

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
HOUARI BOUMEDIENNE



FACULTE DE MATHEMATIQUES

THESE

Présenté pour l'obtention du diplôme de MAGISTER

En : MATHEMATIQUES

Spécialité : Recherche Opérationnelle : Méthodes Stochastiques

Par : DAOUD MALIKA

Thème

**SUR QUELQUES PROBLEMES D'OPTIMISATION
MULTI-OBJECTIFS**

Soutenu publiquement le 01 /03/2008 à 10h30, devant le Jury :

H.AIT HADDADENE	Professeur	Président.
D.CHAABANE	Maître de Conférences	Directeur de thèse.
M.DJEDOUR	Professeur	Examineur.
M.MOULAI	Maître de Conférences	Examineur.

A ma mère

A mon père

A la mémoire de mon oncle Saâd

A toute ma famille

A tout ceux dont les noms sont inscrits sur la page secrète de mon coeur

A tout ceux qui m'ont aidé de près ou de loin à réaliser ce travail

Remerciements

Je tiens à exprimer ma gratitude à mon directeur de mémoire de Magister, le Docteur Djamel Chaâbane de l'Université des Sciences et de la Technologie Houari Boumedienne qui a bien voulu me proposer un sujet de mémoire de Magister et d'avoir suivi le travail jusqu'à son terme. Ses remarques très pertinentes, ses conseils fructueux, m'ont beaucoup aidé à réaliser ce travail.

Je remercie également le Professeur Hacène Aït Haddadène d'avoir accepté de présider ma soutenance.

J'exprime ma sincère reconnaissance aux membres du Jury, Messieurs Hacène Aït Haddadène, Mohamed Djedour et Mustapha Moulaï qui m'ont fait l'honneur d'accepter d'être rapporteurs de mon mémoire.

Je tiens à remercier profondément Monsieur Berrachedi le chef département de Recherche Opérationnelle, et Monsieur Boudhar ainsi que l'ensemble des membres du laboratoire LAID3.

Je remercie enfin pour leurs soutiens les membres de ma famille et j'adresse un grand remerciement à mon père et ma mère qui m'ont encouragé à reprendre mes études.

Résumé

Critiquer la méthode du simplexe et sa forme révisée pour les problèmes d'optimisation mono-objectifs à grande taille, une nouvelle méthode a été développée dite méthode de décomposition, cette méthode a été décrite avec deux exemples illustratifs.

La recherche de l'ensemble de solutions efficaces pour un problème multi-objectif est difficile (peu d'études dans ce cas), même si seulement deux objectifs sont visés, des méthodes exactes ont été développées pour le problème de sac à dos et le problème d'affectation dans le cas bi-objectif basées sur les propriétés de dualité, la méthode Hongroise, la méthode séparation et évaluation et la fonction d'agrégation.

Une adaptation de la méthode de décomposition pour le problème de sac à dos mono-objectif, différente de la méthode séparation et évaluation (Branch et Bound) a été suggérée sous quelques conditions.

Une clarification pour la méthode en deux phases pour résoudre les problèmes d'optimisation bi-objectifs.

Une méthode exacte est suggérée détermine toutes les solutions efficaces d'un problème de sac à dos bi-objectif. Enfin, nous exposons quelques perspectives pour l'avenir.

Mots-clés : Programmation linéaire, Méthode de décomposition, Variables de rentrée simultanée, problème combinatoire, problème d'optimisation multi-objectif (bi-objectif), Méthodes en deux phases.

Table des matières

1	Quelques Problèmes d'Optimisation Combinatoires Uni-critères et Quelques Méthodes de Résolution	6
1.1	Introduction	6
I	Quelques Méthodes de Résolution des Problèmes de Programmation Linéaire	7
1.2	La méthode du simplexe et sa forme révisée	8
1.2.1	Introduction	8
1.2.2	Algorithme du simplexe	10
1.2.3	La forme révisée du simplexe	11
1.2.4	Détermination de la forme produit de l'inverse de la base	12
1.2.5	Algorithme du simplexe révisé	13
1.3	La nouvelle méthode de résolution [6, 11]	14
1.3.1	Formulation mathématique	15
1.3.2	Méthode de décomposition	15
1.4	Les différentes rentrées de variables	21
1.4.1	Variables de rentrée totalement indépendante	21
1.4.2	Variables de rentrée croisée	23
1.4.3	Variables de rentrée liée	23
II	Problème d'Optimisation Combinatoire Uni-critère	29
1.5	Problème d'Optimisation Combinatoire Uni-critère	30
1.5.1	Introduction	30
1.5.2	programmation linéaire en nombres entiers	30
1.5.3	problèmes d'optimisation à variables binaires	30
1.5.4	Le problème de Sac à dos (KP)	31
1.5.5	Le problème d'affectation (AP)	31
1.5.6	Méthodes de résolution	33
1.6	Adaptation de la méthode de décomposition dans le problème de " Sac à Dos" uni-critère	39
1.6.1	La formulation mathématique du problème	39
1.6.2	La formulation mathématique du problème relaxé	40
1.7	Conclusion	47

2	Abécédaire sur Quelques Problèmes d'Optimisation Multi-Objectifs	50
2.1	Introduction	50
2.2	Les méthodes exactes	51
2.2.1	Problèmes d'optimisation multi-objectifs en nombres continus	51
2.2.2	Définitions	52
2.2.3	Méthodes de résolution	54
2.2.4	Catégories de méthodes fondamentales	55
2.2.5	Quelques résultats de base	59
2.2.6	Problèmes d'optimisation multi-objectifs en nombres entiers	59
2.2.7	Méthodes de résolution	62
2.2.8	Problèmes d'optimisation à variables binaires	65
2.2.9	Complexité	65
2.3	Les méthodes métaheuristiques [9, 12]	65
2.3.1	Introduction	66
2.3.2	Voisinage	66
2.3.3	Population	67
3	Problèmes d'Optimisation Combinatoire Bi-Objectifs et la méthode en deux phases	68
3.1	Introduction	68
3.2	La méthode exacte des deux phases dans son contexte général [14, 15]	69
3.2.1	Formulation mathématique du problème bi-AP	69
3.2.2	Détermination des solutions supportées $SE(P)$	70
3.2.3	Détermination des solutions non supportées $E(P) \setminus SE(P)$	73
3.2.4	Illustration de la méthode sur deux exemples	81
3.3	La méthode de résolution proposée pour le problème bi-KP	87
3.3.1	Formulation mathématique du problème bi-KP	87
3.3.2	La première méthode proposée	88
3.3.3	La deuxième méthode proposée	89
3.3.4	Conclusion	91
4	Conclusion générale et perspectives	92
	Bibliographie	94

Introduction générale

Prendre une décision est l'action de tous les jours, tel que le rôle principal de la recherche opérationnelle est d'aider à prendre cette décision en vue d'une gestion efficace, rationnelle et logique.

Longtemps, les modèles d'optimisation se résumaient à trouver une valeur optimale pour un seul objectif. Le simplexe et sa forme révisée étaient les premières méthodes à résoudre ces problèmes. Critiquer ces méthodes pour des problèmes à grande taille, une méthode est née dite méthode de "Décomposition", son but est de remédier quelques uns de ces inconvénients. Afin de s'approcher au maximum de la réalité, des études portant sur l'optimisation multi-critères (multi-objectifs)¹ ont fait leurs apparitions aux années 50. Ces études se sont révélées indispensables car de nombreux problèmes réels sont des modèles multi-critères. En recherche opérationnelle, plusieurs techniques ont été développées au cours des années en vue de traiter des problèmes à objectifs multiples, beaucoup d'approches différentes ont été suggérées, allant d'une combinaison naïve des objectifs en un seul à l'utilisation de la théorie des jeux pour coordonner l'importance relative de chaque objectif et le comportement de différents intervenants. Il existe un nombre important de problèmes réels qui sont des problèmes multi-critères, citons par exemple le problème de sac à dos et le problème d'affectation qui en fait partie, car de nombreux autres objectifs peuvent être optimisés.

L'optimisation multi-objectif consiste à optimiser plusieurs composantes d'un vecteur fonction coût (profit), chaque composante de ce vecteur est une fonction objectif.

Pour les solutions d'un problème multi-objectif (MO), la relation d'ordre n'est pas totale, une solution peut être meilleure qu'une autre sur certains critères et moins bonne sur les autres. Donc il convient de manipuler des populations de solutions dites "Pareto-optimales" (voir le deuxième chapitre), ce concept d'optimalité a réussi à pénétrer en recherche opérationnelle grâce au travail de Koopmans (1951).

Les premières approches de l'optimisation multi-critères n'utilisaient pas ce concept, elles ramenaient les problèmes multi-critères à un seul problème mono-critère² (Théorème de Geoffrion, problème paramétrique, fonction agrégation).

Généralement, les méthodes d'optimisation multi-critères sont regroupées en deux catégories selon la nature du problème multi-objectif :

1. l'analyse multi-critère est utilisée lorsque l'ensemble fini de solutions possibles est fourni explicitement.
2. la programmation mathématique multi-critère (ou optimisation multi-objectif), s'impose lorsque l'ensemble de solutions (de cardinal souvent infini) est défini implicitement par la satisfaction de contraintes.

Notre travail se base sur une étude de quelques problèmes d'optimisation combinatoires multi-objectifs en nombres entiers MOILP (Multiple Objective Integer Linear Programming) (binaires (LPBIL)) où la fonction objectif ainsi que les contraintes sont linéaires en mettant le point sur le problème de sac à dos (Knapsack Problem) et le problème d'affectation (Assignment Problem) bi-objectifs.

L'intérêt de ces problèmes résulte du fait que dans la plupart des problèmes d'optimisation, la

¹consiste à modéliser les problèmes de sorte que tous les objectifs pertinents soient pris en considération.

²par l'intermédiaire de la programmation linéaire paramétrique.

présence des variables discrètes (0,1) est inévitable ; et aussi une majorité de problèmes d'optimisation combinatoires peuvent s'écrire, d'une manière plus concise comme des programmes en nombres entiers. Ces derniers peuvent aussi se formuler comme le problème de sac à dos ou le problème d'affectation.

Ce mémoire comporte quatre chapitres. Le premier chapitre est divisé en deux parties :

Dans la première partie, nous ferons un bref rappel sur la méthode classique du simplexe et sa forme révisée qui ont été développées pour résoudre les problèmes d'optimisation uni-critères. La méthode du simplexe a présenté des insuffisances, critiquer pour des problèmes à grande taille, une méthode dite "décomposition" a été proposée pour résoudre ces problèmes dont la structure est d'une forme bien particulière [11].

Souvent la matrice des contraintes d'un problème d'optimisation à grande taille est creuse, d'où une possibilité d'une adaptation de la méthode qui tient compte de l'entrée simultanée³ de plusieurs variables dans la base. Nous décrirons cette méthode, ensuite nous proposerons un exemple pour la compréhension de la méthode et nous exposerons un exemple détaillé qui a été proposé [11] qui fait l'objet de l'adaptation de l'entrée simultanée des variables.

Dans la deuxième partie, nous présenterons les problèmes d'optimisation combinatoires uni-critères en nombres discrets (Binaires) et quelques méthodes de résolution existantes pour ce genre de problèmes.

Vu le problème de dégénérescence qui a lieu dans le problème de sac à dos, la structure du problème qui converge presque vers la structure des problèmes d'optimisation combinatoires à grande taille dont la matrice des contraintes est creuse⁴. Nous nous sommes intéressés à exploiter la méthode dite décomposition en adaptant l'entrée simultanée des variables qui fût une méthode différente de la méthode séparation et évaluation (Branch and Bound), dans le problème au but de diminuer les étapes inutiles. Ensuite, deux exemples illustratifs seront présentés.

A la fin du premier chapitre, nous suggérons une condition pour faire ajouter les coupes de Gomory dans une seule itération (au même temps), c'est-à-dire utiliser le même principe de l'entrée simultanée des variables dans la base décrit au paravent.

Le deuxième chapitre est entièrement projeté aux problèmes d'optimisation multi-critères. Nous présenterons en premier lieu la structure générale des problèmes MOLP et MOILP, ensuite nous exposerons les méthodes exactes élaborées pour la résolution des problèmes d'optimisation multi-critères en nombres continus, nous citerons quelques résultats de base.

En second lieu, nous illustrerons quelques méthodes développées pour la programmation multi-objectifs discrète⁵(binaires).

Tenant compte de la complexité de la résolution en méthodes exactes des problèmes d'optimisation combinatoire (multi-objectifs), nous terminerons ce chapitre par un bref aperçu sur quelques méthodes méta-heuristiques.

³La définition de l'entrée simultanée est décrite dans la première partie du premier chapitre.

⁴Voir la définition du problème de sac à dos et les problèmes d'optimisation dont la méthode de décomposition est née(les définitions sont dans le mémoire).

⁵les problèmes d'optimisation multi-objectifs en nombres entiers.

Dans le troisième chapitre, et pour le but d'éclaircir une méthode de résolution développée par Ulungu et Teghem, qui a pour but résoudre les problèmes d'optimisation bi-objectifs en variables binaires en deux phases (le problème de sac à dos et le problème d'affectation) , nous décrirons cette dernière dans son contexte général et nous exposerons les deux phases en détails avec des algorithmes qui expliquent les étapes principales de ces deux phases. Ensuite des exemples didactiques seront présentés.

Deux méthodes exactes ont été suggérées pour résoudre le problème de sac à dos bi-critère, la première est basée sur une énumération des variables qui peuvent entrer dans la base, la deuxième est inspirée de la méthode en deux phases [15] pour résoudre le problème de sac à dos bi-objectif. Un exemple didactique sera présenté pour les deux propositions. Nous terminerons par une description de quelques avantages et inconvénients ⁶ des deux méthodes proposées.

Dans le quatrième chapitre, une conclusion résume les principaux résultats obtenus de ce travail, quelques perspectives de recherche pour l'avenir qui certainement feront la suite du travail accompli.

⁶à notre connaissance.

Chapitre 1

Quelques Problèmes d'Optimisation Combinatoires Uni-critères et Quelques Méthodes de Résolution

1.1 Introduction

Bien que le simplexe est la première méthode classique qui a été décrite par G.B.Dantzig en 1947 aisé facile pour résoudre tous les problèmes d'optimisation, quelques critiques de cette méthode feront une piste de recherche d'autres méthodes de résolution.

La forme révisée du simplexe s'impose pour remédier à ses inconvénients en évitant beaucoup de calculs inutiles que fait la méthode classique dite simplexe. L'intérêt premier de l'algorithme révisé est de permettre de travailler à chaque itération avec les données originales du problème, donc éviter la propagation des erreurs d'arrondis d'une itération à la suivante.

D'autre part cette forme révisée ne permet pas toujours de faire des économies de calculs dans des problèmes à grande taille.

Donc, il est indispensable de trouver d'autres méthodes de résolution efficaces et rapides pour couvrir l'insuffisance du simplexe pour les problèmes à grande dimension.

Certains problèmes d'optimisation à grande dimension dont la matrice des contraintes est creuse ont menés les chercheurs à adapter une nouvelle méthode exacte intitulé " méthode de décomposition " elle se base sur l'utilisation de la forme produit de l'inverse de la matrice de base, et sur les sous problèmes extraits du problème initial.

Ainsi l'entrée d'une variable dans la base pour ces problèmes ne modifie pas la structure de la colonne où se trouve une autre variable hors base, celle-ci pouvant à son tour entrer dans la base, donc l'entrée simultanée de deux ou plusieurs variables diminue le nombre d'itération, et le temps de calcul.

On présentera les cas où il y'a une possibilité de faire entrer simultanément des variables dans la base. Enfin deux exemples illustratifs ; le premier montre que l'entrée simultanément des variables dans la base est impossible mais néanmoins la méthode de décomposition est appliquée, le deuxième exemple retiré de [11] montre que l'entrée simultanément des variables est possible. Ensuite, nous adapterons la méthode de décomposition avec l'entrée simultanée des variables dans le problème de sac à dos, un exemple à la fin de l'adaptation.

Une condition nécessaire sera suggérée pour faire entrer les coupes de Gomory simultanément.

Première partie

Quelques Méthodes de Résolution des Problèmes de Programmation Linéaire

1.2 La méthode du simplexe et sa forme révisée

1.2.1 Introduction

La méthode du simplexe développée par G.B.Dantzig en 1947 a été la première méthode qui permet de résoudre aisément les problèmes d'optimisation. On se propose de la présenter brièvement.

Avant ça nous rappelons quelques définitions de types de problèmes traités dans cette thèse.

Programmation Linéaire

La programmation linéaire a pour objet l'étude et la résolution des "programmes linéaires", c'est à dire des problèmes d'optimisation dans lesquels la fonction objective aussi bien que les contraintes sont des fonctions linéaires en variables de décision.

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

Problème d'Optimisation Combinatoire

Un problème d'optimisation combinatoire est défini à partir d'un ensemble fini S et d'une application $f : S \rightarrow \mathbb{R}$, il s'agit de déterminer $\tilde{s} \in S$ tel que $f(\tilde{s}) = (\min_{s \in S} [f(s)])$ ou $f(\tilde{s}) = (\max_{s \in S} [f(s)])$.

La convexité

Un ensemble $C \subset \mathbb{R}^n$ est dit "convexe" si :

$$\begin{cases} x_1, x_2 \in C \\ 0 \leq \lambda \leq 1 \end{cases} \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in C \quad (1.2.1)$$

Une fonction f d'un convexe C à valeurs dans \mathbb{R} est dite convexe si :

$$\begin{cases} x_1, x_2 \in C \\ 0 \leq \lambda \leq 1 \end{cases} \Rightarrow f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (1.2.2)$$

Le programme mathématique $(\max (\min_{s \in S} [f(s)]))$ est un programme "convexe", si C est un ensemble convexe et si f est une fonction convexe.

Remarque 1 *Sans perdre de généralité, nous considérons dans ce chapitre, que le problème traiter est à maximiser.*

La forme générale d'un problème de programmation linéaire est donnée par :

$$(P) \begin{cases} Max Z = Cx \\ Tx = d \\ x \geq 0 \end{cases} \quad (1.2.3)$$

$$T(m, n), \quad C(1, n), \quad x(n, 1), \quad d(m, 1)$$

Supposons que le rang $(T) = m < n$, B est une sous matrice carrée (m, m) et régulière : donc B est une base.

On associe à cette base la décomposition suivante : $T = [B|R] \quad x \begin{pmatrix} x_B \\ x_R \end{pmatrix}$

$x_B = (x_i \quad i \in I)$, I ensemble des indices dans la base.

$x_R = (x_j \quad j \in J)$, J ensemble des indices hors la base.

Remarque 2 1) Une solution de base est admissible si $x_B \geq 0$.

2) Si $x_R = 0$, la solution de base $x = \begin{pmatrix} B^{-1}d \\ 0 \end{pmatrix}$.

3) Une base B est réalisable si la solution de base est réalisable.

4) Une solution de base associée à une base réalisable est appelée solution de base réalisable.

5) Une solution de base est dite dégénérée si x_B a des composantes nulles.

Caractérisation d'une solution de base

Soit le problème ci dessus, en décomposant la matrice T et le vecteur x tel que : $T[B|R]$ et $x \begin{bmatrix} x_B \\ x_R \end{bmatrix}$

on aura : $Bx_R + Rx_R = d \Rightarrow x_B + B^{-1}Rx_R = B^{-1}d$.

On remplace x_B par sa valeur dans la fonction objectif, on obtiendra :

$$MaxZ = C_B B^{-1}d - (C_B B^{-1}R - C_R)x_R$$

Puisque x_B est admissible donc :

$$\begin{cases} MaxZ = C_B B^{-1}d - (C_B B^{-1}R - C_R)x_R \\ \text{tq} \quad B^{-1}d - B^{-1}Rx_R \geq 0 \end{cases} \quad (1.2.4)$$

en posant

$$\begin{aligned} B^{-1}R &= A = (a_{i0} \quad i \in I \quad j \in J) \\ B^{-1}d &= (a_{i0} \quad i \in I) = a_0 \\ C_B B^{-1}R - C_R &= (a_{0j} \quad j \in J) = \alpha_0 \\ B^{-1}t_j &= (a_j \quad j \in J) \end{aligned}$$

la forme équivalente à (1.2.4) est :

$$\begin{cases} MaxZ = a_{00} - \sum_{j \in J} a_{0j}x_j \\ x_i + \sum_{j \in J} a_{ij}x_j = a_{i0} \quad i \in I \\ x_j \geq 0 \end{cases} \quad (1.2.5)$$

Le tableau du simplexe correspondant est le suivant :

		$x_j \quad j \in J$
Z	a_{00}	a_{0j}
$x_i \quad i \in I$	a_{i0}	a_{ij}

TAB. 1.1 – Tableau du simplexe

On appelle $C_B B^{-1}$ vecteur multiplicateur relatif à B .

$C_B = (0, C_B B^{-1}R - C_R)$ vecteur coût relatif à B .

– Si C_B le vecteur coût relatif à B est positif, la solution de base associé à B est solution optimale.

