10} \geq 1 \\ x_7 \leq 0 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_{12} \geq 1 \\ x_7 + x_{10} \leq 0 \end{array} \right.$$

$$F \leftarrow \{P_2, P_5, P_{10}, P_{11}, P_{12}, P_{13}\}.$$

Étape (9)

On sélectionne le problème (P_{11}). Ce dernier est irréalisable.

$$F \leftarrow \{P_2, P_5, P_{10}, P_{12}, P_{13}\}.$$

Étape (10)

On sélectionne le problème (P_{12}). Ce dernier est irréalisable. $F \leftarrow \{P_2, P_5, P_{10}, P_{13}\}$.

Étape (11)

On sélectionne le problème (P_{13}). Ce dernier est irréalisable. $F \leftarrow \{P_2, P_5, P_{10}\}$.

Étape (12)

On sélectionne le problème (P_{10}). Ce dernier est irréalisable. $F \leftarrow \{P_2, P_5\}$.

Étape (13)

On sélectionne le problème (P_5). Ce dernier est irréalisable. $F \leftarrow \{P_2\}$.

Étape (14)

On sélectionne le problème (P_2). La résolution de ce dernier donne le tableau optimal suivant :

B	x_1	x_7	x_9	x_B
x_4	-2	3	1	2
x_5	3	3	-1	2
x_3	0	0	1	1
x_2	0	-1	0	1
x_8	1	0	0	0
x_6	1	1	-3	2
$-z$	-1	0	-2	-2
$-z_1$	2	-2	-1	1
$-z_2$	-3	2	-1	-3

$x^2 = (0, 1, 1)$, le vecteur critère correspondant est $(-1, 3)$.

$ESE(P) \leftarrow \{(0, 0, 2), (1, 0, 1), (2, 0, 0), (0, 1, 1)\}$, $N_2 = \{1, 7, 9\}$, $H_2 = \{1, 7\}$.

On génère deux sous-problèmes (P_{14}) et (P_{15}) en ajoutant respectivement au problème (P_2) , les systèmes de contraintes suivants :

$$\left\{ \begin{array}{l} x_1 \geq 1 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_7 \geq 1 \\ x_1 \leq 0 \end{array} \right.$$

$F \leftarrow \{P_{14}, P_{15}\}$.

Étape (15)

On sélectionne le problème (P_{14}) . Ce dernier est irréalisable. $F \leftarrow \{P_{15}\}$.

Étape (16)

On sélectionne le problème (P_{15}) . Ce dernier est irréalisable. $F \leftarrow \emptyset$. Terminer.

Finalement, les solutions efficaces du problème (P) sont : $(0, 0, 2)$, $(1, 0, 1)$, $(2, 0, 0)$ et $(0, 1, 1)$.

3.3 Implémentation et résultats

En complément de notre étude théorique, nous avons programmé les algorithmes présentés dans ce chapitre ainsi que l'algorithme décrivant la méthode de Sylva et Crema [17] en utilisant le logiciel Matlab version 7 et effectué des expérimentations sur des instances du problème multiobjectif suivant :

$$\text{“Max” } \{Cx \mid Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$$

où A , C et b sont des matrices de taille respective $m \times n$, $k \times n$ et $m \times 1$.

Ces instances ont été construites grâce au générateur aléatoire décrit ci-dessous :

Algorithme 3 Génération aléatoire d'instances pour un problème multiobjectif

Entrées : n : nombre de variables k : nombre de critères m : nombre de contraintes**Sorties :** les matrices A , C et b formant le problème multiobjectif $A \leftarrow \mathbf{randint}(m, n, [10, 30])$ $C \leftarrow \mathbf{randint}(k, n, [-30, 30])$ **Pour** $i = 1$ jusqu'à m **faire**

$$b_i \leftarrow \sum_{j=1}^n A_i^j + \mathbf{randint}(1, 1, [100, 300])$$

Fin pour

$\mathbf{randint}(m, n, [p_1, p_2])$ est une fonction prédéfinie de Matlab qui renvoie une matrice à m lignes et n colonnes dont les éléments sont des entiers générés aléatoirement entre p_1 et p_2 .

