$N°$ D'ORDRE : 27/2009-M/MT

Republique Algerienne Democratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediène

Faculté des Mathématiques

جامعة هواري بومدين للعلوم والتكنولوجيا
U S T H B

Mémoire présenté pour l'obtention du diplôme de MAGISTER

En : MATHÉMATIQUES

Spécialité : Recherche Opérationnelle : Mathématique de Gestion

Par : **MAHDI Sara**

THÈME

Optimisation Discrète Sur L'ensemble Des Solutions Efficaces D'un Problème Multi-Objectifs

Soutenu le 28/06/2009, devant le jury composé de :

M^r.	ABBAS Moncef	Professeur	U.S.T.H.B.	Président.
M^r.	CHAABANE Djamal	Maître de Conférences	U.S.T.H.B.	Directeur de thèse.
M^r.	BERRACHEDI Abdelhafidh	Professeur	U.S.T.H.B.	Examinateur.
M^{me}.	ZIARI Yasmina	Maître de Conférences	U.S.T.H.B.	Examinatrice.

Remerciments

D'abord, je remercie le Bon Dieu qui ma donné la force et le courage pour terminer ce travail.

Je tiens à exprimer mes vifs remerciements à monsieur CHAABANE Djamal, mon directeur de mémoire, Maître de conférences à l'USTHB, de m'avoir encadré, pour l'intéressant sujet qu'il ma proposé, qu'il trouve ici mes vifs et profonds respects, considération, gratitude ainsi reconnaissances.

Mes vifs remerciements vont également à monsieur ABBAS Moncef, professeur à l'USTHB, qu'il ma fait l'honneur d'accepter de présider le jury de ce mémoire.

Je remercie ensuite, monsieur BERRACHEDI Abdelhafidh , professeur à l'USTHB, trouve ici mes grands remerciements et ma s'incère reconnaissance d'avoir accepté de juger ce travail en tant qu'examinateur.

Que Madame ZIARI Yasmina, Maître de conférences à l'USTHB, trouve ici mes vifs remerciements et reconnaissances d'avoir accepté d'examiner ce travail et être dans le jury de ce mémoire.

Je ne terminerai pas sans adresser mes grands remerciements à mes chers parents pour le soutien et l'encouragement qu'ils m'ont apportés, ainsi que mes sœurs et mes frères, et tous ceux qui m'ont de près ou de loin aidé, merci à tous.

Dédicaces

Je dédie ce modeste travail à mes chers parents,

à ma grand-mère,

à mes sœurs et mes frères,

à mes oncles et tantes.

à mes collègues, mes amies et toute ma famille.

Table des matières

Introduction générale

Depuis l'apparition de la programmation mathématique et la rechreche opérationnelle, la plupart des travaux d'optimisation étaient dédiés à l'optimisation d'une seule fonction objectif (problème mono-objectif ou uni-critère)[1], or les problèmes rencontrés dans la pratique sont la plupart du temps de nature multi-objectifs (multi-critères), il y'a généralement plusieurs objectifs conflictuels [2] à satisfaire simultanément (coût à minimiser, quantité à maximiser, temps de réalisation à minimiser, charge de travail à minimiser, ...), ce type de problèmes consiste à les modéliser de sort que tous les objectifs pertinents soient pris en considération.

Pendant longtemps, les méthodes de résolution des problèmes multi-objectifs consistaient principalement à les transformer en problèmes mono-objectif. Depuis quelques années, des travaux prenant en compte directement chacun des objectifs ont été réalisés en utilisant la notion d'optimalité de Pareto, définie initialement dans des travaux en économie au XIX $^{\text{ème}}$ siècle par V.Pareto[23]. Contrairement à l'optimisation mono-objectif la résolution d'un problème multi-objectifs ne conduit pas à une solution optimale, mais un ensemble de solutions offrant un bon compromis entre les différentes fonctions objectifs à optimiser dite les solutions *Pareto-optimales* ou *efficaces*, une solution efficace 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.

Dans la pratique, une classe de problèmes de programmation linéaire multi-objectifs MOLP (Multiple Objective Linear Programming) suscite une importante attention ; il s'agit des problèmes de programmation linéaire multi-objectifs en nombre entiers MOILP (Multiple Objective Integer Linear Programming), où les critères et les contraintes

[1]consiste à trouver une valeur optimale pour un seul objectif.

[2]deux objectifs sont conflictuels lorsque la diminution de l'un entraîne une augmentation de l'autre.

sont tous linéaires et les variables de décision ne peuvent prendre que des valeurs entières. L'intérêt de tels problèmes résulte du fait que dans la plupart d'applications concrètes de programmation mathématique la présence des variables discrètes est inévitable.

Dans certaines situations réelles, les décideurs n'ont pas besoin de l'ensemble des solutions efficaces d'un problème multi-objectifs mais uniquement celles qui réalisent l'optimum d'un objectif différent des objectifs déjà fixés. Ceci nous mène vers la recherche de la solution optimale d'un critère sur l'ensemble des solutions efficaces d'un problème multi-objectifs.

Dans ce mémoire nous avons fixé notre objectif à l'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème MOILP, ce type de problèmes est un cas particulier de l'optimisation non-convexe où la difficulté est principalemment due à la non-convexité du domaine d'admissibilité (l'ensembles de solutions efficaces). Dans ce contexte, nous avons mené une étude détaillée sur la programmation linéaire multi-objectifs en nombres entiers en focalisant notre intérêt sur l'objectif cité ci-dessus, nous avons concrétisé cet objectif par le développement d'une méthode simple évitant le passage systématique par tous les points efficaces.

Ce mémoire s'articule autour de quatre chapitres. Dans le premier chapitre, des notions de base sur l'optimisation multi-objectifs ainsi que quelques méthodes de résolution sont présentées.

Le deuxième chapitre est exposé sous forme de deux parties, la première partie concerne les principaux résultats de la programmation linéaire discrète, où les méthodes classiques de résolution comme la méthode de Gomory fractionnaire et la méthode de séparation et évaluation "branch & bound" sont rappelées. La deuxième partie est entièrement consacré à l'optimisation multi-objectifs en nombres entiers, nous rappelons d'une part la structure générale d'un problème MOILP et quelques résultats comme outils théoriques nécessaires à l'optimisation multi-objectifs discrète, et d'autre part, nous présentons quelques méthodes permettant de déterminer l'ensemble des solutions efficaces d'un tel problème, nous citons parmi autres, la méthode de Sylva & Crema [9] et la méthode de M.Abbas et D.Chaabane [1] avec une illustration numérique de cette dernière.

Dans le troisième chapitre, nous abordons notre contribution dans le domaine multi-

objectifs en mettant le point sur l'optimisation d'un critère linéaire appelé "*critère principal*" sur l'ensemble des points efficaces d'un problème MOILP tout en s'inspirant des algorithmes déjà développés dans la littérature citons parmi lesquels, la méthode de D.Chaabane [7] et celle de Jesús M. Jorge [19], nous proposons une nouvelles méthode de résolution qui représente une combinaison entre les principes utilisés dans ces deux méthodes, nous avons vu la nécessité d'examiner une étape importante dans ce dernier algorithme, il s'agit d'explorer une région lorsque la solution optimale trouvée dans une itération n'est pas efficace pour chercher une solution alternative efficace.
Le quatrième chapitre est consacré à une validation expérimentale de l'algorithme proposé, notre implémentation est testée sur des exemples de la littérature et des instances de problèmes générés aléatoirement à l'aide d'un programme que nous avons développé et implémenté sous l'environnement Matlab 7.0.
Enfin, nous terminons par une conclusion générale dans laquelle on résume les différentes contributions de ce mémoire et quelques perspectives de recherche.

Chapitre 1

La programmation linéaire multi-objectifs

1.1 Introduction

Dans ce chapitre nous présentons les définitions fondamentales et les principaux résultats ainsi que les approches de base de la programmation mathématique multi-objectifs.

1.2 La Programmation Mathématique Multi-objectifs

Un problème multi-objectifs où multi-critères est un problème de décision qui consiste à optimiser (maximiser où minimiser) simultanément p ($p \geq 2$,entier) fonctions réelles notées f_k $(k = 1, 2, ..., p)$ appeleés critères sur un ensemble d'actions noté S.
Un problème de programmation mathématique multi-critères peut être formuler par

$$(MO) \begin{cases} \text{“}opt\text{”} & (f_1, f_2, ..., f_p) \\ t.q & x \in S. \end{cases}$$

Où $S = \{x \in \mathbb{R}^n / g_j(x) \leq 0, j = 1, 2, ..., m\}$.
$f_k(k = 1, 2, ..., p)$ et $g_j(j = 1, 2, ..., m)$ sont des fonctions à valeurs réelles de vecteur de décision $x \in \mathbb{R}^n$.
Dans l'optimisation multi-objectifs il n'est pas possible de trouver dans l'ensemble des actions S, une action qui optimise simultanément les p critères, le problème multi-objectifs est donc mal posé même s'il est correctement formulé par rapport à la réalité concernée par les problèmes de décisions, pour lequel la notion de solution optimale n'a

pas de sens, on cherche alors une action X^* dite solution de meilleur compromis dont le vecteur $f(X^*) = [f_1(X^*), ..., f_p(X^*)]^t$ est bon où acceptable selon les préférences d'un décideur (DM :Decision Maker).

- Si les objectifs $f_k(k = 1, 2, ..., p)$ et les fonctions $g_j(j = 1, 2, ..., m)$ sont linéaires on obtient un problème de programmation linéaire multi-objectifs (Multiple Objective Linear Programming) défini par :

$$(MOLP) \begin{cases} \text{“}opt\text{”} & Z_k = c^k x; k = 1, .., p \\ t.q & x \in S. \end{cases}$$

Où $c^k \in \mathbb{R}^{1\times n}$ pour $k = 1, 2, ..., p$ et $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$
$A \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^n,\ b \in \mathbb{R}^m, m, n \in \mathbb{N}$, si de plus, l'ensemble de décisions est restreint aux entiers positifs, le problème devient un problème de programmation linéaire multi-objectifs en nombres entiers (Multiple Objective Integer Linear Programming) défini par :

$$(MOILP) \begin{cases} \text{“}opt\text{”} & Z_k = c^k x; k = 1, .., p \\ t.q & x \in D. \end{cases}$$

Où $D = S \cap \mathbb{Z}^n$, $\mathbb{Z}$ étant l'ensemble des nombres entiers relatifs.

Remarque. Dans la suite, toute optimisation se rapporte à une maximisation.
Le problème (MOLP) est écrit comme

$$(MOLP) \begin{cases} \text{“}\max\text{”} & Z = Cx \\ t.q & x \in S. \end{cases}$$

Où $Z = (Z_1, Z_2, ..., Z_p)$ et C une matrice de dimension $(p \times n)$ composée de vecteurs lignes $c^k \in \mathbb{R}^{1\times n}$.

1.2.1 Concepts de base

Définition 1.1.

• L'espace de décisions est l'espace $\mathbb{R}^n$ dans lequel se situe l'ensemble des actions S.

• L'espace des critères est l'espace $\mathbb{R}^p$ dans lequel se situe $\mathcal{Z}_S$ l'image de S dans $\mathbb{R}^p$ par l'application linéaire ψ qui associée à chaque vecteur de décision $x \in S$ son image, le vecteur critères $Z(x) = Cx$, dans l'espace des critères $\mathcal{Z}_S$

$$\begin{array}{ll} \psi : \mathbb{R}^n & \mapsto \mathbb{R}^p \\ x = (x_1, ..., x_n) \in S & \to Z(x) = (Z_1(x), ..., Z_p(x)) \in \mathcal{Z}_S. \end{array}$$

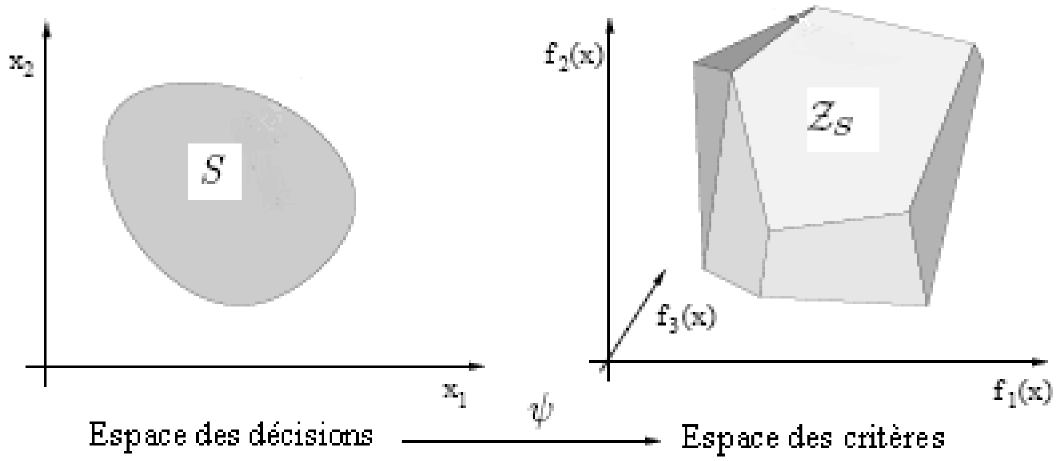


Fig. 1.1 – Problème multi-objectifs avec 2 variables de décision et 3 fonctions objectifs.

1.2.2 Relation de dominance

Toute solution admissible ($x \in S$) ne constitue pas évidemment un bon compromis à considérer, pour qu'elle le soit, on impose généralement une propriété basé sur une relation d'ordre partiel, notée " $>$ " et appelée *dominance* définie sur $\mathbb{R}^p$ par :
$\forall Z, Z' \in \mathcal{Z}_S, Z > Z'$, si et seulement si, $Z_k \geq Z'_k,\ \forall k \in \{1, ..., p\}$ et $Z_k > Z'_k$ pour au moins un k.

Définition 1.2.
• Soient $Z, Z' \in \mathcal{Z}_S$, on dit que Z *domine* Z', si et seulement si, $Z > Z'$.
C'est-à-dire Z est au moins aussi bon que Z' sur tous les critères et strictement meilleur que Z' sur au moins un critère.
• Un vecteur $Z \in \mathcal{Z}_S$ est dit *non dominé*, si et seulement si, il n'existe pas un autre vecteur $Z' \in \mathcal{Z}_S$ tel que $Z' > Z$.

Définition 1.3. $\forall Z, Z' \in \mathcal{Z}_S$
• On dit que Z *domine fortement* Z', si et seulement si, $Z_k > Z'_k,\ \forall k \in \{1, ..., p\}$. [1]

[1] Z est meilleur que Z' sur tous les critères.

• Un vecteur Z est dit *faiblement non dominé* s'il n'existe pas de $Z' \in \mathcal{Z}_S$ tel que Z' *domine fortement* Z.

1.2.3 L'efficacité où Pareto optimalité

Au $XIX^{\text{ème}}$ siècle, Vilfredo Pareto, un mathématicien italien, formule le concept suivant : dans un problème multi-objectifs, il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autre critères, cet équilibre a été appelé ***Optimum de Pareto***.

Définition 1.4.
Une solution $\tilde{x} \in S$ est dite *efficace* où optimale au sens de Pareto (où non inférieure), si seulement si, il n'existe pas de $x \in S$ telle que $Z_k(x) \geq Z_k(\tilde{x}), \forall k = 1, ..., p$ avec au moins une inégalité stricte, et le vecteur critères qui lui est associé, $Z(\tilde{x})$, est dit *solution non dominée.*

Définition 1.5.
• Une solution $\tilde{x} \in S$ est dite *faiblement efficace* s'il n'existe pas de $x \in S$ tel que $Z_k(x) > Z_k(\tilde{x}),\ \forall k = 1, ..., p$.
Autrement dit, une solution $\tilde{x} \in S$ est dite *faiblement efficace* si le vecteur critères qui lui est associé est *faiblement non dominé.*
• Une solution $\tilde{x} \in S$ est dite *fortement efficace* s'il n'existe pas de $x \in S$ tel que $x \neq \tilde{x}$ et $Z(x) \geq Z(\tilde{x})$.

1.2.4 La frontière de Pareto

Le graphe obtenu de l'image des solutions efficaces est appelé la frontière de Pareto (Pareto front).

1.2.5 Le point idéal

Le point idéal $\bar{Z}$ est le point de $\mathbb{R}^p$ défini par

$$\bar{Z} = (\max_{x \in S} Z_1(x), ..., \max_{x \in S} Z_p(x)).$$

1.2.6 Le point Anti-idéal

Le point Anti-idéal $\underline{Z}$ est le point de $\mathbb{R}^p$ défini par

$$\underline{Z} = (\min_{x \in S} Z_1(x), ..., \min_{x \in S} Z_p(x)).$$

Généralement $\bar{Z} et \underline{Z} \notin \mathcal{Z}_S$.

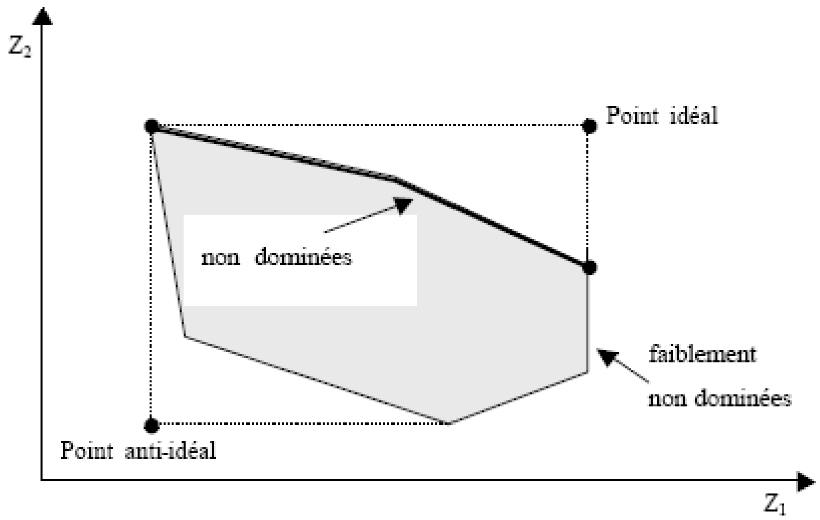


FIG. 1.2 – illustration des définitions

1.2.7 La matrice des gains

Soit $\tilde{x}_j$ une solution optimale obtenue en optimisant le critère Z_j. La matrice carrée de dimension p définie par

$$\begin{pmatrix} \bar{z}_1 & z_{12} & \cdots & z_{1p} \\ z_{21} & \bar{z}_2 & \cdots & z_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ z_{p1} & z_{p2} & \cdots & \bar{z}_p \end{pmatrix}$$

est dite matrice des gains "payoff matrix".

Où :

◇ $\bar{z}_i = Z_i(\tilde{x}_i) = \max_{x \in S} Z_i(x); \forall i = \overline{1,p}.$ [2]

◇ $z_{ij} = Z_i(\tilde{x}_j), \forall i, j \in \{1, ..., p\}$ avec $i \neq j$.

Remarque. La matrice des gains est univoquement déterminée si pour tout critère j, la solution optimale $\tilde{x}_j$ est unique. Sinon, lorsque un critère j possède plusieurs

[2]Les coordonées du point ideal.

solutions optimales, la colonne j de la matrice des gains dépondra de la solution $\tilde{x}_j$ choisie.

1.2.8 Le point nadir

Le point nadir η de $\mathbb{R}^p$ est le point de coordonées

$$\eta_k = \min_{j=1,...,p} z_{kj}, \;\; k = 1, ..., p.$$

où z_{kj} est un élément de la matrice des gains.

1.3 Paramètres de préférences

S'il apparaît naturel de limiter la considération des solutions à celles qui sont efficaces, l'ensemble de celle-ci est généralement très vaste et même infini dans le cas de variables continues. Aussi la sélection d'une solution efficace spécifique comme candidat de meilleur compromis exige une certaine connaissance de la structure de préférences du décideur. Cette information est obtenue directement ou indirectement et peut parfois se traduire en terme de paramètres de préférences, citons brièvement les paramètres qui sont fréquemment utilisés.

$\star$ *Le vecteur de poids* : $(\lambda_1, ..., \lambda_p)$ ou λ_k reflète l'importance relative de chaque critère k, $k = (1, ..., p)$.

Notons que ces paramètres peuvent également être des points cibles tels que :

$\star$ *Le point de référence* qui est défini par des niveaux d'aspiration (valeurs souhaitables) sur chaque critère.

$\star$ *Le point de réservation* qui est défini par des niveaux de réservation (valeurs non souhaitables) sur chaque critère.

1.4 Fonctions scalarisantes

Étant donné un ensemble de paramètres de préférence $\Lambda = \{(\lambda_1, ..., \lambda_p) \in \mathbb{R}^p\}$, il est possible de définir une fonction croissante dite *fonction scalarisante*, qui agrège les valeurs des critères pour chaque solution :

$$s(Z, \lambda) : \mathbb{R}^p \times \Lambda \to \mathbb{R}$$

Où $\lambda \in \Lambda \subset \mathbb{R}^p$ est le vecteur de paramètres choisi.

1.4.1 Caractérisation à l'aide de poids

• La somme pondérée : est largement utilisée dans l'optimisation linéaire multi-objectifs où le problème (MOLP) peut se ramener à un problème de programmtion paramétrique

$$s_1(Z,\lambda) = \sum_{k=1}^{p} \lambda_k Z_k, \text{ avec } \sum_{k=1}^{p} \lambda_k = 1, \lambda_k > 0, \forall k \in \{1,...,p\}.$$
$$s_2(Z,\lambda) = \sum_{k=1}^{p} \lambda_k |Z_k - \bar{z}_k|.$$

Où $\bar{z}_k$ est la $k^{\text{ème}}$ composante du point ideal.
$s_2(Z,\lambda)$ mesure la déviation qui sépare l'évaluation des propositions, qui sont généralement des points efficaces ou faiblement efficaces, du point d'aspiration. Cette déviation peut être mesurée par d'autres normes parmi lesquelles citons :
• Norme L_q pondérée :

$$s_3(Z,\lambda) = \left[\sum_{k=1}^{p} \lambda_k |Z_k - \bar{z}_k|^q\right]^{\frac{1}{q}}, q \in \mathbb{Z}_+^*.$$

• Norme L_∞ de Tchebycheff pondérée :

$$s_4(Z,\lambda) = \max_{1\le k\le p} \{\lambda_k |Z_k - \bar{z}_k|\}.$$

• Norme composée (Tchebycheff pondérée augmentée) :

$$s_5(Z,\lambda) = \max_{1\le k\le p} \{\lambda_k |Z_k - \bar{z}_k|\} + \rho \sum_{k=1}^{p} \lambda_k |Z_k - \bar{z}_k|; \rho > 0.$$

1.4.2 Caractérisation à l'aide de points cibles

• Niveaux d'aspiration : qui peut être obtenue en minimisant la fonction

$$s_6(Z,\lambda) = \left[\sum_{k=1}^{p} \lambda_k |Z_k - \hat{z}_k|^q\right]^{\frac{1}{q}}, q \in \mathbb{Z}_+^*; \hat{z}_k \in \mathcal{Z}_S.$$

Où $(\hat{z}_1, ..., \hat{z}_p)$ est un point cible dont on souhaite s'approcher autant que possible.