– Une base réalisable B tel que le vecteur coût est positif ou nul est appelée base optimale.

1.2.2 Algorithme du simplexe

- 1) Ecrire le problème sous la forme du tableau (Tab.1.1) .
- 2) Déterminer une solution de base initiale correspondante à une solution de base admissible.
Soit I l'ensemble des indices des variables de base, J celui des indices hors base et considérons le tableau du simplexe (Tab. 1.1).
- 3) Tester les a_{0j} .
 - a) Si $a_{0j} \geq 0 \quad \forall j \in J$ la solution considérée est optimale.
 - b) Si $\exists j \in J \quad a_{0j} < 0$ et $a_{0j} \leq 0 \quad \forall i \in I$, il n'y a pas de solution optimale finie et la valeur de la fonction objectif n'est pas bornée ($Z \rightarrow \infty$).
 - c) Autrement, effectuer le changement de base.
- 4) Changement de base :

- a) Déterminer l'indice k de la variable qui entre dans la base,

$$a_{0k} = \min_{j \in J} (a_{0j}) .$$

- b) Déterminer l'indice l de la variable qui sort de la base

$$\frac{a_{l0}}{a_{lk}} = \min_{i \in I} \left\{ \frac{a_{i0}}{a_{ik}} \right\}, \quad a_{ik} > 0.$$

l'élément a_{lk} est appelé " pivot ", si le minimum obtenu est unique (non dégénérescence) , la nouvelle solution aura toutes ses composantes positives, et par conséquent l'ensemble des indices de la nouvelle base sera :

$$I_{nouv} = I \setminus \{l\} \cup \{k\}$$

- c) Calculer le nouveau tableau du simplexe en appliquant les formules de changement de base suivantes :

$$a_{ij}^* = a_{ij} - \frac{a_{ik}a_{lj}}{a_{lk}} \quad i \neq k \quad j \neq l$$

$$a_{kj}^* = \frac{a_{lj}}{a_{lk}} \quad j \neq k$$

$$a_{il}^* = -\frac{a_{ik}}{a_{lk}} \quad i \neq k$$

$$a_{lk}^* = \frac{1}{a_{lk}}$$

5) On arrête l'algorithme lorsque la base obtenue s'avère optimale.

Dualité

A tout problème linéaire on associe un autre problème linéaire appelé dual du problème initial, ce dernier étant appelé par opposition primal.

D'une manière générale on écrit :

Forme standard

$$Primal \begin{cases} MaxZ = Cx \\ Tx = d \\ x \geq 0 \end{cases} \quad Dual \begin{cases} MinW = ud \\ u \in \mathbb{R}^m \\ uT \geq C \end{cases} \quad (1.2.6)$$

Forme canonique

$$Primal \begin{cases} MaxZ = Cx \\ Tx \leq d \\ x \geq 0 \end{cases} \quad Dual \begin{cases} MinW = ud \\ u \geq 0 \\ uT \geq C \end{cases} \quad (1.2.7)$$

Pour la procédure de l'algorithme dual simplexe voir[4].

Critiquer la méthode du simplexe à plusieurs titres, la forme révisée s'impose pour remédier à tous ses inconvénients. Cette forme révisée n'apporte aucune modification par rapport à la méthode du simplexe, c'est sa mise en oeuvre qui a été révisée.

1.2.3 La forme révisée du simplexe

Son rôle principal est de minimiser les erreurs accumulées de calcul et garder la structure des données, en revenant à chaque itération aux données de l'énoncé puisque le Tableau du Simplexe sera calculé en multipliant le tableau initial par la matrice inverse de la base. Cette dernière est obtenue sans effectuer les opérations classiques de l'inverse d'une matrice. Lors du passage d'une base B_{t-1} à la base B_t , une seule colonne est modifiée puisqu'un vecteur t_k prendra la place d'un vecteur t_l , donc il est possible de calculer B_t^{-1} à partir de B_{t-1}^{-1} . L'obtention du tableau simplexe par le produit matriciel permettra de ne calculer du Tableau du Simplexe que les seuls éléments nécessaires à la vérification du test d'arrêt.

Les éléments indispensables du Tableau du Simplexe (TS) sont :

- α_0 la ligne relative à la fonction objectif $C_B B^{-1} R - C_R$.
- a_0 la colonne des termes indépendants $B^{-1} d$.
- a_k la colonne de la variable entrante dans la base, elle correspond au maximum des coefficients a_{0k} .

Construction du Tableau du Simplexe

La forme révisée utilise le tableau du simplexe complet :

	d	$x_i \quad i \in I$	$x_j \quad j \in J$
Z	a_{00}	0	a_{0j}
$x_i \quad i \in I$	a_{i0}	I	a_{ij}

TAB. 1.2 – Tableau du simplexe II

La forme équivalente au problème est :

$$\begin{cases} \max Z - Cx = 0 \\ Tx = d \\ x \geq 0 \end{cases} \quad (1.2.8)$$

Le tableau des données :

$$(\hat{T}, \hat{d}) = \begin{pmatrix} 0 & -C & 1 \\ d & T & 0 \end{pmatrix}$$

Posons :

$$\hat{B} = \begin{pmatrix} 1 & -C_B \\ 0^m & B \end{pmatrix}$$

après calcul, on aura :

$$\hat{B}^{-1} = \begin{pmatrix} 1 & C_B B^{-1} \\ 0^m & B^{-1} \end{pmatrix}$$

Proposition 1.2.3.1 *Le produit est :*

$$(\hat{B}^{-1} \hat{T}, \hat{d}) = \begin{pmatrix} C_B B^{-1} d & C_B B^{-1} T - C & 1 \\ B^{-1} d & B^{-1} T & 0^m \end{pmatrix} = \begin{pmatrix} a_{00} & a_{0j} & 1 \\ a_{i0} & a_{ij} & 0 \end{pmatrix}$$

Le tableau du simplexe résultant est :

	d	$x_j \quad j = 1, \dots, n$	Z
Z	a_{00}	a_{0j}	1
$x_i \quad i \in I$	a_{i0}	a_{ij}	0^m

TAB. 1.3 – Tableau du simplexe III

1.2.4 Détermination de la forme produit de l'inverse de la base

Soit B_{j-1} la base à l'itération $j - 1$ et a_{lk}^{j-1} le pivot déterminé dans le TS est A_{j-1} .

Proposition 1.2.4.1

$$B_j^{-1} = E_j B_{j-1}^{-1} \quad (1.2.9)$$

tel que E_j la matrice carré d'ordre m composé de $m - 1$ vecteur unité e_i ($i \in I_{j-1} - l$) et du vecteur a_l^k ;
 l ligne pivot et j l'itération.

$$\begin{aligned} a_l^j &= (a_{il}^j \quad i \in I) \\ a_{lk}^j &= \frac{1}{a_{lk}^{j-1}} \\ a_{il}^j &= -\frac{a_{ik}^{j-1}}{a_{lk}^{j-1}} \quad i \in I_{j-1} - l \end{aligned}$$

En utilisant (1.2.9) par récurrence on obtient :

$$B_j^{-1} = \prod_{r=1}^j E_r \quad (1.2.10)$$

La relation (1.2.10) est appelée la forme produit de l'inverse.

Généralisation

La formule (1.2.10) peut se généraliser au cas de la matrice \hat{B}_j .
 Définissant la matrice :

$$\hat{E}_j = \begin{pmatrix} 1 & 0 & \dots & a_{0l}^j & 0 & 0 & 0 \\ & & & & & & \\ & & & & & & \\ 0^m & \dots & E_j & 0 & 0 & 0 \end{pmatrix} \text{ avec } a_{0l}^j = -\frac{a_{0k}^{j-1}}{a_{lk}^{j-1}}$$

Nous obtenons :

$$\hat{B}_j^{-1} = \hat{E}_j \hat{B}_{j-1}^{-1}$$

1.2.5 Algorithme du simplexe révisé

Soit un problème de programmation linéaire de maximisation.

- 1) Ecrire le problème sous sa forme standard.
- 2) Déterminer une base initiale correspondante à une solution de base admissible.
 Construire \hat{B}^{-1} .
- 3) Par la multiplication matricielle $\hat{B}^{-1}(\hat{T}, \hat{d})$, déterminer la ligne des coefficients a_{0j} ($j = \overline{1, n}$).
 - a) Si $a_{0j} \geq 0 \quad \forall j \in J$ STOP.

b) sinon déterminer l'indice $k \in J$ vérifiant :

$$a_{0k} = \min_{j \in J} (a_{0j}).$$

4) Déterminer par multiplication matricielle $\hat{B}^{-1}(\hat{T}, \hat{d})$ la colonne des coefficients a_{ik} ($i \in I$) et celles des coefficients a_{i0} ($i \in I$).

a) Si $a_{ik} \leq 0 \quad \forall i \in I$, la valeur de la fonction objectif n'est pas bornée ($Z \rightarrow \infty$).

b) Sinon, déterminer l'indice l de la ligne du pivot

$$\frac{a_{l0}}{a_{lk}} = \min_{i \in I} \left\{ \frac{a_{i0}}{a_{ik}} \right\}, \quad a_{ik} > 0.$$

Construire la matrice \hat{E} et par la forme produit de l'inverse, la nouvelle matrice de base \hat{B}^{-1} .

Reprendre l'algorithme au point (3).

Critique

Quand le nombre d'itération dépasse certain valeur la forme révisée du simplexe ne sera plus avantageuse, c'est à dire qu'elle ne permet pas toujours de faire des économies de calculs. Bien que le simplexe et sa forme révisée ont été les premières méthodes qui permettent de résoudre les problèmes d'optimisation combinatoires, les chercheurs essayent toujours de développer d'autres méthodes meilleures, efficaces et pertinentes comme c'est le cas pour la méthode ci-dessous.

1.3 La nouvelle méthode de résolution [6, 11]

Phénomène

Soit une grande compagnie qui dispose de n filiales, chaque filiale dispose d'une certaine autonomie pour gérer ses niveaux d'activités x_i , en utilisant au mieux les ressources disponibles b_i et en respectant ses contraintes spécifiques $D_i x_i = b_i \quad i = \overline{1, n}$.

Chaque filiale i n'est pas totalement indépendante et la compagnie doit globalement respecter un certain nombre de contraintes appelées *contraintes communes* ou *contraintes de liaisons*.

$$A_1 x_1 + A_2 x_2 + \dots + A_n x_n \leq b_0$$

Ces contraintes sont relatives à des moyens financiers ou matériels (mains d'oeuvres, gros équipements, ..., etc).

Le contexte cité ci-dessus décrit la structure particulière de certain problème linéaire à grande dimension qui autorise la recherche de la solution optimale par décomposition en petits sous problèmes; résoudre ces sous problèmes nous conduit à déterminer la solution optimale du problème principal.

1.3.1 Formulation mathématique

Soit (P) le problème linéaire suivant :

$$(P) \begin{cases} \max Z = \sum_{i=1}^{i=n} C_i^t x_i \\ \sum_{i=1}^{i=n} A_i x_i \leq b_0 \\ D_i x_i = b_i \\ x_i \geq 0 \quad i = \overline{1 \dots n} \end{cases} \quad (1.3.1)$$

$$x^t = (x_1, x_2, \dots, x_n), \quad x_i^t = (x_i^1, x_i^2, \dots, x_i^{n_i}),$$

$$C_i = (C_i^1, C_i^2, \dots, C_i^{n_i})$$

n_i : représente le nombre de variables qui constitue x_i associé à l'activité i définie par $D_i x_i = b_i$.

$$D_i(r_i, n_i);$$

$$b_i(r_i, 1);$$

$C_i(1, n_i)$; C_i : coefficients de la fonction objectif associé à l'activité i .

$$A_i(m, n_i);$$

$$b_0(m, 1).$$

La matrice des contraintes indépendantes, est diagonale par bloc et D_i constituent les éléments diagonaux de cette matrice :

$$\begin{pmatrix} \boxed{D_1} & & & & \\ & \boxed{D_2} & & & \\ & & \ddots & & \\ & & & & \boxed{D_n} \end{pmatrix}$$

Remarque 3 - $b_i > 0 \quad \forall i = \overline{0, n}$

- Si l'inégalité de la contrainte commune est remplacée par une égalité, le problème de programmation linéaire possède $m + r$ contraintes tel que :

$$r = \sum_{i=1}^n r_i$$

- La méthode du simplexe pour ces Problèmes linéaires à grande dimension est évitable ; car elle possède des insuffisances, ainsi que la forme révisée.

- Une méthode exacte a été développée pour ces Problèmes linéaires à grande dimension, il s'agit de la méthode dite méthode de décomposition qui sera détaillée par la suite.

1.3.2 Méthode de décomposition

Principe de la méthode

L'algorithme de décomposition est à titre principal une méthode de résolution des programmes linéaires à grande taille (grand nombres de variables, et grand nombre de contraintes) qui tient compte de la structure particulière des contraintes, plus précisément qui exploite la

présence d'un nombre important de zéro dans la matrice des coefficients des contraintes cités au paravent.

Il existe deux catégories de problèmes de ce genre, la première catégorie est les problèmes sans contraintes communes et la deuxième catégorie est les problèmes avec contraintes communes, comme c'est expliqué ci-dessous.

(a) Problème sans contraintes communes

(a) veut dire que les différentes activités sont totalement indépendantes, le problème se décompose en n sous problèmes SP_i ($i = \overline{1, n}$).

$$(SP_i) \begin{cases} \max Z_i = C_i^t x_i \\ D_i x_i = b_i \\ x_i \geq 0 \end{cases} \quad (1.3.2)$$

La résolution de chaque sous problème (1.3.2) donne une solution optimale partielle w_i^* ; La solution optimale du problème principal est :

$$w^* = (w_1^*, w_2^*, \dots, w_n^*).$$

(b) Problème avec contraintes communes

Le principe de décomposition est basé sur le fait que $\Delta_i = \{x_i \mid D_i x_i = b_i, x_i \geq 0\}$ est un ensemble borné.

(c) Problème maître associé à (P)

Δ_i est un ensemble convexe et borné, il admet donc un nombre fini J_i de points extrêmes y_{ij} ($j = \overline{1, J_i}$), alors :

$$\forall x_i \in \Delta_i \quad \exists (\beta_{ij})_{1 \leq i \leq n} \quad \beta_{ij} \geq 0 \text{ tel que : } x_i = \sum_{j=1}^{J_i} \beta_{ij} y_{ij} \text{ et } \sum_{j=1}^{J_i} \beta_{ij} = 1 \quad \forall i = \overline{1, n}$$

En remplaçant les x_i par leurs valeurs dans le problème (1.3.1), on aura le problème sous sa forme équivalente de B_{ij} suivant :

$$\begin{cases} \max Z = \sum_{i=1}^n \sum_{j=1}^{J_i} C_i^t \beta_{ij} y_{ij} \\ \sum_{i=1}^n \sum_{j=1}^{J_i} A_i \beta_{ij} y_{ij} \leq b_0 \\ \sum_{j=1}^{J_i} \beta_{ij} = 1 \end{cases} \quad (1.3.3)$$

(1.3.3) est appelé le problème maître associé au problème (P), noté (PM).

(d) Résolution du Problème maître

Si $\beta_i = (\beta_{i1}^*, \beta_{i2}^*, \dots, \beta_{iJ_i}^*)$ désigne la solution optimale du problème maître, alors

$x_i^* = \sum_{j=1}^{J_j} \beta_{ij} y_{ij}$ ($i = \overline{1, n}$), désigne la solution optimale du problème principal.

On définit le coût réduit :

$\hat{C}_{ij} = C_B B^{-1} P_{ij} - C_i y_{ij}$ le vecteur colonne associé à β_{ij} est :

$$P_{ij} = \begin{pmatrix} A_i y_{ij} \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \longrightarrow (i+1) \text{ ligne.}$$

Si $\hat{C}_{ij} > 0$ alors la solution optimale est atteinte.

Sinon, on définit β_{ij} ayant le plus petit coût réduit pour entrer dans la base.

Considérons pour tout i la fonction de coût réduit :

$$\hat{C}_i = C_B B^{-1} \begin{pmatrix} A_i x_i \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} - C_i x_i$$

Soit le Problème :

$$(SP_i) \begin{cases} \min \hat{C}_i \\ D_i x_i = b_i \\ x_i \in \Delta_i \\ x_i \geq 0 \end{cases} \quad (1.3.4)$$

la résolution de (SP_i) nous donne le plus petit coût réduit w_i^* pour chaque ensemble Δ_i , si w_i^* le plus petit coût réduit global alors $w^* = \min (w_1^*, w_2^*, \dots, w_n^*)$.

- Si $w^* > 0$, la solution optimale est atteinte .
- Sinon $w^* < 0$, la variable β_{ij} correspondante est choisie pour entrer dans la base.

Remarque 4 Pour déterminer la variable sortante de la base, on utilise la condition de réalisabilité de la méthode révisée du simplexe.

(e) Description de l'algorithme

Etape 1 : Problème maître

- Déterminer le problème maître associé.
- Déterminer une solution initiale de base admissible.

Etape 2 : Sous problèmes

Pour chaque i déterminer la solution w_i^* du sous problème P_i :

Posons :

$$w^* = \min_{i=1,n} w_i^*.$$

si $w^* \geq 0$ la solution courante est optimale, sinon continuer.

Etape 3 : Simplexe révisé

- Introduire la variable sortante et la nouvelle base B^{-1}
- Retourner à l'étape (2).

Remarque 5 *L'optimum d'un sous problème lié à une variable qui vient de rentrer dans la base à une itération donnée vaut zéro à l'itération suivante.*

Remarque 6 *Le problème (PM) comporte $m + \sum_{i=1}^n r_i$ contraintes par contre un nombre de variables est considérable accru puisqu'il est égal à $\sum_{i=1}^n J_i$ nombre total de sommets des polyèdres Δ_i , il est donc exclu que l'on explicite le problème (PM) mais il serait intéressant d'explorer les programmes de bases du problèmes (PM) à la recherche de celui qui est optimal entre (P) et (PM); telle que l'étude qui a été faite par [11] montre que la méthode de décomposition utilisée donne de bons résultats en comparant avec la méthode de résolution classique (simplexe) en nombre d'itérations et en temps de calculs.*

Exemple 1 :

Appliquant la méthode de décomposition :

$$\begin{cases} \max Z = & 2x_1 + 2x_2 \\ x_1 + x_2 \leq & 6 \\ x_1 \leq & 4 \\ x_2 \leq & 3 \end{cases}$$

Solution

Introduisant les vecteurs s_1, s_2 les deux dernières contraintes qui représentent les contraintes indépendantes.

Les points extrêmes associés à chaque Δ_i :

$$\begin{aligned} \Delta_1 &= \{ (x_1, s_1) \mid x_1 + s_1 = 4 \} \\ &= \{ y_{11}(4, 0) \mid y_{12}(0, 4) \} \\ \Delta_2 &= \{ (x_2, s_2) \mid x_2 + s_2 = 3 \} \\ &= \{ y_{21}(3, 0) \mid y_{22}(0, 3) \} \end{aligned}$$

Le problème maître associé :

$$(PM) \begin{cases} \max Z = \sum_{i=1}^2 C_i^t \beta_{i1} y_{i1} \\ \sum_{i=1}^2 A_i \beta_{i1} y_{i1} \leq b_0 \\ \sum_{j=1}^2 \beta_{ij} = 1 \quad \forall i = 1, 2 \end{cases}$$

Les matrices des coût réduits sont :

$$C_1 = (2, 0)^t, \quad C_2 = (2, 0)^t$$

Les matrices des contraintes communes :

$$A_1 = (1, 0), \quad A_2 = (1, 0)$$

$$\begin{cases} \max Z = 8\beta_{11} + 6\beta_{21} \\ 4\beta_{11} + 3\beta_{21} \leq 6 \\ \beta_{11} + \beta_{12} = 1 \\ \beta_{21} + \beta_{22} = 1 \end{cases}$$

La matrice T des contraintes et la matrice des coûts réduits sont : $T = \begin{pmatrix} \beta_{11} & \beta_{21} \\ 4 & 3 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ et

$$C = \begin{pmatrix} 8 & 6 \\ 0 & 0 \end{pmatrix}$$

Etape 1 : Introduisant la variable d'écart dans la contrainte commune et prenons comme base initiale $(\alpha, \beta_{12}, \beta_{22}) = (6, 1, 1)$.

l'inverse de la matrice de base est $B_0^{-1} = I$ $C_B^t = (0, 0, 0)$.

Etape 2 : Recherche de sous problème

Le coût réduit associé à β_{i1}

$$C_i(x_i) = C_B^t B_0^{-1} \begin{pmatrix} A_i x_i \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} - C_i x_i$$

le 1 se situe à la $i^{\text{ème}}$ ligne.

$$(SP_1) \begin{cases} \min \hat{C}_1 = -2x_1 \\ x_1 + s_1 = 4 \end{cases}$$

La solution optimale est atteinte à $y_{11} = (4, 0)$ et $w_1 = -8$.

$$(SP_2) \begin{cases} \min \hat{C}_2 = -2x_2 \\ x_2 + s_2 = 3 \end{cases}$$

La solution optimale est atteinte à $y_{21} = (3, 0)$ et $w_2 = -6$.