Les poids relatifs aux critères sont tous pris égaux à 1.

Notons que la machine sur laquelle nous avons fait tourner les tests est dotée d'un processeur Intel Pentium 4 cadencé à 3.6 GHz avec 1 Go de mémoire vive.

Comparaison entre la méthode décrite en [13] et la version améliorée

Dans les trois tableaux qui suivent, nous illustrons pour 45 instances de petite taille, l'apport de l'amélioration proposée pour la méthode de Abbas et Moumene [13] en termes de gain en nombre de sous-problèmes créés. Le temps de calcul des deux méthodes n'est pas présenté, étant donné que la différence est négligeable.

Problèmes à 1 contrainte (sac à dos)

Instances		Nombre de sous problèmes créés			Nombre de solutions efficaces
k	n	Méthode [13]	Avec amélioration	Différence	
2	3	220	149	71	110
	4	364	264	100	182
	5	270	162	108	135
	6	108	87	21	54
	7	120	80	40	60
	8	228	141	87	114
3	3	348	297	51	116
	4	177	108	69	59
	5	114	66	48	38
	6	105	75	30	35
	7	204	106	98	68
	8	42	33	9	14
4	4	84	47	37	21
	5	1004	515	489	251
	6	420	208	212	105

Problèmes à 2 contraintes

Instances		Nombre de sous problèmes créés			Nombre de solutions efficaces
k	n	Méthode [13]	Avec amélioration	Différence	
2	3	90	61	29	45
	4	84	64	20	42
	5	236	153	83	118
	6	230	161	69	115
	7	82	65	17	41
	8	254	166	88	127
3	3	132	76	56	44
	4	150	101	49	50
	5	81	40	41	27
	6	75	44	31	25
	7	42	19	23	14
	8	42	30	12	14
4	4	216	115	101	54
	5	1676	1124	552	419
	6	188	143	45	47
	7	80	36	44	20

Problèmes à 3 contraintes

Instances		Nombre de sous problèmes créés			Nombre de solutions efficaces
k	n	Méthode [13]	Avec amélioration	Différence	
2	3	114	70	44	57
	4	162	96	66	81
	5	18	9	9	9
	6	62	50	12	31
	7	76	60	16	38
	8	48	37	11	24
3	3	78	37	41	26
	4	231	142	89	77
	5	30	11	19	10
	6	39	23	16	13
	7	42	29	13	14
	8	60	25	35	20
4	4	60	39	21	15
	5	96	53	43	24

Au regard de ces résultats, il apparaît que plus le nombre de solutions efficaces du problème multiobjectif considéré est grand, plus le gain en nombre de sous-problèmes créés est considérable.

Comparaison entre la méthode décrite en [3] et la version améliorée

Etant donné que la méthode de Chergui *et al.* [3] est très coûteuse en temps de calcul, nous n'avons fait des tests que sur 12 instances de petite taille dont les résultats sont donnés dans le tableau suivant :

Instances			Temps de calcul (sec)		Nombre de solutions générées		Nombre de solutions efficaces
m	n	k	Méthode [3]	Avec amélioration	Méthode [3]	Avec amélioration	
2	3	2	5.797	0.109	3002	34	28
2	4	2	5.156	0.203	2306	69	13
2	5	2	2.235	0.172	1023	55	18
2	3	3	15.453	0.234	6605	85	32
2	4	3	32.031	0.125	16795	45	9
2	5	3	20.938	0.281	10527	98	94
3	3	2	3.703	0.141	1932	56	18
3	4	2	71.593	0.125	28998	55	19
3	5	2	4.422	0.157	2015	60	11
3	3	3	2.25	0.156	977	52	5
3	4	3	14.344	0.25	6404	90	18
3	5	3	7.219	0.203	3300	70	10

Ces résultats montrent clairement que le nombre de solutions entières générées par la méthode améliorée ainsi que le temps de calcul sont nettement inférieurs à ceux donnés par la méthode décrite en [3].