• Niveaux de réservation : qui peut être obtenue en maximisant la fonction

$$s_7(Z, \lambda) = \left[\sum_{k=1}^{p} \lambda_k |Z_k - z'_k|^q \right]^{\frac{1}{q}}$$

et des contraintes sur les critères $Z_k \geq z'_k$.

z'_k represente une valeur dont on souhaite s'écarter le plus possible en prenant des valeurs Z_k supérieures ou égale à z'_k.

1.5 Les trois approches fondamentales

Dans la programmation mathématique multi-objectifs on distingue trois principales approches :

1.5.1 Optimisation à priori (décideur → recherche)

Les solution les plus intuitives pour résoudre les problèmes multi-objectifs consistent souvent à combiner les différentes fonctions objectifs en une fonction d'utilité suivant les préférences de décideur. Dans ce cas le décideur est supposé connaître à priori le piods de chaque critère afin de les mélanger dans une fonction unique, cela revient à résoudre un problème uni-critère dont la fonction objectif est une fonction scalarisante :

$$\max_{Z \in \mathcal{Z}_S} s(Z, \lambda)$$

Cependant dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, soit par manque d'expérience ou d'informations.

1.5.2 Optimisation à posteriori (recherche → décideur)

Dans cette approche le décideur prend sa décision d'après un ensemble de solutions calculées par un solveur, dans ce cas la qualité de la décision dépend du choix de la méthode de résolution, celle-ci génère un sous ensemble de solutions efficaces, parmi lesquelles, le décideur peut sélectionner les compromis les plus satisfaisants pour ses préférences.

1.5.3 Optimisation progressive ou interactive (décideur $\leftrightarrow$ recherche)

Cette approche consiste en une alternance de deux étapes :

(a) l'étape de calcul exécutée par l'analyste ou le programme ;

(b) l'étape de dialogue avec le décideur.

La première étape de calcul fournit un ou plusieurs compromis. Ceux-ci sont présentés au décideur qui réagit en apportant des informations complementaires sur ses préférences, ces informations sont injectées sous forme des contraintes dans le programme utilisé et permetant de construire de nouveaux compromis. Le processus d'exploration se termine après un certain nombre d'itérations, soit déterminé par le décideur s'il est satisfait, soit par le programme grâce à une condition d'arrêt.

Utiliser cette approche peut être relativement efficace (en terme d'effort informatique), puisqu'on essaye seulement de produire un sous ensemble de l'ensemble complet de solutions potentielles régies par les préférences du décideur. Cependant, alors que de telles approches reflètent exactement les intentions du décideur, elles ne peuvent pas fonctionner indépendamment, le décideur doit être présent lors l'exécution de l'algorithme.

1.6 Quelques résultats de base

Considérons le problème (MOLP) :

$$(MOLP)\begin{cases} \text{“max”} & Z_k = c^k x;\ k = 1,..,p \\ t.q & x \in S. \end{cases}$$

Soit Λ l'ensemble de tous les vecteurs $\lambda = (\lambda_i), i = 1, ..., p$ défini par :

$$\Lambda = \{\lambda \in \mathbb{R}^p | \sum_{i=1}^{p} \lambda_i = 1, \lambda_i \geq 0, i = 1, ..., p\}$$

Pour $\lambda \in \Lambda$, on définit le problème paramétrique (P_λ) par :

$$(P_\lambda)\begin{cases} \text{“max”} & \sum_{i=1}^{p} \lambda_i Z_i(x) \\ t.q & Z = (Z_1(x), ..., Z_p(x)) \in \mathcal{Z}_S. \end{cases}$$

Théorème 1.1. *(Théorème de Geoffrion [15]). Étant donnée le problème* (P_λ)*, alors* $\tilde{Z}$ *est un vecteur non dominé si et seulement si* $\tilde{Z}$ *est une solution optimale du problème paramétrique* (P_λ).

Autrement dit :

▷ Si $\exists\lambda \in \Lambda$ tel que x est une solution optimale de (P_λ) alors x est efficace pour le problème (MOLP).

▷ Si x est une solution efficace pour (MOLP) et que $\mathcal{Z}_S$ est un ensemble convexe alors il existe $\lambda \in \Lambda$ tel que x solution optimale du problème paramétrique (P_λ).

Définition 1.6. Une solution de base optimale du problème (P_λ), pour un certain $\lambda \in \Lambda$, est une solution efficace, et elle est dite *solution efficace extrême.*

1.7 Résolution du problème multi-objectifs

L'optimisation multi-objectifs consiste à définir des méthodes efficaces pour la résolution de problèmes où tous les objectifs sont pris en compte. Pour les solutions de problème multi-objectifs, la relation d'ordre n'est pas totale, une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres, on ne cherche pas alors une solution optimale mais un ensemble de solutions offrant un bon compromis entre les différantes fonctions objectifs à optimiser pour lesquelles on ne pourra pas effectuer une opération de classement. Les méthodes de résolution des problèmes multi-objectifs sont donc des méthodes d'aide à la décision car le choix final sera laissé au décideur qui choisit parmi les différentes solutions proposées par l'optimiseur la solution de compromis qui lui convient le mieux. Nous présentons brièvement quelques approches proposées au fil des années pour la résolution des problèmes multi-objectifs.

1.7.1 Méthode de la Somme pondérée [25]

Cette méthode consiste à additionnner tous les objectifs en affectant à chacun d'eux un coefficient de poids, ce coefficient représente l'importance relative que le décideur attribue à l'objectif, cela transforme le problème multi-objectifs à un problème mono-objectif de la forme :

$$\max_{x\in S} \sum_{k=1}^{p} \lambda_k Z_k(x), \lambda \in \Lambda.$$

Cette méthode est très efficace et peut être appliquée pour produire une solution non dominée initiale qui peut être employée comme solution initiale pour d'autres techniques, mais la difficulté essentielle de cette approche est de déterminer les poids appropriés quand nous n'avons pas d'informations sur les préférences du décideur, dans ce cas, toute solution optimale obtenue dépend fortement des coefficients utilisés. La plupart des chercheurs choisissent d'utiliser une combinaison linéaire des objectifs et de faire varier les poids de façon à constater l'influence de tel ou tel objectif sur la courbe de gain ("trade-off") des objectifs, cette approche est facile à implémentter mais elle a l'inconvénient de manquer la partie concave de la courbe de gain car tous les résultas obtenus appartiennent à des zones convexes de l'espace des objectifs, ce qui peut poser de sérieux problèmes dans son application aux problèmes réels.

1.7.2 Méthode de programmation par but"Goal programming"

Charnes et Cooper (1961) et Ijiri (1965) sont crédités du développement de la méthode "goal programming" pour un modèle linéaire, Dans cette méthode le décideur doit fixer un but (niveaux d'aspiration) qu'il souhaite à atteindre pour chaque critère, ces valeurs sont ajoutées au problème comme des contraintes supplémentaire, la nouvelle fonction objectif est modéfiée de façon à minimiser la somme des écarts entre les valeurs réalisées des fonctions objectifs et les buts à atteindre. La forme la plus simple de cette méthode peut être formulée comme suit :

$$\min_{x \in S} \{ \sum_{k=1}^{p} |Z_i(x) - \hat{z}_i| \}$$

Où $\hat{z}_i$ représente la valeur à atteindre pour le $i^{\text{ème}}$ critère.
La méthode est très facile à mettre en œuvre mais la définition des buts à atteindre est une question délicate qui détermine l'efficacité de la méthode.

1.7.3 Méthode ε-contrainte

Cette méthode est basée sur la minimisation d'un objectif Z_i en considérant que les autres objectifs Z_j avec $j \neq i$ doivent être inférieurs à une valeur ε_j. En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité.

$$\begin{cases} \min & Z_i(x). \\ t.q & x \in S, \; Z_j(x) \leq \varepsilon_j, \; \forall j \neq i. \end{cases}$$

L'ensemble Pareto optimal peut alors être obtenu en faisant varier les valeurs de ε_j, et en ne résolvant ainsi qu'un ensemble de problèmes d'optimisation mono-objectif, mais la connaissance à priori des intervalles appropriés pour les valeurs de ε_j est exigée pour tous les objectifs.

1.7.4 Méthode lexicographiques [25]

Cette méthode est proposée par Fourman (1985), dans laquelle les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite, la solution optimale est obtenue en optimisant les objectifs l'un après l'autre selon cet ordre en commençant par le plus important, puis dans chaque itération, les objectifs du niveau supérieur sont ajouté sous forme des contraintes en fixant chaqu'un d'eux à sa valeur optimale trouvée précédemment.

Supposons que les objectifs sont rangés par l'ordre suivant :

$Z_1 \succ Z_2 \succ ... \succ Z_p$, le premier problème à résoudre est donné par

$$\begin{cases} \max & Z_1 = c^1 x \\ t.q. & x \in S. \end{cases}$$

Soit x_1^* la solution optimale trouvée avec $Z_1^* = Z_1(x_1^*)$.

Z_1^* devient une nouvelle contrainte qui est ajoutée au deuxième problème

$$\begin{cases} \max & Z_2 = c^2 x \\ t.q. & x \in S. \\ & Z_1(x) = Z_1^* \end{cases}$$

Le $k^{\text{ème}}$ problème sera le suivant :

$$\begin{cases} \max & Z_k = c^k x \\ t.q. & x \in S. \\ & Z_1(x) = Z_1^*, Z_2(x) = Z_2^*, ..., Z_{k-1}(x) = Z_{k-1}^*. \end{cases}$$

La procédure est répétée jusqu'à ce que tous les objectifs soient traités et la solution obtenue à l'étape p sera la solution du problème.

1.8 Conclusion

Durant ces 20 dernières années, les chercheurs ont beaucoup utilisé les techniques de programmation linéaire dans la résolution du problème MOLP. Malheureusement, en ce qui concerne les problèmes MOILP, une intégration des méthodes de programmation linéaire uni-critère en nombres entiers et des méthodes de programmation linéaire multi-objectifs ne conduit pas automatiquement à une méthode de résolution. Nous présentons dans le chapitre suivant la structure générale d'un problème MOILP, en signalant la difficulté d'un tel problème, et quelques approches de résolution.

Chapitre 2

La programmation Linéaire Multi-objectifs en nombres entiers

2.1 Introduction.

Dans de nombreuses situations réelles modélisables par la programmation mathématique, les variables de décision ne peuvent prendre que des valeurs entières, dans ce cas on parle, dans le cas linéaire, de programmation linéaire en nombres entiers ou de programmation linéaire discrète, dont le domaine des solutions réalisables n'est plus convexe, ce qu'il en résulte que la programmation linéaire discrète est plus complexe et plus difficile à traiter.
La programmation linéaire uni-critère reste une source d'inspiration fondamentale pour le développement de méthodes multi-objectifs, nous présentons dans ce chapitre, en premier lieu, les principales notations et les résultats classiques concernant la programmation linéaire en nombres entiers. Ensuite, nous rappelons les concepts de base de l'optimisation discrète multi-objectifs ainsi que quelques méthodes existantes traitant ce genre de problèmes.

2.2 Programmation linéaire uni-critère en nombres entiers

La forme générale d'un problème de programmation linéaire uni-critère peut être donnée par :

$$(LP)\begin{cases} \max & z = cx \\ t.q & x \in S \end{cases}$$

où $S = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}, A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, c \in \mathbb{R}^{1\times n}$.

Dans le cas où seulement quelques variables sont entières, on a alors un problème de programmation linéaire mixte qui s'écrit comme suit :

$$(LPMIX)\begin{cases} \max & (cx + hy) \\ t.q & Ax + Gy = b \\ & x \geq 0. \\ & y \in \mathbb{N} \end{cases}$$

Si toutes les variables sont entières, on aura un programme linéaire en nombres entiers donné par :

$$(ILP)\begin{cases} \max & cx \\ t.q & x \in D \end{cases}$$

où $D = S \cap \mathbb{Z}^n$, $\mathbb{Z}$ est l'ensemble des nombres entiers relatifs.

2.2.1 Complexité

Les problèmes de programmation linéaire en variables discrètes (ILP) sont connus pour être NP-Complets, on n'espère pas, en général, qu'il existe un algorithme en temps polynomial pour ces problèmes. Les deux principales familles de méthodes actuellement connues pour résoudre ces problèmes sont les méthodes arborescentes et les méthodes de troncatures (coupes), en effet ces méthodes exigent un effort calculatoire qui croit exponentiellement avec le nombre de variables du problème.

Remarque. il existe une relation étroite entre (ILP) et (LP). Si par hasard la solution optimale de (LP) est entière, elle est aussi la solution optimale de (ILP), donc la résolution de (LP) peut aider à résoudre (ILP).

2.2.2 Méthodes de résolution d'un programme linéaire en nombres entiers

Il existe généralement deux méthodes exactes pour résoudre un programme linéaire en nombres entiers : les méthodes de coupes et les méthodes par séparation et évaluation "Branch & Bound".

2.2.3 Méthodes de coupes[27]

Les méthodes de coupes sont des algorithmes exacts pour les problèmes (ILP). Ces méthodes résolvent une séquence de relaxations du problème (ILP), les solutions obtenues sont graduellement améliorées pour donner une meilleure approximation de la solution optimale. Pour les grandes instances, le problème (ILP) ne peut pas être résolu à l'optimum, les algorithmes de coupes produisent alors des solutions relativement proches d'une solution optimale en un temps d'exécution raisonnable, avec une garantie sur la valeur de l'approximation, nous décrivons dans cette section deux types de coupes les plus utilisées.

a) Coupe de Dantzig [13]

La coupe de Dantzig a été proposée sur la base que dans le probléme relaxé (LP), le deuxième membre de l'équation matricielle des contraintes est un vecteur positif mais non entier, et une des variables hors-base est strictement positive (supérieure ou égale à 1), cette coupe est formulé par

$$\sum_{j \in N} x_j \geq 1 \tag{2.1}$$

où N est l'ensemble des indices hors-base

La coupe de Dantzig n'est plus utilisée (sauf en cas de nécessité) vu ses conditions d'utilisation .

b) Coupe fractionnaire de Gomory [16]

L'idée principale de cette méthode est d'ajouter des contraintes linéaires qui n'excluent aucun point entier réalisable, une par une jusqu'à ce que la solution optimale de la relaxation soit entière.

Dans une première étape, on résout le programme relaxé (LP), on cherche une solution de base optimale en utilisant la méthode de simplexe, si elle existe, on choisit une variable de base non entière et on génère une inéquation sur la contrainte associée à cette variable afin de couper la région de faisabilité courante.

Étant donnée une base optimale B du problème relaxé (LP), le tableau optimal correspondant est donné par

x_B	$\hat{A} = B^{-1}A$	$\hat{b} = B^{-1}b$
$-Z$	$\hat{c} = c - \pi A$	$z - c_B B^{-1} b$

Tab. 2.1 – Tableau simplexe optimal associé à la base B.

Où :

$\pi = c_B B^{-1}$: est dit *vecteur multiplicateur relatif à la base* B.

$\hat{c} = c - \pi A$: est dit *vecteur coût réduit* relatif à la base B, avec $\hat{c}_B = 0$.

Si la solution optimale de (LP) est entière, elle est une solution optimale du problème (ILP). Sinon, parmi les variables de base, choisissons $x_i,\ i \in B$ dont la valeur est fractionnaire.
La $i^{\text{ème}}$ ligne du tableau optimal est donnée par

$$x_i + \sum_{j \in N} \hat{a}_{ij} x_j = \hat{b}_i. \tag{2.2}$$

Où :

$\hat{a}_{ij}$: est un élément de la matrice optimale des contraintes $\hat{A}$.

N : est l'ensemble des indices hors-base.

Notations : étant donné un nombre réel α, on désigne par :
$\lfloor \alpha \rfloor$: le plus grand entier inférieur ou égal à α.
$\langle \alpha \rangle = \alpha - \lfloor \alpha \rfloor$ est appelée la partie fractionnaire de α et $\lfloor \alpha \rfloor$ sa partie entière.
Puisque toutes les variables sont positives ou nulles, on a :

$$\sum_{j \in N} \lfloor \hat{a}_{ij} \rfloor x_j \leq \sum_{j \in N} \hat{a}_{ij} x_j.$$

de (2.2) on a :

$$x_i + \sum_{j \in N} \lfloor \hat{a}_{ij} \rfloor x_j \leq \hat{b}_i.$$

Comme le membre gauche est entier dans cette inégalité, la partie droite (second membre) peut être remplacée par sa partie entière

$$x_i + \sum_{j \in N} \lfloor \hat{a}_{ij} \rfloor x_j \leq \lfloor \hat{b}_i \rfloor. \tag{2.3}$$

en soustrayant (2.3) de (2.2) on obtient

$$\sum_{j \in N} \langle \hat{a}_{ij} \rangle x_j \geq \langle \hat{b}_i \rangle$$

En ajoutant une variable d'écart x_s à cette dernière inéquation, on obtient la coupe de Gomory définie par

$$-\sum_{j \in N} \langle \hat{a}_{ij} \rangle x_j + x_s = -\langle \hat{b}_i \rangle.$$

Cette contrainte est introduite dans le tableau simplexe optimal et le nouveau problème formé peut être resolu en utilisant la méthode duale du simplexe. Après un nombre fini d'itérations, ou bien on obtient une solution optimale entière, ou bien le problème devient impossible.

2.2.4 Méthode par séparation et évaluation "Branch & Bound"

La méthode de "Branch & Bound" a été spécialement élaborée pour des problèmes en variables discètes (ILP), le principe de cette méthode consiste à subdiviser l'ensemble S (l'ensembles de solutions admissibles) en un nombre fini de sous-ensembles S^i, généralement on prend

$$\bigcup_{i=1} S^i = S \qquad avec \qquad S^i \cap S^j = \emptyset, \forall (i,j), i \neq j$$

Ces subdivisions successives sont représentées à l'aide d'une arborescence de racine S et des "nœuds" S^i présentant les sous-ensembles de solutions effectués, notons que S^i peut être vide.

Le déroulement de la méthode est constitué principalement de trois procédures :

1) Procédure de séparation

La phase de séparation consiste à diviser le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solution réalisables. Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir resolu le problème initial, ce principe de séparation peut être appliqué de manière recurssive à chacun des sous-ensembles de solutions obtenus.

2) Procédure d'évaluation

l'évaluation d'un nœud de l'arborescence a pour but de déterminer l'optimum (une borne supérieure pour un problème de maximisation) de l'ensemble des solutions réalisables associé au nœud en question, ou au contraire, de prouver mathématiquement que cet ensemble ne contient pas de solution optimale pour le problème initial.
La solution optimale du sous-problème associé à un nœud donné est appelée *solution partielle.*

3) Procédure Stérilisation

Le but de cette procédure est d'éviter l'examination de tous les nœuds de l'arborescence. Dans le cas où l'évaluation de la solution optimale du sous-probmème traité est inférieur à la borne supérieure globale (ie. la meilleure solution trouvée jusqu'a present), on est certain que toute solution réalisable de ce sous-problème ne sera pas meilleure que l'optimum global courant, il est donc inutile d'éffectuer la séparation de son ensemble de solutions réalisables, ou au contraire, si l'évaluation de la solution optimale de ce sous-problème est supérieur à celle de l'optimum global courant, dans ce cas, ce dernier peut être actualisé. On peut également arrêter la recherche dans un nœud lorsque le sous-problème qui est lui associé est non réalisable.

2.3 Programmation linéaire multi-objectifs en nombres entiers

La programmation linéaire multi-objectifs en nombres entiers fait partie du cadre de l'aide à la décision multi-critère (MCDM) à décisions discrètes. Ces dernières décennies, plusieurs méthodes ont été développées pour la résolution des problèmes (MOILP), que ce soit des méthodes donnant l'ensemble de toutes les solutions efficaces, ou des méthodes générant un sous-ensemble de points efficaces selon les préférences de décideur. Dans cette section nous rappelons dans un premier temps la structure générale d'un problème (MOILP) et quelques résultats relatifs, ainsi que quelques méthodes de résolution existantes dans la littérature.

2.3.1 Formulation mathématique d'un problème MOILP

Un programme linéaire multi-objectifs en nombre entiers (MOILP) est constitué d'un système de contraintes linéaires définissant un domaine discret de solutions réalisables et d'un ensemble de fonctions linéaires à maximiser où minimiser définissant des critères conflictuels, le problème (MOILP) consiste à déterminer toute solution realisable entière x telle qu'il n'existe aucune autre solution realisable y qui fournisse des valeurs au mois aussi bonne que celles de x sur chaque critère et meilleur sur au mois l'un des critères.

Un problème (MOILP) peut être formulé par :

$$(MOILP) \begin{cases} \text{“max”} & Z_k = c^k x; k = 1, .., p \\ t.q & x \in D. \end{cases}$$

Où $D = S \cap \mathbb{Z}^n$, $\mathbb{Z}$ étant l'ensemble des nombres entiers relatifs.
et $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$, $A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, m, n \in \mathbb{N}$
Rappelons d'abord la définition d'une solution efficace.

Définition 2.1. une solution réalisable $x^\star$ du problème (MOILP) est une solution *efficace*, si et seulement si, il n'existe pas d'autre solution réalisable x telle que $Z_k(x) \geq Z_k(x^\star), k = 1, ..., p$, avec au moins une inégalité stricte et le vecteur critères correspondant, $Z(x^\star)$, est dit *solution non dominée.*

Notons ***IE*** l'ensemble de toutes les solutions efficaces du problème (MOILP).

Théorème 2.1. *[9] Si $x^\star$ est une solution optimale du problème (mono-critère)*

$$(P_\lambda) : \max\{\lambda^t Z(x) : x \in S\}$$

alors pour tout $\lambda \in \mathbb{R}^p, \lambda > 0$, $x^\star$ est une solution efficace du problème multi-critère :

$$\text{“max”}\{Z(x) : x \in S\}.$$

2.3.2 Solutions efficaces supportées et non supportées

Les problèmes MOLP et MOILP sont de nature différente, la différence est due au fait que l'ensemble S est convexe [1] par contre l'ensemble D n'est plus convexe, pour cette raison le principe de Geoffrion qui permet de caractériser l'ensemble des solutions efficaces par les solutions optimales du problème paramétrique (P_λ) n'est pas valable pour les problèmes (MOILP), ceci car certaines solutions efficaces peuvent ne pas être obtenues pour aucun $\lambda > 0$. Pour cela, l'ensemble de solutions efficaces peut être partagé en deux sous-ensembles :

* l'ensemble des solutions efficaces obtenues en résolevant le problème paramétrique (P_λ) dite *solutions efficaces supportées* (supported efficient solutions) notée ***SE***.
* l'ensemble des solutions efficaces qui ne sont pas solutions du problème (P_λ) pour aucun λ dite *solutions efficaces non supportées* (non-supported efficient solutions) notée ***NSE***.

2.3.3 Détection graphique des solutions efficaces

Nous donnons quelques concepts utilisés pour détecter géométriquement les solutions efficaces d'un problème MOILP.