La recherche de la variable qui doit entrer dans la base :

$$\min_{i=1,2} \{w_i\} = \min \{w_1, w_2\} = w_1 = -8.$$

donc la variable correspondante à l'optimum w_1 est β_{11} . Alors β_{11} entre dans la base, le vecteur

colonne associé à β_{11} est $P_{11} = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix}$ et on a : $d_0 = (6, 1, 1)$.

on a $\min\{\frac{6}{4}, 1\} = 1$ donc la variable β_{11} entre au niveau de la deuxième ligne et β_{21} est la variable sortante.

Alors l'inverse de la matrice de base est obtenue par :

$$B_1^{-1} = E_0 B_0^{-1}.$$

E_0 est obtenue en remplaçant la deuxième colonne de B_0^{-1} par le pseudo inverse de P_{11} ,

$\xi_2 = (-4, 1, 0)$ on trouve :

$$B_1^{-1} = E_0 B_0^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$d_1 = B_1^{-1} d_0 = B_1^{-1} \cdot \begin{pmatrix} 6 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

$$\begin{pmatrix} \alpha_1 \\ B_{11} \\ B_{22} \end{pmatrix} = \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

La nouvelle matrice de coût réduit associé aux variables de base est $C_{B_1}^t = (0, C_1^t Y_{11}, 0)$

Etape 3 :

β_{11} vient de rentrer dans la base, l'optimum correspondant à SP_1 est nul, il est inutile de chercher SP_1 . Retrouvons le sous problème associé à β_{21} .

$$\hat{C}_2 = C_{B_1}^t B_1^{-1} \begin{pmatrix} A_2 x_2 \\ 0 \\ 1 \end{pmatrix} - C_2 x_2$$

$$C_{B_1}^t B_1^{-1} = (0, 8, 0) \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ 0 \\ 1 \end{pmatrix} - 2x_2$$

$$\hat{C}_2 = (0, 8, 0) \begin{pmatrix} x_2 \\ 0 \\ 1 \end{pmatrix} - 2x_2$$

$$(SP_2) \begin{cases} \min \hat{C}_2 & = -2x_2 \\ x_2 + s_2 & = 3 \end{cases}$$

$w_2 = -6 < 0$ donc la variable correspondante à la solution w_2 est β_{21} . Alors β_{21} entre dans la base, le vecteur colonne associé à β_{21} est :

$$P_{21} : B_1^{-1} \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} \text{ et } d_1 = B_1^{-1} d_0 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

on a $\min\{\frac{2}{3}, 1\} = \frac{2}{3}$ donc β_{21} entre au niveau de la première ligne. Alors α est la variable sortante.

alors l'inverse de la matrice de base est obtenu par $B_2^{-1} = E_1 B_1^{-1}$. E_1 est obtenue en remplaçant la première colonne de B_0^{-1} par le pseudo-inverse de P_{21} $\xi_1 = (\frac{1}{3}, 0, \frac{-1}{3})$ on trouve

$$E_1 = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-1}{3} & 0 & 1 \end{pmatrix}$$

$$B_2^{-1} = E_1 B_1^{-1} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$B_2^{-1} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

on aura :

$$\begin{pmatrix} B_{21} \\ B_{11} \\ B_{22} \end{pmatrix} = B_2^{-1} d_0 = \begin{pmatrix} \frac{1}{3} & -\frac{4}{3} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{3} & \frac{4}{3} & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} \\ 1 \\ \frac{1}{3} \end{pmatrix}$$

B_{21} vient d'entrer dans la base et donc l'optimum est nul, il est inutile de chercher SP_2 .

On a :

$$\min_{i=1,2} \{w_i\} = 0$$

donc la solution optimale est atteinte : $x_1^* = (4, 0)^t$ et $x_2^* = (3, 0)^t$

$$x_1^* = \beta_{11} y_{11} = 1(4, 0)^t + \beta_{12} y_{12} = 1(4, 0)^t = (4, 0)^t$$

$$x_2^* = \beta_{21} y_{21} + \beta_{22} y_{22} = \frac{2}{3}(3, 0)^t = (2, 0)^t$$

$$w^* = (2, 0)(4, 0)^t + (2, 0)(2, 0)^t = 8 + 4 = 12$$

La nouvelle matrice de coût réduit associé aux variables de bases $(\beta_{21}, \beta_{11}, \beta_{22})$ est

$$C_{B_2}^t = (C_2^t Y_{21}, C_1^t Y_{11}, 0).$$

Remarquons que cette méthode de résolution à pour objectif, résoudre le problème initial, même s'il est à grande dimension ; c'est résoudre des sous problèmes simples à une seule variable.

La méthode de décomposition est un outil efficace pour résoudre les problèmes d'optimisation à grande dimension, la matrice des contraintes de ces problèmes est souvent creuse, donc il peut arriver que l'entrée d'une variable ne modifie pas les vecteurs associés à certaines variables. Il serait intéressant de voir est-ce qu'il y aura une possibilité de faire entrer plusieurs variables dans la base simultanément ? Si oui, avec ce cheminement le temps de calcul diminuera certainement, ainsi que les erreurs accumulées.

1.4 Les différentes rentrées de variables

Nous décrirons la définition des différentes rentrées de variables dans la base.

1.4.1 Variables de rentrée totalement indépendante

Soit le problème linéaire suivant :

$$(P) \begin{cases} \max \sum_{i=1}^{i=n} C_i^t x_i \\ \sum_{j=1}^{j=n} a_{ij} x_j + s_i = b_i \quad i = \overline{1, n} \end{cases}$$

Toutes les définitions sont considérées par rapport à (P)

Définition 1 Soit x_p une variable de vecteur colonne $P_{x_p} = (a_{1p}, a_{2p}, \dots, a_{mp})^t$.

On dira que le coefficient : a_{i_0p} ($i_0 \in \{\overline{1, m}\}$) est qualifié au pivot à l'itération t_0 si quelque soit l'itération $t < t_0$ on a :

$$-a_{i_0} > 0.$$

-la valeur de a_{i_0p} reste inchangée par rapport à celle qu'il avait initialement.

Définition 2 Soit x_p une variable de vecteur colonne associé $P_{x_p} = (a_{1p}, a_{2p}, \dots, a_{mp})^t$

On appelle pseudo inverse de P_{x_p} le vecteur défini par :

$$\xi_p = \left(-\frac{a_{1p}}{a_{sp}}, -\frac{a_{2p}}{a_{sp}}, \dots, \frac{1}{a_{sp}}, -\frac{a_{s+1p}}{a_{sp}}, \dots, -\frac{a_{mp}}{a_{sp}} \right)^t, \text{ où } a_{sp} \text{ est le pivot à l'itération considérée.}$$

Définition 3 Deux variables hors base x_p et x_q sont dites de rentrée totalement indépendante à une itération t_0 si à cette itération leurs vecteurs colonnes respectifs associés

$$P_{x_p} = (a_{1p}, a_{2p}, \dots, a_{mp})^t \text{ et } P_{x_q} = (a_{1q}, a_{2q}, \dots, a_{mq})^t \text{ vérifient : } a_{ip} \neq 0 \Rightarrow a_{iq} = 0.$$

De tels vecteurs se présentent sous la forme :

$$P_{x_p} = \begin{pmatrix} 0 \\ \vdots \\ a_{lp} \\ 0 \\ a_{(l+2)p} \\ \vdots \\ a_{ip} \\ 0 \\ \vdots \\ a_{(i+k)p} \\ 0 \\ \vdots \end{pmatrix} \quad \text{et} \quad P_{x_q} = \begin{pmatrix} a_{1p} \\ \vdots \\ 0 \\ a_{(l+1)q} \\ 0 \\ \vdots \\ 0 \\ a_{(i+1)q} \\ \vdots \\ 0 \\ a_{(i+k+1)q} \\ \vdots \end{pmatrix}$$

Définition 4 Une variable x_p de vecteur colonne $P_{x_p} = (a_{1p}, a_{2p}, \dots, a_{mp})^t$ est dite libre à la ligne i_0 ($i_0 \in \{1, \dots, m\}$) si $a_{i_0p} = 0$.

Généralisation de la forme produit

Considérons les variables x_p, x_q de vecteurs colonnes respectifs à P_{x_p}, P_{x_q} et ayant respectivement comme pivot a_{i_0p}, a_{j_0q} à l'itération t_0 . Si elles sont de rentrée totalement indépendante à cette itération alors elles peuvent entrer simultanément dans la base.

Si de plus, pour toute itération $t < t_0$, les variables situées à la ligne i_0 (j_0) des différentes bases sont libres (sauf la variable de base s_{i_0} (s_{j_0})), dans ce cas l'inverse de la matrice de base courante B^{-1} résultante est obtenue en remplaçant les colonnes numéro i_0 et j_0 de $B_{t_0-1}^{-1}$ respectivement par les pseudo inverses de P_{x_p} , et P_{x_q} .

Preuve : Voir[11].

1.4.2 Variables de rentrée croisée

Définition 5 Deux variables x_p et x_q de vecteurs colonnes P_{x_p} et P_{x_q} sont de rentrée croisée à l'itération t_0 si à cette itération :

- elles sont libres respectivement aux lignes j_0 et i_0 ($i_0 \neq j_0$).
- les coefficients a_{i_0} et a_{j_0} se présentent comme pivots respectifs de x_p et x_q .

En d'autres termes ; elles sont de rentrées croisées à l'itération t_0 si :

- Le pivot de x_p se trouve sur une ligne où x_q est libre ;
- Celui de x_q se trouve sur une ligne où x_p est libre.

Les vecteurs colonnes P_{x_p} et P_{x_q} s'écrivent :

$$P_{x_p} = \begin{pmatrix} a_{1p} \\ \vdots \\ a_{i_0 p} \\ a_{(i_0+1)p} \\ \vdots \\ 0 \\ a_{(j_0+1)p} \\ \vdots \end{pmatrix} \quad \text{et} \quad P_{x_q} = \begin{pmatrix} a_{1q} \\ \vdots \\ 0 \\ a_{(i_0+1)q} \\ \vdots \\ a_{j_0 q} \\ a_{(j_0+1)q} \\ \vdots \end{pmatrix}$$

Généralisation de la forme produit

Deux variables hors base x_p et x_q de vecteurs colonnes respectifs P_{x_p} et P_{x_q} de rentrée croisée à l'itération t_0 peuvent rentrer simultanément dans la base.

Notons i_0 et j_0 les lignes au niveau x_p et x_q entrent, si pour toute itération $t < t_0$ les variables des différentes bases sont libres i_0 et j_0 (sauf respectivement les variables de base s_{i_0} et s_{j_0}) alors l'inverse de la matrice de base courante $B_{i_0}^{-1}$ résultant est obtenu en remplaçant les colonnes numéros i_0 et j_0 de $B_{i_0-1}^{-1}$ par les pseudo inverses de P_{x_p} et P_{x_q} .

Preuve : Voir [11]

1.4.3 Variables de rentrée liée

Définition 6 Etant donné deux variables hors base x_p et x_q de vecteurs colonnes respectifs $P_{x_p} = (a_{1p}, a_{2p}, \dots, a_{mp})^t$ et $P_{x_q} = (a_{1q}, a_{2q}, \dots, a_{mq})^t$.

On dira que x_q est simplement liée à l'itération t_0 si à cette itération le pivot de x_p se trouve sur une ligne où x_q est libre.

Les vecteurs colonnes de telles variables se présentent sous la forme :

$$P_{xp} = \begin{pmatrix} a_{1p} \\ a_{2p} \\ \vdots \\ a_{i_0p} \\ a_{i_0+1p} \\ \vdots \\ a_{mp} \end{pmatrix} \quad \text{et} \quad P_{xq} = \begin{pmatrix} a_{1q} \\ a_{2q} \\ \vdots \\ 0 \\ a_{i_0+1q} \\ \vdots \\ a_{mq} \end{pmatrix}$$

où a_{i_0p} le pivot de x_q .

Généralisation de la forme produit

Deux variables hors base x_p et x_q de vecteurs colonnes respectifs P_{xp} et P_{xq} de rentrée croisée à l'itération t_0 peuvent rentrer simultanément dans la base. Notons i_0 et i_0 les lignes au niveau desquelles entrent x_p et x_q , si pour toute itération $t < t_0$ les variables des différentes bases sont libres i_0 et j_0 (sauf respectivement les variables de base s_{i_0} et s_{j_0}) alors l'inverse de la matrice de base courante $B_{t_0}^{-1}$ résultant est obtenu en remplaçant les colonnes numéros i_0 et j_0 de $B_{t_0-1}^{-1}$ par les pseudo inverses de P_{xp} et P_{xq} .

L'exemple ci-dessous exploitera l'entrée simultanément croisée des variables.

Exemple 2 soit le problème linéaire suivant :

$$(P) \left\{ \begin{array}{l} \max Z = 5x_1 - x_2 - x_3 + 3x_4 + x_5 \\ 2x_1 + x_2 + 3x_3 + x_4 + 4x_5 \leq 20 \\ x_1 \leq 3 \\ \quad x_2 \leq 2 \\ \quad \quad x_3 \leq 3 \\ \quad \quad \quad x_4 \leq 2 \\ \quad \quad \quad \quad x_5 \leq 1 \\ x_i \geq 0 \quad i = \overline{1, n} \end{array} \right.$$

Solution

Introduisons les variables d'écart s_1, s_2, s_3, s_4, s_5 dans les cinq dernières contraintes qui représentent les contraintes indépendantes. Les points extrêmes associées à chaque ensemble Δ_i :

$$\begin{aligned} \Delta_1 &= \{(x_1, s_1) \mid x_1 + s_1 = 3\} = \{y_{11}(3, 0)^t, y_{12}(0, 3)^t\} \\ \Delta_2 &= \{(x_2, s_2) \mid x_2 + s_2 = 2\} = \{y_{21}(2, 0)^t, y_{22}(0, 2)^t\} \\ \Delta_3 &= \{(x_3, s_3) \mid x_3 + s_3 = 3\} = \{y_{31}(3, 0)^t, y_{32}(0, 3)^t\} \\ \Delta_4 &= \{(x_4, s_4) \mid x_4 + s_4 = 2\} = \{y_{41}(2, 0)^t, y_{42}(0, 2)^t\} \\ \Delta_5 &= \{(x_5, s_5) \mid x_5 + s_5 = 1\} = \{y_{51}(1, 0)^t, y_{52}(0, 1)^t\} \end{aligned}$$

Le problème maître associé s'écrit :

$$(PM) \left\{ \begin{array}{l} \max Z = \sum_{i=1}^5 C_i^t \beta_{i1} y_{i1} \\ \sum_{i=1}^5 A_i \beta_{i1} y_{i1} \leq b_0 \\ \sum_{j=1}^2 \beta_{ij} = 1 \quad \forall i = \overline{1,5} \end{array} \right.$$

les matrices des coûts réduits sont :

$$C_1 = (5, 0)^t, \quad C_2 = (-1, 0)^t, \quad C_3 = (-1, 0)^t, \quad C_4 = (3, 0)^t, \quad C_5 = (1, 0)^t.$$

les matrices de contraintes communes :

$$A_1 = (2, 0) \quad A_2 = (1, 0) \quad A_3 = (3, 0) \quad A_4 = (1, 0) \quad A_5 = (4, 0).$$

$$\left\{ \begin{array}{l} \max Z = 15\beta_{11} - 2\beta_{21} - 3\beta_{31} + 6\beta_{41} + \beta_{51} \\ 6\beta_{11} + 2\beta_{21} + 9\beta_{31} + 2\beta_{41} + 4\beta_{51} \leq 20 \\ \beta_{11} + \beta_{12} = 1 \\ \beta_{21} + \beta_{22} = 1 \\ \beta_{31} + \beta_{32} = 1 \\ \beta_{41} + \beta_{42} = 1 \\ \beta_{51} + \beta_{52} = 1 \end{array} \right.$$

La matrice des contraintes :

$$T = \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} & \beta_{41} & \beta_{51} \\ 6 & 2 & 9 & 2 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

la matrice des coûts réduits :

$$C = \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} & \beta_{41} & \beta_{51} \\ 15 & -2 & -3 & 6 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Etape 0 :

On introduit la variable d'écart α , dans la première contrainte et on prends comme base initiale $(\alpha, \beta_{12}, \beta_{22}, \beta_{32}, \beta_{42}, \beta_{52})^t$, la solution correspondante est

$(\alpha, \beta_{12}, \beta_{22}, \beta_{32}, \beta_{42}, \beta_{52})^t = (20, 1, 1, 1, 1, 1)^t$ l'inverse de la matrice de base courante est $B_0^{-1} = I$, et $C_{B_0} = (0, 0, 0, 0, 0, 0)$

Etape 1 : Le coût réduit associé à β_{i1} est :

$$C_i(x_i) = C_{B_0}^t B_0^{-1} \begin{pmatrix} A_i x_i \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} - C_i x_i$$

1 se situe à la i ème ligne.

Le premier sous problème est :

$$(SP_1) \begin{cases} \min \hat{C}_1 = -5x_1 \\ x_1 + s_1 = 3 \end{cases}$$

la solution optimale est atteinte pour $(x_1, s_1) = y_{11} = (3, 0)$ et, $w_1 = -15$.

Le deuxième sous problème est :

$$(SP_2) \begin{cases} \min \hat{C}_2 = x_2 \\ x_2 + s_1 = 2 \end{cases}$$

la solution optimale est atteinte pour $(x_2, s_2) = y_{22} = (0, 2)$ et, $w_2 = 0$.

Le troisième sous problème est :

$$(SP_3) \begin{cases} \min \hat{C}_3 = x_3 \\ x_3 + s_3 = 3 \end{cases}$$

la solution optimale est atteinte pour $(x_3, s_3) = y_{32} = (0, 3)$ et, $w_3 = 0$.

Le quatrième sous problème est :

$$(SP_4) \begin{cases} \min \hat{C}_4 = -3x_4 \\ x_4 + s_4 = 2 \end{cases}$$

la solution optimale est atteinte pour $(x_4, s_4) = y_{41} = (2, 0)$ et, $w_4 = -6$.

Le cinquième sous problème est :

$$(SP_5) \begin{cases} \min \hat{C}_5 = -x_5 \\ x_5 + s_5 = 1 \end{cases}$$

la solution optimale est atteinte pour $(x_5, s_5) = y_{51} = (1, 0)$ et, $w_5 = -1$.

Recherche de la variable qui doit entrer dans la base :

On a $\min \{w_1, w_2, w_3, w_4, w_5\} = w_1 = -15$ alors la variable β_{11} doit entrer dans la base, le vecteur colonne associé est $P_{11} = (6, 1, 0, 0, 0)^t$ et $B_0^{-1}P_{11} = (6, 1, 0, 0, 0)^t$

$b_0 = B_0^{-1}b = (20, 1, 1, 1, 1)^t$ on a $\min\{\frac{20}{6}, 1\} = 1$, alors β_{11} entre au niveau de la deuxième ligne donc la variable sortante de la base est β_{12} . Comme P_{41} et P_{51} les vecteurs colonne associés aux variables β_{41} et β_{51} sont nuls à cette contrainte alors cherchons si β_{41} , β_{11} , β_{51} sont de rentrée croisée afin de les faire entrer ensemble dans la base. L'inverse de la matrice courante est :

$$B_1^{-1} = \begin{pmatrix} 1 & -6 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b_1 = B_1^{-1}b_0 = \begin{pmatrix} 14 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$w_4^* < w_5^*$, β_{41} entre dans la base

$$P_{41} = B_1^{-1}P_{41} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$\min\{\frac{14}{2}, 1\} = 1$ donc β_{41} entre au niveau de la cinquième ligne, alors β_{42} est la variable sortante. L'entrée de β_{41} ne modifie pas les vecteurs colonnes P_{11} et P_{51} associés aux variables β_{11} et β_{51} car β_{41} entre à la première ligne où les autres vecteurs sont libres. L'inverse de la matrice base courante est obtenue en remplaçant le cinquième vecteur colonne de B_1^{-1} par le pseudo-inverse de P_{41} , $\xi_5 = (-2, 0, 0, 0, 1, 0)$

$$B_2^{-1} = \begin{pmatrix} 1 & -6 & 0 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad B_2^{-1}P_{51} = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad B_2^{-1}P_{51} = \begin{pmatrix} 12 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$\min\{\frac{12}{4}, 1\} = 1$ β_{51} entre au niveau de la 6^{ème} ligne, l'entrée de β_{51} ne modifie pas P_{41}, P_{11} donc $\beta_{41}, \beta_{11}, \beta_{51}$ peuvent entrer simultanément dans la base, donc l'inverse de la matrice courante est :

$$B_3^{-1} = \begin{pmatrix} 1 & -6 & 0 & 0 & -2 & -4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

alors :

$$b_3 = \begin{pmatrix} \alpha_1 \\ B_{11} \\ B_{22} \\ B_{32} \\ B_{41} \\ B_{51} \end{pmatrix} = B_3^{-1} \begin{pmatrix} 20 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Etape2 : $\beta_{41}, \beta_{11}, \beta_{51}$ viennent de rentrer dans la base. Les optimums des $(SP_4), (SP_1), (SP_5)$ sont nuls, il est inutile de chercher les sous problèmes associés. Retrouvons les optimums de $(SP_2), (SP_3)$

$$\hat{C}_2 = C_{B_3}^t B_3^{-1} \begin{pmatrix} A_2 x_2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - C_2 x_2$$

$$(SP_2) : \begin{cases} \min \hat{C}_2 = x_2 \\ x_2 + S_2 = 2 \end{cases}$$

$w_2^* = 0$ alors y_{22} est la solution optimale.

$$(SP_3) : \begin{cases} \min \hat{C}_3 & = x_3 \\ x_3 + S_3 & = 3 \end{cases}$$

$w_3^* = 0$ alors y_{32} est la solution optimale.