Comparaison entre les deux méthodes améliorées

Pour cette série d'expérimentations, chaque instance étant résolue dix fois par chaque algorithme, les valeurs du tableau suivant représentent la moyenne sur les dix exécutions du temps de calcul, du nombre de coupes de Gomory utilisées, du nombre de solutions entières générées, du nombre de solutions réalisables et du nombre de solutions efficaces.

Instances			Méthode améliorée de Abbas et Moumene			Méthode améliorée de Chergui <i>et al.</i>			Nombre de solutions efficaces	Nombre de solutions réalisables
n	m	k	Temps de calcul (sec)	Nombre de coupes	Nombre de solutions générées	Temps de calcul (sec)	Nombre de coupes	Nombre de solutions générées		
3	2	2	0.2158	56.5	20.2	0.4015	3.3	148.2	20.2	425.8
4	2	2	3.35	138.7	29.9	0.9719	28.1	302.9	29.9	2939.5
5	2	2	0.8329	77.6	21.1	1.8032	33.8	474.5	21.1	6338.5
6	2	2	2.781	167.9	37.3	7.6751	84.4	1640.3	37.3	70640.4
3	2	3	13.4856	1483.9	137.6	1.4452	8.7	332.8	116.5	425.8
4	2	3	7.8	484.8	61.6	2.4423	67.4	496.2	61.6	2939.5
5	2	3	2.7187	532.8	50.8	9.0327	219.2	1991.7	49.2	6338.5
3	3	2	2.3518	145.4	16.7	0.2701	20.8	100.2	16.7	336.6
4	3	2	1.536	89.1	15.3	0.7122	52.3	192.2	15.3	1218.8
5	3	2	0.7296	103.7	26	1.7468	64.4	335.5	26	8039.2
6	3	2	1.1922	115.8	24.9	27.3095	596.5	2847.2	24.9	34742
3	3	3	4.3626	730	58.4	0.7344	7	250.6	43.6	336.6
4	3	3	2.2078	393.2	51.4	33.4546	212.5	946	49.7	1218.8
5	3	3	4.6751	336.5	46.8	3.9187	118.2	755.2	46.6	8039.2

Tout d'abord, aucune des deux méthodes considérées dans cette section n'est clairement plus performante que l'autre en termes de temps de calcul. Par contre, en comparant le nombre de solutions entières générées, on constate que les résultats sont en faveur de la méthode améliorée de Abbas et Moumene sur toutes les instances. Il est même égal au nombre de solutions efficaces pour neuf d'entre elles, ce qui n'est pas le cas pour la méthode améliorée de Chergui *et al.*. Néanmoins, cette dernière évite l'exploration d'une bonne partie de l'ensemble des solutions réalisables qui ne contient que des solutions inefficaces. Notons en dernier, que la performance de ces méthodes dépend fortement de celle de la méthode duale fractionnaire de Gomory qui d'un point de vue opérationnel, donne des résultats décevants car elle nécessite en général un grand nombre d'itérations.

Comparaison entre la méthode améliorée de Abbas et Moumene et celle de Sylva et Crema

Le tableau suivant donne pour chaque instance et pour chaque algorithme, la moyenne sur cinq exécutions indépendantes, du temps de calcul et du nombre de solutions efficaces obtenues.

Instances			Temps de calcul (sec)		Nombre de solutions efficaces
n	m	k	Méthode améliorée de Abbas et Moumene	Méthode de Sylva et Crema	
3	2	2	0.2876	14.5594	27.6
4	2	2	4.9498	33.9218	28.8
5	2	2	4.2408	94.8032	33.8
6	2	2	3.509	69.044	31
3	2	3	2.6654	1229.2904	44.6
4	2	3	11.1624	921.9716	44.2
3	3	2	3.95	4.3594	18.6
4	3	2	2.8594	40.7282	22.6
5	3	2	1.8156	1524.7752	44.6
6	3	2	1.6	120.1378	28.6
3	3	3	0.556	259.7064	31.6
4	3	3	1.472	1363.684	37.6

Les résultats obtenus par cette dernière série d'expérimentations montrent que le temps de calcul donné par la méthode améliorée de Abbas et Moumene est inférieur à celui donné par la méthode de Sylva et Crema sur toutes les instances traitées. Cette différence s'accroît pour les instances incluant un nombre important de solutions efficaces. Nous pouvons donc conclure que, contrairement à la méthode améliorée de Abbas et Moumene, la performance de la méthode de Sylva et Crema est intimement liée au nombre de solutions efficaces du problème multiobjectif considéré, ce qui constitue une limite pour la méthode.