Définition 2.2. (Cône).
Soit $V \in \mathbb{R}^n, V \neq \emptyset$, V est un Cône si et seulement si $\alpha v \in V$ pour tout scalaire $\alpha \geq 0$ et tout $v \in V$. Le vecteur d'origine $0 \in \mathbb{R}^n$ est contenu dans chaque Cône.

Définition 2.3. (Cône polaire).
Soit $V \in \mathbb{R}^n$ un Cône. Le cône polaire non négatif de V (noté $V^{\geq}$) est le cône convexe[2] $V^{\geq} = \{y \in \mathbb{R}^n |\ y^t v \geq 0\ \forall v \in V\}$.

Définition 2.4. (Vecteurs générateurs d'un cône).
Soit un ensemble de vecteurs $\{v^1, ..., v^p\}$ de $\mathbb{R}^n$ et l'ensemble V tel que :

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

[1] Un ensemble $A \subset \mathbb{R}^n$ est dit convexe si : $\forall x_1, x_2 \in A, \forall \lambda \in [0, 1] \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in A$.
[2] Un cône convexe est un cône qui est également un ensemble convexe.

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

Définition 2.5. (Cône semi-polaire positif).
Soit $V \in \mathbb{R}^n$ un cône convexe généré par $\{v^1, v^2, ..., v^p\}$. Alors, le cône semi-polaire positif (noté $V^>$) est le cône convexe
$V^> = \{y \in \mathbb{R}^n | \ y^t v^i \geq 0, \text{ pour tout } i \text{ et } y^t v^i > 0 \text{ pour au moins un } i\} \bigcup \{0_{\mathbb{R}_n}\}$.

Définition 2.6. (Ensemble de dominance).
Soit $\bar{x} \in S$ et $C^>$ le cône semi-polaire positif généré par les gradients des p fonctions objectifs i.e. :
$C^> = \{y \in \mathbb{R}^n | c^i y \geq 0, \text{pour tout } i \text{ et } c^i y > 0 \text{ pour au moins un } i\} \bigcup \{0_{\mathbb{R}_n}\}$.
L'ensemble de dominance de $\bar{x}$ est donné par :

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

L'ensemble de dominance en $\bar{x}$ contient tous les points dont les vecteurs critères dominent le vecteur critère de $\bar{x}$. Le théorème suivant montre l'importance de cet ensemble dans la détection des solutions efficaces :

Théorème 2.2. *(Steuer,1986). Soit $D_{\bar{x}}$ l'ensemble de dominance de $\bar{x} \in S$, $\bar{x}$ est efficace si et seulement si $D_{\bar{x}} \cap S = \{\bar{x}\}$.*

2.4 Résolution du problème MOILP

Il existe plusieurs méthodes pour la résolution du problème MOILP, génénalement, ces méthodes peuvent être classées en deux catégories, les méthodes interactives et les méthodes non interactives, nous présentons dans la suite quelques méthodes des deux catégories.

2.5 Les méthodes interactives

2.5.1 Méthode de Klein & Hannan [21]

La technique est proposée pour générer séquentiellement un sous-ensemble efficace ou l'ensemble de toutes les solutions efficaces pour le problème MOILP, elle consiste à résoudre une suite de programmes linéaires en nombres entiers (IPL) en rajoutant progressivement des contraintes supplémentaires assurant que les nouvelles solutions seront meilleurs que toutes les solutions déjà trouvées sur au moins un critère.

L'algorithme de la méthode :

Étape 1 : Choisir arbitrairement un critère $i, i \in 1, 2, ..., p$ du problème (MOILP) et résoudre le problème uni-critère suivant :

$$(P_0)\begin{cases} \max & Z_i = c^i x \\ t.q & x \in D. \end{cases}$$

Si la solution de (P_0) est unique, alors elle est efficace et elle est l'unique élément dans la liste initiale des solutions efficaces $\boldsymbol{IE}_0$, sinon soit $\zeta(P_0)$ l'ensemble des solutions optimales de (P_0), par comparaison deux à deux des vecteurs critères associés, retenir celles qui sont non dominées pour construire $\boldsymbol{IE}_0$ l'ensemble des solutions efficaces correspondant à $\zeta(P_0)$.

Étape j : $(j \geq 1)$ résoudre le problème (P_j) défini par :

$$(P_j)\begin{cases} \max & Z_i = c^i x \\ t.q & x \in D. \\ & \bigwedge_{k=1}^{r}\left(\bigvee_{s=1,s\neq i}^{p}(c^s x \geq c^s \tilde{x}_k + \epsilon^s)\right). \end{cases}$$

où $0 < \epsilon^s \leq 1$ et $\tilde{x}_k,\ k = 1, ..., r$ sont les solutions efficaces obtenues dans les itérations $0, 1, ..., j-1$.

Si $\epsilon^s < 1$ la méthode génère un sous-ensemble de l'ensemble des solutions efficaces, et si $\epsilon^s = 1$ la procédure donne toutes les solutions efficaces. La liste des solutions efficaces obtenue à l'étape j est

$$IE_j = \bigcup_{k=0}^{j-1} IE_k.$$

la procédure s'arrête lorsque le problème (P_j) devient irréalisable. Il est clair que la procédure est finie étant donné qu'on élimine au moins une solution à chaque étape et qu'il existe un nombre finie (si D est borné) de solutions admissibles.

2.5.2 Méthode de Gonzales, Reeves & Franz[17]

Dans cette méthode, les préférences du décideur sont requises afin de mettre à jour l'ensemble des solutions efficaces sélectionnées à chaque étape, la méthode est donc

interactive et elle s'applique aux problèmes MOILP sans restriction, elle est principalement composée de deux phases :
- La première phase consiste à sélectionner un sous-ensemble $\tilde{S} \subset SE(MOILP)$ de solutions efficaces (solutions supportées) selon les préférences du décideur.
- Dans la deuxième phase, les solutions non supportées seront considérées et seulement celles préférées par le décideur (DM) seront introduites dans l'ensemble $\tilde{S}$. Donc, à chaque étape, une nouvelle solution efficace est présentée au décideur qui doit la comparer avec les éléments de $\tilde{S}$ pour actualiser l'ensemble de solutions efficaces préférées. L'algorithme de la méthode est décrit en détail dans [8].

2.5.3 Méthode de Steuer & Choo[25]

Cette méthode est une procédure générale applicable à tout problème de programmation mathématique multi-critère linéaire ou non, avec le cas échéant des variables entières, la procédure se décompose en trois étapes principales :
Etape 1 : Calculer un point cible, par exemple le point idéal, la recherche des solutions efficaces se fait par quadrillage de l'ensemble D au moyen d'un ensemble diversifié de valeurs du vecteur de paramètres λ, pour chaque valeur de λ, la distance de Tchebytcheff calculée par rapport au point cible est minimisée, ceci donne un sous-ensemble de solutions efficaces supportées, soit x^1 la solution efficace choisie par le décideur.
Etape 2 : Déterminer un deuxième ensemble de valeurs du vecteur λ de façon à quadriller le voisinage élargi de x^1 . Une nouvelle solution efficace, soit x^2, est désignée par le décideur comme en étape 1.
Etape 3 : Continuer la procédure mais en focalisant le paramètre λ dans le voisinage restreint du nouveau compromis.

2.6 Les méthodes non interactives

2.6.1 Méthode de Chalmet, Lemondis & Elzinga [12]

La méthode est proposée pour la résolution des problèmes bi-critère en variables discrètes, dans une première étape le problème paramétrique

$$\max_{x \in D}\{\lambda Z_1(x) + (1-\lambda)Z_2(x)\}$$

est résoulu pour générer l'ensemble des solutions supportées $\boldsymbol{SE}$, en suite l'ensemble des solutions non supportées est déterminé de la façon suivante :
Soients $Z^1 = (Z_1^1, Z_2^1)$ et $Z^2 = (Z_1^2, Z_2^2)$ deux points non dominés, adjacents parmi les points déjà obtenus. On cherche s'il existe d'autres solutions non-dominées telles que :

$$Z_1^1 \leq Z_1 \leq Z_1^2 \text{ et } Z_2^2 \leq Z_2 \leq Z_2^1$$

en résolvant le problème :

$$(P')\begin{cases} \max & \lambda Z_1(x) + (1-\lambda)Z_2(x) \\ t.q & Z_1(x) \geq Z_1^1 \cdots\cdots (i) \\ & Z_2(x) \geq Z_2^2 \cdots\cdots (ii) \\ & x \in D, \lambda \in]0,1[\end{cases}$$

Graphiquement, on a la représentation suivante

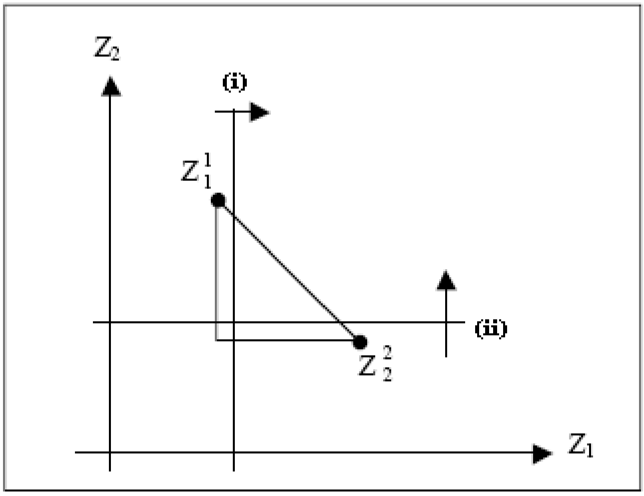


Fig. 2.1 – Illustration de la méthode

S'il existe une solution de (P'), Soit Z^l (un vecteur non-dominé) la même procédure est appliquée ensuite avec les paires Z^1, Z^l et Z^l, Z^2. Sinon, s'il n'existe pas de solution pour le problème (P') alors Z^1 et Z^2 sont deux solutions non dominées adjacentes. La procédure prend fin lorsque toutes les paires adjacentes ont été examinées.

2.6.2 Méthode de Bitran [4]

Une classe très intéressente du problèmes multi-objectifs en nombres entiers sont les problèmes pour lesquels les variables ne peuvent prendre que deux valeurs 0 ou 1.

Ce type de problèmes est noté MOBLP (Multiple Objective Binary Linear Programming)
La formulation mathématique d'un tel problème est donnée par :

$$(MOBLP)\{\text{“}\max_{x\in S'}\text{”} \quad z_k = c^k x, k = 1, ..., p\}$$

où $S' = \{x \in \{0,1\}^n | Ax \leq b\}$.

Cette méthode identifie toutes les solutions efficaces et elle est spécifique aux problèmes à variables binaires. Bitran a considéré d'abord le problème :

$$(P_{B_1})\{\text{“}\max_{x\in S_1}\text{”} \quad z_k = c^k x, k = 1, ..., p\}$$

Avec $S_1 = \{0,1\}^n$: l'ensemble des sommets de l'hypercube unité dans $\mathbb{R}^n$. Les relations suivantes ont été établies entre (MOBLP) et (P_{B_1}) :

• $x \in E(P_{B_1}) \cap S \Rightarrow x \in E(MOBLP)$.

C'est-à-dire une solution efficace pour le problème (P_{B_1}) et réalisable pour le problème (MOBLP) est aussi efficace pour le problème (MOBLP).

• $x \notin E(P_{B_1}) \cap S \nRightarrow x \notin E(MOBLP)$.

C'est-à-dire une solution qui n'est pas efficace pour le problème (P_{B_1}) n'est pas forcement non efficace pour le problème (MOBLP).

L'algorithme de la méthode comporte trois étapes :

(a) Caractériser l'ensemble des solutions efficaces du problème (P_{B_1}).

(b) Parmi ces solutions, déterminer celles qui sont réalisables pour le problème (MOBLP).

(c) Adjoindre ensuite les autres solutions efficaces de (MOBLP) non trouvées dans l'étape (a).

(a) Caractérisation de (P_{B_1}) :

▷ Soit $V = \{v^t \in \mathbb{R}^n, t \in T | Cv^t \geq 0, v^t_j \in \{0, 1, -1\} \forall j\}$ est dit l'ensemble de directions de préférence, où T est un ensemble d'indices.

On dit que x^1 domine x^2 dans la direction v^t, si et seulement si, $x^1 = x^2 + v^t$, donc $Cx^1 \geq Cx^2$.

▷ Soit $M(v^t)$ l'ensemble des points de S_1 dominés dans la direction v^t par un autre point de S_1. On a $M(v^t) = \{x^{t,r} | r = 1, ..., R\}$ avec :

$$x^{t,r} = \begin{cases} 0 & si \quad v_j^t = 1 \\ 1 & si \quad v_j^t = -1 \\ 0 \; ou \; 1 & si \quad v_j^t = 0 \end{cases}$$

Bitran a montré que cet ensemble peut être déterminé par l'ensemble des solutions optimales du problème $\min\limits_{x \in S_1'} v^t x$
où $S_1^{'} = \{x \in \mathbb{R}^n | 0 \leq x_j \leq 1, \forall j\}$ est la relaxation linéaire de S_1.
la résolution successive des problèmes relaxés, notés (RP_{B_1}), pour différentes directions $v^t, t \in T$ permet de déterminer successivement $M(v^t)$ puis $E(P_{B_1}) = \overline{\bigcup\limits_{t \in T} M(v^t)}$ [3].
Notons que quelques directions seulement sont examinées.
(b) Détermination de l'ensemble $E_1 = E(P_{B_1}) \cap S$:
Pour cela il suffit de tester l'admissibilité des solutions de $E(P_{B_1})$.
(c) Caratérisation de l'ensemble E(MOBLP) :
Pour déterminer l'ensemble des solutions non efficaces dans (P_{B_1}) et efficaces dans (MOBLP), noté E_2, on utilise la propriété suivante :

$$x \in E_2 \Rightarrow x + v^t \notin S, \forall v^t, \; t \in T \quad t.q. \; x \in M(v^t)$$

Finalement, $E(MOBLP) = E_1 \cup E_2$

2.6.3 Méthode de J. Sylva & A. Crema [9]

La méthode proposée par les auteurs est une variante de celle de Klein & Hannan présentée précédemment, ils ont élaborés un algorithme d'énumération de tous les vecteurs non dominés d'un problèmes MOILP en utilisant une approche théorique simple et directe, le problème est résolu en résolvant une suite de programmes linéaires en nombres entiers maximisant une combinaison positive de toutes les fonctions objectifs, en rajoutant dans chaque étape des contraintes assurant la détection d'une nouvelle solution efficace.
L'algorithme génère aussi des sous-ensembles de solutions efficaces qui peuvent être utiles dans la construction des méthodes interactives pour des problèmes concrets de grande dimension.

[3] $\overline{A}$ est le complément de l'ensemble A

Rappelons qu'un problème MOILP est défini par

$$(P)\begin{cases} \text{“max”} & Z = Cx \\ t.q. & x \in D = \{Ax = b, x \in \mathbb{Z}^n, x \geq 0\}. \end{cases}$$

Où $C \in \mathbb{Z}^{p\times n}, A \in \mathbb{R}^{m\times n}$ et $b \in \mathbb{R}^m$.

Théorème 2.3. *[9] Si $x^\star$ est une solution optimale du problème (mono-critère) :*

$$\max\{\lambda^t Cx : x \in S\}$$

Alors pour tout $\lambda \in \mathbb{R}^p,\ \lambda > 0$, $x^\star$ est une solution efficace du problème (P).

Proposition 2.1. *[9] Soit $x^1, x^2, ..., x^l$ des solutions efficaces pour le problème (P) et $D_s = \{x \in \mathbb{Z}^n : Cx \leq Cx^s, s \in \{1, ..., l\}\}$.*
Si x^* solution efficace pour le problème

$$(P_l)\begin{cases} \text{“max”} & Z = Cx \\ t.q. & x \in (D - \bigcup_{s=1}^{l} D_s). \end{cases}$$

Alors x^*est une solution efficace pour le problème (P).
De plus, si le problème (P_l) est non réalisable, alors $\{Cx^s\}_{s=1}^{l}$ est l'ensemble de tous les vecteurs critères non dominés pour le problème (P).

Corrolaire 2.1. *[9] Soit $x^1, x^2, ..., x^l$ des solutions efficaces pour le problème (P), $D_s = \{x \in \mathbb{Z}^n | Cx \leq Cx^s, s \in \{1, ..., l\}\}$.*
Si x^* est une solution optimale du problème

$$(P_l^\lambda)\begin{cases} \max & \lambda^t Cx \\ t.q. & x \in (D - \bigcup_{s=1}^{l} D_s). \end{cases}$$

pour un certain vecteur $\lambda \in \mathbb{R}^p, \lambda > 0$, alors x^* est une solution efficace pour le problème (P).

L'algorithme de la méthode

Étape 1 : Après avoir choisi le vecteur $\lambda, \lambda > 0$, la première étape consiste à résoudre le problème

$$(P_0^\lambda)\begin{cases} \max & \lambda^t Cx \\ t.q. & x \in D. \end{cases}$$

Si ce problème n'est pas réalisable, alors le problème (P) est aussi non réalisable. Sinon, une solution optimale x^1 est trouvée et elle est efficace pour le problème (P) d'après le théorème précédent. Ensuite, une suite de problèmes linéaires en nombres entiers augmentés par un ensemble de contraintes éliminant les solutions efficaces déjà trouvées sont résolus progressivement.

Après l itérations du processus, si le problème (P_{l-1}^{λ}) n'est pas réalisable, l'algorithme prend fin, sinon, une nouvelle solution efficace x^l est trouvée et un nouveau problème (P_l^{λ}) est défini en éliminant de l'ensemble d'admissibilité de (P_{l-1}^{λ}) toutes les solutions vérifiant $Cx \leq Cx^l$, ceci peut être traduit mathématiquement par les contraintes suivantes :

$$\begin{cases} (Cx)_k \geq ((Cx^l)_k + 1)y_k^l - M_k(1 - y_k^l), k = 1, 2, ..., p. \\ \sum_{k=1}^{p} y_k^l \geq 1, y_k^l \in \{0, 1\}, k = 1, 2, ..., p. \end{cases}$$

Où $-M_k$ est la borne inférieure pour les valeurs réalisables de la $k^{\text{ème}}$ fonction objectif (Par exemple, dans le cas où la matrice des critères C est positive, M_k peut être fixée à 0 pour tout k).

Étape *l* : Résoudre le problème :

$$(P_l^{\lambda}) \begin{cases} \max & \lambda^t Cx \\ t.q. & x \in D. \\ & (Cx)_k \geq ((Cx^s)_k + 1)y_k^s - M_k(1 - y_k^s). \\ & \sum_{k=1}^{p} y_k^s \geq 1, y_k^s \in \{0, 1\} \quad \text{pour } s = \overline{1, l}; k = \overline{1, p}. \end{cases}$$

Remarque. Pour des problèmes de grande taille, l'énumération de tous les vecteurs non dominés peut ne pas être possible, Dans ce cas l'intérêt se porte sur une partie seulement des solutions efficaces, cette partie peut être obtenue en changeant le problème (P_l) en :

$$(P_l^{\lambda}) \begin{cases} \max & \lambda^t Cx \\ t.q. & x \in D. \\ & (Cx)_k \geq ((Cx^s)_k + f_k)y_k^s - M_k(1 - y_k^s). \\ & \sum_{k=1}^{p} y_k^s \geq 1, y_k^s \in \{0, 1\}; s = \overline{1, l}; k = \overline{1, p}. \end{cases}$$

Où f_k représente l'écart souhaité entre deux vecteurs non dominés successivement trouvés pour chaque fonction objectif k.

La procédure continue jusqu'à ce que le problème (P_l) devient irréalisable. A la fin, on obtient l'ensemble de toute les solutions efficaces ou seulement une partie qui intéresse le décideur.
Un autre algorithme est proposé par **Sylva et Crema** [11] pour énumerer tous les vecteurs non dominés d'un programme linéaire multi-objectifs en variables mixtes. Où à partir d'un premier vecteur non dominé, la procedure cherche à chaque itération le premier point qui maximise une distance définie par la norme infini de l'ensemble dominé par les solutions precédament trouvées, lorsque toutes les variables sont entiers, l'algorithme génère l'ensemble de tous les vecteurs non dominés, l'algorithme de la méthode est décrit en détail dans [11].

2.6.4 Méthode de D.Chaabane & M.Abbas "SEEVD"[1]

"SEEVD" est l'abréviation de "Méthode de détermination des Solutions Efficaces dans l'Espace des Variables Discrètes", cette méthode est une forme améliorée de la méthode de Gupta et Malhotra [18] où le test d'arrêt est corrigé pour déterminer toutes les solutions efficaces du problème MOILP sans en manquer aucune.
Dans une première étape, une solution initiale est déterminée en résolvant un problème mono-critère en nombre entier, cette solution constitue le premier élément de la liste des solutions efficaces, puis une séquence de coupes est appliquée après avoir exploré les arêtes incidentes à cette solution selon une direction précise définie par un ensemble d'indices hors-base, ceci nous permet de générer une nouvelle solution efficace ajoutée à la liste précédente de solutions efficaces et de réduire le domaine de recherche jusqu'à ce qu'il devient vide. Dans ce cas le processus s'arrêt avec une liste finale contient toutes les solutions efficaces du problème traité. Nous commençons d'abord par quelques définitions et théorèmes nécessaire pour la compréhension du fonctionnement de l'algorithme.
Considérons le problème mono-critère (où uni-critère) suivant

$$(ILP_1)\begin{cases} \max & Z_1 = c^1x \\ t.q. & x \in D. \end{cases}$$

où $D = \{x \in \mathbb{N}^n | Ax = b\}$.

Notations

– $Z_1^1 = Z_1^*$: est la valeur optimale de Z_1 obtenue dans le problème (ILP_1).

– X_1^* : est la solution optimale correspondant à Z_1^*.

– $(Z_1^1, Z_2^1, ..., Z_p^1)$: est le premier p-uplet non dominé, avec $Z_i^1 = Z_i(X_1^*)$ pour $i \in \{2, ..., p\}$.

Pour $k \geq 1$:

– $X_k = (x_{k,j})$: est la solution efficace correspondant à $(Z_1^k, Z_2^k, ..., Z_p^k)$

– B_k : est la base associée à la solution X_k.

– $a_{k,j}$: est le vecteur d'activité de $x_{k,j}$ par rapport à la région tronquée.

– $y_{k,j} = (B_k)^{-1} a_{k,j}$.

– $I_k = \{j | a_{k,j} \in B_k\}$.

– $N_k = \{j | a_{k,j} \notin B_k\}$.

– $\Gamma_k = \{j \in N_k | Z_{k,j}^1 - c_j^1 > 0$ et $Z_{k,j}^i - c_j^i > 0$ pour au moins un $i \in \{2, 3, ..., p\}\}$.

Où $Z_{k,j}^i = c_{B_k}^i y_{k,j}$ et c_j^i est la $j^{\text{ème}}$ composante du vecteur c^i et $c_{B_k}^i$ est le vecteur des coefficients coûts des variables de base associées à B_k du vecteur c^i.