On a $\min_{i=1,5} \{w_i\} = 0$, alors la solution optimale est atteinte :

$$\begin{aligned} x_1^* = y_{11} &= (3, 0)^t, & x_2^* = y_{22} &= (0, 2)^t, & x_3^* = y_{32} &= (0, 3)^t, & x_4^* = y_{41} &= (2, 0)^t, \\ x_5^* = y_{51} &= (1, 0)^t, & & & & & & \text{et la solution optimale du problème est } w^* = 22. \end{aligned}$$

Conclusion

La méthode de décomposition est une méthode de résolution des problèmes d'optimisation à grande dimension, son rôle principal est de décomposer le problème en sous problème, trouver la solution optimale de ces sous problèmes nous permet de trouver la solution optimale du problème initial. Ainsi, les problèmes d'optimisation à grande dimension ont souvent une matrice de contrainte creuse, telle que l'entrée d'une variable ne modifie pas d'autres colonnes qui vont à leurs tours entrer dans la base, d'où vient l'idée de faire entrer simultanément plusieurs variables dans la base. Cette démarche certainement diminuera le nombre d'itérations et beaucoup de calculs.

Notre premier objectif est d'appliquer cette méthode en adaptant l'entrée simultanée de plusieurs variables pour le problème fameux d'optimisation combinatoire, il s'agit du problème de sac à dos.

Deuxième partie

Problème d'Optimisation Combinatoire Uni-critère

1.5 Problème d'Optimisation Combinatoire Uni-critère

1.5.1 Introduction

Dans de nombreuses situations réelles modélisables par la programmation mathématique toutes les variables sont astreintes à être entières, c'est le cas où on parle d'un programme linéaire en nombres entiers tel que la présence des variables discrètes est inévitable. Leur domaine des solutions réalisables n'est plus convexe contrairement aux programmes linéaires, il en résulte que la programmation linéaire en nombres entiers est plus complexe que la programmation linéaire. Des algorithmes ont été développés pour résoudre les programmes linéaires en nombres entiers (méthode par séparation et évaluation, méthodes de coupes,... etc), il serait intéressant de produire des méthodes de résolution spécialisées pour ces programmes.

On verra par la suite les différentes méthodes de résolution concernant ce type de problème.

1.5.2 programmation linéaire en nombres entiers

La formulation d'un programme linéaire est donnée par :

$$(LP) \begin{cases} \max Z = C^t x \\ Ax = b \\ x \geq 0 \end{cases} \quad (1.5.1)$$

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

$$(ILP) \begin{cases} \max Z = C^t x \\ Ax = b \\ x_j \in \mathbb{N} \end{cases} \quad (1.5.2)$$

Dans le cas particulier où les contraintes $x_j \in \mathbb{N}$ sont remplacées par $x_j \in \{0, 1\}^n$, on dit qu'on a "un programme linéaire 0, 1" ou "problème linéaire en variables bivalents" écrit sous la forme suivante :

$$(LPBI) \begin{cases} \max Z = C^t x \\ Ax = b \\ x \in \{0, 1\}^n \end{cases} \quad (1.5.3)$$

Remarque 7 A, b, C sont des matrices entières ou peuvent être ramenés à des entiers. (Consulter [7]).

1.5.3 problèmes d'optimisation à variables binaires

Il existe un nombre considérable de problèmes en nombres entiers qui est formulé comme problèmes à variables qui ne peuvent prendre que deux variables 0 ou 1, ce type de problème est noté (BLP), citons par exemple le problème d'affectation (AP), le problème de sac à dos (KP), le problème de couverture d'ensembles (SCP), production d'une pièce, ..., etc.

On s'intéressera qu'à deux problèmes fameux, le problème d'affectation (AP) et le problème de sac à dos (KP) et leurs méthodes de résolution.

1.5.4 Le problème de Sac à dos (KP)

On dispose de n objets ayant chacun un poids A_j et un profit C_j ($j = \overline{1, n}$). Il faut effectuer une sélection dont le poids total soit inférieur ou égal à un nombre donné b et dont la valeur, somme des valeurs des objets sélectionnés soit maximum, on cherche $J \subset \{1, 2, \dots, n\}$ tel que $\sum_{j \in J} C_j$ soit maximum sous la contrainte $\sum_{j \in J} A_j \leq b$.

Associons à chaque objet j une variable x_j qui prend la valeur 1 si le $j^{\text{ème}}$ objet fait partie de la sélection ($j \in J$), 0 sinon.

Le problème de sac à dos se formule de la manière suivante :

$$(KP) \begin{cases} \max(Z) = \sum_{j=1}^n C_j x_j \\ \sum_{j=1}^n A_j x_j \leq b \\ x_j \in \{0, 1\} \quad j = \overline{1, n} \end{cases} \quad (1.5.4)$$

Ce modèle théorique est particulièrement intéressant, il est le plus utilisé dans l'optimisation combinatoire, comme il a de nombreuses applications pratiques. Pour cette raison plusieurs formes de ce problème (formes particulières, formes générales) ont été examinées et étudiées par un ensemble de chercheurs.

La forme généralisée

La forme est présentée comme suit :

$$(KPg) \begin{cases} \max(Z) = \sum_{j=1}^n C_j x_j \\ \sum_{j=1}^n A_j x_j \leq b \\ x_j \in \{0, 1, \dots, b\} \quad j = \overline{1, n} \end{cases} \quad (1.5.5)$$

1.5.5 Le problème d'affectation (AP)

Le problème d'affectation est présenté sous le contexte suivant, n personnes doivent être affectées à n travaux, chaque personne doit s'effectuer à un travail (ou une tâche) et un(e) seul(e), le **coût de formation** de la $i^{\text{ème}}$ personne lorsqu'elle est affectée au $j^{\text{ème}}$ travail est c_{ij} . Il s'agit d'affecter les personnes aux travaux de sorte que la somme des coûts de formation soit minimum.

Ce problème peut se formuler comme un programme linéaire à variables bivalentes en posant :

$$x_{ij} = \begin{cases} 1 & \text{si la } i^{\text{ème}} \text{ personne est affectée au } j^{\text{ème}} \text{ travail} \\ 0 & \text{sinon} \end{cases} \quad (1.5.6)$$

On a alors :

$$(AP) \begin{cases} \min(Z) = \sum_{j=1}^n \sum_{i=1}^n C_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} = 1 \quad i = \overline{1, n} \\ \sum_{i=1}^n x_{ij} = 1 \quad j = \overline{1, n} \\ x_{ij} \in \{0, 1\} \quad i, j = \overline{1, n} \end{cases} \quad (1.5.7)$$

$\sum_{j=1}^n x_{ij} = 1$ veut dire que chaque tâche est prise en charge par une personne et une seule.

$\sum_{i=1}^n x_{ij} = 1$ veut dire que chaque personne s'est vue affecter à un travail et un seul.

(1.5.7) est le problème primal, son dual est écrit de la manière suivante :

$$(D) \begin{cases} \max \left\{ \sum_{i=1}^n u_i + \sum_{j=1}^n v_j \right\} \\ u_i + v_j \leq C_{ij} \quad \forall i, j = \overline{1, n} \end{cases} \quad (1.5.8)$$

Remarque 8 Le problème (AP) peut être résolu en ignorant les contraintes d'intégrité sur les variables $x_{ij} \in \{0, 1\}$. C'est à dire que toute solution optimale de base de (AP') :

$$(AP') \begin{cases} \min(Z) = \sum_{j=1}^n \sum_{i=1}^n C_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} = 1 \quad i = \overline{1, n} \\ \sum_{i=1}^n x_{ij} = 1 \quad j = \overline{1, n} \\ x_{ij} \geq 0 \quad i, j = \overline{1, n} \end{cases} \quad (1.5.9)$$

est solution optimale de (AP).

Remarque 9 Pour toute solution réalisable de base du problème d'affectation, on aura $(2n - 1)$ variables dans la base et seulement n variables égales à 1. Le problème d'affectation est très dégénérée (toujours $n - 1$ variables de base égale à 0).

Définition 7 Problème de dégénérescence

La solution de base dégénérée : Une solution de base est dite dégénérée si l'une au moins des variables de base est égale à zéro, c'est à dire si les vecteurs e_1, \dots, e_m forment une base de l'espace à m dimensions des contraintes, la combinaison linéaire à coefficients non négatif :

$$e_0 = x_1 e_1 + \dots + x_n e_n.$$

En l'absence de dégénérescence, l'algorithme du simplexe converge vers la solution optimale en un nombre fini de pas. En effet, il n'y a qu'un nombre fini de solutions de base réalisables (aucune base ne se présente deux fois).

En cas de dégénérescence, ces arguments ne sont plus valables, il pourrait en effet arriver que l'on effectue plusieurs changements de base successifs sans que la valeur de la fonction économique ne varie, et que l'on retrouve après un certain nombre de bases une solution de base réalisable déjà rencontrée. L'algorithme recommencera alors indéfiniment le cycle des changements de base et ne convergera plus vers la solution optimale. Le cyclage est donc une séquence accidentelle de la dégénérescence.

Dans les problèmes pratiques de tailles raisonnables, cette situation ne se présente pas. Il est très fréquent de rencontrer dans les grands problèmes de très longues suites de changement de base sans augmentation de la fonction objectif, ce qu'on appelle des pseudo cycle. Plusieurs méthodes ont été proposées pour éviter ce cyclage. La plus utilisée est la méthode de perturbation décrite dans [17] et dans d'autres références .

1.5.6 Méthodes de résolution

Pour les programmes linéaires en nombres entiers : Lorsqu'on relâche les contraintes d'intégrité sur les variables $x_j \in \mathbb{N}$ dans le programme linéaire en nombres entiers (ILP). On obtient le programme linéaire :

$$(LP') \begin{cases} \max Z = C^t x \\ Ax = b \\ x \geq 0 \end{cases} \quad (1.5.10)$$

Par construction , il existe une relation étroite entre (ILP) et (LP'). Si par hasard la solution optimale de (LP') est entière, c'est aussi la solution optimale de(ILP). Donc la résolution de (LP') peut aider à résoudre(ILP), et on verra la méthode dans laquelle la résolution d'un programme linéaire en nombres entiers passe par la résolution du programme linéaire associé.

Il existe deux types principaux de méthodes pour résoudre un programme linéaire en nombres entiers :

- 1) Les méthodes d'optimisation par coupe (Coupe de Dantzig, de Gomory, ...).
- 2) La méthode de séparation et évaluation (Branch and Bound).

Les méthodes de coupe de Gomory

Principe des méthodes de coupe

Il est possible de résoudre un (ILP) en passant par la résolution d'un programme linéaire en variables continues.

La 1ère méthode

Une première méthode se base sur le théorème suivant :

Theorem 1 Si D désigne l'ensemble des solutions réalisables d'un (ILP) et $D+$ la fermeture convexe de D , le (LP) ayant $D+$ comme domaine réalisable admet la même solution optimale que le (ILP) initial. (la recherche de $D+$ souvent n'est pas un problème simple)

2ème méthode

Les méthodes de coupe :

$$(ILP) \begin{cases} \max Z = C^t x \\ Ax = b \\ x \in \mathbb{N} \end{cases} \quad (1.5.11)$$

Le principe de la méthode consiste à :

Ajouter au (ILP) et au (LP) associé une suite d'inégalités supplémentaires, appelées coupes, qui sont nécessairement satisfaites par les solutions entières qui ne modifient donc pas l'ensemble des solutions réalisables du (ILP) mais qui amputent le domaine des solutions réalisables du (LP), et ce jusqu'à l'obtention pour le (LP) d'un sommet optimal entier (qui est donc solution optimale du (ILP)). Il existe plusieurs méthodes de coupe, rappelons la méthode des coupes de Gomory.

Théorème de base de l'algorithme de Gomory

En résolvant le (LP) par le simplexe, nous obtenons le tableau de simplexe final du type :

	d	$x_i \quad i \in I$	$x_j \quad j \in J$
Z	a_{00}	0	a_{0j}
$x_i \quad i \in I$	a_{i0}	I	a_{ij}

TAB. 1.4 –

$I = \{1, \dots, m\}$ l'ensemble des indices des variables de base B.

$J = \{m + 1, \dots, n\}$ l'ensemble des indices des variables hors base B.

Dans cette base, (ILP) et (LP) s'écrivent sous la forme suivante :

$$(ILP) \begin{cases} \max Z = a_{00} - \sum_{j \in J} a_{0j} x_j \\ x_i + \sum_{j \in J} a_{ij} x_j \leq a_{i0} \quad i \in I \\ x_j \in \mathbb{N} \end{cases} \quad (1.5.12)$$

$$(LP) \begin{cases} \max Z = a_{00} - \sum_{j \in J} a_{0j} x_j \\ x_i + \sum_{j \in J} a_{ij} x_j \leq a_{i0} \quad i \in I \\ x_j \geq 0 \end{cases} \quad (1.5.13)$$

Les contraintes du (ILP) s'écrivent sous la forme $x_i + \sum_{j \in J} a_{ij} x_j \leq a_{i0}$ pour ($i \in I$) avec $x_j \geq 0$

et entière, J l'ensemble des indices hors base.

Si la solution de base optimale est non entière, il existe une ligne i avec $a_{i0} \notin \mathbb{N}$, cette dernière équation peut être écrite comme :

$$x_i = [a_{i0}] + f_{i0} - \sum_{j \in J} [a_{ij}] x_j - \sum_{j \in J} f_{ij} x_j.$$

(On utilise $a_{i0} = [a_{i0}] + f_{i0}$ tel que $0 \leq f_{i0} < 1$).

$[a_{i0}]$: la partie entière de a_{i0} .

f_{i0} : la partie fractionnaire de a_{i0} .

$$\begin{cases} a_{ij} = [a_{ij}] + f_{ij} \\ 0 \leq f_{i0} < 1 \end{cases} \quad (1.5.14)$$

Theorem 2 Une condition nécessaire pour qu'une solution réalisable du (LP) soit entière est que :

$$f_{i0} + S_i = \sum_{j \in J} f_{ij} x_j, \quad i \in I, \quad (S_i) \text{ est une variable d'écart } (S_i \in \mathbb{N}).$$

Démonstration (Voir[1])

Déroulement de l'algorithme de Gomory

Etape 1

Résoudre le (LP) associé par l'algorithme du simplexe. On envisage deux cas :

- 1) Si les composantes de cette solution sont entières, passer à l'étape (3).
- 2) Si l'une au moins des composantes de cette solution n'est pas entière, passer à l'étape (2).

Etape 2

La solution optimale du (LP) n'est pas réalisable pour (ILP). Soit x_λ^0 une composante non entière de cette solution, A partir de cette composante non entière x_λ^0 , définir et ajouter au (ILP) et (LP) une nouvelle contrainte appelée coupe de Gomory.

Pour (ILP) :

$$\begin{cases} f_{\lambda 0} = \sum_{j \in J} f_{\lambda j} x_j - s_\lambda \\ s_\lambda \in \mathbb{N} \end{cases} \quad (1.5.15)$$

Pour (LP) :

$$\begin{cases} f_{\lambda 0} = \sum_{j \in J} f_{\lambda j} x_j - s_\lambda \\ s_\lambda \geq 0 \end{cases} \quad (1.5.16)$$

On examine l'effet de cette contrainte supplémentaire :

Le (ILP) ainsi complété optimale est équivalent au (ILP) initial, en effet, cette contrainte est toujours satisfaite par les solutions réalisables du (ILP) puisque c'est une condition nécessaire d'intégrité.

Le (LP) ainsi complété est plus restrictif que (LP) initial, en effet, cette contrainte n'est pas satisfaite par la solution optimale a_0 du (LP) puisque $f_{\lambda 0}$ et $x_j = 0$ ($j \in J$).

-Retourner à l'étape (1) et recommencer la procédure avec cette fois le (ILP) et (ILP) complété par la contrainte de Gomory.

Etape 3

La solution optimale du (LP) est réalisable et donc optimale pour (ILP) Fin.

Si on définit, à chaque itération une coupe de Gomory à partir d'une "bonne" composante non entière a_{λ_0} il est possible de démontrer que l'algorithme fournit toujours la solution entière, si elle existe et ce en nombre fini de pas.

Détails de l'algorithme

- 1) Comme la solution optimale du (LP) peut avoir plusieurs composante a_{λ_0} non entière, on a intérêt à prendre la coupe la plus profonde possible dans le polyèdre des solutions réalisables, or la coupe est d'autant plus profonde dans la direction x_j ($j \in J$) que x_j est grand, c'est-à-dire, que f_{λ_0} est grand et f_{λ_j} est petit, on a intérêt à définir la coupe à partir de la composante non entière a_{λ_0} tel que f_{λ_0} soit maximum et f_{λ_j} soit minimum, le choix de la composante non entière a_{λ}^0 qui maximise f_{i_0} :

$$f_{\lambda_0} = \max \{f_{i_0} \mid f_{i_0} > 0\}$$

- 2) Pour résoudre le (LP) complété par contrainte de Gomory, on peut utiliser l'algorithme du simplexe. En pratique, il est plus simple d'ajouter au tableau simplexe final de l'étape précédente la contrainte de Gomory :

$$-f_{i_0} = - \sum_{j \in \bar{J}} f_{\lambda_j} x_j + s_{\lambda}.$$

Et d'appliquer l'algorithme dual-simplexe.

- 3) Toute contrainte de Gomory peut être supprimée si la variable d'écart associé à s_{λ} est en base dans le tableau simplexe final d'une des itérations, en d'autres termes, à la fin de chaque itération seules les variables d'écart s_{λ} hors base ont conservées.
- 4) Si au cours d'une itération l'algorithme dual-simplexe au (LP) augmenté montre que ce programme n'admet pas une solution réalisable cela implique que le (ILP) n'a pas de solutions réalisables, mais le (LP) initial peut avoir des solutions réalisables.
- 5) Le principal inconvénient de cette méthode est que, même pour des problèmes raisonnables, l'algorithme peut converger très lentement vers la solution optimale.

La méthode de Séparation et évaluation

Les solutions d'un programme linéaire en nombres entiers sont, en général fini.

Une méthode simple pour résoudre un problème linéaire en nombres entiers consiste à :

- a) Enumérer toutes les solutions satisfaites aux contraintes individuelles du problème.
- b) Sélectionner celles qui sont réalisables.
- c) Calculer pour chacune d'elles la valeur correspondante de la fonction économique (la fonction objectif) et en déduire la (les) solutions optimale(s).

Il existe des méthodes d'énumération plus raffinées pour la résolution des problèmes linéaires en nombres entiers. On s'intéressera à une seule d'entre elles, elle s'applique particulièrement aux programmes linéaires en variables discrètes.

Principe de la méthode

C'est une méthode développée pour des problèmes en variables discrètes; tel qu'il est aisé, en général, de déterminer l'ensemble des solutions satisfaites aux contraintes individuelles du problème. Malheureusement cet ensemble est généralement trop vaste pour qu'il soit possible d'en extraire immédiatement la solution optimale, en conséquent on procède à une séparation de cet ensemble en sous-ensembles de plus en plus petits, jusqu'à l'obtention de sous-ensembles suffisamment restreint pour que toutes l'information nécessaire (à l'obtention de la solution optimale) puisse être extraite, on parle dans ce cas de sous-ensembles sondable (sondé) de solutions. La procédure est constituée de trois procédures :

1) Procédure de séparation

Le mode de séparation choisi doit satisfaire aux trois conditions suivantes :

- a) Le premier ensemble de solutions à séparer est l'ensemble d'admissibilité S .
- b) La réunion des sous-ensembles S_i (noté sous-ensemble sondables de solutions) obtenus lors d'une séparation est l'ensemble séparé S .
- c) Le nombre total de séparation doit être fini pour tout problème.