Le tableau ci-dessous résume quelques avantages et inconvénients des méthodes étudiées dans ce chapitre :

	Avantages	Inconvénients
Méthode améliorée de Abbas et Moumene	<ul style="list-style-type: none"> – Elle génère un nombre très réduit de solutions inefficaces. 	<ul style="list-style-type: none"> – La résolution d'un sous-problème (P_r) ne donne pas nécessairement une solution efficace. – A le défaut des méthodes basées sur les coupes de Gomory pour la résolution des programmes linéaires en nombres entiers. – Détermine exactement une solution efficace pour chaque point non dominé.
Méthode améliorée de Chergui <i>et al.</i>	<ul style="list-style-type: none"> – Toutes les solutions efficaces ayant le même vecteur critère sont déterminées (si elles existent). – Utilisation modérée des coupes de Gomory compte tenu de la nature des contraintes utilisées $x_{j_l} \geq 1, j_l \in H_l$. – En général, elle évite l'exploration d'une bonne partie de l'ensemble des solutions réalisables qui ne contient que des solutions inefficaces. 	<ul style="list-style-type: none"> – Si à chaque étape, $H_l = N_l$, alors la méthode doit passer en revue toutes les solutions réalisables entières.
Méthode de Sylva et Crema	<ul style="list-style-type: none"> – La résolution de chaque problème (P_r) engendre forcément une solution efficace. – Elle peut générer des sous-ensembles de solutions efficaces. 	<ul style="list-style-type: none"> – Le nombre de programmes linéaires en nombres entiers à résoudre est égal au nombre de solutions non dominées de l'instance traitée plus un. De plus, la taille de ces problèmes augmente à chaque étape de k variables et $k + 1$ contraintes (k étant le nombre de critères). – Détermine exactement une solution efficace pour chaque point non dominé.

Application : Réalisation d'un plan optimal d'insémination artificielle

Dans ce dernier chapitre, nous présentons une application concrète pouvant se modéliser par la programmation linéaire multiobjectif discrète. Il s'agit de réaliser un plan d'insémination artificielle pour une ferme d'élevage laitier satisfaisant au mieux les objectifs tracés par l'éleveur. Plusieurs études ont été menées dans ce sens incluant des modèles d'optimisation linéaire [5], des modèles d'optimisation linéaire discrète [11] et des modèles d'optimisation linéaire multiobjectif discrète [18].

Sommaire

4.1	Position du problème	65
4.2	Modélisation mathématique	65
4.3	Résolution du problème et résultats	68
4.3.1	Jeux de données	68
4.3.2	Analyse des résultats et commentaires	69

4.1 Position du problème

L'insémination artificielle consiste à déposer une dose de semence dans les voies génitales de la femelle pour obtenir une fécondation. Cette technique s'est aujourd'hui imposée en élevage bovin dans le monde entier ; ceci se justifie par le fait qu'elle met à la disposition des éleveurs, des semences sélectionnées génétiquement afin d'apporter un plus à la nouvelle génération au niveau du critère de sélection choisi.

Les semences des taureaux améliorateurs sont conservées dans des paillettes et congelées dans l'azote liquide. Un catalogue répertorie les caractéristiques moyennes des descendantes femelles de l'animal. Nous citons entre autres, les caractéristiques liées à la production (index de production) telles que la quantité de lait, de matière grasse et de matière protéique.

Considérant le nombre important de taureaux améliorateurs disponibles ainsi que la complexité d'informations données par taureau, l'éleveur ayant recours à l'insémination artificielle se trouve souvent perplexe devant le choix d'un taureau inséminateur pour combler les faiblesses de son troupeau. Dans une telle situation, un problème d'optimisation peut bien sûr apparaître.