Un problème (ILP_1) peut avoir plusieurs solutions optimales, nous introduisons la notion de solution alternative dans la définition suivante :

Définition 2.7. Soit x^* une solution optimale du problème (ILP_1). Une solution réalisable $\tilde{x} \in D$ est dite *solution alternative* à x^* si $Z(x^*) = Z(\tilde{x})$.

Définition 2.8. Soit X_k une solution optimale de (ILP_1) et soit $j_k \in N_k$, l'arête E_{j_k} incidente à la solution X_k est définie par l'ensemble

$$E_{j_k} = \left\{ (x_i \in \mathbb{R}^{|I_k|+|N_k|}) \;\middle|\; \begin{array}{l} x_i = x_{k,i} - \theta_{j_k} y_{k,ij_k} \text{ pour } i \in I_k \\ x_{j_k} = \theta_{j_k} \\ x_\alpha = 0, \forall \alpha \in N_k \backslash \{j_k\} \end{array} \right\}$$

où $0 < \theta_{j_k} \leq \min_{i \in \mathrm{I}_k} \{\frac{x_{k,i}}{y_{k,ij_k}}; y_{k,ij_k} > 0\}$, θ_{j_k} est un entier positif et $\theta_{j_k} \times y_{k,ij_k}$ est un entier $\forall i \in I_k$.

Le théorème suivant génère des coupes correspondant aux directions définies par les indices de Γ_1.

Théorème 2.4. *Toute solution réalisable du problème* (ILP_1) *qui n'est pas sur l'arête* $E_{j_1}, j_1 \in \Gamma_1$*, émanant de la solution optimale* X_1 *du problème* (IPL_1) *appartient au demi-espace*

$$\sum_{j \in N_1 \backslash \{j_1\}} x_j \geq 1$$

Ce théorème se généralise sans difficulté pour les étapes successives $k \geq 2$, la démonstration peut être trouvée dans [8].

L'algorithme de la méthode

Étape 1. Résoudre le problème (ILP_1), soit X_1^* une solution optimale dont le p-uplet correspondant est $(Z_1^1, Z_2^1, ..., Z_p^1)$. Au lieu de (IPL_1), on peut résoudre un des problèmes $(IPL_i, i = 2, 3, ..., p)$ qui maximise $Z_i = c^i x$.

• Si $J_1 = \{j \in N_1 | Z_j^1 - c_j^1 = 0\} = \emptyset$, alors la solution optimale est unique, enregistrer le premier vecteur non dominé $(Z_1^1, Z_2^1, ..., Z_p^1)$ et former la liste initiale des solutions non dominées $Opt_1 = \{(Z_1^1, Z_2^1, ..., Z_p^1)\}$. Aller à l'étape 2.

• Si $J_1 \neq \emptyset$, la solution optimale peut ne pas être unique, pour chaque $j \in J_1$ calculer $\Theta_j = \lfloor \min_{i \in I_1} \{ \frac{x_{1,i}}{y_{1,ij}}, y_{1,ij} > 0 \} \rfloor$ [4].

(a) Si pour tout $j \in J_1$ on a $\Theta_j < 1$, alors il n'existe pas de solutions alternatives à la solution X_1, poser $Opt_1 = \{(Z_1^1, Z_2^1, ..., Z_p^1)\}$ et aller à l'étape 2.

(b) Sinon, tant qu'il existe au moins un $j \in J_1$ tel que $\Theta_j \geq 1$ faire ;

▷ Explorer l'arête

$$E_j = \left\{ (x_i \in \mathbb{R}^{|I_1| + |N_1|}) \;\middle|\; \begin{array}{l} x_{1,i} = x_{1,i} - \theta \times y_{1,ij} \text{ pour } \; i \in I_1 \\ x_{1,j} = \theta \\ x_{1,\alpha} = 0, \forall \alpha \in N_1 \backslash \{j\} \end{array} \right\}$$

pour θ entier variant entre 1 et Θ_j et $\theta \times y_{1,ij}$ entiers.

▷ Sur chacune des solutions réalisables trouvées sur une arête, évaluer les critères, et ajouter à la liste des vecteurs critères non dominés(une comparaison deux à deux est effectuée).

▷ Choisir arbitrairemet un $j_1 \in J_1$ et aller à l'étape 2.2. (on peut choisir j_1 tel que Θ_{j_1} soit maximal).

[4] $\lfloor \alpha \rfloor$: est le plus grand entier inférieur ou égal à α

Étape 2 Soit $k = 1$

(2.1) Construire l'ensemble

$\Gamma_k = \{j \in N_k | Z^1_{k,j} - c^1_j > 0$ et $\exists\ i \in \{2, 3, ..., p\}$tel que $Z^i_{k,j} - c^i_j > 0\}$.

(2.1.1) Si $\Gamma_k = \emptyset$, aller à l'étape 2.2, (la coupe effectuée est une coupe de Dantzig $\sum_{j \in N_k} x_j \geq 1$).

(2.1.2) Sinon, soit $\gamma = \Gamma_k$, et aller à (α).

(α) Si $\gamma = \emptyset$, choisir un indice $j_k \in \Gamma_k$ et aller à l'étape 2.2. Sinon, soit $j_k \in \gamma$, calculer $\theta^k_{j_k} = \lfloor \min_{i \in I_k} \{\frac{x_{k,i}}{y_{k,ij}}, y_{k,ij} > 0\} \rfloor$.

- Si $\theta^k_{j_k} = 0$, alors il n'y a aucune solution réalisable sur l'arête E_{j_k}, faire $\gamma := \gamma \backslash \{j_k\}$ et aller à (α).
- Sinon, aller à **(2.1.3)**.

(2.1.3) Pour $\theta = 1, 2, ..., \theta^k_{j_k}$, calculer toutes les solutions réalisables entière sur l'arête E_{j_k} en utilisant

$$\begin{cases} x_i = x_{k,i} - \theta \times y_{k,ij_k} \text{ pour } i \in I_k \\ x_{j_k} = \theta \\ x_\alpha = 0, \forall \alpha \in N_k \backslash \{j_k\} \end{cases}$$

Évaluer les critères sur chaque solution, éliminer les vecteurs critères dominés et introduire les nouveaux vecteurs critères potentiellement non dominés (non dominés par rapport aux solutions de la liste de l'itération k) dans la liste Opt_k. Choisir un indice j_k de Γ_k et aller à l'étape 2.2.

(2.2) Ajouter la coupe $\sum_{j \in N_k \backslash \{j_k\}} x_j \geq 1$ et appliquer la méthode dual simplexe et les coupes de Gomory si nécessaire. Soit X_{k+1} une solution optimale du problème augmenté.

Évaluer tous les critères en cette solution. Si le vecteur critères correspondant est dominé par l'un des éléments de la liste Opt_k, l'ignorer. Sinon, l'ajouter à la liste des vecteurs non dominés pour produire la liste Opt_{k+1}. Faire $k = k + 1$ et aller à l'étape 2.1

Étape 3 La procédure prend fin quand l'opération pivot est impossible, C'est-à-dire, le problème est devenu irréalisable dans la nouvelle région tronquée.

L'algorithme se termine par une liste finale Opt_{k+1} représente l'ensemble de toutes les

solutions non dominées du problème.
Gupta & Malhotra ont donnés une autre définition de l'ensemble Γ_k, leur définition favorise la recherche sur des arêtes correspondant à des directions améliorantes pour au moins un des critères :

$$\Gamma_k = \{j \in N_k | Z^1_{k,j} - c^1_j > 0 \; et \; Z^i_{k,j} - c^i_j < 0 \text{ pour au moins un } i \in \{2, 3, ..., p\}\}.$$

L'expérimentation numérique semble indiquer qu'il n'y a pas d'avantage à choisir une définition que l'autre, nous illustrons l'algorithme de la méthode sur un exemple numérique proposé par Sylva & crema [9].

Illustration numérique

Considérons le probmlème $MOILP$

$$(P) \begin{cases} \max & Z_1 = x_1 - 2x_2 \\ \max & Z_2 = -x_1 + 3x_2 \\ t.q. & x_1 - 2x_2 \leq 0 \\ & x_1 \leq 2 \\ & x_2 \leq 2 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

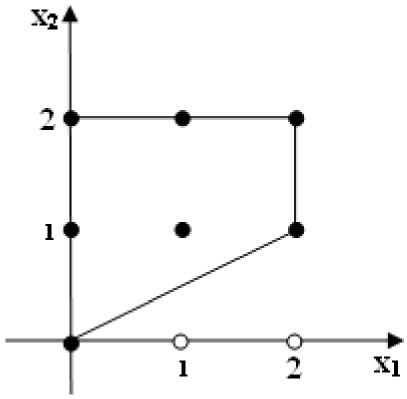


Fig. 2.2 – Domaine d'admissibilité

La résolution de (IPL_1) donne la solution $X_1 = X^*_1 = (0, 0)$, les résultats sont représentés dans le tableau suivant

Base	x_1	x_2	x_3	x_4	x_5	variables de base
x_1	1	-2	1	0	0	0
x_4	0	2	-1	1	0	2
x_5	0	1	0	0	1	2
$Z_j^1 - c_j^1$	0	0	1	0	0	0
$Z_j^2 - c_j^2$	0	-1	-1	0	0	0

Tab. 2.2 – Tableau optimal 1.

$I_1 = \{1, 4, 5\}$, $N_1 = \{2, 3\}$;
L'ensemble $J_1 = \{j \in N_1 / Z_j^1 - c_j^1 = 0\} = \{2\} \neq \emptyset$, la solution X_1 peut ne pas être unique, calculer alors Φ_2
$\Phi_2 = \lfloor \min\{\frac{2}{2}, \frac{2}{1}\} \rfloor = 1$
Explorer l'arête E_2

$$E_2 = \begin{cases} x_1^1 = 0 - (-2) = 2 \\ x_2^1 = 1 \\ x_3^1 = 0 \\ x_4^1 = 2 - 2 = 0 \\ x_5^1 = 2 - 1 = 1 \end{cases}$$

La solution entière sur l'arête E_2 est $X_1' = (2, 1)$ dont le vecteur critères correspondant est $(Z_1^1, Z_2^1) = (0, 1)$ qui est non dominé, initialiser alors l'ensemble des solutions non dominées par $Opt_1 = \{(0, 1)\}$.
Tronquer l'arête E_2 par la coupe : $\sum\limits_{j \in N_1 \setminus \{2\}} x_j \geq 1 \Leftrightarrow x_3 \geq 1$ (ou bien $-x_1 + 2x_2 \geq 1$).
Cette coupe est ajoutée au tableau optimal précédent, et par application de la méthode dual simplexe et les coupes de Gomory si nécessaire, on obtient le tableau optimal suivant

$Base$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	variables de base
x_2	0	1	0	0	0	0	-1	1
x_4	0	0	0	1	0	-1	2	1
x_5	0	0	0	0	1	0	1	1
x_3	0	0	1	0	0	-1	0	1
x_1	1	0	0	0	0	1	-2	1
$Z_j^1 - c_j^1$	0	0	0	0	0	1	0	-1
$Z_j^2 - c_j^2$	0	0	0	0	0	-1	-1	2

TAB. 2.3 – Tableau optimal 2.

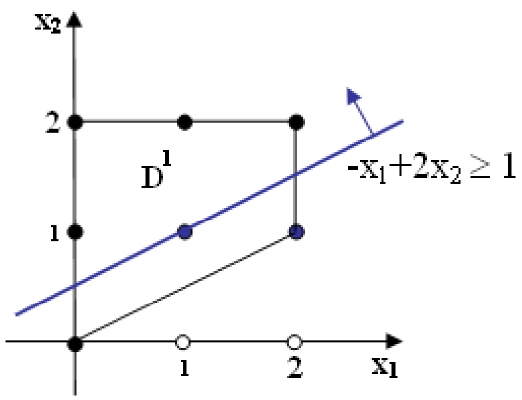


FIG. 2.3 – Région réduite D^1

La solution optimale est $X_2 = (1,1)$ dont le vecteur critères correspondant est $(Z_1^2, Z_2^2) = (-1,2)$ qui est non dominé, alors
$Opt_2 = Opt_1 \cup \{(-1,2)\}$.
$I_2 = \{2,4,5,3,1\}$, $N_2 = \{6,7\}$.
$\Gamma_2 = \{j \in N_2 / Z_j^1 - c_j^1 > 0 \ et \ Z_j^2 - c_j^2 < 0\} = \{6\} \neq \emptyset$
$\theta_6^2 = \lfloor \min\{\frac{1}{1}\} \rfloor = 1$, la solution entière sur l'arête E_6 est :

$$E_6 = \begin{cases} x_1^2 = 1 - 1 = 0 \\ x_2^2 = 1 - 0 = 1 \\ x_3^2 = 1 - (-1) = 2 \\ x_4^2 = 1 - (-1) = 2 \\ x_5^2 = 1 - 0 = 1 \\ x_6^2 = 1 \\ x_7^2 = 0 \end{cases}$$

le veteur critères correspondant est $(-2, 3)$ qui n'est pas dominé par aucun élément de l'ensemble Opt_2, donc l'ensemble des solutions non dominées devient
$Opt_2 = Opt_2 \cup \{(-2, 3)\}$.
Tronquer E_6 par la coupe $\sum_{j \in N_2 \setminus \{6\}} x_j \geq 1 \Leftrightarrow x_7 \geq 1$ ou bien $x_2 \geq 2$.

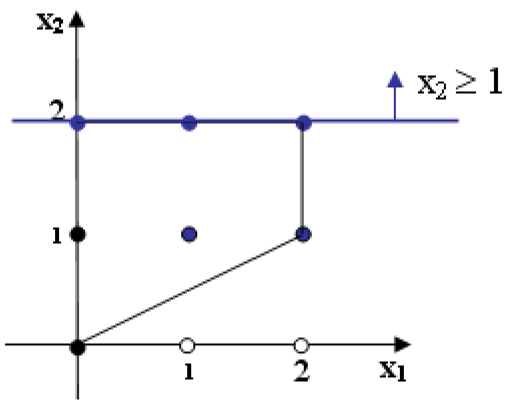


FIG. 2.4 – Région réduite D^2

La solution trouvée dans la région tronquée est présentée dans le tableau suivant

Base	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	variables de base
x_2	0	1	0	0	0	0	0	-1	2
x_6	0	0	0	-1	0	1	0	-2	1
x_5	0	0	0	0	1	0	0	1	0
x_3	0	0	1	-1	0	0	0	-2	2
x_1	1	0	0	1	0	0	0	0	2
x_7	0	0	0	0	0	0	1	-1	1
$Z_j^1 - c_j^1$	0	0	0	1	0	0	0	2	-2
$Z_j^2 - c_j^2$	0	0	0	-1	0	0	0	-3	4

TAB. 2.4 – Tableau optimal 3.

$X_3 = (2, 2)$, $(Z_1^3, Z_2^3) = (-2, 4)$, en comparant (Z_1^3, Z_2^3) avec les éléments de Opt_3, on trouve que le point $(-2, 3)$ est dominé par $(-2, 4)$, on obtient alors
$Opt_3 = \{opt_2 \setminus \{(-2, 3)\}\} \cup \{(-2, 4)\}$.
$I_3 = \{2, 6, 5, 3, 1, 7\}$, $N_3 = \{4, 8\}$
$\Gamma_3 = \{4, 8\} \neq \emptyset$.
pour $j = 4$, $\theta_4^3 = 2$, calculer les solutions entières sur l'arête E_4 :
$\theta = 1$; $\{x_1^3 = 1,\ x_2^3 = 2,\ x_3^3 = 3,\ x_4^3 = 1,\ x_5^3 = 0,\ x_6^3 = 2,\ x_7^3 = 1\}$,
le vecteur critères correspondant est $(-3, 5)$ qui est non dominé, donc
$Opt_3 = \{(0, 1),\ (-1, 2),\ (-2, 4),\ (-3, 5)\}$.
$\theta = 2$; $\{x_1^3 = 0,\ x_2^3 = 2,\ x_3^3 = 4,\ x_4^3 = 2,\ x_5^3 = 0,\ x_6^3 = 3,\ x_7^3 = 1\}$,
le vecteur critères correspondant est $(-4, 6)$ qui est non dominé, donc
$Opt_3 = \{(0, 1),\ (-1, 2),\ (-2, 4),\ (-3, 5), (-4, 6)\}$.
De la même manière, on ajoute la coupe $x_8 \geq 1$ au tableau optimal précédent, le problème devient impossible. L'algorithme prend fin avec l'ensemble de toute les solutions non dominées Opt_3, et l'ensemble de tous les points efficaces est donné par :

$$\boldsymbol{IE}(P) = \{(2, 1), (1, 1), (2, 2), (1, 2), (0, 2)\}.$$

Chapitre 3

L'optimisation d'une fonction linéaire sur un domaine efficace discret.

3.1 Introduction.

Depuis les années 1970, des auteurs se sont intéressés à l'optimisation d'une fonction sur l'ensemble des solutions efficaces d'un problème de programmation linéaire multi-objectifs. Dans certaines situations, les décideurs n'ont pas besoin d'énumérer l'ensemble de tous les points efficaces mais seulement celles qui optimisent un certain critère de compromis - généralement linéaire - dit *critère principal*, ces problèmes sont particulièrement difficile à traiter en raison de la nature non convexe de l'ensemble de faisabilité (l'ensemble des points efficaces). Le problème a été étudié la première fois dans le cas continu en 1972 par Philip [24], qui a décrit un algorithme basé sur le déplacement sur les sommets efficaces adjacents et un bon nombre de chercheurs ont suivi cette voie (Benson [3],Isermann et Steuer ,Yoshitsugu Yamamoto [29],Ecker, J. G. et Song, H.G. [14], Sayin 2000, et autre).

Les algorithmes existants pour résoudre ce problème peuvent être classés en plusieurs groupes selon leur principe de fonctionnement ; citons

- les algorithmes de recherche de sommet adjacent.
- les algorithmes de recherche de sommets non adjacents.
- les algorithmes basés sur la méthode de séparation et évaluation.
- les algorithmes basés sur la méthode de relaxation Lagrangienne.
- les algorithmes basés sur la méthode duale et de bisection.

De notre part, nous nous sommes particulièrement intéressés à la maximisation d'une fonction linéaire notée ϕ sur l'ensemble de solutions efficaces E d'un problème MOILP, Une approche directe consisterait à déterminer l'ensemble E et par suite à optimiser ϕ sur cette ensemble, cette approche n'est pas appropriée pour des raisons d'ordre pratique (difficulté de déterminer E qui est peut être un ensemble de taille exponentielle en nombre de variables).

Contrairement au cas continu qui a été complètement étudiée par plusieurs auteurs, le cas discret n'a pas vu beaucoup de développements semblable, une méthode donne seulement une borne supérieure de la fonction objectif ϕ est proposée par N.C. Nguyen [22], la première méthode proposée pour l'optimisation sur l'ensemble efficace d'un problème MOILP en évitant l'énumération explicite de tous les points efficaces est celle de M.Abbas et D.Chaabane [2], où différents types de coupes sont imposées de telle manière que l'amélioration de la valeur optimale de la fonction objectif à chaque itération soit garantie, plus une autre méthode exacte proposée par D.Chaabane [7] pour résoudre le même problème dans l'espace des critères en optimisant une somme pondérée des critères initiaux, suivie par l'algorithme de Jesùs M.Jorge [19] qui est basé sur l'analyse d'un ordre approprié des problèmes linéaires en nombres entiers pour éliminer successivement les solutions moins bonnes sur le critère principal.

Dans ce chapitre, nous allons détailler et illustrer les méthodes proposées par D.Chaabane et Jesùs M.Jorge pour raison qu'à partir de ses principes, nous avons élaboré une nouvelles méthode de résolution, où on a utilisé, d'une part, un ensemble de coupes en nombres entiers pour réduire le domaine d'admissibilité en éliminant les solution dominées par une solution efficace trouvée, et d'autre part, une exploration des arêtes incidentes à la solution optimale courante est réalisée si elle est non efficace pour chercher une solution alternative efficace.

Rappelons d'abord quelques définitions et théorèmes nécessaires pour justifier le fonctionement des algorithmes présentés.

Considérons le problème MOILP défini par :

$$(P(D)) \begin{cases} \text{“max”} & Z_i = c^i x, i = 1, 2, ..., p. \\ t.q & x \in D \end{cases} \tag{3.1}$$

Où : $D = S \cap \mathbb{Z}^n, S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}, A \in \mathbb{Z}^{m\times n}, b \in \mathbb{Z}^m, c^1, c^2, ..., c^p \in \mathbb{Z}^n$, sont des vecteurs lignes, et $p \geq 2$, $\mathbb{Z}$ est l'ensemble des entiers relatifs.
On suppose que D est non vide et que S est un polyèdre convexe et borné et l'ensemble de toutes les solutions efficaces entières du problème $(P(D))$ est noté E. Le problème principal est décrit par :

$$(P_E)\begin{cases} \max & \phi(x) = dx \\ t.q & x \in E \end{cases} \tag{3.2}$$

Où d est un vecteur ligne de dimension n de composantes entières.
Le problème relaxé (P_R) est donné par

$$(P_R)\begin{cases} \max & \phi(x) = dx \\ t.q & x \in D \end{cases} \tag{3.3}$$

Soit le problème mono-objectif $(P_i(D)), i \in \{1, 2, ..., p\}$

$$(P_i(D))\begin{cases} \max & Z_i = c^i x \\ t.q & x \in D \end{cases} \tag{3.4}$$

Un problème $(P_i(D))$ peut avoir plusieurs solutions optimales, nous rappelons la notion de solution alternative dans la définition suivante :

Définition 3.1. Soit x^* une solution optimale du problème $(P_i(D))$, une solution réalisable $\tilde{x} \in D$ est dite *solution alternative* à x^* si $Z_i(x^*) = Z_i(\tilde{x})$.

Dans l'analyse et la résolution des problèmes multi-objectifs, le concept de points efficaces joue un rôle important, rappelons la signification d'un point efficace dans la définition suivante :

Définition 3.2. un point $x \in D$ est dit *solution efficace* du problème $(P(D))$, si et seulement si, il n'existe pas d'autre solution $x' \in D$ tel que $Cx' \geq Cx$ et $Cx' \neq Cx$, et le vecteur critères $Z(x)$ est dit *solution non dominée.*

Soit le problème paramétrique (P_λ^0) :

$$(P_\lambda^0)\begin{cases} \max & Z_\lambda = \displaystyle\sum_{k=1}^{p} \lambda_k Z_k(x) \\ t.q & x \in D. \end{cases} \tag{3.5}$$

la condition nécessaire du théorème de Geoffrion est une conséquence directe de l'efficacité d'une solution optimale de (P_λ^0) où les coefficients de pondération $\lambda_k > 0$, $\forall k \in \{1, 2, ..., p\}$.