A chaque itération, l'ensemble d'admissibilité S est subdivisé en un nombre fini de sous-ensembles S_i noté sous-ensembles sondables de solutions, généralement on écrit :

$$S = \bigcup_i S_i$$

avec

$$S_i \cap S_j = \phi \quad \forall i \neq j$$

Les séparations successives se présentent ainsi sous forme d'un graphe particulier, une arborescence de racine

$$S = \{\text{ensembles de solutions admissibles}\}$$

On notera : sous-ensembles de solutions : "Sommet"

2) Procédure d'évaluation

L'un des sommets obtenus par séparation est examiné, si le sommet se révèle sondé il n'a plus besoin d'être séparé, et on passe à l'opération suivante, si le sommet n'est pas sondable, il convient de le séparer encore, et on recommence l'opération (1) .

3) Choix d'un sommet pendant

On appelle sommet pendant tout sommet non encore sondé de l'arborescence; si les tests ont montré que le sommet dont on s'occupe est sondé, il faut poursuivre la procédure de séparation en examinant un nouveau sommet, non encore sondé (un sommet pendant), s'il n'en existe plus, la procédure de séparation est terminée.

NB :

Pour plus de détails voir[17].

Programmation Dynamique

La programmation dynamique est une méthode de résolution des problèmes d'optimisation basé sur une énumération implicite des solutions.

Pour pouvoir s'appliquer avec intérêt, elle exige que les problèmes traités aient une structure particulière, de type séquentielle.

La plupart des problèmes combinatoires peuvent se mettre sous cette forme et être résolus par programmation dynamique.

Le principe de base de la programmation dynamique est le suivant :

- 1) On prolonge le problème dans une famille de problème de même nature.
- 2) On relie par une relation de récurrence les solutions optimales de ces problèmes.

Exemple 3 Résolution du problème de sac à dos

$$(KP) \begin{cases} \max(Z) = \sum_{j=1}^n C_j x_j & C_j \geq 0 \\ \sum_{j=1}^n A_j x_j \leq b & b \text{ et } A_j \text{ des entiers positifs} \\ x_j = 0 \text{ ou } 1 & j = \overline{1, n} \end{cases} \quad (1.5.17)$$

Pour tout k de 1 à n et pour tout y de 0 à b .

Considérons les problèmes $P_k(y)$:

$$P_k(y) \begin{cases} \max(Z) = \sum_{j=1}^n C_j x_j \\ \sum_{j=1}^n A_j x_j \leq y \\ x_j = 0 \text{ ou } 1 & j = \overline{1, n} \end{cases} \quad (1.5.18)$$

appelons $Z_k(y)$ la valeur de la solution optimale de $P_k(y)$, on a alors :

$Z_{k+1}(y) = \max \{Z_k(y), C_{k+1} + Z_k(y - A_{k+1})\}$ qui exprime le fait que $x_{k+1} = 0$ ou 1

Il existe un algorithme [16, 17] qui permet d'obtenir la valeur de la solution optimale, et aussi les valeurs des variables de la solution optimale $x_k(y)$ tel que :

$$x_k(y) = \begin{cases} 0 & \text{si } Z_{k+1}(y) = Z_k(y) \\ 1 & \text{sinon} \end{cases} \quad (1.5.19)$$

Méthode Hongroise

La méthode hongroise

Cette méthode a été conçu par les deux mathématiciens hongrois Egervary et Kônig, elle se base sur la matrice coût.

Le déroulement de l'algorithme de la méthode :

- 1) Retrancher le minimum de chaque ligne de la matrice coût .

- 2) Retrancher le minimum de chaque colonne de la matrice coût.
 - 3) Trouver une solution réalisable en affectant un seul zéro par ligne et par colonne.
Si cette solution existe alors c'est la solution optimale; s'il n'existe pas assez de zéros pour une affectation complète, utiliser l'étape (4)
 - 4) a) Marquer les lignes ayant un zéro non affecter.
 - b) Marquer les colonnes ayant un zéro non affecter sur une ligne marquée.
 - c) Marquer les lignes non encore marquées ayant un zéro affecté dans une colonne marquée.
 - d) Barrer les lignes non marquées et les colonnes marquées.
 - e) Choisir le minimum des éléments non barrés, le retrancher des éléments non barrés et le rajouter aux éléments doublement barrés.
- On obtient le tableau optimal.
Chercher une solution réalisable on affectant un seul zéro par ligne et par colonne. Si elles existent, c'est une solution optimale.
- 5) Aller à l'étape (3)

1.6 Adaptation de la méthode de décomposition dans le problème de " Sac à Dos" uni-critère

1.6.1 La formulation mathématique du problème

Bien que la méthode de Branch et Bound est une première méthode décrite pour la résolution des problèmes d'optimisation en nombres entiers l'arbre de résolution augmente d'une façon rapide même pour des problèmes à petite taille, donc c'est intéressant d'exploiter d'autres méthodes.

Soit le problème de Sac à dos

$$(KP) \begin{cases} \max(Z) = \sum_{i=1}^n C_i x_i \\ \sum_{i=1}^n A_i x_i \leq b \\ x_i \in \{0, 1\} \quad i = \overline{1, n} \end{cases} \quad (1.6.1)$$

Pour qu'on adapte la méthode de décomposition dans le problème de Sac à dos, il faut vérifier que le problème traité est de la forme P.

En effectuant quelques changements au problème sans perdre aucun points de l'ensemble de solutions, à ce moment le problème est prêt à être appliquer par cette méthode.

1.6.2 La formulation mathématique du problème relaxé

Soit le problème de Sac à dos relaxé :

$$(P_R) \begin{cases} \max(Z) = \sum_{i=1}^n C_i x_i \\ \sum_{i=1}^n A_i x_i \leq b \\ x_i \leq 1 \\ x_i \geq 0 \end{cases} \quad (1.6.2)$$

Le déroulement de la méthode de résolution du problème

On introduit les variables d'écart s_1, s_2, \dots, s_n dans les n dernières contraintes qui représentent les contraintes indépendantes. Les points extrêmes associés à chaque ensemble Δ_i ($i = 1, 2, \dots, n$) dans ce cas sont entières.

$$\begin{aligned} \Delta_1 &= \{ (x_1, s_1) \mid x_1 + s_1 = 1 \} \\ &= \{ y_{11} = (1, 0)^t, y_{12} = (0, 1)^t \} \\ \Delta_2 &= \{ (x_2, s_2) \mid x_2 + s_2 = 1 \} \\ &= \{ y_{21} = (1, 0)^t, y_{22} = (0, 1)^t \} \\ &\vdots \\ &\vdots \\ \Delta_n &= \{ (x_n, s_n) \mid x_n + s_n = 1 \} \\ &= \{ y_{1n} = (1, 0)^t, y_{1n} = (0, 1)^t \} \end{aligned}$$

Le problème maître associé s'écrit donc :

$$(P_{RM}) \begin{cases} \max(Z) = \sum_{i=1}^n C_i^t \beta_{i1} Y_{i1} \\ \sum_{i=1}^n A_i \beta_{i1} Y_{i1} \leq b \\ \beta_{i1} + \beta_{i2} = 1 \quad i = \overline{1, n} \end{cases} \quad (1.6.3)$$

On cherche la solution optimale β_i^* du problème (P_{RM})

Deux cas apparaissent à la résolution du problème (P_{RM}) :

- 1) Si tous les β_{ij}^* sont entières alors x_i sont entières ($x_i = 0$ ou 1).
- 2) Sinon, on fait appelle à la méthode de séparation et évaluation avec moins de variables traitées.

Notre premier objectif est d'utiliser la méthode de décomposition et avoir une solution entière. Pour que ça soit vérifié, il faut que le problème traité satisfait les conditions citées ultérieurement :

On suivant les étapes de l'algorithme précédent, jusqu'à l'étape (3) qui introduit les variables sortantes, on distingue trois possibilités :
Etant donné :

$$\hat{b} = \begin{pmatrix} \hat{b} \\ 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ 1 \end{pmatrix} \quad \text{et} \quad P_{x_s} = \begin{pmatrix} \hat{a}_{1s} \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \longrightarrow (s+1) \text{ ligne.}$$

\hat{a}_{1s} est la première composante de A_{1s} .

\hat{b} veut dire qu'on est dans une itération quelconque, P_{x_s} vecteur colonne associé à x_s , s : colonne de pivot correspond à une itération courante.

Pour déterminer la ligne de pivot, on constate trois différents cas :

Cas1

On aura une rentrée simultanée

Si $\frac{\hat{b}}{\hat{a}_{1s}} \geq 1$, donc $\min \left\{ \frac{\hat{b}}{\hat{a}_{1s}}, 1 \right\} = 1$, la ligne de pivot sera $s+1$ le pivot=1 (il n'y aura pas un changement dans l'opération pivotage) .

Conclusion 1

Méthode de décomposition + rentrée simultanée des variables \rightarrow solution entière.

Donc le problème de Sac à dos peut être résolu efficacement par cette méthode, car la solution optimale du (P_R) est entière.

Cas2

On n'aura pas une rentrée simultanée :

Si $0 < \frac{\hat{b}}{\hat{a}_{1s}} < 1$, donc $\min \left\{ \frac{\hat{b}}{\hat{a}_{1s}}, 1 \right\} = \frac{\hat{b}}{\hat{a}_{1s}}$, le pivot se trouve à la première ligne qui correspond à la contrainte commune, dans ce cas on n'aura pas une rentrée simultanée car on retrouve des coefficients \hat{a}_{1j} non nuls $\hat{a}_{1j} > 0$ (la condition qui nécessite une rentrée simultanée).

Le pivot= \hat{a}_{1s} (il y aura un changement dans la matrice de pivotage car les $(\hat{a}_i \neq 0 \text{ et } \hat{a}_i \neq 1)$, mais on peut toujours utiliser la méthode de décomposition avec plus d'étapes et avoir certainement une solution optimale non entière qui sera solution optimale du problème (P_R) et pas pour (P) (cas non désirable), c'est-à-dire la solution non entière (la solution est dans \mathbb{R}) n'est pas une solution du problème de sac à dos.

Remarque 10 Dans le cas où les $\hat{a}_{1j} = 1$ ($j = 1, \dots, n$) on aura une matrice dont le déterminant égal à 1, 0, -1, donc même si on fait pas entrer les variables simultanément, la solution optimale sera entière.

Dans ce deuxième cas la matrice inverse sera remplie par les vecteurs pseudo inverse dont ses éléments sont différents de zéro.

Cas3

Si $\frac{\hat{b}}{\hat{a}_{1s}} < 0$ la variable choisie pour entrer dans la base n'est pas réalisable, elle est rejeter.

Illustration de la méthode par deux exemples didactiques de problèmes de sac à dos

Exemple 4

$$(KP) \begin{cases} \max(Z) = 11x_1 + 5x_2 + 7x_3 + 13x_4 + 3x_5 \\ 4x_1 + 2x_2 + 8x_3 + 7x_4 + 5x_5 \leq 16 \\ x_i \in \{0, 1\} \quad i = \overline{1, 5} \end{cases} \quad (1.6.4)$$

Soit le problème relaxé :

$$KP_R \begin{cases} \max Z = & 11x_1 & +5x_2 & +7x_3 & +13x_4 & +3x_5 \\ & 4x_1 & +2x_2 & +8x_3 & +7x_4 & +5x_5 & \leq 16 \\ & x_1 & & & & & \leq 1 \\ & & x_2 & & & & \leq 1 \\ & & & x_3 & & & \leq 1 \\ & & & & x_4 & & \leq 1 \\ & & & & & x_5 & \leq 1 \\ & & & & & & i = \overline{1, 5} \\ & & & & & & x_i \geq 0 \end{cases}$$

Solution :

On introduit les variables d'écart s_1, s_2, \dots, s_5 dans les 5 dernières contraintes qui représentent les contraintes indépendantes. Les points extrêmes associés à chaque ensemble Δ_i ($i = 1, 2, \dots, 5$) (dans ce cas sont entières).

$$\begin{aligned} \Delta_1 &= \{ (x_1, s_1) \mid x_1 + s_1 = 1 \} \\ &= \{ y_{11} = (1, 0)^t, y_{12} = (0, 1)^t \} \\ \Delta_2 &= \{ (x_2, s_2) \mid x_2 + s_2 = 1 \} \\ &= \{ y_{21} = (1, 0)^t, y_{22} = (0, 1)^t \} \\ &\vdots \\ &\vdots \\ \Delta_5 &= \{ (x_5, s_5) \mid x_5 + s_5 = 1 \} \\ &= \{ y_{15} = (1, 0)^t, y_{15} = (0, 1)^t \} \end{aligned}$$

Le problème maître associé s'écrit donc :

$$(P_R M) \begin{cases} \max Z = C_1^t \beta_{11} Y_{11} + C_2^t \beta_{21} Y_{21} + C_3^t \beta_{31} Y_{31} + C_4^t \beta_{41} Y_{41} + C_5^t \beta_{51} Y_{51} \\ A_1 \beta_{11} Y_{11} + A_2 \beta_{21} Y_{21} + A_3 \beta_{31} Y_{31} + A_4 \beta_{41} Y_{41} + A_5 \beta_{51} Y_{51} \leq 16 \\ \beta_{11} + \beta_{12} = 1 \\ \beta_{21} + \beta_{22} = 1 \\ \beta_{31} + \beta_{32} = 1 \\ \beta_{41} + \beta_{42} = 1 \\ \beta_{51} + \beta_{52} = 1 \end{cases} \quad (1.6.5)$$

Les matrices des coûts réduits sont :

$$C_1 = (11, 0)^t, C_2 = (5, 0)^t, C_3 = (7, 0)^t, C_4 = (13, 0)^t, C_5 = (3, 0)^t.$$

Les matrices des contraintes communes :

$$A_1 = (4, 0), \quad A_2 = (2, 0), \quad A_3 = (8, 0), \quad A_4 = (7, 0), \quad A_5 = (5, 0).$$

On remplaçant les A_i et les C_i par leurs valeurs ; le problème maître associé s'écrit donc :

$$(KPRM) \begin{cases} \max Z = 11\beta_{11} + 5\beta_{21} + 7\beta_{31} + 13\beta_{41} + 3\beta_{51} \\ 4\beta_{11} + 2\beta_{21} + 8\beta_{31} + 7\beta_{41} + 5\beta_{51} \leq 16 \\ \beta_{11} + \beta_{12} = 1 \\ \beta_{21} + \beta_{22} = 1 \\ \beta_{31} + \beta_{32} = 1 \\ \beta_{41} + \beta_{42} = 1 \\ \beta_{51} + \beta_{52} = 1 \end{cases} \quad (1.6.6)$$

La matrice des contraintes :

$$T = \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} & \beta_{41} & \beta_{51} \\ 4 & 2 & 8 & 7 & 5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

la matrice des coûts réduits :

$$C = \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} & \beta_{41} & \beta_{51} \\ 11 & 5 & 7 & 13 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Etape 1

Introduisons une variable d'écart α pour la première contrainte, la base initiale sera :

$$(\alpha, \beta_{12}, \beta_{22}, \beta_{32}, \beta_{42}, \beta_{52}).$$

La solution correspondante est $(\alpha, \beta_{12}, \beta_{22}, \beta_{32}, \beta_{42}, \beta_{52}) = (16, 1, 1, 1, 1, 1)^t$;

La matrice de base correspondante est $B = I$ (matrice d'identité d'ordre 6) , $B^{-1} = I$ et $C_B = (0, 0, 0, 0, 0, 0)$.

Etape 2

Recherche de sous problème

Le coût réduit associé à β_{i1} est défini par :

$$C_i(x_i) = C_{B_0}^t B_0^{-1} \begin{pmatrix} A_i x_i \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} - C_i x_i$$

1 se situe à la $i^{\text{ème}}$ ligne.

Le premier sous problème est :

$$(SP_1) \begin{cases} \min \hat{C}_1 = -11x_1 \\ x_1 + s_1 = 1 \end{cases}$$

La solution optimale est atteinte pour $Y_{11} = (1, 0)$ et $w_1 = -11$.

Le deuxième sous problème est :

$$(SP_2) \begin{cases} \min \hat{C}_2 = -5x_2 \\ x_2 + s_2 = 1 \end{cases}$$

La solution optimale est atteinte pour $Y_{21} = (1, 0)$ et $w_2 = -5$.

Le troisième sous problème est :

$$(SP_3) \begin{cases} \min \hat{C}_3 = -7x_1 \\ x_3 + s_3 = 1 \end{cases}$$

La solution optimale est atteinte pour $Y_{31} = (1, 0)$ et $w_3 = -7$.

Le quatrième sous problème est :

$$(SP_4) \begin{cases} \min \hat{C}_4 = -13x_4 \\ x_4 + s_4 = 1 \end{cases}$$

La solution optimale est atteinte pour $Y_{41} = (1, 0)$ et $w_4 = -13$.

Le cinquième sous problème est :

$$(SP_5) \begin{cases} \min \hat{C}_5 = -3x_5 \\ x_5 + s_5 = 1 \end{cases}$$

La solution optimale est atteinte pour $Y_{51} = (1, 0)$ et $w_5 = -3$.

La variable entrante dans la base est β_{41} car :

$\min \{w_1, w_2, \dots, w_5\} = -13 = w_4$; le vecteur colonne associé est $P_{41} = (7, 0, 0, 0, 1, 0)^t$ et

$$B_0^{-1}P_{41} = \begin{pmatrix} 7 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$b_0 = \begin{pmatrix} 16 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

On a $\min \left\{ \frac{16}{7}, 1 \right\} = 1$; donc la variable sortante est β_{42} ; la variable β_{41} entre au niveau de la cinquième ligne; comme les coefficients des vecteurs colonnes P_{11} et P_{31} associés respectivement aux variables hors base β_{11} et β_{31} sont nuls à cette contrainte alors cherchons si β_{41} et β_{11} et β_{31} sont de rentrée croisée.

L'inverse de la matrice de base courante est obtenue en remplaçant le cinquième vecteur colonne de B_0^{-1} par le pseudo inverse de P_{41} , $\xi_4 = (-7, 0, 0, 0, 1, 0)^t$ on aura :

$$B_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & -7 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b_1 = B_1^{-1} b_0 = \begin{pmatrix} 9 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

comme $w_1 < w_3$, on détermine en premier lieu la variable que β_{11} doit remplacer dans la base.

Le vecteur colonne P_{11} associé à β_{11} est $P_{11} = (4, 1, 0, 0, 0, 0)^t$

$$B_0^{-1} P_{11} = \begin{pmatrix} 4 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$b_1 = \begin{pmatrix} 9 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

On a $\min \left\{ \frac{9}{4}, 1 \right\} = 1$, alors la variable sortante est β_{12} , donc β_{11} entre à la place de β_{12} ; l'entrée de β_{11} ne modifie pas les vecteurs colonnes P_{41} et P_{31} associés à β_{41} et β_{31} ; β_{11} entre à une ligne où les autres vecteurs sont libres.

Examinons de la même manière le cas de β_{31} .

L'inverse de la matrice de base courante est obtenue en remplaçant le deuxième vecteur colonne de B_1^{-1} par le pseudo inverse de $\xi_2 = (-4, 1, 0, 0, 0, 0)^t$ on aura :

$$B_2^{-1} = \begin{pmatrix} 1 & -4 & 0 & 0 & -7 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b_2 = B_2^{-1} b_0 = \begin{pmatrix} 5 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Pour w_3 , on détermine en premier lieu la variable que β_{31} doit remplacer dans la base.

Le vecteur colonne P_{31} associé à β_{31} est $P_{31} = (8, 0, 0, 1, 0, 0)^t$

$$B_0^{-1} P_{31} = \begin{pmatrix} 8 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

On a $\min \left\{ \frac{5}{8}, 1 \right\} = \frac{5}{8}$, donc la variable sortante est α , alors β_{31} entre à la place de α ; puisque la variable entre au niveau de la première ligne donc il n'y aura pas une rentrée simultanée; aussi la condition de la contrainte commune n'est pas vérifiée donc la variable β_{31} ne peut pas entrer dans la base car la solution ($\beta_{41}=\beta_{11}=\beta_{31}=1$) n'est pas réalisable (ne vérifie pas la contrainte commune).

Alors, β_{11}, β_{41} viennent d'entrer dans la base donc $w_1 = w_4 = 0$; la variable β_{31} n'est pas réalisable; on cherche la variable entrante suivante, $\min \{w_2, w_5\} = w_2 = -5$; donc la solution optimale est atteinte pour $Y_{21} = (1, 0)$

La variable entrante est β_{21} , on cherche la variable sortante :

Le vecteur colonne P_{21} associé à β_{21} est $P_{21} = (2, 0, 1, 0, 0, 0)^t$

$$B_0^{-1}P_{21} = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

On a $\min \left\{ \frac{5}{2}, 1 \right\} = 1$, donc la variable sortante est β_{22} , la variable β_{21} entre au niveau de la troisième ligne; L'inverse de la matrice de base courante est obtenu en remplaçant le troisième vecteur colonne de B_2^{-1} par le pseudo inverse de $\xi_3 = (-2, 0, 1, 0, 0, 0)^t$ on aura :

$$B_3^{-1} = \begin{pmatrix} 1 & -4 & -2 & 0 & -7 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b_3 = B_3^{-1}b_0 = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

La capacité résiduelle du problème est positive ($16 - (a_1 + a_2 + a_4) > 0$).