Dans le paragraphe qui suit, nous décrivons un modèle mathématique permettant de dresser un plan d'insémination artificielle pour une ferme d'élevage laitier¹ tout en prenant comme objectif l'amélioration du lait des futures vaches en quantité et en qualité.

4.2 Modélisation mathématique

Notations

Le modèle qui sera présenté fera appel aux notations suivantes :

N : nombre de paillettes nécessaires pour l'insémination artificielle et est égale au nombre de vaches à inséminer ;

n : nombre de taureaux améliorateurs dont la semence est disponible ;

¹Elevage dont la production principale est le lait.

$disp_j$: nombre disponible de paillettes de semence du taureau j ;

$Lait_j$: amélioration en quantité de lait observée sur les filles du taureau j durant une lactation (10 mois) exprimée en kg ;

MG_j (resp. MP_j) : amélioration en matière grasse (resp. en matière protéique) observée dans le lait des filles du taureau j en kg ;

CD_j : coefficient mesurant la précision des index de production relatifs au taureau j ($Lait_j, MG_j, MP_j$), plus il est proche de 100, plus la valeur de l'index est sûre.

La solution cherchée sera par ailleurs décrite au moyen des variables entières x_j , $j = \overline{1, n}$, où :

x_j : nombre de paillettes de semence du taureau j utilisées.

Modèle mathématique

Le problème de réalisation d'un plan optimal d'insémination artificielle peut se modéliser comme un programme linéaire multiobjectif en nombres entiers. En utilisant les notations décrites ci-dessus, voici la formulation mathématique proposée :

$$\text{Maximiser } \sum_{j=1}^n Lait_j x_j \quad (4.1)$$

$$\text{Maximiser } \sum_{j=1}^n MG_j x_j \quad (4.2)$$

$$\text{Maximiser } \sum_{j=1}^n MP_j x_j \quad (4.3)$$

$$\text{Maximiser } \sum_{j=1}^n CD_j x_j \quad (4.4)$$

sous les contraintes :

$$\sum_{j=1}^n x_j = N \quad (4.5)$$

$$\forall j = \overline{1, n}, \quad x_j \leq disp_j \quad (4.6)$$

$$\forall j = \overline{1, n}, \quad x_j \leq \frac{\alpha}{100} \times N \quad (4.7)$$

$$\forall j = \overline{1, n}, \quad x_j \geq 0, \quad x_j \text{ entier}$$

Pour que le plan d'insémination artificielle soit rentable pour l'éleveur, nous devons favoriser l'utilisation des semences des taureaux faisant le plus améliorer la quantité et la qualité de lait de la prochaine génération, ce qui est exprimé par les fonctions objectifs (4.1), (4.2) et (4.3).

Bien que certains taureaux présentent des index de production élevés, il n'en est pas de même pour leur exactitude. Un critère important à prendre en considération est en effet, la précision des index relatifs à chaque taureau utilisé qu'on cherchera à maximiser, ce qui est donné par la fonction (4.4).

La contrainte (4.5) impose l'achat du nombre de paillettes nécessaires pour l'insémination. Le stock de paillettes de chaque taureau étant limité, il est indispensable de s'assurer que le nombre réservé pour l'insémination soit disponible, ce qui est garanti par la contrainte (4.6). Enfin, la contrainte (4.7) assure qu'au plus $\alpha\%$ ($0 < \alpha < 100$) de la totalité des vaches peuvent être inséminées par le même taureau. Ceci réduit le risque de consanguinité et favorise la diversification dans le troupeau.

Remarques

- En Algérie, le coût des paillettes est le même pour tous les taureaux, c'est pourquoi ce paramètre n'a pas été pris en considération dans cette étude.
- Pour un éleveur, c'est la teneur en protéines et en matières grasses qui fait la qualité du lait de ses vaches. Étant donné que le lait produit localement dans notre pays n'en contient qu'en faibles quantités, nous avons considéré l'augmentation de la qualité de lait l'un des objectifs du problème.
- Puisque toutes les variables x_j , $j = \overline{1, n}$, sont entières, le terme $\frac{\alpha}{100} \times N$ dans (4.7) peut être remplacé par $\lfloor \frac{\alpha}{100} \times N \rfloor$.