Théorème 3.1. *[25].Si $\bar{x}$ une solution optimale du problème uni-critère (P_λ^0) pour un certain $\lambda > 0$ alors, $\bar{x}$ est une solution efficace du problème multi-objectifs $(P(D))$.*

Comme D n'est pas convexe, la réciproque de ce théorème n'est pas vraie pour les problèmes $MOILP$, ceci car certaines solutins efficaces peuvent ne pas être obtenues pour aucun $\lambda > 0$.

3.2 Notations et résultats de base.

On utilisera tout au long de ce chapitre les notations suivantes.
- $Z_1, Z_2, ..., Z_p$ sont les critères initiaux et ϕ est le critère principal à optimiser ;
- X_{opt} est la solution optimale du problème (P_E).
- $\phi_{opt} = dX_{opt}$ est la valeur optimale du critère ϕ.
- $Z_i(X_{opt})$ est l'évaluation du critère i en la solution X_{opt}.
- $H^0 = D = S \cap \mathbb{Z}^n$.
- $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m, x \in \mathbb{Z}^n$.

Pour $k \geq 1$ on a :
- I_k : l'ensemble des indices de base du tableau optimale de l'itération k.
- N_k : l'ensemble des indices hors-base du tableau optimale de l'itération k.

$$H^k = H^{k-1} \cap \left\{ \begin{array}{ll} x \in D | Z_i(x) \geq (Z_i(X_{opt}) + 1)y_i^k - M_i(1 - y_i^k) & (*) \\ \text{pour } i = 1, 2, ..., p. & \\ \sum_{i=1}^{p} y_i^k \geq 1, y_i^k \in \{0, 1\}. & (**) \end{array} \right\}$$

où $-M_i$ est la borne inférieure de la $i^{\text{ème}}$ fonction objectif dans le domaine D et à chaque critère Z_i nous associons une variable binaire y_i^k définie par :

$$y_i^k = \begin{cases} 1 & \text{si le critère } Z_i \text{ est strictement amélioré par rapport à } Z_i(X_{opt}). \\ 0 & sinon. \end{cases}$$

Remarque. Dans le cas où $y_i^k = 0$, la contrainte $(*)$ donne $Z_i(x) \geq -M_i$, qui n'est pas restrictive et la contrainte $(**)$ veut dire qu'au moins un des critères est amélioré.

3.3 Test d'efficacité

Dans les procédures présentées, on est appelé à tester l'efficacité d'une solution réalisable, nous avons jugé utile de montrer que le résultat déjà appliqué pour le cas des variables de décision continues [14], est vrai aussi pour le cas des variables discrètes, son utilisation est très simple, à chaque fois qu'on veut tester l'efficacité d'une solution donnée, on résout un problème de programmation linéaire en variables mixtes inchangé dans tous ces paramètres sauf le vecteur second membre des contraintes, qu'il faut le remplacer par la solution en question, le théorème suivant est utilisé dans les différentes étapes des algorithmes pour tester l'efficacité d'une solution réalisable donnée.

Théorème 3.2. *Soit X^* un élément arbitraire de la région D. $X^* \in E(P)$ si et seulement si, Θ^*, la valeur optimale de la fonction objectif, est nulle dans le problème de programmation linéaire mixte suivant :*

$$P(X^*) \begin{cases} \max & \Theta = \sum_{i=1}^{p} \psi_i \\ \textit{tel que} & Cx - I\Psi = CX^* \\ & x \in D, \ \psi_i \in \mathbb{R}^+; \forall i = \overline{1,p}. \end{cases} \tag{3.6}$$

où C est une matrice $p \times n$, dont la $i^{\text{ème}}$ ligne correspond à c^i, $i = 1, 2, ..., p$, I est la matrice identité $(p \times p)$ et $\Psi = (\psi_i)_{i=1,...,p}$.

Preuve 3.1.
($\Rightarrow$) *Soit $X^* \in E(P)$. Comme $\psi_i \geq 0$ pour tout $i \in \{1, 2, ..., p\}$, alors $Cx \geq CX^*$ dans $P(X^*)$. Supposons que, $\Theta^* \neq 0$, alors il existe i tel que $\psi_i > 0, I\Psi \geq 0$ et $I\Psi \neq 0$ ce qui implique $Cx > CX^*$. Ainsi, il existe $x \in D$ tel que $Cx \geq CX^*$ et $Cx \neq CX^*$, ce qui est en contradiction avec l'hypothèse selon laquelle X^* est efficace.*
($\Leftarrow$) *Soit $\Theta^* = 0$, supposons que X^* est non efficace, il existe un point $x \in D$ tel que $Cx \geq CX^*$ et $Cx \neq CX^*$. Par conséquent, il existe $x \in S$ tel que $Cx - CX^* \geq 0$ et $Cx - CX^* \neq 0$, d'où $I\Psi > 0$, et donc, il existe i tel que $\psi_i > 0$ qui est en contradiction avec l'hypothèse $\Theta^* = 0$.* □

3.4 Coupes de type I

L'algorithme proposé par D.Chaabane est basé principalement sur l'exploration des arêtes incidentes à une solution trouvée et l'utilisation de coupes éliminant une arête contenant des solutions admissibles au lieu d'un seul point admissible.

Proposition 3.1. *Soit X_k une solution optimale de $(P_1(D_k))$. Supposons que $j_k \in \mathrm{N}_k$. une arête E_{j_k} incidente à la solution X_k est définie par l'ensemble*

$$E_{j_k} = \left\{ \quad (x_i \in D_k) \left| \begin{array}{l} x_i = x_{k,i} - \theta_{j_k} y_{k,ij_k} \textit{ pour } i \in I_k \\ x_{jk} = \theta_{j_k} \\ x_\alpha = 0, \forall \alpha \in N_k \backslash \{j_k\} \end{array} \right. \right\} \tag{3.7}$$

où D_k est la région tronquée dans l'itération k, $x_{k,i}$ est la $i^{\text{ème}}$ composante de la solution X_k, y_{k,ij_k} est l'élément de la ligne i et la colonne j_k dans le tableau optimale de X_k. θ_{j_k} est un entier positif tel que $0 < \theta_{j_k} \leq \min\limits_{i \in I_k} \{ \dfrac{x_{k,i}}{y_{k,ij_k}}; y_{k,ij_k} > 0\}$ et $\theta_{j_k} \times y_{k,ij_k}$ est un entier $\forall i \in I_k$ si de telles valeurs de θ_{j_k} existent.

Le théorème suivant suggère une coupe qui peut être considéré comme une généralisation de la coupe de Dantzig (voir (2.1)) dite **coupe de type I**

Théorème 3.3. *[18].Une solution réalisable entière du problème $(P_1(D_k))$ qui n'est pas sur l'arête E_{j_k} et incidente à X_k de la région tronquée D_k, se situe dans le demi espace fermé*

$$\sum_{j \in N_k \backslash \{j_k\}} x_j \geq 1, \quad k \geq 1 \tag{3.8}$$

L'avantage d'utiliser cette coupe est de tronquer toutes les solutions réalisables entières du probléme $P_1(D_k)$ qui se trouvent sur l'arête E_{j_k} issue de la solution optimale X_k, tandis que, la coupe de Dantzig ne tronque qu'un seul point du domaine d'admissibilité à savoir X_k.

3.5 Coupes de type II

Après avoir trouvé une nouvelle solution efficace de $(P(D))$, qui améliore la meilleure valeur de ϕ obtenue jusqu'à present notée ϕ_{opt}, nous imposons une coupe dite **coupe de type II** pour chercher une nouvelle solution efficace dont la valeur ϕ est meilleure ou égale à ϕ_{opt}.

$$dx \geq \phi_{opt}. \tag{3.9}$$

3.6 Résolution du problème dans l'espace des critères.

3.6.1 La méthode de D.Chaabane.[7]

L'auteur a dévelpppé une méthode pour résoudre le problème (P_E) dans l'éspace des critères, dont la fonction objectif est donné par une somme pondéré des critères initiaux. Le principe de la méthode est le suivant :
Dans une première étape la condition nécessaire du theorème de Geoffrion est utilisée pour déterminer une solution efficace initiale en résolevant le problème paramétrique (P_λ^0), en suite, dans chaque itération k, la region d'admissibilité est réduite en ajoutant des contraintes éliminant les solutions dominées par la solution efficace courante, ainsi qu'une contrainte plus stricte que la coupe (3.9) pour éliminer toutes les solutions moins bonnes sur le critère principal en résolvant le problème (P_λ^k) défini par :

$$(P_\lambda^k)\begin{cases} \max \quad Z_\lambda = \sum_{k=1}^{p} \lambda_k Z_k(x) \\ t.q \quad x \in D^k \end{cases} \tag{3.10}$$

où $\lambda > 0$ pour tout $\{i = 1, 2, ..., p\}$ et D^k est le domaine réduit défini par

$$D^k = \big(D \backslash (\bigcup_{\upsilon=0}^{k-1} \Delta_\upsilon)\big) \bigcap D_{X_{opt}}. \tag{3.11}$$

Où $D_{X_{opt}} = \{x \in D | dx \geq \phi_{opt} + 1\}$.
$\Delta = \{x \in \mathbb{Z}^n | Cx \leq CX_\upsilon\}$; avec $X_0, X_1, ..., X_{k-1}$ sont les solutions efficaces obtenues en résolvant les problèmes $(P_\lambda^0), (P_\lambda^1), ..., (P_\lambda^{k-1})$ respectivement.
Ceci peut être traduit mathématiquement par les contraintes additionnelles suivantes :

$$H^k = H^{k-1} \cap \left\{ \begin{array}{l} x \in D | Z_i(x) \geq (Z_i(X_{opt}) + 1) y_i^k - M_i(1 - y_i^k) \\ \text{pour } i = 1, 2, ..., p. \\ \sum_{i=1}^{p} y_i^k \geq 1, y_i^k \in \{0, 1\}. \end{array} \right\}$$

Si une solution optimale est trouvée non efficace, une exploration des arêtes incidentes est faite pour déterminer une nouvelle solution efficace meilleure sur le critère principal, si aucune solution efficace n'est trouvée sur les arêtes incidentes, on choisit une arête contenant le plus grand nombre possible de solutions admissibles et on la coupe de la région courante en utilisant une coupe de **type I**. La region d'admissibilité est réduite d'une itération à l'autre jusqu'à ce qu'elle devient vide.

(a) L'algorithme de la méthode.

Étape 1 : Résoudre le problème relaxé (P_R), la solution obtenue X^* est testée pour l'efficacité en résolvant le problème $P(X^*)$ (voir 3.6).

- Si X^* est efficace, alors la procedure prend fin et X^*est une solution optimale de (P_E).
- Sinon, aller à l'etape 2.

Étape 2 : poser $k = 0$, $H^0 = D$ et résoudre le problème (P_λ^0) défini précédemment . Soit X_0 une solution optimale de (P_λ^0), cette solution est efficace selon la condition nécéssaire d'efficacité du théorème de Geoffrion.
Faire $Sol_{eff} := Sol_{eff} \cup \{X_0\}, X_{opt} = X_0 \ et \ \phi_{opt} = dX_0$, et aller à l'étape 3.

Étape 3 : Faire $k = k + 1$, et résoudre le problème (P_λ^k) défini par

$$(P_\lambda^k) \begin{cases} \max \quad Z_\lambda = \displaystyle\sum_{i=1}^{p} \lambda_i Z_i(x) \\ t.q \quad x \in D^k \end{cases}$$

Où $D^k = H^k \cap \{x \in D | dx \geq \phi_{opt} + 1\}$.

- Si $D^k = \emptyset$, aller à l'étape 6.
- Sinon, soit X_k une solution optimale de (P_λ^k).
  - Si X_k est efficace, alors $Sol_{eff} := \ Sol_{eff} \cup \{X_k\}, \ X_{opt} = X_k, \ \phi_{opt} = dX_k$ et aller à l'étape 3.
  - Sinon , aller à l'etape 4.

Étape 4 : On explore toutes les arêtes E_{jk} incidentes à X_k.
Soit $J_k = \{j \in N_k | Z_{\lambda,j} - c_j = 0\}$.

- Si $J_k = \emptyset$, aller à l'etape 5 (dans ce cas la coupe utilisée dans l'etape 5 devient une coupe de Dantzig).
- Sinon, soit $\gamma = J_k$ et aller à l'etape (4.1)

(4-1) Si $\gamma = \emptyset$, choisir $j_k \in J_k$ et aller à l'étape 5. Sinon, sélectionner $j_k \in J_k$ et calculer $\theta^0_{j_k} = \lfloor \min_{i \in I_k} \{ \frac{x_{k,i}}{y_{k,ij_k}} | y_{k,ij_k} > 0 \} \rfloor$.[1]

- Si $\theta^0_{j_k} = 0$, alors il n'existe aucune solution entière sur l'arête E_{j_k}, poser $\gamma = \gamma - \{j_k\}$ et aller à l'etape (4-1).
- Si $\theta^0_{j_k} \geq 1$, aller à l'etape (4.2).

(4-2) chercher une solution entière efficace sur l'arête E_{j_k} correspondant à θ commençant de $\theta = 1$ jusqu'à $\theta^0_{j_k}$ (θ est un entier positif) et tester pour l'efficacité.

- Si une solution efficace entière est rencontrée, soit X'_k telle que $dX' > \phi_{opt}$, poser $Sol_{eff} := Sol_{eff} \cup \{X'_k\}, X_{opt} = X'_k$ et $\phi_{opt} = dX'_k$ et aller à l'étape 3.
- S'il n'existe aucune solution efficace entière l'arête E_{jk} poser $\gamma = \gamma \backslash \{j_k\}$ et aller à l'etape (4.1).

Étape 5 : Soit $k = k + 1$.

La nouvelle région tronquée (D^k), est un sous ensemble de D^{k-1} (ou D, si $k = 1$) en appliquant une coupe de **type I** où une coupe de Dantzig et utilisant la méthode dual simplexe et les coupes de Gomory si nécessaire pour obtenir une nouvelle solution optimale X_k. Si cette solution est efficace (faire le test) et $dX_k > \phi_{opt}$, faire $Sol_{eff} := Sol_{eff} \cup \{X_k\}, X_{opt} = X_k$, $\phi_{opt} = dX_k$ et aller à l'étape 3. Sinon aller à l'étape 4.

Étape 6 : (Étape finale) La solution optimale est donc X_{opt} et la valeur de ϕ correspondante est ϕ_{opt} et la liste des solutions efficaces parcourues améliorant ϕ est Sol_{eff} (un sous-ensemble de l'ensemble des solutions efficaces de $P(D)$ qui peut intéresser le décideur).

(b) Illustration numérique :

Considérons l'exemple suivant

$$(P(D)) \begin{cases} \max & Z_1 = x_1 - 3x_2 \\ \max & Z_2 = x_1 + 3x_2 \\ t.q. & x_1 + 2x_2 \leq 8, \\ & 2x_1 + x_2 \leq 7, \\ & x_1 - 2x_2 \leq 1, \\ & x_1, x_2 \in \mathbb{N}^+ \end{cases}$$

[1] $\lfloor \alpha \rfloor$: est le plus grand entier inférieur ou égal à α.

Le problème principal est

$$(P_E)\begin{cases} \max & \phi = -3x_1 - 2x_2 \\ t.q. & x_1, x_2 \in E. \end{cases}$$

On prend les bornes inférieures des deux fonctions objectifs $-M_1 = -12, -M_2 = 0$ et $\lambda_1 = \frac{1}{2}, \lambda_2 = \frac{1}{2}$, les poids de la somme pondérée $Z_\lambda = \frac{1}{2}Z_1 + \frac{1}{2}Z_2 = x_1$.

Étape 1 : la solution optimale du problème relaxé (P_R) est $(0,0)$ qui n'est pas efficace.

Étape 2 : Posons $k = 0$, $H^0 = D$ et cherchons une première solution efficace en résolvant le problème (P_λ^0)

$$(P_\lambda^0)\begin{cases} Max & Z_\lambda = x_1 \\ t.q & (x_1, x_2) \in H^0 \end{cases}$$

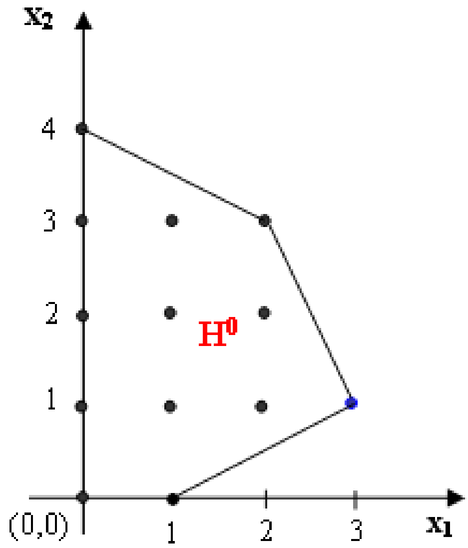


FIG. 3.1 – Région d'admisibilité H^0

$X_0 = (3,1)'$ est une solution optimale de (P_λ^0) dont le vecteur critères correspondant est $Z(X_0) = (0,6)$, elle est efficace d'après le théorème de Geoffrion, on la considère comme première solution efficace et on pose $Sol_{eff} := \{(3,1)'\}, X_{opt} = (3,1)'$ et $\phi_{opt} = dX_{opt} = -11$.

Étape 3 : Faire $k = k + 1 = 1$ et formuler le problème (P_λ^1) augmenté par l'addition des contraintes suivantes :

Pour $i = 1$, on a

$$
\begin{aligned}
Z_1(x) \geq (Z_1(X_{opt}) + 1)y_1^1 - M_1(1 - y_1^1) &\Leftrightarrow x_1 - 3x_2 \geq (0+1)y_1^1 - 12(1 - y_1^1) \\
&\Leftrightarrow x_1 - 3x_2 \geq -12 + 13y_1^1 \\
&\Leftrightarrow -x_1 + 3x_2 + 13y_1^1 \leq 12.
\end{aligned}
$$

Pour $i = 2$, on a :

$$
\begin{aligned}
Z_2(x) \geq (Z_2(X_{opt}) + 1)y_2^1 - M_2(1 - y_2^1) &\Leftrightarrow x_1 + 3x_2 \geq (6+1)y_2^1 - 0(1 - y_2^1) \\
&\Leftrightarrow x_1 + 3x_2 \geq 7y_2^1 \\
&\Leftrightarrow -x_1 - 3x_2 + 7y_2^1 \leq 0.
\end{aligned}
$$

avec :

$$\sum_{i=1}^{2} y_i^1 \geq 1 \quad \Leftrightarrow y_1^1 + y_2^1 \geq 1, y_i^1 \in \{0, 1\}.$$

Ainsi que la contrainte :

$$
\begin{aligned}
dx \geq \phi_{opt} + 1 \quad &\Leftrightarrow -3x_1 - 2_2 \geq -11 + 1 \\
&\Leftrightarrow -3x_1 - 2_2 \geq -10.
\end{aligned}
$$

le nouveau domaine réduit est donné par

$$H^1 = H^0 \cap \left\{ \quad x \in D \;\middle|\; \begin{array}{l} -x_1 + 3x_2 + 13y_1^1 \leq 12 \\ -x_1 - 3x_2 + 7y_2^1 \leq 0 \\ y_1^1 + y_2^1 \geq 1 \\ (y_1^1, y_2^1) \in \{0, 1\}. \end{array} \right\}.$$

On a donc le problème

$$(P_\lambda^1) \begin{cases} Max & Z_\lambda = x_1 \\ t.q. & (x_1, x_2) \in H^1 \\ & -3x_1 - 2_2 \geq -10. \end{cases}$$

$X_1 = (2, 2)', (y_1^1, y_2^1) = (1, 1)$ est une solution optimale et le vecteur critères correspondant est $Z(X_1) = (-4, 8)$, on teste son efficacité par la résolution du problème $P(X_1)$ défini par

$$P(X_1) \begin{cases} Max & \Theta = \psi_1 + \psi_2 \\ t.q. & x_1 + 2x_2 \leq 8, \\ & 2x_1 + x_2 \leq 7, \\ & x_1 - 2x_2 \leq 1, \\ & x_1 - 3x_2 - \psi_1 = -4, \\ & x_1 + 3x_2 - \psi_2 = 8, \\ & (x_1, x_2) \in \mathbb{N}^+, \psi_i \in \mathbb{R}^+, i = 1, 2. \end{cases}$$

Nous obtenons $\psi_1 = \psi_2 = 0, \Theta^* = 0$, donc elle est efficace et comme $dX_1 > \phi_{opt}$, on pose $Sol_{eff} := Sol_{eff} \cup \{X_1\}, X_{opt} = X_1 = (2,2)'$ et $\phi_{opt} = dX_1 = -10$.

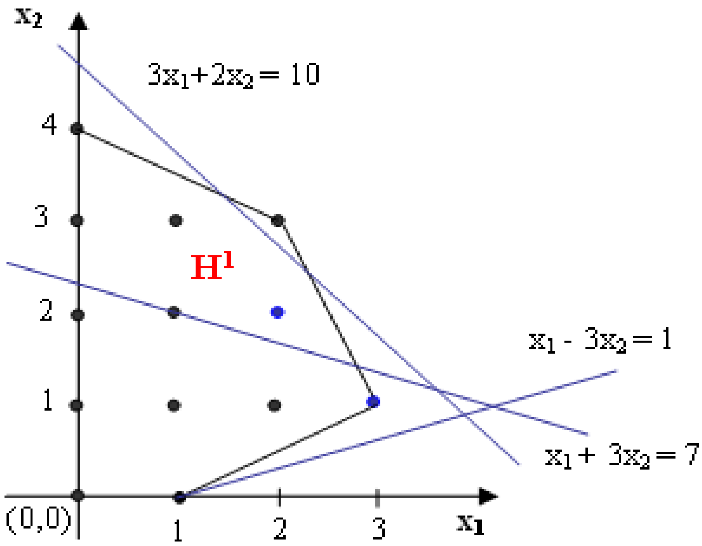


Fig. 3.2 – La région réduite H^1

Étape 4 : Soit $k = k+1 = 2$, de la même façon qu'on a déterminé H^1, on définit H^2 par

$$H^2 = H^1 \cap \left\{ \quad x \in D \left| \begin{array}{l} -x_1 + 3x_2 + 9y_1^2 \leq 12 \\ -x_1 - 3x_2 + 9y_2^2 \leq 0 \\ y_1^2 + y_2^2 \geq 1 \\ (y_1^2, y_2^2) \in \{0,1\}. \end{array} \right. \right\}.$$

le problème (P_λ^2) est

$$(P_\lambda^2) \begin{cases} Max & Z_\lambda = x_1 \\ t.q. & (x_1, x_2) \in H^2 \\ & -3x_1 - 2x_2 \geq -9. \end{cases}$$

Soit $X_2 = (1,0)'$ une solution optimale de (P_λ^2), le vecteur critères correspondant est $Z(X_2) = (1,1)$, on teste son efficacité (résoudre $P(X_2)$), on trouve qu'elle est efficace et $dX_2 = -3 > \phi_{opt}$, on pose alors $Sol_{eff} := Sol_{eff} \cup \{X_2\}, X_{opt} = X_2 = (1,0)'$ et $\phi_{opt} = dX_2$.