Pour w_5 la variable β_{51} va être rejetée car $16 - (a_1 + a_2 + a_4 + a_5) < 0$;

La solution optimale du problème (P_{RM}) est $(\alpha, \beta_{11}, \beta_{21}, \beta_{32}, \beta_{41}, \beta_{52}) = (3, 1, 1, 1, 1, 1)$;

La solution optimale du problème (P_R) est $x_1^* = \beta_{11}Y_{11} + \beta_{12}Y_{12}$; $x_2^* = \beta_{21}Y_{21} + \beta_{22}Y_{22}$ et $x_3^* = \beta_{31}Y_{31} + \beta_{32}Y_{32}$ et $x_4^* = \beta_{41}Y_{41} + \beta_{42}Y_{42}$ et $x_5^* = \beta_{51}Y_{51} + \beta_{52}Y_{52}$.

Donc $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 1, 0)$; elle est entière donc, c'est une solution optimale pour le problème de sac à dos.

Conclusion 2

Pour que le problème de sac à dos peut être résolu par la méthode de décomposition en adaptant l'entrée simultanée des variables, il faut que sa capacité résiduel est $b - \sum_{i \in B} a_i$ soit positif et assez grande (B est la base courante) et par cette condition, on retombe jamais dans le cas où $\frac{\hat{b}}{\hat{a}_{1s}} < 1$.

Pour l'exemple (1), on a distingué le troisième cas et le premier cas :

$\frac{\hat{b}}{\hat{a}_{1s}} < 0$ la variable choisit pour entrer dans la base n'est pas réalisable, elle est rejeter et ça pour le cas de la variable β_{31} .

la solution du problème est entière car l'entrée des variables a été simultanée (dû à $\frac{\hat{b}}{\hat{a}_{1s}} \geq 1$) à chaque étape.

Exemple 5

$$(KP_1) \begin{cases} \max Z = 10x_1 + 8x_2 + 5x_3 \\ 6x_1 + 5x_2 + 4x_3 \leq 9 \\ x_i \in \{0, 1\} \quad i = \overline{1, 3} \end{cases} \quad (1.6.7)$$

Soit le problème relaxé :

$$KP_{1R} : \begin{cases} \max Z = & 10x_1 & +8x_2 & +5x_3 \\ & 6x_1 & +5x_2 & +4x_3 & \leq 9 \\ & x_1 & & & \leq 1 \\ & & x_2 & & \leq 1 \\ & & & x_3 & \leq 1 \\ x_i \geq 0 & & & & i = \overline{1, 3} \end{cases}$$

On remarque que la solution optimale de ce deuxième exemple est $(x_1, x_2, x_3) = (1, 3/5, 0)$ qui n'est pas entière car c'était impossible de faire entrer des variables dans la base de façon simultanée, donc ce n'est pas une solution optimale pour KP_1 (la solution $\notin \{0, 1\}$).

1.7 Conclusion

Dans cette partie, nous avons présenté la méthode de décomposition en adaptant l'entrée simultanée des variables.

Le problème de sac à dos a fait l'objet de nombreuses études, l'une d'elles était la méthode de séparation et évaluation. Notre premier objectif était d'utiliser la méthode présentée au paravent pour résoudre ce problème.

Après avoir ajouter les contraintes $x_i \leq 1$ dans la structure du problème, on a pu résoudre le problème relaxé de sac à dos, en adaptant cette méthode. On a constaté qu'elle résout le problème de sac à dos sous certaines conditions, la première si l'entrée simultanée des variables est vérifiée, on aura une solution entière. La deuxième condition si $0 < \frac{\hat{b}}{\hat{a}_{1s}} < 1$, alors $\min \left\{ \frac{\hat{b}}{\hat{a}_{1s}}, 1 \right\} = \frac{\hat{b}}{\hat{a}_{1s}}$, le pivot se trouve à la première ligne qui correspond à la contrainte commune, dans ce cas on n'aura pas une rentrée simultanée, mais il se peut que la solution soit entière dans le cas où les coefficients de la contrainte commune égal à 1, comme il se peut que la solution soit non entière si ces coefficients sont différents de 1.

La troisième condition si $\frac{\hat{b}}{\hat{a}_{1s}} < 0$, la variable choisie pour entrer dans la base n'est pas réalisable, elle est rejetée, alors le problème de sac à dos n'est pas résolu par cette méthode, c'est à dire la solution obtenue est non entière.

Proposition 1.7.0.1 *Au cas où la méthode de décomposition est utilisée pour résoudre un problème d'optimisation combinatoire uni-critère, on constate deux différents cas (voir précédemment);*

- 1) la résolution donne une solution entière (Terminer).
- 2) la résolution donne une solution non entière, dans ce cas on fait entrer les coupes de Gomory.

Pour qu'on puisse utiliser le même principe, l'entrée simultanément des variables dans la base, on doit faire entrer la coupe de Gomory pour toutes les contraintes dont le second membre x_i^0 $i = p, q$ est non entier.

Pour que ça soit vérifié, il faut que :

Soient $P_{x_p^0}$ et $P_{x_q^0}$ les vecteurs lignes associés respectivement aux x_p^0 et x_q^0 tels que :

$$P_{x_p^0} = (f_{pm+1}, f_{pm+2}, \dots, f_{pn})$$

$$P_{x_q^0} = (f_{qm+1}, f_{qm+2}, \dots, f_{qn})$$

Pour qu'on fait ajouter la coupe de Gomory pour exemple deux lignes p et q.

$$\begin{cases} f_{p0} + S_p = \sum_{j \in J} f_{pj} x_j & p \in I \\ f_{q0} + S_q = \sum_{j \in J} f_{qj} x_j & q \in I \end{cases} \quad S_p \text{ et } S_q \text{ variables d'écart de Gomory} \quad (1.7.1)$$

Il y a deux possibilités :

1) **Une rentrée totalement indépendante des coupes**

Deux coupes f_{p0} et f_{q0} sont dites de rentrée totalement indépendante à une itération t_0 si à cette itération leurs vecteurs lignes respectifs associés $P_{f_{p0}} = (f_{pm+1}, f_{pm+2}, \dots, f_{pn}, -1, 0)$ et $P_{f_{q0}} = (f_{qm+1}, f_{qm+2}, \dots, f_{qn}, 0, -1)$ vérifient :

$$f_{pj} = 0 \Rightarrow f_{qj} \neq 0$$

2) **Une rentrée croisée des coupes**

Deux coupes $P_{f_{p0}}$ et $P_{f_{q0}}$ sont dites de rentrée croisée à une itération t_0 si à cette itération leurs vecteurs lignes respectifs associés $P_{f_{p0}} = (f_{p(m+1)}, f_{p(m+2)}, \dots, f_{pn}, -1, 0)$ et $P_{f_{q0}} = (f_{q(m+1)}, f_{q(m+2)}, \dots, f_{qn}, 0, -1)$ vérifient :

– Le pivot de $P_{f_{p0}}$ se trouve sur une colonne où $P_{f_{q0}}$ est libre.

– Celui de $P_{f_{q_0}}$ se trouve sur une colonne où $P_{f_{p_0}}$ est libre.

Les vecteurs lignes $P_{f_{p_0}}$ et $P_{f_{q_0}}$ s'écrivent :

$$P_{f_{p_0}} = (f_{p(m+1)}, f_{p(m+2)}, \dots, f_{p i_0}, \dots, 0, \dots, f_{pn}, -1, 0).$$

$$P_{f_{q_0}} = (f_{q(m+1)}, f_{q(m+2)}, \dots, 0, \dots, f_{q j_0}, \dots, f_{qn}, 0, -1).$$

Remarque 11 *La plupart des programmes linéaires modélisant des situations concrètes, la matrice de contraintes est creuse, si la matrice reste creuse pendant le traitement, le nombre d'opérations de pivotage¹ peut être diminué de manière rapide. Mais ce n'est jamais le cas, les différentes matrices engendrées lors des opérations de pivotage n'ont aucune raison de rester creuses, ces colonnes se remplissent par les pseudo inverses des vecteurs colonnes des variables entrantes dans la base, ce qui fait qu'on pourra jamais faire entrer les coupes de Gomory simultanément, mais il reste à prouver que ce cas peut se réaliser.*

¹Opération de pivot

Chapitre 2

Abécédaire sur Quelques Problèmes d'Optimisation Multi-Objectifs

2.1 Introduction

La recherche opérationnelle occupe une place éminente dans la théorie économique, son but essentiel est d'aider à prendre des décisions en parcourant des méthodes exactes et efficaces. Les modèles d'optimisation à objectif unique était une première modélisation des problèmes de recherche opérationnelle. A travers le temps on remarque que ce procédé manque de réalisme, il serait impossible de combiner tous les points désirés en une seule fonction objectif, et donc on néglige un nombre fini d'objectifs et on s'éloigne dans ce cas de la solution réaliste. Une telle situation ne modélise pas correctement un problème.

Afin de s'approcher au maximum de la réalité, de nouvelles méthodes caractérisées par la prise en considération de tous les objectifs apparaient. Cette multiplicité d'objectifs semble plus logique pour la modélisation des problèmes.

L'optimisation multi-objectif (MO) ou multi-critère est un domaine dont l'importance est justifiée par la nature multi-objectif des problèmes d'optimisation à résoudre dans la réalité. Elle se propose de traiter les problèmes qui nécessitent la satisfaction de plusieurs critères en même temps. Ces critères sont parfois conflictuels et parfois complémentaires.

Un des aspects caractérisant l'optimisation multi-objectif est qu'elle fournit un ensemble de solutions réalisant le meilleur compromis entre les critères considérés. Cet ensemble de solutions est appelé l'ensemble optimal de Pareto.

Plusieurs techniques ont été développés pour traiter ces problèmes à objectifs multiples, la plus utilisée est l'algorithme qui maximise plusieurs objectifs par l'intermédiaire de la programmation linéaire paramétrique.

Il existe deux types principaux de méthodes de résolution des problèmes multi-objectifs, les méthodes exactes et les méthodes métaheuristiques.

On esseyera dans ce chapitre de donner un aperçu sur quelques méthodes de résolution des problèmes multi-objectifs en nombres continus et des problèmes multi-objectifs en nombres entiers.

2.2 Les méthodes exactes

Les méthodes exactes ont la particularité de garantir l'optimalité de la solution obtenue. Il existe plusieurs méthodes qui résolvent le cas d'un problème bi-objectif ; par contre, il existe que très peu de travaux concernant le cas multi-objectif.

2.2.1 Problèmes d'optimisation multi-objectifs en nombres continus

L'optimisation multi-objectif ou la programmation multi-objectif (l'optimisation multi-critère) est un processus d'optimisation de deux ou plusieurs objectifs conflictuels (maximiser la performance d'un véhicule et minimiser la consommation des carburants), simultanément qui vérifie certaines contraintes. On trouve l'optimisation dans de divers champs de Finance et Gestion, où la décision optimale sera choisie dans la présence de (Trade-offs), compromis (échange de questions et réponses).

Le problème linéaire multi-objectif (multi-critère) est défini comme un problème dont on cherche la solution ou le vecteur de décision $x = (x_1, x_2, \dots, x_n)$ qui satisfait un ensemble de contraintes et optimise un vecteur de fonction objectif $Z = (Z_1, Z_2, \dots, Z_p)$

Le schéma qui suit montre une modélisation d'un problème multi-objectif :

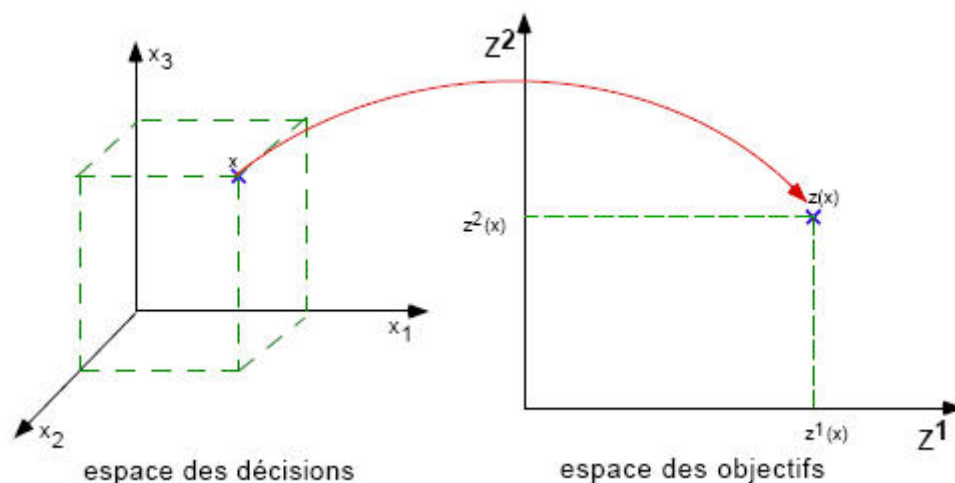


FIG. 2.1 – Modélisation d'un problème multi-objectif

Formulation mathématique

Le problème se présente comme suit :

$$(MOLP) \begin{cases} \text{''Optimiser'' } Z_k = C^k x & k = \overline{1, p} \\ \text{t.q. } x \in S \end{cases} \quad (2.2.1)$$

où l'ensemble S est déterminé par des contraintes (inéquations, équations) linéaires

$$S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}, \quad A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \quad C^k \in \mathbb{R}^{1 \times n} \quad k = \overline{1, p}.$$

Si de plus, l'ensemble des décisions est restreint aux entiers positifs, le problème devient un problème de programmation linéaire multi-objectif entier défini par :

$$(MOILP) \begin{cases} \text{"Optimiser"} Z_k = C^k x & k = \overline{1, p} \\ \text{t.q} \\ x \in D \end{cases} \quad (2.2.2)$$

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

Remarque 12 Dans la suite du chapitre, nous considérons, sans perte de généralité, que toutes les fonctions objectifs sont à maximiser.

Remarque 13 Dans le cas d'un problème minimiser les objectifs, on inverse les inégalités. Une écriture équivalente du problème MOILP se présente comme suit :

$$(MOILP) \begin{cases} \text{"max"} Z = C^k x & k = \overline{1, p} \\ \text{t.q} \\ x \in D \end{cases} \quad (2.2.3)$$

2.2.2 Définitions

Soit le problème(MOLP) :

$$(MOLP) \begin{cases} \text{"max"} Z_k = C^k x & k = \overline{1, p} \\ \text{t.q} \\ x \in S \end{cases} \quad (2.2.4)$$

Considérons l'application linéaire Ψ qui associe à chaque vecteur de décision $x \in S$ son image $Z = \Psi(x) = (c^1 x, \dots, c^p x)$ dans l'espace des critères.

Définition 8 L'espace \mathbb{R}^p dans lequel se situe l'ensemble des actions S ($S \subseteq \mathbb{R}^n$) est appelé espace de décisions.

Par espace des critères on entend l'espace \mathbb{R}^p dans lequel se situe $\Psi(S)$.

Définition 9 *Pareto optimale (Efficacité aux sens de Pareto) (Pareto efficiency "Pareto optimality"*

Lorsqu'une solution est associée à une seule valeur, on parle de problème mono-objectif, lorsqu'elle est associée à plusieurs valeurs, on parle d'un problème multi-objectif (ou multi-critère). Dans ce dernier cas, on cherche un ensemble de solutions non dominé, parmi lesquelles on ne peut décider si une solution est meilleure qu'une autre, aucune n'étant systématiquement supérieure aux autres sur tous les objectifs.

On dit que $\tilde{x} \in S$ est une solution efficace s'il n'existe pas de $x \in S$ tel que $Z_k(x) \geq Z_k(\tilde{x})$ $\forall k = \overline{1, p}$ avec au moins une inégalité stricte.

Définition 10 *Dominance*

Soit Z^1 et $Z^2 \in \mathbb{R}^p$, on dit que Z^1 domine Z^2 si et seulement si $Z_k^1 \geq Z_k^2 \quad \forall k = \overline{1, p}$ et $Z^1 > Z^2$ pour au moins un k . Autrement dit : une solution \tilde{x} sera dite efficace (optimale au sens de Pareto, non dominée) si et seulement s'il n'existe pas de solution x telle que x domine \tilde{x} .

Définition 11 la frontière de Pareto

Le graphe obtenu de l'image des solutions efficaces est appelé la frontière efficace (Pareto front of problem) dans le cas continu.

Le schéma ci-dessous nous montre la frontière efficace au cas d'un problème à minimiser.

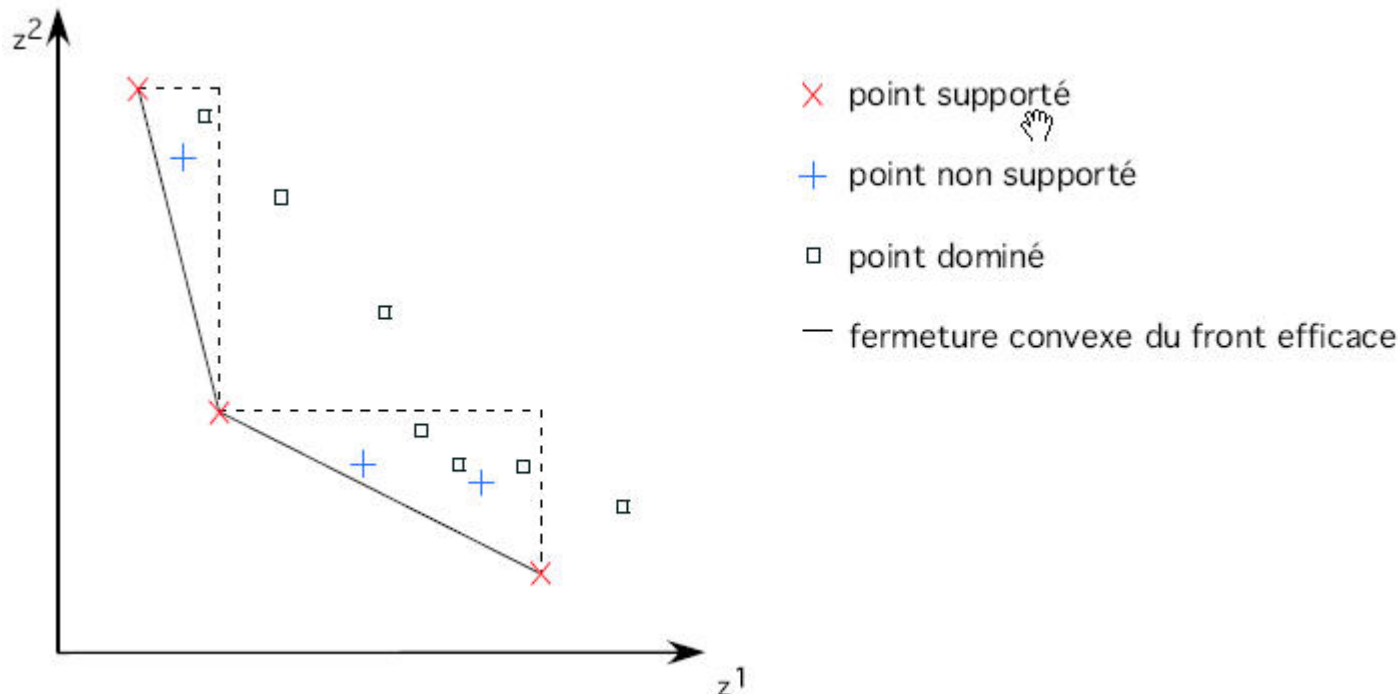


FIG. 2.2 – La frontière de Pareto

Définition 12 Point idéal

Le point idéal \bar{Z} est le point de \mathbb{R}^p de coordonnées :

$$\bar{Z}_k = \max_{x \in S} (Z_k(x)) \quad k = 1, \dots, p.$$

Définition 13 Matrice des gains

Soit \hat{x}^j une solution optimale du critère Z_j .

La matrice $(p \times p)$ formée des éléments de $Z_{kj} = Z_k(\hat{x}^j)$ est dite matrice des gains.

$$\begin{pmatrix} \bar{Z}_1 & \dots & Z_{1j} & \dots & Z_{1p} \\ \vdots & & \vdots & & \vdots \\ Z_{k1} & \dots & Z_{kj} & \dots & Z_{kp} \\ \vdots & & \vdots & & \vdots \\ Z_{p1} & \dots & Z_{pj} & \dots & \bar{Z}_p \end{pmatrix}$$

La diagonale principale de la matrice des gains est formée des coordonnées du point idéal.