Hypothèse

Pour que le modèle ci-dessus admette au moins une solution, on suppose que :

$$\sum_{j=1}^n \min(\text{disp}_j, \lfloor \frac{\alpha}{100} \times N \rfloor) \geq N$$

4.3 Résolution du problème et résultats**4.3.1 Jeux de données**

Pour valider le modèle décrit précédemment, on se propose de réaliser un plan optimal d'insémination artificielle pour une ferme d'élevage laitier dans la commune de Mazouna, wilaya de Relizane, possédant 38 vaches de race Holstein à inséminer. Pour réaliser ce plan, nous allons considérer 8 taureaux améliorateurs de la même race dont la disponibilité de semence, les index de production et le coefficient de précision de ces derniers sont résumés dans le tableau suivant :

j	Nom du taureau j	disp_j	$Lait_j(kg)$	$MG_j(kg)$	$MP_j(kg)$	CD_j (%)
1	Jubibelt	20	736	47	22	76
2	Heros	10	1405	35	37	82
3	Melos	5	562	14	24	75
4	Jaboto	9	1459	75	45	72
5	Pander	13	1388	58	35	86
6	Allwis	8	1008	49	29	88
7	Luminor	15	1880	42	46	74
8	Ravens	5	1485	30	37	83

Excepté la disponibilité de semence, les données présentées dans le tableau ci-dessus sont reprises depuis le catalogue des taureaux du C.N.I.A.G (Centre National d'Insémination et d'Amélioration Génétique) d'Alger.

En prenant $\alpha = 20$, le modèle d'optimisation correspondant est le suivant :

$$(P) \left\{ \begin{array}{l} \text{Max } 736x_1 + 1405x_2 + 562x_3 + 1459x_4 + 1388x_5 + 1008x_6 + 1880x_7 + 1485x_8 \\ \text{Max } 47x_1 + 35x_2 + 14x_3 + 75x_4 + 58x_5 + 49x_6 + 42x_7 + 30x_8 \\ \text{Max } 22x_1 + 37x_2 + 24x_3 + 45x_4 + 35x_5 + 29x_6 + 46x_7 + 37x_8 \\ \text{Max } 76x_1 + 82x_2 + 75x_3 + 72x_4 + 86x_5 + 88x_6 + 74x_7 + 83x_8 \\ \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 38 \\ \\ 0 \leq x_1 \leq 7, \text{ entier} \\ 0 \leq x_2 \leq 7, \text{ entier} \\ 0 \leq x_3 \leq 5, \text{ entier} \\ 0 \leq x_4 \leq 7, \text{ entier} \\ 0 \leq x_5 \leq 7, \text{ entier} \\ 0 \leq x_6 \leq 7, \text{ entier} \\ 0 \leq x_7 \leq 7, \text{ entier} \\ 0 \leq x_8 \leq 5, \text{ entier} \end{array} \right.$$

On a bien $\sum_{j=1}^n \min(\text{disp}_j, [0.2 \times 38]) = 52 > 38$, donc (P) admet au moins une solution.

4.3.2 Analyse des résultats et commentaires

Pour déterminer l'ensemble des solutions efficaces de (P), nous avons utilisé les deux méthodes améliorées présentées dans le chapitre précédent et 257 solutions efficaces ont été trouvées. La méthode de Sylva et Crema [17] s'est montrée très coûteuse en temps et en espace mémoire, ce qui a entraîné l'arrêt de l'exécution après 24 heures de calcul. Le cardinal de l'ensemble des solutions efficaces étant important, le choix d'une solution parmi celles-ci qui satisfait l'éleveur n'est plus une tâche facile. Pour palier à ce problème, nous fixons un poids pour chaque critère suivant les préférences de l'éleveur afin de construire une combinaison à coefficients positifs des critères considérés et résolvons le programme linéaire en nombres entiers obtenu. En effet, après discussion avec les spécialistes du domaine, les valeurs suivantes des

poinds ont été retenues :

$\lambda_1 = 0.35$, $\lambda_2 = 0.2$, $\lambda_3 = 0.1$ et $\lambda_4 = 0.35$.