Étape 5 : Soit $k = 3$, le problème (P_λ^3) est

$$(P_\lambda^3) \begin{cases} Max & Z_\lambda = x_1 \\ t.q. & (x_1, x_2) \in H^3 \\ & -3x_1 - 2x_2 \geq -2. \end{cases}$$

Où

$$H^3 = H^2 \cap \left\{ \quad x \in D \quad \middle| \quad \begin{array}{l} -x_1 + 3x_2 + 14y_1^3 \leq 12 \\ -x_1 - 3x_2 + 2y_2^3 \leq 0 \\ y_1^3 + y_2^3 \geq 1 \\ (y_1^3, y_2^3) \in \{0, 1\}. \end{array} \right\}.$$

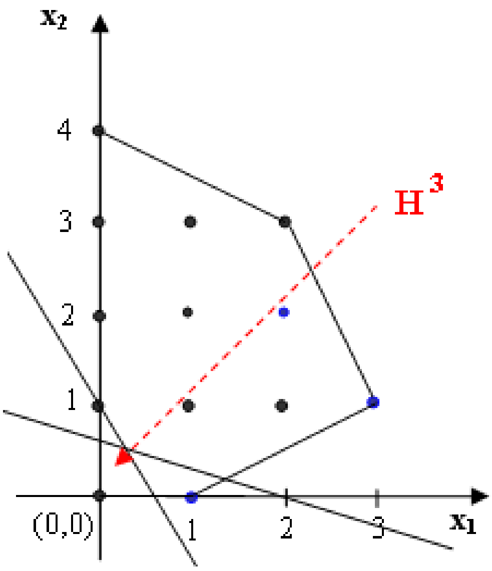


Fig. 3.3 – La région réduite $H^3 \cup \mathbb{Z}^2 = \emptyset$

Le problème devient impossible, la procedure prend fin avec $X_{opt} = (1, 0)'$, $\phi_{opt} = -3$ et le sous-ensemble des solution efficaces examinées est $Sol_{eff} = \{(3, 1)', (2, 2)', (1, 0)'\}$. Cette méthode construit un ensemble notée Sol_{eff} contient toutes les solutions efficaces rencontrées lors de l'optimisation du critère ϕ, elle est efficace dans le sens où elle peut fournir un court chemin pour atteindre la solution optimale de (P_E) sans devoir énumérer toutes les solutions efficaces, mais dans chaque itération il y a $(2p + 1)$ contraintes à ajouter, ce qui cause une explosion en terme de contraintes surtôt dans les problèmes de grande taille.

3.6.2 La méthode de Jesús[19]

La méthode est proposée pour produire une solution optimale globale de (P_E) sans devoir énumérer l'ensemble de toutes les solutions efficaces, la procédure commence à résoudre le problème relaxé (P_R). Évidemment, seulement dans un nombre limité de cas spéciaux la solution optimale de (P_R) fournit une solution optimale de (P_E). Ainsi, si ce n'était pas le cas, une solution efficace domine la solution optimale de (P_R) est alors obtenue. Par suite, dans chaque itération, le critère principal est optimisé sur le domaine restreint par des contraintes en nombres entiers qui sont inclues progressivement pour éléminer les solutions dominées par la solution efficace courante, afin de fournir une solution non dominée par les solutions détectées antérieurement jusqu'à ce qu'une solution optimale soit finalement trouvée.

(a) Formulation de l'algorithme :

Étape 0 : (initialisation)

Poser $\phi_{inf} = -\infty, \phi_{sup} = +\infty, l = 1$ et résoudre le problème relaxé (P_R)

- Si (P_R) n'est pas réalisable. Alors terminer, le problème (P_E) n'a pas de solution.
- Autrement, soit x^l solution optimale de (P_R).

Étape 1 : Tester l'efficacité de x^l.

- Si x^l est efficace, l'algorithme prend fin et $X_{opt} = x^l,\ \phi_{opt} = dx^l$.
- Sinon, poser $\phi_{sup} = dx^l$ et aller à l'étape 2.

Étape 2 :

Soit $\hat{x}^l \in E$ une solution optimale du test d'éfficacité dont le vecteur critères domine celui de x^l.

Dans l'espace des critères, plusieurs solutions efficaces peuvent avoir le même vecteur critères, pour cela le problème (T_l) est résolus pour optimiser le critère principal sur toutes les solutions équivalentes à $\hat{x}^l$.[2]

Le problème (T_l) est défini par

$$(T_l) : \max\{dx | Cx = C\hat{x}^l, x \in D\}. \tag{3.12}$$

soit $\bar{x}^l$ une solution optimale du (T_l)

- Si $d\bar{x}^l > \phi_{inf}$, poser $\phi_{inf} = d\bar{x}^l$ et $X_{opt} = \bar{x}^l$.

[2]Les solutions ayant le même vecteur critères.

- Si $\phi_{inf} = \phi_{sup}$. Terminer, X_{opt} est la solution optimale de (P_E).

Étape 3 : Résoudre le problème (P_l) défini par

$$(P_l) : \max\{dx | x \in D - \bigcup_{s=1}^{l} D_s\}. \tag{3.13}$$

où $D^s = \{x \in \mathbb{Z}^n | Cx \leq C\bar{x}^s\}$; avec $\bar{x}^1, \bar{x}^2, ..., \bar{x}^l$ sont les solutions optimales des problèmes $(T_1), (T_2), ..., (T_l)$ respectivement.

- Si (P_l) n'est pas réalisable. Terminer, X_{opt} est une solution optimale de (P_E).
- Autrement, soit x^{l+1} solution optimale de (P_l).

▷ Si $dx^{l+1} \leq \phi_{inf}$. Terminer, X_{opt} est une solution optimale de (P_E).

▷ Sinon, poser $l = l + 1$ et aller à l'étape 1.

Proposition 3.2. *Soit x^{l+1} une solution optimale du problème (P_l) telle que $\phi(x^{l+1}) > \max_{s \in \{1,...,l\}} \{\phi(\bar{x}^s)\}$. Si $x^{l+1} \in E$ alors x^{l+1} est une solution optimale de (P_E).*

Preuve :

Supposons que x^{l+1} n'est pas optimale pour le problème (P_E), alors, il existe une solution réalisable $\hat{x} \in E$ telle que $\phi(\hat{x}) > \phi(x^{l+1})$. Comme x^{l+1} solution optimale de (P_l) alors $\hat{x} \in \bigcup_{s=1}^{l} D_s$. Ceci implique que, $\exists s \in \{1, ..., l\}$ tel que $\hat{x} \in D_s$, C'est-à-dire, $C\hat{x} \leq C\bar{x}^s$ car $D_s = \{x \in \mathbb{Z}^n / Cx \leq C\bar{x}^s\}$.
Comme $\hat{x} \in E$ nous aurons certainement $C\hat{x} = C\bar{x}^s$ ($\hat{x}$ solution équivalente à $\bar{x}^s$) donc $\phi(\hat{x}) \leq \phi(\bar{x}^s)$ car $\bar{x}^s$ est une solution optimale du problème (T_l).
par conséquence, $\phi(x^{l+1}) < \phi(\hat{x}) \leq \phi(\bar{x}^s)$ qui est en contradiction avec l'hypothèse $\phi(x^{l+1}) > \max_{s \in \{1,...,l\}} \{\phi(\bar{x}^s)\}$. □

Proposition 3.3. *Soient $\bar{x}^1, ..., \bar{x}^l \in E$, si (P_l) est irréalisable, alors l'ensemble de toutes les solutions non dominées est* $\mathrm{Z}(E) = \{C\bar{x}^1, ..., C\bar{x}^l\}$.

Preuve :

$(\supseteq)$ est vérifiée par hypothèse.

$(\subseteq)$ Si (P_l) n'est pas réalisable, alors $D \subseteq \bigcup_{s=1}^{l} D_s$, donc $\forall x \in D, \exists s \in \{1, ..., l\}$ tel que $x \in D_s$. Particulièrement, soit $\bar{x} \in E$, il existe $s \in \{1, ..., l\}$ tel que $C\bar{x}^s \geq C\bar{x}$, ce qui implique $C\bar{x}^s = C\bar{x}$ car $\bar{x} \in E$. □

(b) Illustration numérique :

Considérons l'exemple suivant

$$(P(D))\begin{cases} \max & Z_1 = x_1 - 2x_2 \\ \max & Z_2 = -x_1 + 4x_2 \\ t.q. & -2x_1 + x_2 \leq 0, \\ & x_1 \leq 3, \\ & x_2 \leq 2, \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

Le problème principal est

$$(P_E)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & x_1, x_2 \in E. \end{cases}$$

On prend les bornes inférieures des deux fonctions objectifs $-M_1 = -3, -M_2 = -3$

Itération 1

Étape 0 : Poser $\phi_{inf} = -\infty, \phi_{sup} = +\infty, l = 1.$

Résoudre le problème relaxé

$$(P_R)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & x_1, x_2 \in D. \end{cases}$$

soit $x^1 = (0,0)$ une solution optimale de (P_R) dont le vecteur critères $Z(x^1) = (0,0)$.

Étape 1 : tester l'efficacité de x^1 en résolvant le problème

$$P(x^1)\begin{cases} \max & \Theta = \psi_1 + \psi_2 \\ t.q. & (x_1, x_2) \in D. \\ & x_1 - 2x_2 - \psi_1 = 0 \\ & -x_1 + 4x_2 - \psi_2 = 0 \\ & \psi_i \geq 0, i = 1, 2. \end{cases}$$

$\Theta^* \neq 0$ donc x^1 n'est pas efficace, poser $\phi_{sup} = dx^1 = 0$ et aller à l'etape 2.

Étape 2 : Soit $\hat{x}^1 = (2,1)$ une solution optimale de $P(x^1)$, qui est efficace, le vecteur critères correspondant est $Z(\hat{x}^1) = (0,2)$.

Résoudre le problème

$$(T_1)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & (x_1, x_2) \in D. \\ & x_1 - 2x_2 = 0 \\ & -x_1 + 4x_2 = 2 \end{cases}$$

La solution optimale de (T_1) est $\bar{x}^1 = \hat{x}^1 = (2,1)$ et comme $\phi(\bar{x}^1) = -4 > \phi_{inf} = -\infty$, poser $\phi_{inf} = -4$ et $X_{opt} = \bar{x}^1$.

$\phi_{inf} \neq \phi_{sup}$, aller à l'etape 3.

Étape 3 :

Résoudre le problème (P_1) défini par

$$(P_1)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & (x_1, x_2) \in D; \\ & Z_i(x) \geq (Z_i(X_{opt}) + 1)y_i^1 - M_i(1 - y_i^1); \\ & \sum_{i=1}^{2} y_i^1 \geq 1; y_i^1 \in \{0,1\}, i = 1, 2. \end{cases}$$

qui est formuler par

$$(P_1)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & (x_1, x_2) \in D. \\ & x_1 - 2x_2 \geq y_1^1 - 3(1 - y_1^1) \qquad (1) \\ & -x_1 + 4x_2 \geq 3y_2^1 - 3(1 - y_2^1) \qquad (2) \\ & y_1^1 + y_2^1 \geq 1, (y_1^1, y_2^1) \in \{0,1\}. \end{cases}$$

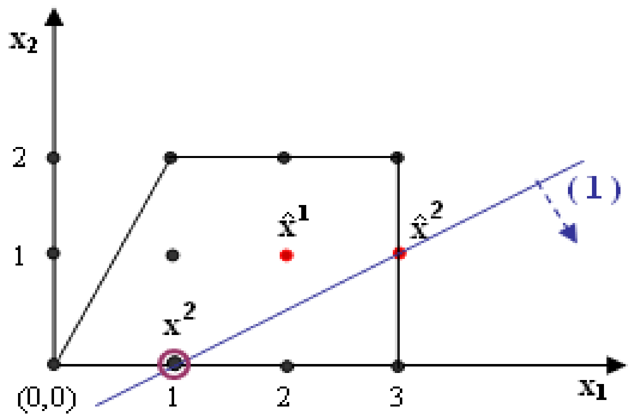


FIG. 3.4 – La région réduite D^1

soit $x^2 = (1,0), y = (1,0)$ une solution optimale de (P_1) et $Z(x^2) = (1,-1)$.

comme $\phi(x^2) = -1 > \phi_{inf}$, poser $l = l+1 = 2$ et aller à l'etape 1.

Itération 2

Étape 1 : La solution x^2 est testée pour l'efficacité en résolvant le problème $P(x^2)$, on trouve qu'elle n'est pas efficace ($\Theta^* \neq 0$), poser $\phi_{sup} = dx^2 = -1$ et aller à l'etape 2.

Étape 2 : $\hat{x}^2 = (3,1)$ une solution optimale de $P(x^2)$ qui est efficace, $Z(\hat{x}^2) = (1,1)$.

Résoudre le problème

$$(T_2)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & (x_1, x_2) \in D. \\ & x_1 - 2x_2 = 1 \\ & -x_1 + 4x_2 = 1 \end{cases}$$

La solution optimale de (T_2) est $\bar{x}^2 = \hat{x}^2 = (3,1)$.

comme $\phi(\bar{x}^2) = -5 < \phi_{inf} = -4$, aller à l'etape 3 sans faire la mise à jour.

Étape 3 : Résoudre le problème (P_2)

$$(P_2)\begin{cases} \max & \phi = -x_1 - 2x_2 \\ t.q. & x_1, x_2 \in D. \\ & x_1 - 2x_2 \geq y_1^1 - 3(1 - y_1^1) \\ & -x_1 + 4x_2 \geq 3y_2^1 - 3(1 - y_2^1) \\ & y_1^1 + y_2^1 \geq 1, (y_1^1, y_2^1) \in \{0,1\} \\ & x_1 - 2x_2 \geq 2y_1^2 - 3(1 - y_1^2) \qquad (3) \\ & -x_1 + 4x_2 \geq 2y_2^2 - 3(1 - y_2^2) \qquad (4) \\ & y_1^2 + y_2^2 \geq 1, (y_1^2, y_2^2) \in \{0,1\} \end{cases}$$

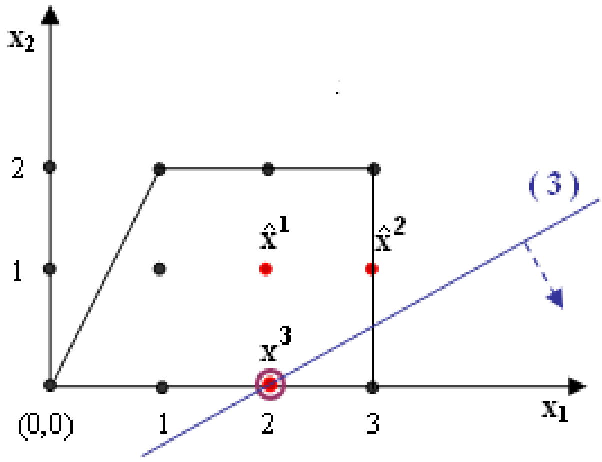


FIG. 3.5 – La région réduite D^2

$x^3 = (2,0), y = (1,0,1,0)$ une solution optimale de (P_2) et $Z(x^3) = (2,-2)$.

$\phi(x^3) = -2 > \phi_{inf} = -4$, poser $l = l + 1 = 3$ et aller à l'etape 1.

Itération 3

La solution x^3 est testée pour l'efficacité en résoulevant le problème $P(x^3)$, on trouve qu'elle est efficace($\Theta^* = 0$), donc l'algorithme prend fin.

$X_{opt} = (2,0)$ solution optimale de (P_E) et $\phi_{opt} = -2$ est la valeur optimale du critère parincipal.

3.7 Une méthode de résolution proposée : Méthode ODSE

"Optimisation Discrète sur l'ensemble des Solutions Efficaces"

La méthode proposée est une combinaison entre les principes utilisés dans les deux algorithmes présentés précédemment, le principe de la méthode est le suivant :

Dans une première étape, on cherche une solution efficace initiale, on part de la solution optimale du problème relaxé (P_R), si elle existe, cette solution est testé pour l'efficacité, pour voir si par hasard, on peut résoudre le problème (P_E) juste en résolvant son problème relaxé, si ce n'etait pas le cas, la solution optimale de test d'efficacité, soit $\hat{x}$, est utilisée pour optimiser le critère principal sur les solutions admissibles ayant un même vecteur critères en résolvant le problème (T_l) (voir (3.12)), la solution optimale de ce problème, soit $\bar{x}$, est considérée comme une première solution efficace, on initialise alors $X_{opt} = \bar{x}$ et $\phi_{opt} = d\bar{x}$.

Ensuite, dans chaque itération, le domaine de recherche est réduit progressivement par des contraintes éliminant toutes les solutions dominées par la solution $\bar{x}$ en résolvant le problème (P_l) (voir (3.13)), la solution optimale trouvée, soit x^l, produit une valeur maximum du critère ϕ dans le domaine réduit, si elle est efficace, la procédure se termine avec x^l une solution optimale du problème, sinon, en utilisant le tableau optimal correspodant, une exploration des arêtes incidentes à cette solution est réalisée pour chercher une solution entière alternative efficace. Si une telle solution existe, elle est la solution optimale de (P_E), sinon, l'algorithme va poursuivre par une alternance des étapes de réduction et d'exploration jusqu'à ce qu'une solution améliore la valeur de ϕ soit trouvée où le domaine de recherche devient vide.

Dans la suite, nous donnons une déscription formelle de l'algorithme.

(a) L'algorithme de la méthode

Étape 0 : Résoudre le problème relaxé (P_R).

- Si (P_R) est irréalisable, alors la procédure prend fin, le problème princial (P_E) n'a pas de solution.
- Sinon, soit x^0 une solution optimale de (P_R).

Étape 1 : Poser $l = 0, \phi_{opt} = -\infty, H^0 = D$.

- Si x^0 est efficace ; terminer, x^0 est une solution optimale de (P_E).
- Sinon, soit $\hat{x}^0$ solution efficace dont le vecteur critères correspond domine celui de x^0(solution optimale du test d'effecacité de x^0).

Étape 2 : Résoudre le problème (T_l) défini par

$$(T_l)\begin{cases} \max & \phi(x) \\ t.q & x \in D. \\ & Cx = C\hat{x}^l \end{cases}$$

Soit $\bar{x}^l$ une solution optimale de (T_l).

- Si $\phi(\bar{x}^l) > \phi_{opt}$; poser $X_{opt} = \bar{x}^l, \phi_{opt} = d\bar{x}^l$ et aller à l'étape 3.

Étape 3 : Poser $l = l + 1$ et résoudre le problème (P_l) défini par

$$(P_l)\begin{cases} \max & \phi(x) \\ t.q & x \in H^l. \end{cases}$$

$$H^l = H^{l-1} \cap \left\{ \begin{array}{l} x \in D | Z_i(x) \geq (Z_i(X_{opt}) + 1)y_i^l - M_i(1 - y_i^l) \\ \text{pour } i = 1, 2, ..., p. \\ \sum_{i=1}^{p} y_i^l \geq 1, y_i^l \in \{0, 1\}. \end{array} \right\}$$

- Si $H^l = \emptyset$; terminer, X_{opt} est la solution optimale de (P_E).
- Sinon ; soit x^l solution optimale de (P_l).
  - Si $\phi(x^l) \leq \phi_{opt}$; terminer, X_{opt} est la solution optimale de (P_E).
  - Sinon,
    - ▷ Si x^l est efficace ; terminer $X_{opt} = x^l, \phi_{opt} = dx^l$.
    - ▷ Sinon, soit $\hat{x}^l$ solution efficace domine x^l(solution optimale du teste d'effecacité de x^l) ; aller à l'étape 4.

Étape 4 : Construire $J_l = \{j \in N_l | \phi_j - d_j = 0\}$

Où N_l est l'ensemble des indice hors base dans le tableau optimal correspodant à x^l.

(4.1) Si $J_l = \emptyset$, aller à l'étape 2.

(4.2) Sinon, chercher une solution entière efficace x'^l alternative à x^l en explorant les arêtes incidentes. choisir $j_l \in J_l$ et calculer $\theta_{j_l}^0 = \lfloor \min_{i \in I_l} \{ \frac{x_{l,i}}{y_{l,ij_l}}, y_{l,ij_l} > 0 \} \rfloor$.

- Si $\theta_{j_l}^0 = 0$, aucune solution entière ne peut être existe sur l'arête E_{j_l}, poser $J_l = J_l \setminus \{j_l\}$ et aller à (4-1).
- Sinon, chercher une solution entière efficace sur l'arêt E_{j_l} correspondant à $\theta_{j_l}^0$ commençant de $\theta = 1$ jusqu'à $\theta = \theta_{j_l}^0$ (θ est un entier positif) et tester pour l'efficacité.
  - Si $\exists$ x'^l solution entière efficace alternative à x^l ; terminer, $X_{opt} = x'^l, \phi_{opt} = dx'^l$.
  - Sinon, poser $J_l = J_l \setminus \{j_l\}$ aller à l'étape (4.1).

Proposition 3.4. *Soit x'^l une solution alternative à x^l la solution optimale du problème (P_l) avec $\phi(x^l) > \max_{s \in \{1,...,l\}} \{\phi(\bar{x}^s)\}$.*
Si $x'^l \in E$ alors x'^l est une solution optimale de (P_E).

Preuve :

En ajoutant la définition d'une solution alternative (Voir Définition 3.1) à la démonstration de la proposition 3.2, nous obtenons une conséquence directe que si x'^l, une solution alternative à x^l avec $\phi(x^l) > \max\limits_{s\in\{1,\ldots,l\}}\{\phi(\bar{x}^s)\}$, est efficace alors $\phi(x'^l) > \max\limits_{s\in\{1,\ldots,l\}}\{\phi(\bar{x}^s)\}$ car $\phi(x'^l) = \phi(x^l)$, donc x'^l est une solution optimale de (P_E). □

Proposition 3.5. *L'algorithme converge en un nombre fini d'itérations et donne une solution optimale du problème* (P_E).

Preuve :

Par l'hypothèse que la région d'admissibilité S est bornée (il y a un nombre limité de solutions entières) et D est supposé non vide, on aura au moins une solution entière ; dans chaque itération de l'algorithme, où bien une solution efficace améliore la valeur de ϕ est trouvée où bien le domaine de recherche est réduit progréssivement jusqu'à ce qu'il devient vide. □

(b) Illustration numérique : Nous illustrons le fonctionnemnt de l'algorithme par deux exemples didactiques.

Exemple1 : Considérons l'exemple présenté précédement pour illustrer la méthode de D.Chaabane

Étape 0 : Rappelons que les bornes inférieures des deux fonctions objectifs sont $-M_1 = -12, -M_2 = 0$ respectivement, la solution optimale du problème relaxé (P_R) est $x^0 = (0, 0)$.

Étape 1 : $l = 0, \phi_{opt} = -\infty$, $H^0 = D$, la solution x^0 n'est pas efficace, $\hat{x}^0 = (3, 1)$ est la solution obtenue en résolvant le problème de test d'efficacité $P(x^0)$ (voir 3.6).