Définition 14 Point anti-idéal

Le point anti-idéal \underline{Z} est le point de \mathbb{R}^p de coordonnées :

$$\underline{Z}_k = \min_{x \in S} (Z_k(x)) \quad k = 1, \dots, p.$$

Définition 15 *Point Nadir*

Le point Nadir η de \mathbb{R}^p de coordonnées

$$\eta_k = \min_{j=1, \dots, p} (Z_{kj}) \quad k = 1, \dots, p.$$

2.2.3 Méthodes de résolution

Dans de nombreux problèmes, on doit prendre en considération deux ou plusieurs critères conflictuels, ces objectifs expriment de différentes unités. La nature de ces problèmes n'a pas une seule solution mais un ensemble de solutions car c'est impossible (difficile) de définir une solution "bonne solution" qui satisfait à toutes les fonctions objectifs. En réalité les problèmes multi-objectifs sont mal posés, contrairement aux problèmes uni-critères, il n'existe pas en général un unique vecteur de valeur qui soit meilleur que tous les autres. Le plus souvent, résoudre le problème (MO) consiste à trouver l'ensemble des solutions optimales au sens de Pareto, c'est-à-dire les solutions telles qu'il n'existe pas de solution au moins aussi bonne sur tous les critères et meilleure sur au moins un critère. D'où vient le sens de la solution optimale au sens de Pareto définie au paravent. Le décideur peut exprimer le fait qu'une solution est préférable qu'une autre, mais il n'existe pas une solution meilleure que toutes les autres solutions. Les méthodes de résolution des problèmes multi-objectifs sont des méthodes d'aides à la décision car le choix final sera laissé au décideur.

Il existe deux types d'approches pour résoudre un problème multi-objectif :

- 1) Ramener le problème multi-objectif à un seul objectif au risque d'enlever toutes significations au problème.
PbMO \rightarrow Préférence du Décideur \rightarrow Détermination du poids de chaque objectif \rightarrow Problème à objectif unique \rightarrow Solveur pour objectif unique \rightarrow Solution du problème.
- 2) Tenter d'apporter des réponses au problème tenant compte de tous les objectifs (calcul de la matrice gain pour arriver à toutes les réponses)
PbMO \rightarrow Solveur multi-objectif \rightarrow Ensemble Pareto optimal \rightarrow Préférence du décideur \rightarrow choix de la solution la plus appropriée \rightarrow solution du problème.

Donc pour sélectionner une solution efficace (solution optimale, bonne solution) s'a demande et nécessite une certaine connaissance de la structure de préférence des décideurs, cette information est traduite en termes de paramètres. Les méthodes existantes sont classées selon la façon dont le décideur articule des préférences.

Définition 16 1) Un vecteur $\Lambda = (\lambda_1, \dots, \lambda_p)$ (les λ_k étant appelés coefficients d'importance des p fonctions objectifs ou paramètres de préférence).

- 2) Des vecteurs de performances ayant des significations particulières, comme le point de référence qui est défini par des niveaux d'aspiration (valeurs souhaitables) sur chaque critère et point de réservation qui est défini par des niveaux de réservation (valeurs non souhaitables) sur chaque critère.

2.2.4 Catégories de méthodes fondamentales

Dans la programmation mathématique multi-objectif, on distingue trois principales méthodes :

Les méthodes a priori

Le décideur indique l'importance relative des fonctions objectifs, il connaît a priori le poids de chaque objectif avant de combiner les différentes fonctions objectifs en une seule fonction d'utilité suivant les préférences du décideur (Théorie l'utilité multiattribut), donnée par :

Fonction scalarisante

Etant donné un ensemble de paramètre de préférence $\Lambda = (\lambda_1, \dots, \lambda_p)$ (les λ_k étant appelés coefficients d'importances des fonctions objectifs).

On définit une fonction croissante des objectifs et agrègeons les valeurs des objectifs pour chaque solution :

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

où $(\Lambda \subset \mathbb{R}^p)$ est l'ensemble des paramètres. Les fonctions d'agrégations les plus employées sont :

Utilisation de la fonction d'utilité

L'ensemble des méthodes de la théorie de l'utilité multi-critère repose sur l'axiome suivant : Tout décideur essaye d'optimiser une fonction d'utilité

$$S_1(\lambda, Z) = U_k(Z_k) \quad k = \overline{1, p} \quad (p \text{ fonctions objectifs})$$

Les modèles les plus utilisés sont le modèle additif et le modèle multiplicatif. U_k représente une fonction de mise à l'échelle du $k^{\text{ème}}$ objectif.

Critique

Dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, soit par manque d'expérience, soit par manque d'information, l'utilisation de ces modèles impose que les objectifs soient compatibles.

Il est donc très difficile d'utiliser ces techniques lorsque l'ensemble des objectifs est composé à la fois d'objectifs qualitatifs et quantitatifs.

Caractérisation à l'aide de poids :

La somme pondérée

Cette méthode est largement utilisée en optimisation linéaire multi-objectif, elle consiste à additionner tous les objectifs en affectant à chacun un coefficient de poids. Ce coefficient représente l'importance relative que le décideur attribut à l'objectif, donc le problème d'optimisation multi-objectif (MOLP) peut se ramener à un problème de programmation paramétrique mono-objectif de la forme :

$$\bullet S_2(Z, \lambda) = \max_{x \in S} \sum_{k=1}^p \lambda_k Z_k(x)$$

$$\lambda_k \in \Lambda \quad \Lambda = \left\{ \lambda_k \mid \sum_{k=1}^p \lambda_k = 1 \quad \lambda_k > 0 \quad k = \overline{1, p} \right\}$$

La fonction scalarisante $\sum_{k=1}^p \lambda_k Z_k(x)$ permet de caractériser entièrement ou partiellement l'ensemble de solutions efficaces.

Critique

Cette procédure est efficace et très simple à mettre en oeuvre mais se heurte à quelques problèmes, tels que le choix des poids des objectifs et de la compensation entre les différents objectifs.

Choix des poids des objectifs

Une solution à ce problème est 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 le résultat.

La compensation entre les différents objectifs

Une valeur de base d'un objectif peut être composée par les valeurs des autres objectifs, cela peut conduire le décideur à faire le mauvais choix.

$$\bullet S_3(Z, \lambda) = \max \sum_{k=1}^p \lambda_k |Z_k - \bar{Z}_k|$$

\bar{Z}_k est la $k^{\text{ème}}$ composante du point idéal et $\sum_{k=1}^p \lambda_k |Z_k - \bar{Z}_k|$ mesure la déviation qui sépare l'évaluation des propositions, qui sont généralement sur des points efficaces ou faiblement efficaces, du point d'aspiration ; cette déviation peut être mesurée par d'autres normes :

- Norme L_q pondérée :

$$\bullet S_4(Z, \lambda) = \max \left[\sum_{k=1}^p \lambda_k |Z_k - \bar{Z}_k|^q \right]^{1/q} \quad q \in \mathbb{Z}_+^*$$

- Norme L_q de tchebycheff pondérée :

$$\bullet S_5(Z, \lambda) = \max_{1 \leq k \leq p} \{ \lambda_k |Z_k - \bar{Z}_k| \}$$

- Norme composée (Tchebycheff pondérée augmentée)

$$\bullet S_6(Z, \lambda) = \max_{1 \leq k \leq p} \{ \lambda_k |Z_k - \bar{Z}_k| \} + \rho \sum_{k=1}^p \lambda_k |Z_k - \bar{Z}_k| \quad \rho > 0$$

Caractérisation à l'aide de points cibles

- Niveau d'aspiration

$$\bullet S_7(Z, \lambda) = \left[\sum_{k=1}^p \lambda_k |Z_k - \hat{Z}_k|^q \right]^{1/q} \quad q \in \mathbb{Z}_+^* \quad \hat{Z}_k \in \Psi(S)$$

\hat{Z}_k est un point cible qu'on souhaite être très près.

- Niveau de réservation

$$\bullet S_8(Z, \lambda) = \left[\sum_{k=1}^p \lambda_k |Z_k - Z'_k|^q \right]^{1/q}$$

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

Z'_k représente un plus petit seuil.

Méthode lexicographie

Dans cette méthode les critères sont rangés par ordre d'importance par le concepteur, la solution optimale est alors obtenue en optimisant les fonctions objectifs commençant par les plus importantes et le processus continu ainsi selon l'ordre d'importance affecté aux objectifs.

On suppose que les indices des fonctions objectifs indiquent non seulement le nombre de critères mais également la priorité de l'objectif, donc représentent les plus et les moins importants critères respectivement. Le premier problème à résoudre peut être formulé par :

$$\begin{cases} \text{"Max"} Z_1(x) \\ g_i(x) \geq 0 \quad i = \overline{1, m} \end{cases} \quad (2.2.5)$$

Dans notre cas les fonctions Z_1 et g_i sont linéaires.

Les critères sont rangés en ordre de préférence puis optimiser l'un après l'autre dans cette ordre.

A l'itération k on résoud le problème :

$$\begin{cases} \text{"Max"} C^k(x) \quad 1 \leq k \leq p \\ x \in S_k \end{cases} \quad (2.2.6)$$

avec $S_1 = S$ et $S_k = \{x \mid x \in S_{k-1}, C^{k-1} \geq \overline{Z}_{k-1} - h_{k-1}\}$

La solution finale dépend de l'ordre de préférence et la définition de h_k et elle n'est pas nécessairement efficace.

Méthode de la programmation par but " Goal programming "

Dans cette méthode les décideurs doivent affecter des cibles (niveaux d'aspiration) ou des buts qu'ils souhaitent réaliser pour chacun des objectifs, ces valeurs sont incorporé au problème en tant que contraintes additionnelles, la fonction objectif essayera de réduire au minimum les déviations absolues des cibles aux objectifs. La forme la plus simple de cette méthode peut être formulé comme suit :

$$\min_{x \in S} \left\{ \sum_{k=1}^p \lambda_k |Z_i - Z_i^*| \right\}$$

où Z_i^* représente la cible fixé par les décideurs pour le $i^{\text{ème}}$ critère. Le critère doit réduire au minimum la somme des valeurs absolues de la différence entre les valeurs à atteindre et les valeurs réellement réalisées. Il est possible d'utiliser cette technique de manière interactive en modifiant à chaque itération les niveaux d'aspiration.

Une formulation plus générale de la programmation but est de remplacer la fonction objectif ci-dessus par la $p^{\text{ème}}$ puissance de la déviation $|Z_i - Z_i^*|$. Une telle formulation est appelée la programmation but généralisée.

Différentes applications de cette technique ont été proposé dans Ignizio(1983).

Critique

Nous pouvons reprendre la critique faite pour la somme pondérée, la méthode est facile à mettre en oeuvre mais la définition des poids et des buts à atteindre est une question délicate qui détermine l'efficacité de la méthode.

Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs buts non réalisables.

Méthode des seuils de satisfaction

Cette méthode est basée sur la maximisation d'un objectif Z_l en considérant que toutes les autres objectifs Z_k avec $k \neq l$ doivent être supérieur à une valeur ϵ_k . En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité.

$$P(s) \begin{cases} \text{Max}_{x \in S} Z_l(x) \\ Z_k(x) \geq \epsilon_k \quad \forall k \neq l \end{cases} \quad (2.2.7)$$

Le décideur peut ensuite réitérer ce processus sur un objectif différent jusqu'à ce qu'il trouve une solution satisfaisante.

Critique

La connaissance a priori des intervalles appropriés pour les valeurs de ϵ_k est exigée pour tous les objectifs.

Il existe d'autres méthodes, la méthode de Vincke, Vanderpooten, mini-max, méthode de Zionts et Wallenius et d'autres [5].

Les méthodes à posteriori

Le décideur prend sa décision dans un ensemble de solutions calculé par un solveur multi-objectif. Dans ce cas, la qualité de la décision dépend du choix de la méthode de résolution, car celle-ci va devoir donner un ensemble de résultats le plus représentatif de l'espace des objectifs efficaces. L'inconvénient de ces méthodes est que le nombre de solutions efficaces générés peut être tellement grand de sorte que l'on doit avoir recours à une procédure d'agrégation pour dégager la où les meilleures solutions.

En général, ces approches comportent toutes trois étapes :

- Recherche d'une solution initiale réalisable.
- Recherche d'une solution initiale efficace.
- Recherche des solutions efficaces.

Les méthodes interactives

Le déroulement de la plupart des méthodes interactives se résume par une alternance d'étapes de dialogue et de calcul.

Critique

Dans une méthode interactive, c'est de la dynamique de l'interaction qu'est issue l'aide de la décision. Ces modèles exigent donc une connaissance approfondie, de la part du décideur, des utiles utilisés. Ce n'est malheureusement pas souvent le cas. Leur domaine d'application se résume à des situations où le nombre d'actions est très élevé.

2.2.5 Quelques résultats de base

Considérons le problème (MOLP) :

$$(MOLP) \begin{cases} \text{"max"} Z_k = C^k x & k = \overline{1, p} \\ \text{t.q} & x \in S \end{cases} \quad (2.2.8)$$

Définition 17 Soit Λ l'ensemble de tous les vecteurs $\lambda = \lambda_i, \quad i = \overline{1, p}$ définis par :

$$\Lambda = \left\{ \lambda \in \mathbb{R}^p \mid \sum_{i=1}^p \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = \overline{1, p} \right\}$$

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

$$P(\lambda) \begin{cases} \text{Max} \sum_{i=1}^p \lambda_i Z_i(x) \\ \text{t.q} \quad Z = (Z_1(x), \dots, Z_p(x))' \in \Psi(S). \end{cases} \quad (2.2.9)$$

Théorème de Geoffrion(1968)

Soit le problème paramétrique $P(\lambda)$ défini ci-dessus :

- Si x est une solution optimale de $P(\lambda)$, x est efficace pour le problème (MOLP)
- Si x est une solution efficace pour (MOLP) et que $\Psi(S)$ est un ensemble convexe, il existe $\lambda \in \Lambda$ tel que x est solution optimale du problème paramétrique $P(\lambda)$.

Théorème de Soland (1979)

Un point $x \in S$ est une solution efficace pour le problème (MOLP) si et seulement si x est une solution optimale du problème $P(s)$ défini dans (2.2.7).

2.2.6 Problèmes d'optimisation multi-objectifs en nombres entiers

Lorsqu'on s'impose aux variables de décision du (MOLP) d'être entières, nous aurons le modèle :

$$(MOILP) \begin{cases} \text{"optimiser"} Z_k = C^k x & k = \overline{1, p} \\ t.q \\ x \in D \end{cases} \quad (2.2.10)$$

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

dont la structure est complètement différente de celle du (MOLP), la différence est dû au fait que l'ensemble S est convexe par contre l'ensemble D n'est pas convexe ; et donc le principe de Geoffrion qui permet de caractériser l'ensemble des solutions efficaces par les solutions optimales du problème paramétrique n'est plus valable, tel qu'il représente qu'une partie des solutions efficaces appelé les solutions supportées "supported solutions", sur cette base l'idée d'efficacité peut être élargie en répartissant l'ensemble des solutions efficaces en deux sous-ensembles. L'ensemble de solutions optimales du $P(\lambda)$ où S est remplacé par D , on le note par $SE(MOILP)$ l'ensemble des solutions efficaces supportées et $NSE(MOILP)$ l'ensemble des solutions efficaces non supportées.

Notons $IE(MOILP)$ l'ensemble de toutes les solutions efficaces du problème MOILP, et par $E(MOILP)$ l'ensemble des solutions efficaces du problème MOLP qui est caractérisé par la solution du problème $P(\lambda)$.

Illustrerons par un exemple numérique la notion des solutions supportées et non supportées (exemple introduit par Bowman[5]).

Exemple

Soit :

$$(P^{Bow}) \begin{cases} \text{"max"} Z_1 = 6x_1 + 3x_2 + x_3 \\ \text{"max"} Z_2 = x_1 + 3x_2 + 6x_3 \\ x_1 + x_2 + x_3 \leq 1 \\ x_1, x_2, x_3 \in \{0, 1\} \end{cases} \quad (2.2.11)$$

Il est clair d'après le théorème de Geoffrion que les solutions optimales du problème ($P^{Bow}(\lambda)$) :

$$P^{Bow}(\lambda) \begin{cases} \text{max} Z = \lambda(6x_1 + 3x_2 + x_3) + (1 - \lambda)(x_1 + 3x_2 + 6x_3) \\ t.q \quad x_1 + x_2 + x_3 \leq 1 \\ x_1, x_2, x_3 \in \{0, 1\} \quad \text{avec} \quad 0 < \lambda < 1 \end{cases} \quad (2.2.12)$$

sont des solutions efficaces.

L'ensemble des solutions admissibles du problème est formulé de trois solutions $X_1 = (1, 0, 0)$, $X_2 = (0, 1, 0)$, et $X_3 = (0, 0, 1)$.

La table ci-dessous résume l'énumération de toutes les solutions efficaces et les valeurs correspondantes des objectifs.

Z	Z_1	Z_2
X_1	6	1
X_2	3	3
X_3	1	6

TAB. 2.1 – Solutions efficaces

X_1 et X_3 sont des solutions efficaces, obtenus du problème $(P^{BOW}(\lambda))$, mais la solution X_2 , qui est aussi efficace, n'est pas une solution optimale du problème $(P^{BOW}(\lambda))$ voir ci-dessous.

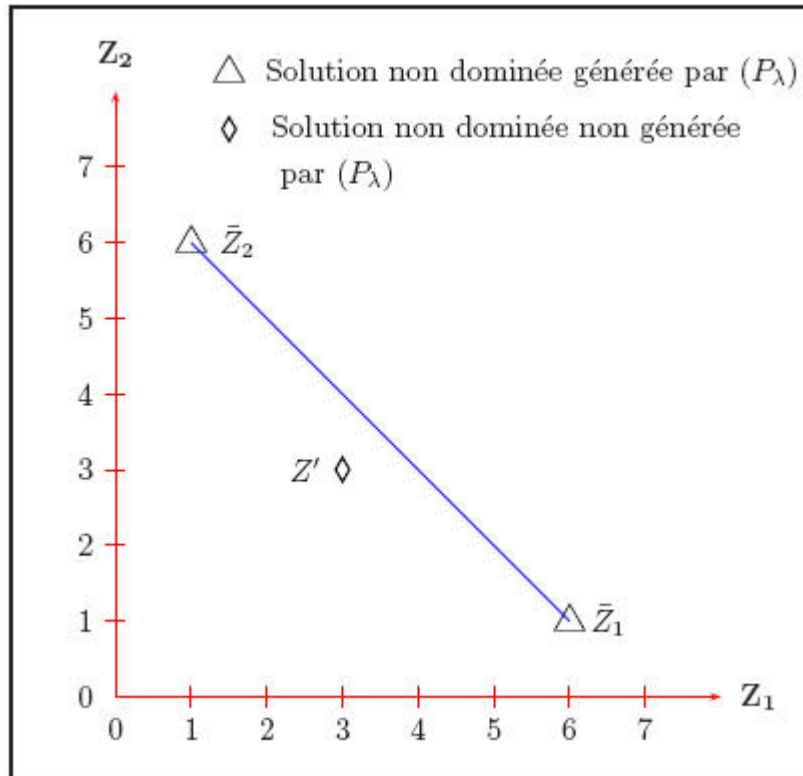


FIG. 2.3 –

Alors, les solutions efficaces sont réparties en deux sous-ensembles :

- Ensemble des solutions efficaces supportées SE(MOILP), qui est l'ensemble des solutions efficaces obtenu en résolvant le problème paramétrique.
- Ensemble des solutions efficaces non supportées NSE(MOILP).

En conclusion, la difficulté rencontrée dans la résolution des problèmes MOILP, est la détermination de l'ensemble NSE(MOILP) des solutions efficaces qui ne sont pas solutions du problème paramétrique pour aucun paramètre λ .

Dans la section suivante, nous allons présenter quelques méthodes de résolution pour les problèmes d'optimisation multi-objectif en nombre entiers.

2.2.7 Méthodes de résolution

Il existe un nombre très grand de méthodes faite juste pour résoudre les problèmes d'optimisation en nombres entiers, citons quelques unes :

Méthode de Klein et Hannan(1982)

Cette méthode consiste à résoudre progressivement une séquence de programmes linéaires en nombres entiers, avec des contraintes ajoutées à chaque itération.

La méthode génère des solutions efficaces de façon que l'utilisateur n'est pas obligé de déterminer entièrement l'ensemble des solutions efficaces s'il ne s'intéresse qu'à quelques unes.

L'algorithme de la méthode est décrit dans[4].

Méthode de Villareal-Karwan

Cette méthode est une extension des techniques de résolution du problème de Sac à dos à l'aide de la programmation dynamique.

Dans le problème (MOILP), les coefficients des matrices C et A sont supposés entiers positifs et N_j ($j = 1, 2, \dots, n$) est un entier positif considéré comme borne supérieure de la variable x_j .

En posant $Y^i = \sum_{j=1}^i a^j x_j$ où $a^j(m, 1)$ est la $j^{\text{ème}}$ colonne de la matrice A, introduisons le problème suivant :

A la $i^{\text{ème}}$ itération :

$$(P_i(Y^i)) \begin{cases} \text{"max"} \sum_{j=1}^i C^j x_j \\ t.q \\ \sum_{j=1}^i a_j x_j \leq Y^i \\ 0 \leq x_j \leq N_j, \quad j = 1, \dots, i \end{cases} \quad (2.2.13)$$

où $0 \leq Y^i \leq b$ et $IE(P_i(Y^i))$ est l'ensemble de solutions efficaces correspondant.