La solution de meilleur compromis $(0, 7, 0, 7, 7, 5, 7, 5)$ est trouvée en utilisant l'une quelconque des méthodes citées plus haut où la matrice des critères est réduite au seul critère d'agrégation. Cette solution correspond à l'achat de 7 paillettes de semence de Heros, 7 de Jaboto, 7 de Pander, 5 de Allwis, 7 de Luminor et 5 de Ravens.

Les résultats obtenus à travers cette étude, montrent encore une fois que le nombre de solutions efficaces peut être exponentiel même pour un problème de petite taille. L'énumération complète de ces dernières n'a plus d'intérêt pratique et il est plus avantageux de se tourner vers des méthodes interactives où le décideur intervient dans la construction d'une solution de bon compromis.

Conclusion et perspectives

De nombreux problèmes rencontrés dans la pratique, dans divers secteurs liés à l'industrie, l'économie, etc. nécessitent la prise en charge de plusieurs objectifs qui, très souvent, sont conflictuels. Pour ce type de problèmes, il ne s'agit plus de rechercher une solution optimale, mais un ensemble de solutions Pareto optimales qui se distinguent par les différents compromis réalisés entre les différentes fonctions objectifs considérées.

Dans les trente dernières années, la plupart des travaux réalisés dans ce domaine ont porté sur la programmation linéaire multiobjectif en variables continues. Les raisons principales de cet intérêt sont d'une part le développement de la programmation linéaire monoobjectif en recherche opérationnelle et la facilité relative de traiter de tels problèmes, et d'autre part l'abondance de cas pratiques pouvant être formulés sous cette forme. Dans cette étude, nous nous sommes particulièrement intéressés aux problèmes de programmation linéaire multiobjectif en nombres entiers pour lesquels relativement peu de travaux ont été réalisés.

Nous avons commencé ce travail par introduire les notions de base concernant l'optimisation multiobjectif, ce qui a fait l'objet du premier chapitre de ce mémoire.

Après une étude détaillée de quelques méthodes existantes d'optimisation multiobjectif discrète dans le deuxième chapitre, en l'occurrence, la méthode décrite

par Klein et Hannan [9], Gupta et Malhotra [8], Sylva et Crema [17], Abbas et Moumene [13] et Chergui *et al.* [3], nous avons proposé des améliorations pour les deux dernières. Ces améliorations ont été exposées, illustrées sur des exemples et approuvées par une étude expérimentale dans le chapitre trois. À cet effet, nous avons programmé en utilisant le logiciel Matlab version 7, toutes les procédures nécessaires intervenant dans les méthodes citées plus haut, entre autres, les méthodes du simplexe et duale du simplexe avec adjonction de coupes de Gomory. Une étude comparative des méthodes étudiées a été également faite.

Pour finir, une modélisation multiobjectif pour un problème concret a été abordée. Le problème consiste à dresser un plan d'insémination artificielle pour une ferme d'élevage laitier tout en prenant comme objectif l'amélioration du lait des futures vaches en quantité et en qualité. Le modèle obtenu a été validé en considérant des données réelles d'une ferme de la commune de Mazouna, wilaya de Relizane, possédant 38 vaches de race Holstein à inséminer.

Nous avons constaté que toutes les méthodes utilisant les coupes de Gomory convergent très lentement, ce qui nous a contraint à traiter des instances de petite taille. Nous proposons donc, comme perspectives, de remplacer les coupes de Gomory par d'autres techniques de programmation linéaire en nombres entiers exactes ou approchées telles que Branch & Bound et Branch & cut ou des métaheuristiques ayant fait leur preuve dans le domaine de l'optimisation combinatoire multiobjectif telles que les algorithmes évolutionnaires.