Étape 2 : Résoudre le problème (T_0) défini par

$$(T_0)\begin{cases} \max & \phi = -3x_1 - 2x_2 \\ t.q. & (x_1, x_2) \in D. \\ & x_1 - 3x_2 = 0 \\ & x_1 + 3x_2 = 6 \end{cases}$$

La solution optimale de (T_0) est $\bar{x}^0 = (3, 1)$, $\phi(\bar{x}^0) = -11 > \phi_{opt}$, initialisons alors $\phi_{opt} = \phi(\bar{x}^0) = -11, X_{opt} = (3, 1)$ et aller à l'étape 3.

Étape 3 : $l = l + 1 = 1$, résoudre le problème (P_1)

$$(P_1)\begin{cases} \max & \phi = -3x_1 - 2x_2 \\ t.q. & x_1 + 2x_2 \leq 8 \\ & 2x_1 + x_2 \leq 7 \\ & x_1 - 2x_2 \leq 1 \\ & x_1 - 3x_2 \geq y_1^1 - 12(1 - y_1^1) \\ & x_1 + 3x_2 \geq 7y_2^1 \\ & y_1^1 + y_2^1 \geq 1 \\ & (x_1, x_2) \in \mathbb{N}^+, (y_1^1, y_2^1) \in \{0, 1\}. \end{cases}$$

La solution optimale de ce problème est $x^1 = (1,0); (y_1^1, y_2^1) = (1,0); Z(x^1) = (1,1)$ et $\phi(x^1) = -3$.

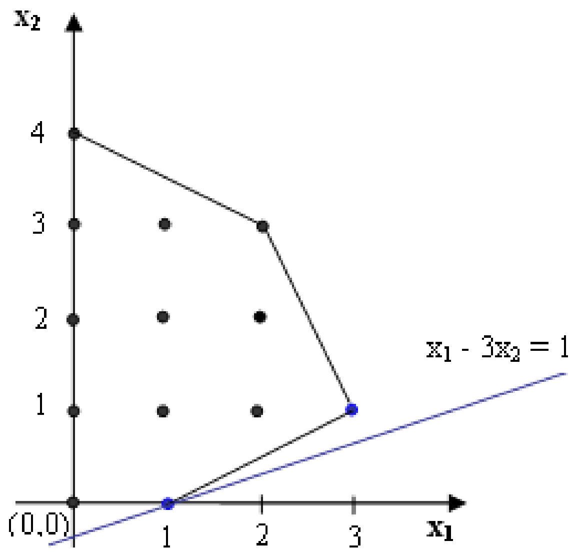


FIG. 3.6 – La région réduite D^1

et comme $\phi(x^1) > \phi_{opt}$, cette solution est testé pour l'éfficacité en résolvant le problème

$$(P(x^1))\begin{cases} \max & \Theta = \psi_1 + \psi_2 \\ t.q. & x_1 + 2x_2 \leq 8, \\ & 2x_1 + x_2 \leq 7, \\ & x_1 - 2x_2 \leq 1, \\ & x_1 - 3x_2 - \psi_1 = 1, \\ & x_1 + 3x_2 - \psi_2 = 1, \\ & (x_1, x_2) \in \mathbb{N}^+, \psi_i \in \mathbb{R}^+, i = 1, 2. \end{cases}$$

nous obtenons $\Theta^* = 0$, donc elle est efficace.

La procédure prend fin avec $X_{opt} = (1, 0)$ et $\phi_{opt} = -3$.

Exemple2 : Considérons l'exemple proposé par J. Philip [24]

$$(P(D)) \begin{cases} \max & Z_1 = 2x_1 - x_2 \\ \max & Z_2 = -x_1 + 2x_2 \\ t.q. & x_1 + 2x_2 \leq 8, \\ & x_1 \leq 5, \\ & x_2 \leq 7, \\ & x_1 + x_2 \leq 10, \\ & x_1, x_2 \in \mathbb{N}^+ \end{cases}$$

Le problème principal est donné par

$$(P_E) \begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & x_1, x_2 \in E. \end{cases}$$

Étape 0 : Résoudre le problème relaxé (P_R)

$$(P_R) \begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & x_1 + 2x_2 \leq 8, \\ & x_1 \leq 5, \\ & x_2 \leq 7, \\ & x_1 + x_2 \leq 10, \\ & x_1, x_2 \in \mathbb{N}^+ \end{cases}$$

La solution optimale de (P_R) est $x^0 = (0, 0)$, les bornes inférieures des deux fonctions objectifs sont $-M_1 = -7, -M_2 = -5$ respectivement.

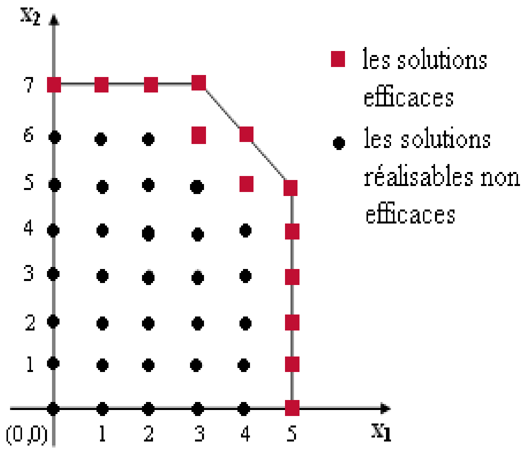


FIG. 3.7 – Le domaine d'admissibilité D

Étape 1 : $l = 0, \phi_{opt} = -\infty$, $H^0 = D$, la solution x^0 n'est pas efficace, $\hat{x}^0 = (5, 5)$ est la solution obtenue en résolvant le problème de test d'efficacité $P(x^0)$ dont le vecteur critères correspondant est $Z(\hat{x}^0) = (5, 5)$.

Étape 2 : Résoudre le problème (T_0)

$$(T_0)\begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D. \\ & 2x_1 - x_2 = 5 \\ & -x_1 + 2x_2 = 5 \end{cases}$$

La solution optimale est $\bar{x}^0 = (5, 5)$, $\phi(\bar{x}^0) = -20 > \phi_{opt}$,

faire alors $X_{opt} = (5, 5), \phi_{opt} = -20$.

Étape 3 : $l = l + 1 = 1$, résoudre le problème (P_1)

$$(P_1)\begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D \\ & 2x_1 - x_2 \geq 6y_1^1 - 7(1 - y_1^1) \qquad (1) \\ & -x_1 + 2x_2 \geq 6y_2^1 - 5(1 - y_2^1) \qquad (2) \\ & y_1^1 + y_2^1 \geq 1 \\ & (x_1, x_2) \in \mathbb{N}^+, (y_1^1, y_2^1) \in \{0, 1\}. \end{cases}$$

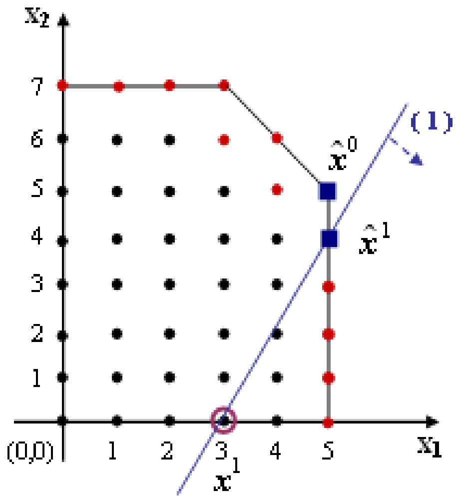


FIG. 3.8 – La région réduite D^1

La solution optimale de ce problème est $x^1 = (3,0), \phi(x^1) = -3$, $Z(x^1) = (6,-3)$ et $y = (1,0)$.

Comme $\phi(x^1) > \phi_{opt}$ cette solution est testé pour l'efficacité en résolvant le problème

$$(P(x^1)) \begin{cases} \max & \Theta = \psi_1 + \psi_2 \\ t.q. & x_1 \leq 5, \\ & x_2 \leq 7, \\ & x_1 + x_2 \leq 10, \\ & 2x_1 - x_2 - \psi_1 = 6, \\ & -x_1 + 2x_2 - \psi_2 = -3, \\ & (x_1, x_2) \in \mathbb{N}^+, \psi_i \in \mathbb{R}^+, i = 1, 2. \end{cases}$$

On obtient $\Theta^* \neq 0$ et $\hat{x}^1 = (5,4)$ solution optimale de $(P(x^1))$ dont le vecteur critères correspondant $Z(\hat{x}^1) = (6,3)$.

Étape 4 : l'ensemble $J_1 = \{j \in N_1 | c_j - d_j = 0\} = \emptyset$ (en utilisant le tableau optimale de x^1), aller alors l'étape 2.

iltération 2

Étape 2 : Résoudre le problème (T_1)

$$(T_1) \begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D. \\ & 2x_1 - x_2 = 6 \\ & -x_1 + 2x_2 = 3 \end{cases}$$

$\bar{x}^1 = (5,4)$, $\phi(\bar{x}^1) = -17 > \phi_{opt}$, faire $X_{opt} = (5,4), \phi_{opt} = -17$.

Étape 3 : $l = l + 1 = 2$, résoudre le problème (P_2)

$$(P_2)\begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D^1 \\ & 2x_1 - x_2 \geq 7y_1^2 - 7(1 - y_1^2) \qquad (3) \\ & -x_1 + 2x_2 \geq 4y_2^2 - 5(1 - y_2^2) \qquad (4) \\ & y_1^2 + y_2^2 \geq 1 \\ & (x_1, x_2) \in \mathbb{N}^+, (y_1^2, y_2^2) \in \{0, 1\}. \end{cases}$$

La solution optimale est $x^2 = (4,0)$ qui n'est pas efficace avec $\phi(x^2) = -4$, $Z(x^2) = (8,-4)$ et $Y = (1,0,1,0)$. Soit $\hat{x}^2 = (5,2), Z(\hat{x}^2) = (8,-1)$ solution optimale du test d'efficacité de x^2.

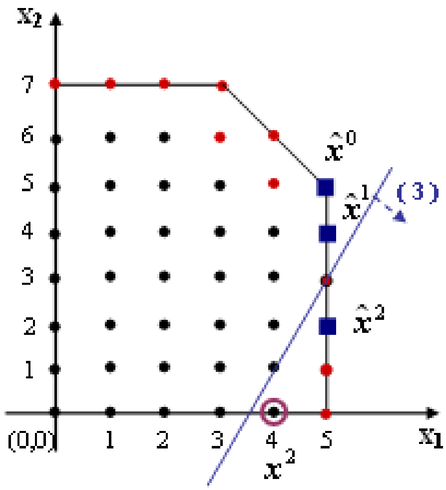


FIG. 3.9 – La région réduite D^2

Étape 4 : l'ensemble $J_2 = \{j \in N_2 | c_j - d_j = 0\} = \emptyset$, aller alors l'étape 2.

iltération 3

Étape 2 : Résoudre le problème (T_2)

$$(T_2)\begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D. \\ & 2x_1 - x_2 = 8 \\ & -x_1 + 2x_2 = -1 \end{cases}$$

$\bar{x}^2 = (5,2), \phi(\bar{x}^2) = -11 > \phi_{opt}$, faire alors $X_{opt} = (5,2), \phi_{opt} = -11$.

Étape 3 $l = l + 1 = 3$, résoudre le problème (P_3)

$$(P_3)\begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D^2 \\ & 2x_1 - x_2 \geq 9y_1^3 - 7(1 - y_1^3) \qquad (5) \\ & -x_1 + 2x_2 \geq -5(1 - y_2^3) \qquad (6) \\ & y_1^3 + y_2^3 \geq 1 \\ & (x_1, x_2) \in \mathbb{N}^+, (y_1^3, y_2^3) \in \{0, 1\}. \end{cases}$$

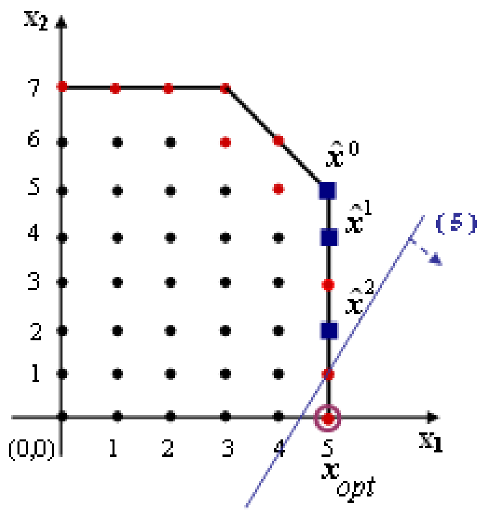


FIG. 3.10 – La région réduite D^3

$x^3 = (5, 0), \phi(x^3) = -5 > \phi_{opt}$, après le test d'efficacité de x^3 on trouve qu'elle est efficace, la procédure prend fin avec $X_{opt} = (5, 0)$ et $\phi_{opt} = -5$.

L'ensemble de toutes les solutions efficaces de ce problème est

$E = \{(0, 7), (1, 7), (2, 7), (3, 7), (3, 6), (4, 6), (4, 5), (5, 5), (5, 4), (5, 3), (5, 2), (5, 1), (5, 0)\}$.

Tandis que l'algorithme proposé optimise le critère ϕ sans devoir passer par toutes ces solutions mais seulement par $E' = \{(5, 5), (5, 4), (5, 2), (5, 0)\}$.

Chapitre 4

Expérimentation et résultats

4.1 Introduction

Dans ce chapitre, nous complétons notre étude par des résultats expérimentaux obtenus lors de l'implémentation de la nouvelle méthode proposée pour l'optimisation d'un critère linéaire sur l'ensemble des solutions efficaces d'un problème MOILP, à l'aide d'un programme qu'on a réalisé en utilisant le logiciel MATLAB 7.0 "**Mat**rix **Lab**oratory" qui est un logiciel de calcul numérique, utilisé dans de nombreux domaines d'application, basé sur le calcul matriciel.

4.2 Implantation et description des codes

4.2.1 Génération aléatoire des instances

Nous avons programmé un algorithme générateur pour construire des instances aléatoires du problème multi-objectifs suivant :

$$\text{“}\max\text{”}\{Cx|Ax \leq b, x \geq 0\}.$$

Où $A \in \mathbb{Z}^{m\times n},\ C \in \mathbb{Z}^{p\times n},\ x \in \mathbb{Z}^{n}$ et $b \in \mathbb{Z}^{m\times 1}$.
Les éléments des matrices $A,\ C\ et\ b$ ainsi que le vecteur de coefficients $d \in \mathbb{Z}^{1\times n}$ du critère principal ϕ sont générer à l'aide d'une fonction prédéfinie en MATLAB 7.0 appelée ***randint*** $(l, k, [\upsilon_{max}, \upsilon_{min}])$
qui renvoie une matrice à l lignes et k colonnes dont les éléments sont des entiers générés

aléatoirement entre v_{max} et v_{min}.

La procédure de ce programme est la suivante :

Algorithm 1 Génération aléatoire des instances.

Entrées :
n : nombre de variables.
p : nombre de critères.
m : nombre de contraintes.
Soties : les matrices $A,\ b,\ C$ et d.
$A \leftarrow \textbf{randint}(m, n, [0, 30])$.
$C \leftarrow \textbf{randint}(p, n, [-20, 20])$.
$d \leftarrow \textbf{randint}(1, n, [-10, 10])$.
pour $i = 1$ jusqu'à m **faire**

$$b_i \leftarrow k \times \sum_{j=1}^{n} A_i^j + \textbf{randint}(1,\ 1,\ [0, 30]).$$

Avec k est fixé à 2 (le domaine possède au moins deux points admissibles).

Fin pour.

4.2.2 Programme implémentant la méthode "ODSE"

La méthode proposée est basée sur l'optimisation d'un critère linéaire sur un ensemble des points entiers, avant d'appeler le programme principal, les données nécessaires : $Type$ = "*max*" , la matrice des critères $C(p \times n)$, la matrice des contraintes $A(m \times n)$, le vecteur second membre $b(m \times 1)$ et le vecteur du critère principal $d(1 \times n)$ sont introduits d'une manière interactive ou à partir d'un fichier "data".

Les principales procédures utilisées sont les suivantes :

- Procédure de simplexe : Cette procédure est appelée au niveau de deux étapes, la prmière étape afin de résoudre le problème relaxé (P_R) du problème principal en maximisant le critère linéaire $\phi(x) = dx$ sur le domaine d'admissibilité, et la deuxième étape pour déterminer une borne inférieure pour chaque critère qu'on va les utiliser dans les itérations ultérieures lorsqu'on ajoute les coupes données dans la section (3.2).
- Algorithme de "Branch & bound" : La méthode proposée est basée sur la résolution des problèmes de programmation linéaire en nombres entiers, pour cela, cet algorithme est appelé dans chaque itération pour déterminer une solution optimale entière.
- Algorithme dual simplexe : Cette procédure est appelée pour résoudre le problème (P_l) qui optimise le critère principal ϕ dans le domaine réduit par des contraintes éliminant toutes les solutions dominées par la solution efficace courante.

- Procédure de test d'efficacité qui consiste à résoudre un problème de programmation linéaire mixte (Voir 3.6) pour tester l'efficacité d'une solution optimale trouvée.
- Procédure consiste à résoudre le problème $T(x)$ pour optimiser le critère principal ϕ sur toutes les solutions équivalentes à la solution efficace courante, si elles existent.
- Procédure de comparaison : dans chaque itération, cette procédure est appelée pour comparer l'évaluation du critère principal ϕ en la solution efficace courante avec la valeur la plus récente de ϕ, puis, si la nouvelle solution efficace parcourue est meilleur que la précédente sur le critère principal, on fait la mise à jour de la solution optimale X_{opt} et la valeur optimale ϕ_{opt}.
- Procédure d'exploration : Si la solution optimale de la région réduite est trouvée non efficace, cette procédure est appelée pour chercher une solution alternative efficace en explorant les arêtes incidentes à cette solution.

4.3 Résultats pour la méthode ODSE

Nous avons réalisé notre implémentation sur une machine Pentium 4 à 2.4 GHZ avec 512 MO de mémoire vive, notre algorithme est testé sur des exemples traités par des chercheurs dans différents articles avec une série des instances générées aléatoirement.

4.3.1 Tests sur quelques exemples traités dans la littérature

Le tableau suivant résume les résultas des tests sur quelques exemples résolus par des chercheurs du domaine, nous présentons pour chaque exemple le nombre totale des solution efficaces suivi par le nombre des solution efficaces parcourues par l'algorithme proposé.

Instances	n	m	p	#itér	durée(sec)	# sol-eff	# sol-eff parcourues
Jesús,M.Jorge[19]	2	3	2	3	0.062	7	3
Ecker,J.G.& Song,H.G.[14]	2	3	2	2	0.031	5	2
Philip,J.[24]	2	3	2	4	0.125	13	4
Bowman,V.J.[5]	3	4	2	1	0.000	3	1
Sylva,J.& Crema,A.[9]	3	4	2	1	0.000	4	1
Sylva,J.& Crema,A.[10]	4	6	2	2	0.030	4	2
Gupta,R.& Malhotra,R.[18]	2	3	3	5	0.149	9	5
Karaivanova,J.N.,al.[20]	3	3	3	5	0.547	14	4
Klein,D.& Hannan,E.[21]	4	6	3	2	0.034	6	2

Tab. 4.1 – Tableau 1.

Où :
∗ $n, m, p(p \geq 2) \in \mathbb{N}^+$: sont réspéctivement le nombre des variables, le nombre des contraintes, le nombre des critères.
∗ #itér : est le nombre d'itérations efféctuées, notons qu'il est égal au nombre de programmes linéaires (P_l) résolus au cours de l'exécution plus l'itération initiale (résolution du problème relaxé).
∗ durée : la durée d'exécution en secondes (*sec*).
∗ # sol-eff : le nombre de solutions efficaces du problème $MOILP$.
∗ # sol-eff parcourues : le nombre des solutions efficaces parcourues au cours de l'exécution.
Nous constatons dans ce tableau que l'algorithme proposé génère une solution optimale exacte sans devoir examiner l'ensemble de tous les points efficaces du problème (qui peut être très grand même pour un problème de petite taille) mais uniquement un sous-ensembles de points efficaces maximisant le critère principal, ce qui réduit le temps d'exécution échoué et le nombre d'itérations effectuées.

4.3.2 Tests sur des exemples générés aléatoirement

Dans cette série d'expérimentations, des instances aléatoires de différentes tailles ont été générés par la procédure décrite ci-dessus, les résultats obtenus pour chaque instance représentent la moyenne sur dix exécutions indépendantes du temps de calcul et du nombre d'itérations efféctuées. Ainsi que la valeur minimale et maximamle de ces deux derniers dans les dix exécutions.
Evidement, les difficultés rencontrés dans la résolution du problème (P_E) sont étroitement liées à leur dimension, le tableau suivant montre clairement que même pour les problèmes de tailles relativement grandes le temps moyen requis par le programme et le nombre d'itérations exigées restent toujours petit par rapport à la taille du problème étudié ce qui confirme l'efficacité de la méthode.

p	$n \times m$	5×5	10×10	20×15	30×15	40×15	50×15
3	durée (sec)	0.89 [0.005; 3]	2.14 [0.02; 6.5]	18.23 [0.2; 97]	25.85 [0.5; 119]	104.04 [0.225; 352.5]	87 [0.105; 707]
	#iter	3.39 [1; 5]	3.23 [1; 6]	3.37 [1; 6]	4.23 [2; 7]	3.69 [2; 7]	4.7 [1; 10]
5	durée (sec)	6.3 [0.015; 43]	9.67 [0.1; 71.5]	49.82 [0.012; 559]	142.75 [0.02; 885.5]	86.38 [0.125; 759]	273.18 [0.4; 1500]
	#iter	3.9 [1; 7]	3.58 [2; 6]	4.38 [2; 9]	4.16 [2; 11]	3.41 [1; 8]	4.8 [1; 10]
8	durée (sec)	1.545 [0.02; 8]	11.28 [0.075; 35.6]	52.56 [0.02; 225]	219 [0.5; 1550]	196 [0.5; 982]	147 [2.02; 658]
	#iter	2.37 [1; 5]	2.76 [1; 6]	3.66 [2; 7]	4.22 [1; 10]	4.34 [1; 11]	3.9 [2; 9]

Tab. 4.2 – Tableau 2.

Nous nous intéressons dans ces résultats, au nombre d'itérations effectuées au cours de l'exécution, nous remarquons dans le tableau ci-dessus que le nombre de solutions efficaces examinées, qui est égal au nombre d'itérations réalisées, est relativement petit par rapport aux tailles des instances traitées. En effet, après l'itération initiale (résolution du problème relaxé), chaque itération réalisée consiste à résoudre un problème (P_l) augmenté de p variables et $(2p + 1)$ contraintes (p étant le nombre de critères) ce qui cause une explosion en terme de temps et de mémoire, notre algorithme de sa part, a l'avantage de produire une solution optimale exacte en évitant de parcourir la totalité de solutions admissibles en utilisant deux techniques : L'optimisation du critère principal dans chaque itération qui nous dirige continuellement vers la solution qui accorde une meilleure valeur de ce critère dans le domaine réduit, de plus, si cette solution est efficace, elle constitue une solution optimale du problème, si ce n'est pas le cas , l'exploration des arêtes incidentes (pour chercher une solution alternative efficace) qui est effectuée d'une manière optimiste (choisir les arêtes contenant un nombre maximum de solutions réalisables) peut être nous déplacé vers une solution optimale sans devoir réaliser des autres itérations ce qui minimise le nombre d'itérations réalisées et par conséquence le nombre de problèmes (P_l) à résoudre.