Bibliographie

- [1] V.J. Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez & S. Zionts (eds), MCDM, Springer-Verlag, Berlin, pages 76-85, 1976.
- [2] D. Chaabane. Optimisation multicritère en nombres entiers. Thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, 2005.
- [3] M.E.A. Chergui, M. Moulaï et M. Abbas. Génération des solutions efficaces d'un programme linéaire multiobjectifs en nombres entiers. Actes du colloque COSI'05, Béjaïa, Algérie, pages 462-472, 2005.
- [4] G.B. Dantzig. Linear programming and extensions. Princeton University Press, Princeton, 1963.
- [5] D.T. Galligan and J.D. Ferguson. Application of linear programming in bull selection for a dairy herd. JAVMA 206, pages 173-176, 1995.
- [6] A.M. Geoffrion. Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications 22, pages 618-630, 1968.
- [7] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society 64, pages 275-278, 1958.
- [8] R. Gupta and R. Malhotra. Multi-criteria integer linear programming problem. Cahiers de CERO 34, pages 51-68, 1992.

- [9] D. Klein and E. Hannan. An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research* 9, pages 378-385, 1982.
- [10] C.E. Lemke. The dual method for solving the linear programming problem. *Naval Research Logistic Quarterly* 1, pages 36-47, 1954.
- [11] M.B. McConnel and D.T. Galligan. The use of integer programming to select bulls across breeding companies with volume price discounts. *J. Dairy Sci.* 87, pages 3542-3549, 2004.
- [12] M. Moulai. Optimisation multicritère fractionnaire linéaire en nombres entiers. Thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, 2002.
- [13] A. Moumene. Contribution algorithmique pour la résolution d'un programme linéaire multi-objectifs en nombres entiers. Mémoire de Magister, Université des Sciences et de la Technologie Houari Boumediene, Algérie, 2002.
- [14] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*. Wiley, New York, 1988.
- [15] M. Sakarovitch. *Optimisation combinatoire, programmation discrète volume 2*. Hermann, 1984.
- [16] R.E. Steuer. *Multiple criteria optimization : theory, computation and application*. John Wiley & Sons, New York, 1985.
- [17] J. Sylva and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research* 158, pages 46-55, 2004.
- [18] P.R. Tozer and J.R. Stokes. Using multiple objective programming in a dairy cow breeding program. *J. Dairy Sci.* 84, pages 2782-2788, 2001.

Contribution algorithmique en programmation multiobjectif discrète

Résumé

La plupart des problèmes réels rencontrés en pratique requièrent l'optimisation simultanée de plusieurs objectifs souvent conflictuels. La notion d'optimalité disparaît pour les problèmes de ce type au profit de la notion d'efficacité. Une solution efficace est une solution à partir de laquelle, il est impossible d'augmenter la valeur d'un critère sans diminuer celle d'au moins un autre. Dans ce mémoire, nous nous sommes intéressés à l'étude de quelques méthodes permettant de déterminer l'ensemble des solutions efficaces d'un problème de programmation linéaire multiobjectif en nombres entiers. Ce travail de synthèse nous a permis d'apporter des améliorations à deux d'entre elles, à savoir, la méthode de Abbas et Moumene et celle de Chergui *et al.* en réduisant de façon significative le temps d'exécution. Afin de mettre en valeur notre apport, des résultats expérimentaux issus de l'implémentation des deux méthodes améliorées sont présentés et commentés.

Mots clés : optimisation multiobjectif, programmation linéaire en nombres entiers, solution efficace.

An algorithmic contribution in discrete multiobjective programming

Abstract

Most real life problems encountered in practice require the simultaneous optimization of multiple, often conflicting, objectives. For such problems, the concept of optimality is replaced with that of efficiency. An efficient solution is a solution from which it is impossible to increase the value of one criterion without deteriorating that of at least one other. In this thesis, we were interested in the study of some methods that produce the efficient set of a multiple objective integer linear programming problem. This synthesis has enabled us to improve two of them, namely, Abbas and Moumene's method and Chergui *et al.*'s one by reducing significantly the execution time. To highlight our contribution, experimental results from the implementation of the two improved methods are presented and discussed.

Keywords : multiobjective optimization, integer linear programming, efficient solution.