4.4 Description algorithmique Nous exposons dans cette partie une description algorithmique de l'algorithme principal de la méthode proposée ainsi que les différentes procédures utilisées dans cet algorithme.

Algorithm 2 l'optimisation discrète d'un critère linéaire ϕ sur l'ensemble efficace d'un problème multi-objectifs.

Entrées :
$Type$: max ;
$A_{(n\times m)}$: matrice des contraintes ;
$b_{(n\times 1)}$: vecteur second mombre ;
$d_{(1\times n)}$: vecteur du critère principal ;
$Cr_{(p\times n)}$: matrice des critères ;
Sorties :
X_{opt} : solution optimale du problème (P_E).
ϕ_{opt} : valeur optimale du critère ϕ.
Initialisation : $\phi_{opt} \leftarrow -inf$, $l \leftarrow 1$, recherche $\leftarrow 1$.
Résoudre le problème relaxé (P_R).
$[x_0, z_0, real]$ = Pb_relaxé $(Type, d, A, b)$.

Si $real = 0$ **alors** le probmème (P_E) est irréalisable.
Sinon
    $[x_t, z_t] = test\_efficace\ (Cr, A, b, x_0)$.
    Si $z_t = 0$ **alors** $X_{opt} \leftarrow x_0$, $\phi_{opt} \leftarrow z_0$.
    Sinon, poser $x_{ef} \leftarrow x_t$, $(P_0) \leftarrow (P_R)$.
        Tant que $recherche = 1$ **faire**
            $[x_{eq}, z_{eq}] = opt\ (Type, Cr, A, b, d, x_{ef})$.
            Si $z_{eq} > \phi_{opt}$ **alors** $X_{opt} \leftarrow x_{eq}$, $\phi_{opt} \leftarrow z_{eq}$.
            Finsi
            $[x_l, z_l, Dom, table_{opt}] = resol\_P_l\ (Type, d_{l-1}, A_{l-1}, b_{l-1}, x_{ef})$.
            Si $Dom = 0$ **Ou** $z_l < \phi_{opt}$ **alors** $recherche \leftarrow 0$.
            Sinon
                $[x_t^l, z_t^l] = test\_efficace(Cr, A, b, x_l)$.
                Si $z_t^l = 0$ **alors** $X_{opt} \leftarrow x_l$, $\phi_{opt} \leftarrow z_l$, $recherchre \leftarrow 0$.
                sinon
                    Construire l'ensemble $J_l = \{j \in N_l / \phi_j - d_j = 0\}$.
                    où N_l est l'ensemble des indices des variables hors base de x_l.
                    Si $J_l \neq \emptyset$ **alors**
                        $[x_{exp}, z_{exp}, existe] = Explor(Cr, A, b, d, table_{opt}, J_l)$.
                        Si $existe = 1$ **alors**
                        $X_{opt} \leftarrow x_{exp}$, $\phi_{opt} \leftarrow z_{exp}$, $rechereche \leftarrow 0$.
                        Finsi
                    Finsi
                Finsi
            Finsi
            $l \leftarrow l + 1, x_{ef} \leftarrow x_t^l$.
        Fin tant que
    Finsi
Finsi

Test d'efficacité d'une solution réalisable x :

Algorithm 3 $[x_t, z_t] = test\_efficace(Cr, A, b, x)$.

Résoudre un problème linéaire mono-objectif en variables mixte $P(x)$ en utilisant la méthode de deux phases et la méthode de séparation et évaluation.
soit (x_t, z_t) solution optimale du problème

Si $z_t = 0$ **alors** x est une solution efficace du problème multi-objectifs.

Sinon x_t est une solution efficace dont le vecteur critères domine celui de x.

Finsi.

Résolution du problème relaxé (P_R) :

Algorithm 4 $[x_0, z_0, real] = \text{Pb\_relaxé}\ (Type, d, A, b)$.

Initialisation : $real \leftarrow 1$

Résoudre le problème linéaire en nombre entiers (P_R) en utilisant l'algorithme de simplexe et la méthode par séparation et évaluation.

Si (P_R) est irréalisable **alors** $real \leftarrow 0$.

Sinon (x_0, z_0) solution optimale du problème relaxé(P_R).

Finsi.

Résolution du problème (P_l) **:**

Algorithm 5 $[x_l, z_l, Dom, table_{opt}] = resol\_P_l(Type, d_{l-1}, A_{l-1}, b_{l-1}, x_{ef})$.

Créer un nouveau problème (P_l) à partir de (P_{l-1}) en lui ajoutant les contraintes

$$\begin{cases} Z_i(x) \geq (Z_i(x_{ef}) + 1)y_i^k + M_i(1 - y_i^k), \quad i = 1, 2, ..., p. \\ \sum_{i=1}^{p} y_i^k \geq 1, y_i^k \in \{0, 1\}. \end{cases}$$

où M_i est la borne inférieure du i^{me} critère.

pour $i = 1$ **à** p **faire**

  $[M_i] =$ Pb_relaxé $(\min, Cr_i, A, b)$.

fin pour

Résoudre le problème (P_l) en utilisant la méthode de simplexe, duale du simplexe, la séparation et évaluation.

Si (P_l) est irréalisable **alors** $Dom \leftarrow 0$.

Sinon $Dom \leftarrow 1$, (x_l, z_l) solution optimale du problème et $table_{opt}$ le tableau optimale correspondant.

Finsi.

Algorithme d'exploration des arêtes incidentes à une solution x_l en cherchant une solution alternative efficace :

Algorithm 6 $[x_{exp}, z_{exp}, existe] = Explor(Cr, A, b, d, table_{opt}, J_l)$.

Initialisation : $existe \leftarrow 0$, $explor \leftarrow 1$, $i \leftarrow 1$.

Tant que $J_l \neq \emptyset$ **et** $explor = 1$ **faire**
calculer $\hat{\theta}$,
Si $\hat{\theta} = 0$ **alors** $J_l \leftarrow J_l \setminus \{J_l(i)\}$.
Sinon $j \leftarrow \hat{\theta}$
Tant que $j > 0$ **et** $existe = 0$ **faire**.
déterminer une solution entières x_j qui se trouve sur l'arete E_j et tester son efficacité.
$[x_t, z_t] = test\_efficace(Cr, A, b, x_j)$
Si $z_t = 0$ **alors**
$existe \leftarrow 1,\ x_{exp} \leftarrow x_j,\ z_{exp} \leftarrow c(x_j)$
Sinon $j \leftarrow j - 1$.
Finsi.
Fin tant que.
Finsi.
$i \leftarrow i + 1$.
Fin tant que.

Chapitre 5

Conclusion et perspectives

Les problèmes d'optimisation issus de la réalité sont la plupart du temps de nature multi-objectifs car plusieurs critères souvent contradictoires sont à considérer simultanément. Pour ce type de problèmes, la notion d'optimalité disparaît au profit de la notion d'efficacité. Il s'agit de chercher un ensemble de solutions réalisant le meilleur compromis entre les critères considérés.

Dans cette étude, nous nous sommes intéressés aux problèmes de programmation linéaire multi-objectifs en nombres entiers. L'intérêt de tels problèmes résulte du fait que dans de nombreuses situations réelles modélisables par la programmation mathématique, les variables de décision ne peuvent prendre que des valeurs entières.

Ensuite, Nous avons focalisés notre attention à l'optimisation d'un critère linéaire sur l'ensemble de solutions efficaces d'un tel problème, dans certaines situations pratiques les décideurs ne sont pas intéressés par toutes les solutions efficaces, mais seulement par celles qui optimisent la valeur d'un critère prédéterminé, ce type de problèmes est en principe très difficile à résoudre vu que son ensemble d'admissibilité (l'ensemble des solutions efficaces) n'est pas convexe.

Nous avons commencé ce travail par introduire les notions de base concernant l'optimisation multi-objectifs et quelques méthodes de résolution existantes dans la littérature dans le premier chapitre.

Après, nous avons réalisé une description détaillée sur l'optimisation multi-objectifs discrète dans le deuxième chapitre afin de passer à l'analyse de l'optimisation d'un critère linéaire sur l'ensemble des points efficaces d'un problème multi-objectifs en nombre entier dans le chapitre 3, nous nous sommes intéressés à deux méthodes de résolution les plus récentes dans la littérature à notre connaissance. Il s'agit de la méthode de

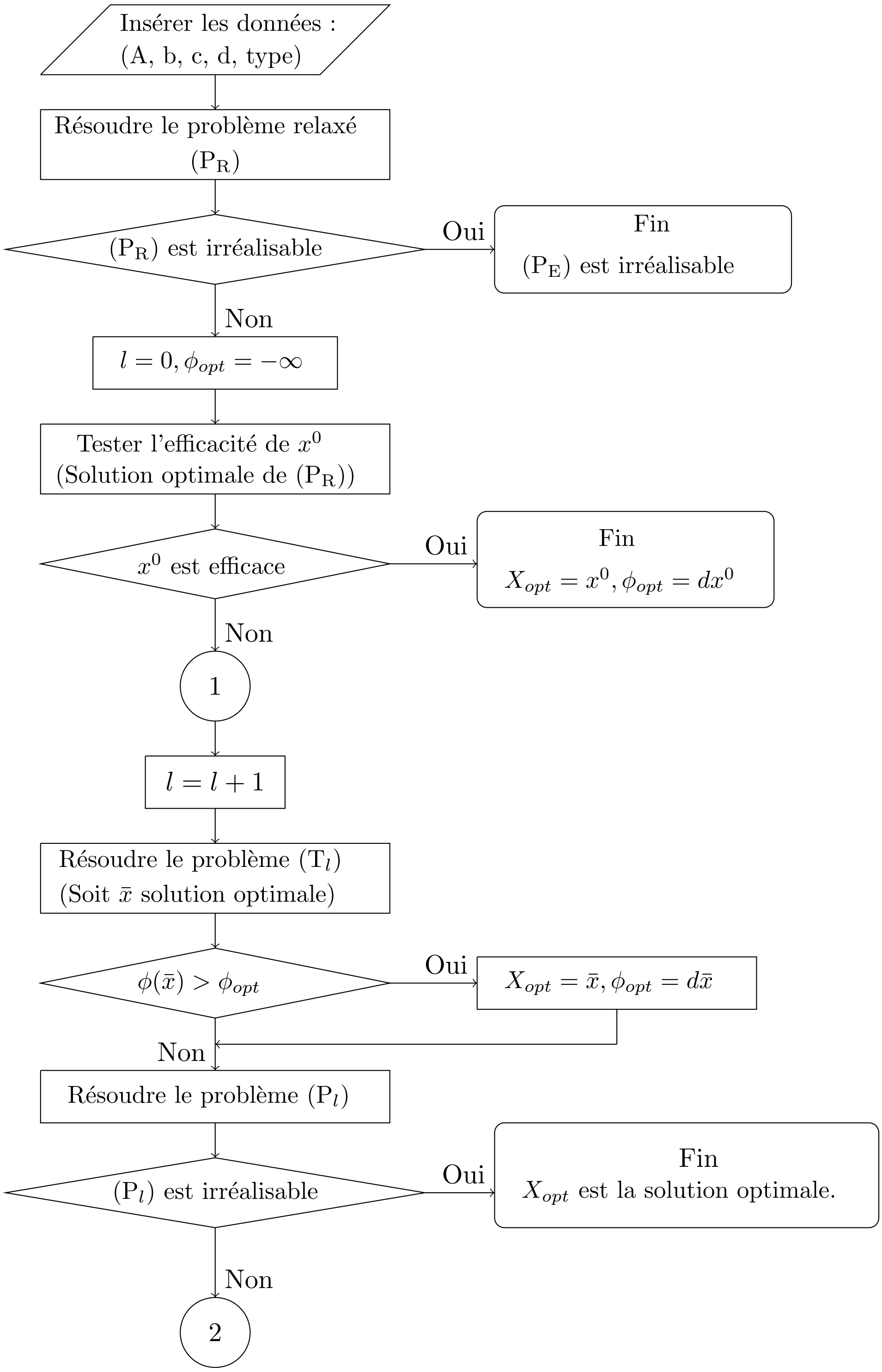
D.Chaabane [7] et celle de Jesùs M.Jorge [19]. À partir de lesquelles nous avons élaboré une nouvelle méthode de résolution où on a combiné entre les principes utilisés dans ces deux méthodes, nous avons exposé et illustré les trois méthodes par des exemples didactiques, puis nous avons testé l'efficacité de la nouvelle méthode par une étude expérimentale réalisée à l'aide d'un programme qu'on a implémenté sous l'environnement Matlab 7.0

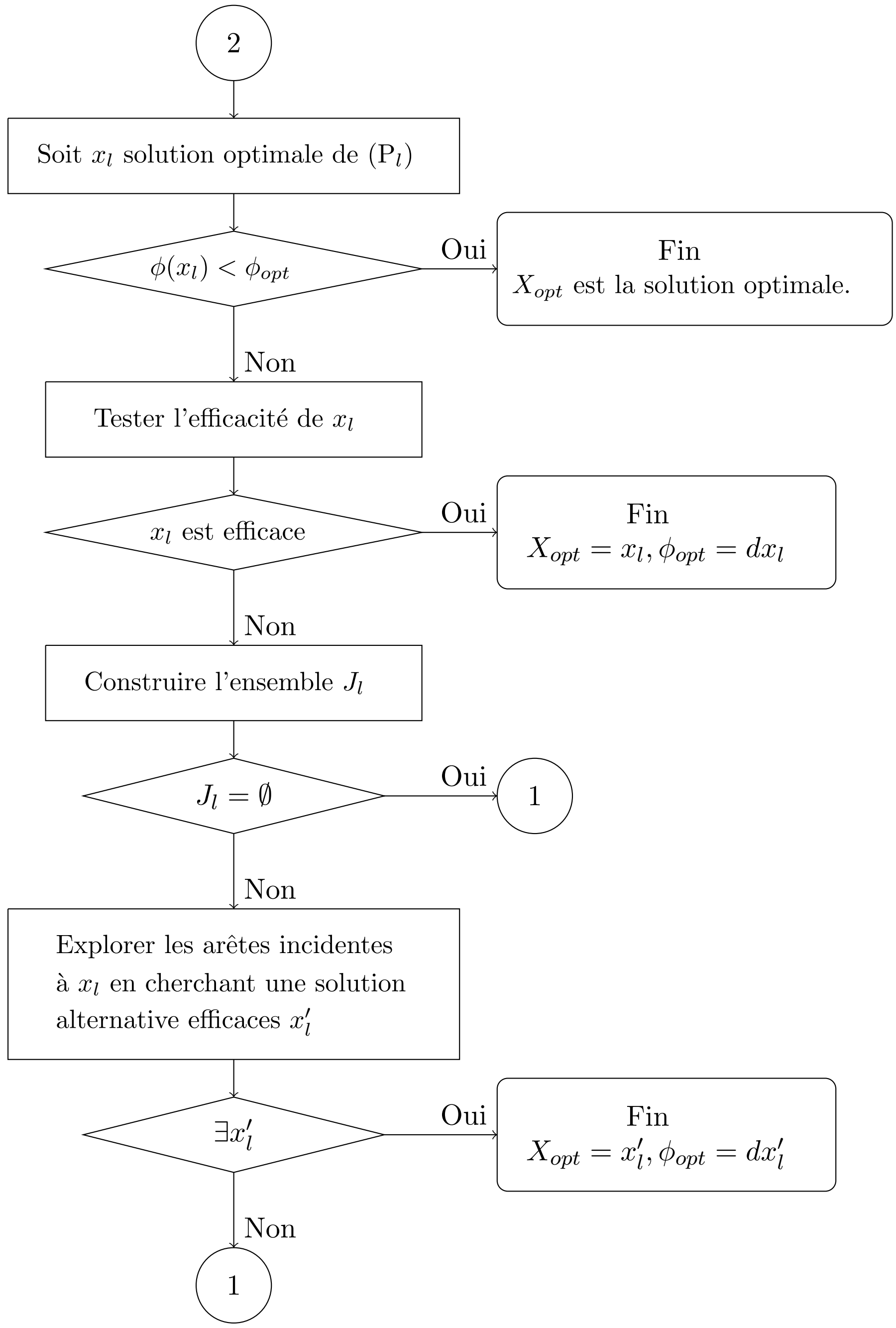
D'après notre modeste étude, nous avons constaté que l'algorithme proposé est très efficace en terme de gain précieux du nombre d'itérations effectuées (le nombre de points efficaces examinés) ce qui lui permet d'aller très loin dans la taille des problèmes traités, nous proposons comme perspectives de recherche :

- Appliquer des métha-heuristiques pour la résolution de ce problème.
- L'optimisation d'un critère linéaire sur l'ensemble de solutions efficace d'un problème multi-objectif en variables mixtes.
- Résoudre les problèmes d'optimisation des flots maximaux.
- L'optimisation d'un critère non linéaire sur les solutions efficaces d'un problème multi-objectifs, nous pensons particulièrement au cas fractionnaires et quadratique.
- L'optimisation d'un critère de compromis sur l'ensemble des solution efficaces d'un problème stochastique.

Annexe

Organigramme de la méthode ODSE




2
Soit x_l solution optimale de (P_l)
$\phi(x_l) < \phi_{opt}$
Oui
Fin
X_{opt} est la solution optimale.
Non
Tester l'efficacité de x_l
x_l est efficace
Oui
Fin
$X_{opt} = x_l, \phi_{opt} = dx_l$
Non
Construire l'ensemble J_l
$J_l = \emptyset$
Oui
1
Non
Explorer les arêtes incidentes à x_l en cherchant une solution alternative efficaces x'_l
$\exists x'_l$
Oui
Fin
$X_{opt} = x'_l, \phi_{opt} = dx'_l$
Non
1

Bibliographie

[1] M. Abbas and D. Chaabane, *An algorithm for solving multiple objective integer linear programming problem*, RAIRO Operations Research 36, pp. 351-364, (2002).

[2] M. Abbas and D. Chaabane, *Optimizing a Linear Function Over an Integer Efficient Set*, European Journal of Operations Research, 174, pp. 1140-1161, (2006).

[3] H.P. Benson, *Optimizing Over the Efficient Set*, Journal of Mathematical Analysis and Applications, 98, pp.562-580, (1984).

[4] G. Bitran, *Theory and Algorithms for Linear Multiple Objective Programs with Zero - One Variables*, Mathematical Programming, 17,pp. 362-390,(1979).

[5] V.J. Bowman, *On the Relationship of the Tchebytcheff Norm and the Efficient Frontier of Multiple-Criteria Objectives*, in Thiriez H. & Zionts S. (Eds), Multiple Criteria Decision Making, Springer-Verlag,pp. 76-85,(1976).

[6] D. Chaabane, *Solving Multiple Objective Integer Linear Programming Problem*, prépublication n°089, O.R.Dept, Université des Sciences et de la Technologie Houari Boumediene, Algérie, (2002).

[7] D. Chaabane, *A Method for optimizing over the Integer Efficient Set in th criteria space*, Second International Engineering Conference Proceeding. Ryadh, pp. 19-21, (2004).

[8] D. Chaabane,*CONTRIBUTION A L'OPTIMISATION MULTICRITERE EN VARIABLES DISCRETES* :Dissertation soumise à la Faculté Polytechnique de Mons.en vue l'obtention du titre de Docteur en Sciences Appliquées.

[9] A. Crema and J. Sylva, *A method for finding the set of non-dominated vectors for multiple objective integer linear programs*, European Journal of Operational Research, in press, (2003).

[10] A. Crema J and Sylva *A method for finding the set of non-dominated vectors for multiple objective integer linear programs*, European Journal of Operational Research, 158, pp. 46-55,(2004).

[11] A. Crema and J. Sylva, *A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs*, European Journal of Operational Research, in press, (2006).

[12] Chalmet L.G., Lemondis L. and Elzinga D.J., *An Algorithm for the Bi-Criterion Integer Programming Problem*, European Journal of Operational Research, 25,pp. 292-300,(1986).

[13] G.B. Dantzig,*On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem*, Operations Research, 7, pp.58-66,(1959).

[14] J. G. Ecker, H.G. Song, *Optimizing a Linear Function over an Efficient Set*, Journal of Optimization Theory and Applications 83 (3),pp 541-563,(1994).

[15] A.M.Geoffrion,*Proper Efficiency and the theory of Vector Maximization*, Journal of Mathematical Analysis and Applications ,22, pp.618-630,(1968).

[16] R.E. Gomory, Outline of an Algorithm for Integer Solutions to Linear Programs,Bulletin of the AMS 64,pp. 275-278.

[17] J.J. Gonzales, G.R. Reeves, L.S. Franz, *An Interactive Procedure for Solving Multiple Objective Integer Linear Programming Problems*, in Haimes Y. and Chankong(eds), Decision Making with Multiple Objectives, Springer-Verlag, Berlin, pp.250-260,(1985).

[18] R. Gupta, R. Malhotra, *Multi-Criteria Integer Linear Programming Problem*, Cahiers du CERO 34,pp. 51-68,(1992).

[19] J.M.Jorge, *An algorithm for Optimizing a linear function over an Integer Efficient Set*, European Journal of Operational Research, (2008).

[20] Karaivanova, J. N., Narula, S. C., Vassilev, V.*An interactive procedure for multiple objective integer linear programming problems*, European Journal of Operational Research ,68, pp.344-351,(1993).

[21] D. Klein, E. Hannan, *An Algorithm for Multiple Objective Integer Linear Programming Problem*, European Journal of Operational Research 9,pp. 378-385,(1982).

[22] N.C. Nguyen, *An algorithm for optimizing a linear function over the Integer Efficient Set*, Konrad-Zuse-Zentrum fur Informationstechnik. Berlin, (1992).

[23] V.Pareto,*Cours d'économie politique*,1-2,F.Rouge, Lausanne, (1896).

[24] J. Philip, *Algorithms for the Vector Maximization Problem*, Mathematical Programming 2,pp. 207-229, (1972).

[25] R. Steuer, *Multiple Criteria Optimization : Theory, Computation and Applications*, John Wiley & Sons, New-York (1985).

[26] J. Teghem, P.L. Kunsch, *Interactive Method for Multiobjective Integer Linear Programming*, in : G. Fandel et al. (Eds.), Large Scale Modelling and InteractiveDecision Analysis, Springer Verlag, pp 75-87,(1986).

[27] J. Teghem, *Programmation Linéaire* , Editions de l'Université de Bruxelles SMA deuxième édition (2003).

[28] D.J. White, *the maximization of a function over the Efficient Set via a penalty function Approch*, European Journal of Operational Research, 94, pp. 143-153, (1996).

[29] Y.Yamamoto, *Optimization over the Efficient Set*, Overview,Kluwer Academic Publishers, (2001).