Ce problème peut être résolu en considérant la programmation dynamique multi-objectifs à l'itération i

$$\begin{cases} H_i(Y^i) = \text{"max"}(c^i x_i \oplus H_{i-1}(Y^{i-1})) \\ a_i \leq Y^i - Y^{i-1} \\ 0 \leq x_j \leq N_j \end{cases} \quad (2.2.14)$$

c^i est le $i^{\text{ème}}$ vecteur de la matrice C et $H_{i-1}(Y^{i-1})$ correspond à l'ensemble de solutions de l'étape $i - 1$, avec un second membre de contraintes Y^{i-1} .

\oplus indique que la somme s'effectue sur tous les éléments de $H_{i-1}(Y^{i-1})$.

Ces équations récursives sont appliquées pour chaque vecteur second membre Y^i , $0 \leq Y^i \leq b$ et pour chaque $i^{\text{ème}}$ itération du problème.

Dans l'étape finale n, $IE(P) = IE(P_n(b))$.

Méthode de Chalmet

Cette méthode a été proposée par Chalmet et al pour résoudre un problème bi-objectif en variables discrètes.

Le problème uni-critère $\max_{x \in D} \{\lambda Z_1(x) + (1 - \lambda)Z_2(x)\}$ est résolu afin de générer l'ensemble des solutions supportées (SE) de (MOILP), puis l'ensemble des solutions non supportées qui sont déterminées comme suit :

Soient $Z^1 = (Z_1^1, Z_2^1)$ et $Z^2 = (Z_1^2, Z_2^2)$ deux points non dominés, adjacents parmi les points déjà obtenus.

On cherche s'il y a une solution non dominée entre ces points en résolvant le problème suivant :

$$(P') \left\{ \begin{array}{l} \max(\lambda Z_1(x) + (1 - \lambda)Z_2(x)) \\ t.q \quad Z_1(x) \geq Z_1^1 \\ \quad \quad Z_2(x) \geq Z_2^2 \\ \quad \quad x \in D \\ \quad \quad \lambda \in [0, 1]. \end{array} \right. \quad (2.2.15)$$

Soit Z^l la solution efficace s'il y a des solutions à ce problème.

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

Méthode de Gonzales, Reeves et Franz :

Cette méthode est interactive, elle comporte deux phases :

- Un sous-ensemble E de solutions efficaces (supported solutions) est d'abord déterminée (ensembles de solutions préférées par le décideur).
- Les solutions non supportées sont ensuite présentées l'une après l'autre au décideur, si x^* (est l'une d'elles) est préférable à une solution x de E, x est rejetée de E et elle est remplacée par x^* (L'algorithme de la méthode est décrit dans[4]).

Méthode de Alves et Climaco

Cette méthode est interactive, elle combine l'utilisation de la distance de Tchebycheff avec la technique des coupes de Gomory(1963), à chaque itération, l'algorithme proposé fournit la solution non dominée la plus proche du point de référence fixé par le décideur. L'analyse des informations données par le décideur, à chaque étape de dialogue, permet d'ajuster le point de référence de l'itération suivante.

Méthode MOMIX de Teghem et Kunch(1986)

C'est une version interactive de la méthode Branch and Bound. Elle comporte deux phases :

La phase 1

C'est une phase d'initialisation, il s'agit de déterminer le premier point efficace x^0 en minimisant la norme de Tchebychev. Le problème suivant est résolu pour $m=0$.

$$(P_m) \left\{ \begin{array}{l} \min \delta \\ t.q. \quad \beta_{m,k}(M_{m,k} - c^k x) \leq \delta \quad k = 1, 2, \dots, p \\ \quad \quad x \in S_m \end{array} \right. \quad (2.2.16)$$

où $S_0 = S$; $M_{m,k}$ la valeur optimale de la $k^{\text{ème}}$ fonction objectif sur S_m et $\beta_{m,k} = \frac{\alpha_{m,k}}{\sum_{i=1}^p \alpha_{m,k}}$ le poids de la $k^{\text{ème}}$ fonction objectif par exemple $\alpha_{m,k} = \frac{M_{m,k} - n_{m,k}}{\max(|n_{m,k}|, |M_{m,k}|) \|C^k\|}$; où $n_{m,k}$ est la $k^{\text{ème}}$ composante du point nadir sur S_m .

La phase 2

La méthode Branch and Bound interactive est utilisée en deux étapes :

Première étape (Procédure descendante "depth first") :

Soit la $m^{\text{ème}}$ itération.

- Soit x^{m-1} le $m^{\text{ème}}$ compromis et $Z_k = C^k x^{m-1}$ la valeur correspondante pour la $i^{\text{ème}}$ fonction objectif.
- Le DM indique le critère $l_m(1) \in \{1, 2, \dots, p\}$ à améliorer en priorité.
- Un nouveau compromis est obtenu en résolvant le problème (P_m) avec $S_m = S_{m-1} \cap \{x | Z_{l_m(1)}(x) > Z_{l_m(1), (m-1)}\}$ de façon que le critère $l_m(1)$ est amélioré.

Test d'arrêt

Des tests d'arrêts ont été définis, on cite quelques uns :

- a) $S_m = \phi$.
- b) $M_{k,m} - m_{k,m} \leq \epsilon_k$ pour toutes les valeurs k ($\epsilon_k > 0$, fixé).
- c) \hat{Z} , le vecteur des meilleures valeurs des fonctions objectifs fournies par le meilleur compromis trouvé est préféré au vecteur idéal M_m .
- d) Un nombre Q d'itérations a été effectué.

Si aucune amélioration n'a été apporté à \hat{Z} après un nombre Q d'itérations fixé par le décideur, la première étape de la procédure s'arrête.

Deuxième étape (Procédure remontante "backtracking") :

La région d'admissibilité négligée à chaque itération de la procédure descendante est scanner pour un éventuel meilleur compromis que \hat{Z} dans cette région. Les différentes opérations sont :

- Le noeud correspondant au compromis x^{m-1} est séparé en p branches.
- Soit $l_m(k)$; $k = 1, 2, \dots, p$ l'ordre de priorité dans lequel le décideur veut voir améliorer les p critères par rapport à ce compromis ;
- Les p sous noeuds sont obtenus par adjonction respective des contraintes suivantes :

- 1) $Z_{l_m(1)} > Z_{l_m}^{m-1}(1)$.
- 2) $Z_{l_m(2)} > Z_{l_m}^{m-1}(2)$, $Z_{l_m(1)} \geq Z_{l_m}^{m-1}(1)$.
- ⋮
- p) $Z_{l_m(p)} > Z_{l_m}^{m-1}(p)$, $Z_{l_m(k)} \geq Z_{l_m}^{m-1}(k)$. $k = 1, 2, \dots, p$.

L'ensemble de solutions admissibles est partitionné et à chaque sous noeud on résoud le problème (P_m) .

Les mêmes tests d'arrêts sont utilisables.

Il reste à citer la méthode de Stuer and Choo, la méthode de Macott and Soland, l'algorithme de Moulaï et l'algorithme de Chaâbane et d'autres voir [4, 5].

2.2.8 Problèmes d'optimisation à variables binaires

Un nombre considérable de problèmes en nombres entiers est formulé comme problèmes à variables qui ne peuvent prendre que deux valeurs 0 ou 1. Ce type de problèmes est noté MOBLP (Multiple Objective Binary Linear Programming), par exemple le problème de sac à dos, le problème d'affectation, le problème de voyageur de commerce multi-critère, ..., etc.

La formulation mathématique d'un problème d'optimisation à variables binaires est donnée par :

$$(MOBLP) \{ \text{"max"}_{x \in S} Z_k = C^k x, \quad k = 1, 2, \dots, p \}$$

où

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

Il serait intéressant de développer quelques méthodes de résolution concernant ce type de problèmes, malgré qu'il existe quelques unes citons la méthode la plus récente, celle de Ulungu et Teghem et aussi la méthode de Bitran, Méthode de J.Sylva et A.Crema Voir [5].

2.2.9 Complexité

Résoudre des problèmes d'optimisation combinatoire (multi-objectifs) NP-difficiles est très ardu, en effet pour de tels problèmes, les méthodes exactes exigent un effort calculatoire qui croit exponentiellement avec la taille des instances du problème, et rapidement, les méthodes métaheuristiques deviennent l'unique moyen d'obtenir une bonne solution en un temps raisonnable. Ces méthodes métaheuristiques sont principalement basées sur des méthodes de recherche locale (méthodes de descente, tabou, recuit simulé) ou des méthodes évolutives (algorithmes génétiques, colonies de fourmis, Scatter Search).

Nous proposons d'exposer quelques méthodes métaheuristiques pour la résolution des problèmes d'optimisation combinatoires multi-objectifs NP-difficiles.

2.3 Les méthodes métaheuristiques [9, 12]

Contrairement aux méthodes exactes, les méthodes métaheuristiques ne garantissent pas de trouver de manière exacte l'ensemble des solutions Pareto optimales mais une approximation aussi bonne que possible de cet ensemble dans un temps acceptable. Cet approximation est également appelée ensemble de solutions potentiellement efficaces.

Il existe deux types de métaheuristiques ; le premier concerne les méthodes de voisinage dont le but est de créer une solution à partir d'une autre solution grâce à une fonction de voisinage ; le second type regroupe les méthodes à base d'évolution de population qui travaillent avec un ensemble de solutions évoluant pour un mieux vers un ensemble de bonnes solutions.

Les chercheurs ont vu que les méthodes exactes prennent assez de temps et parfois on peut pas avoir une solution exacte donc ils ont pensé aux méthodes Méta-Heuristiques (Travaux de Gandibleux et autres chercheurs).

2.3.1 Introduction

Généralités

Les métaheuristiques forment une famille d'algorithmes d'optimisation qui résolvent des problèmes d'optimisation difficile (souvent issus des domaines de la recherche opérationnelle); elles sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, apprennent les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

Les métaheuristiques sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particuliers).

Le but d'une métaheuristique est de résoudre un problème d'optimisation donné, elle cherche un objet mathématique (une permutation, un vecteur, ..., etc.) minimisant (ou maximisant) une fonction objectif, qui décrit la qualité d'une solution au problème. L'ensemble des solutions possibles forme l'espace de recherche; qui est au minimum borné, mais peut être également limité par un ensemble de contraintes.

2.3.2 Voisinage

Le principe général le plus utilisé dans les méthodes métaheuristiques est celui de voisinage; cette dernière est un impact important sur le comportement des métaheuristiques pour les problèmes combinatoires; tel que à chaque solution d'un problème, on associe un sous-ensemble de solutions. Ce sous-ensemble qu'il est possible d'atteindre par une série de transformations données. Par extension on désigne parfois par le terme " voisinage " l'ensemble des transformations considérées, peut être statique, comme dans le cas du recuit simulé ou des méthodes de bruitage, mais aussi dynamique et dépendre du temps t (ou plus précisément de l'itération à laquelle on se trouve).

Un voisinage simple pour le problème du voyageur de commerce sera, par exemple, l'ensemble des solutions qu'il est possible de construire en permutant deux villes dans une solution donnée. Bien qu'il n'existe que très peu de résultats théoriques sur l'adéquation entre un voisinage et un problème discret donné, il peut être possible d'en calculer des indicateurs empiriques. Les techniques les plus classiques concernant la définition d'un voisinage tournent autour des notions de permutations, de chaînes d'éjections et d'optimisation partielles.

Voisinage sur une permutation

De nombreux problèmes d'optimisation combinatoire peuvent s'exprimer sous la forme d'une recherche de permutations, tous ces problèmes pourraient utiliser le même voisinage puisque leurs solutions se représentent sous la même forme (Problème d'affectation, Voyageur de commerce et d'autres). L'inversion de deux éléments adjacents est un des voisinages les plus simples

pour une permutation. La taille de ce voisinage est en $O(n)$ pour un problème à n objets (villes, sites, tâches) et c'est certainement un des plus petits que l'on puisse imaginer fonctionner en pratique (il possède la propriété de connexité pour ces problèmes).

2.3.3 Population

Principe général des métaheuristiques à population :

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors importante.

Les plus connues dans cette classe sont le recuit simulé, la recherche avec tabous, la recherche à voisinage variable, la méthode GRASP ou encore les méthodes de bruitage.

La notion de population

La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération ; on peut citer les algorithmes génétiques, l'optimisation par essais particuliers, les algorithmes de colonies de fourmis. La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

Autrement un ensemble de solutions, ou population de solutions est une des plus anciennes formes de mémoire utilisée dans les métaheuristiques. En effet, on peut considérer qu'une population de solutions forme une mémoire traduisant l'évolution du processus au cours du temps. Les principales différences qu'il y a entre les algorithmes génétiques, la recherche par dispersion où les évolutions de ces algorithmes résident essentiellement dans la gestion de la population et de son utilisation pour construire de nouvelles solutions.

Chapitre 3

Problèmes d'Optimisation Combinatoire Bi-Objectifs et la méthode en deux phases

3.1 Introduction

L'optimisation multi-critère consiste à optimiser deux ou plusieurs composantes d'un vecteur de fonction, chaque composante de ce vecteur est une fonction objectif. Pour les solutions du problème multi-critère, la relation d'ordre n'est pas totale, une solution peut être meilleure qu'une autre sur certains critères et moins bonne sur les autres. Donc il convient de travailler dans des populations de solutions dites "Pareto-optimales"¹.

Il existe plusieurs problèmes classiques d'optimisation combinatoire bi-objectifs en variables binaires qu'on rencontre souvent dans la réalité, citons quelques uns, le problème d'affectation "Assignment Problem"(AP), le problème de Sac à Dos "Knapsack"(KP), le problème de couverture d'ensemble "Set Covering"(SCP), le problème de "Set Paking"(SPP), le problème de transport, le problème de location, le problème plus court chemin, ..., etc.

Les premières approches de l'optimisation multi-critère n'utilisaient pas ces notions de dominance et de la solution Pareto optimale, elle ramenaient les problèmes multi-critères à un seul problème mono-critère (agrégation), par contre, ces approches ont négligé une partie de l'ensemble de solutions; ils se sont intéressé qu'à la détermination de sous ensembles de solutions supportées (Théorème de Geoffrion).

Ensuite, des méthodes exactes se sont révélées, consiste à la détermination d'un ensemble complet, c'est à dire pour chaque point non-dominé au moins une solution efficace. Parmi ces méthodes, séquence de pivots du simplexe (Malhotra et al, 1982); la méthode en deux phases : une procédure générale pour résoudre des problèmes combinatoires bi-objectifs (Ulungu et Teghem (1995)), la méthode dichotomique en utilisant un solveur (Gandibleux et Degoutin, 2002). Dans ce chapitre, nous nous intéresserons à la description de la méthode en deux phases de Ulungu et Teghem dans son contexte général, en l'adaptant au problème de de Sac à Dos et le problème d'affectation.

Nous suggérons ensuite une méthode pour la recherche de l'ensemble de solutions efficace pour le problème de Sac à Dos "Knapsack"(KP) bi-critère différente de la méthode séparation et évaluation (Branch and Bound).

¹Voir chapitre 2

3.2 La méthode exacte des deux phases dans son contexte général [14, 15]

La méthode des deux phases pour résoudre les problèmes d'optimisation bi-objectifs en nombres binaires a été développée par Ulungu et Teghem(1994),son principe est de résoudre le problème en deux phases.

L'ensemble de solutions efficaces noté $IE(P)$ d'un problème bi-objectif en nombres entiers est obtenu en deux étapes :

Les solutions efficaces supportées (SE(P))

Celles qui se situent sur l'enveloppe convexe de l'ensemble des solutions, elles peuvent être obtenues par résolution du problème d'optimisation combinatoire uni-critère paramétrique obtenu par agrégation linéaire des différents objectifs. Ce problème se présente comme suit :

$$\min \left(\sum_{i=1}^2 \lambda_i Z_i(x) \right)$$

avec

$$\lambda \in \Lambda = \left\{ \lambda_i > 0, i = 1, 2 \quad \mid \quad \sum_{i=1}^2 \lambda_i = 1 \right\}$$

Les solutions efficaces non-supportées (NSE(P))

Cette partie de solutions ne peuvent pas être obtenues par résolution du problème uni-critère, ces solutions sont localisées dans les triangles formés par les solutions supportées $\Delta Z^r Z^s$ appelées "zones potentielles de solutions efficaces".

Le choix final d'une solution efficace est effectué par le décideur. Ce dernier peut ne pas être satisfait du seul choix apporté par l'ensemble de solutions supportées, en effet ces derniers peuvent ne représenter qu'une faible partie des solutions efficaces.

3.2.1 Formulation mathématique du problème bi-AP

L'ensemble de solutions d'un problème d'affectation multi-objectifs est restreint à un problème bi-objectifs. Pour plus de deux objectifs, c'est un problème très difficile, il vient d'être traité par Anthony Przybylski, Xavier Gandibleux, Matthias Ehrgott. Avec un objectif il est clair qu'un problème d'affectation peut être résolu comme un programme linéaire car l'unimodularité de la matrice de contraintes garantie une solution optimale entière. On peut aussi résoudre le problème avec la méthode hongroise. Ces méthodes ne peuvent pas s'appliquer dans le cas multi-objectif, on peut seulement trouver l'ensemble de solutions d'une fonction d'agrégation des deux objectifs. La section suivante montre qu'il peut y avoir d'autres solutions différentes de celles obtenues par l'agrégation, ces solutions n'étaient pas prise en considération par quelques chercheurs.

Dans la section suivante nous allons appliqués la méthode en deux phases dans le problème d'affectation bi-objectif.

La formulation générale du problème d'affectation bi-objectif est la suivante :

Soit $C_{ij}^p \in \mathbb{N}^+$ et $X = (x_{11}, x_{12}, \dots, x_{nn})$.

$$(bi - AP) \left\{ \begin{array}{l} \text{"min"} Z_k(X) = \sum_{i=1}^n \sum_{j=1}^n C_{ij}^k x_{ij} \quad k = 1, 2 \\ \sum_{j=1}^n x_{ij} = 1 \quad i = \overline{1, n} \\ \sum_{i=1}^n x_{ij} = 1 \quad j = \overline{1, n} \\ x_{ij} \in \{0, 1\} \end{array} \right. \quad (3.2.1)$$

La méthode en deux phases est basée sur un problème uni-critère obtenu par agrégation des deux critères, la première phase détermine toutes les solutions efficaces supportées, la seconde détermine toutes les solutions efficaces non supportées en se basant sur les solutions supportées déjà trouvées.

Remarque 14 Pour simplifier l'écriture le problème (bi-AP) est noté P dans la recherche de l'ensemble de solutions efficaces.

3.2.2 Détermination des solutions supportées SE(P)

Au début, les chercheurs se contentent de cette seule phase pour déterminer toutes les solutions efficaces d'un problème, après d'autres chercheurs ont montré que cette phase ne détermine qu'un sous ensemble de solutions efficaces dites "solutions efficaces supportées".

La première phase consiste à déterminer pour chaque objectif une solution optimale.

Soit \tilde{X}_1 la solution optimale du critère Z_1 .

Soit \tilde{X}_2 la solution optimale du critère Z_2 .

Soit \tilde{S} la liste des solutions efficaces supportées trouvées ; cette liste est initialiser par les deux solutions optimales \tilde{X}_1 et \tilde{X}_2 respectivement des deux critères Z_1 et Z_2 où leurs valeurs dans l'espace des critères sont $(\tilde{Z}_1, \tilde{Z}_2)$ et (\hat{Z}_1, \hat{Z}_2) .

Les solutions de \tilde{S} sont rangées par ordre croissant du critère Z_1 .

Soient X^r et X^s deux solutions consécutives de \tilde{S} ; on a donc :

$$Z_1^{(r)} < Z_1^{(s)}$$

et

$$Z_2^{(r)} > Z_2^{(s)}$$

Une agrégation des deux critères est réalisée à l'aide des poids :

$$\lambda_1^{(t)} = Z_2^{(r)} - Z_2^{(s)} \quad \text{et} \quad \lambda_2^{(t)} = Z_1^{(s)} - Z_1^{(r)}$$

Le problème uni-critère $P^{(t)}$, avec la matrice $C^{(t)} = [\lambda_1^{(t)} c_{ij}^{(1)} + \lambda_2^{(t)} c_{ij}^{(2)}]$ est considéré et optimisé :

$$(P^{(t)}) \left[\begin{array}{l} \min Z^{(t)} = \lambda_1^{(t)} Z_1(x) + \lambda_2^{(t)} Z_2(x) \\ \text{avec} \left\{ \begin{array}{l} \lambda_1^{(t)} = Z_2^{(r)} - Z_2^{(s)} \\ \lambda_2^{(t)} = Z_1^{(s)} - Z_1^{(r)} \end{array} \right. \end{array} \right] \quad (3.2.2)$$

NB :

La direction de la fonction économique déterminée par la matrice $C^{(t)}$ n'est rien d'autre que la concrétisation, pour le problème bi-critère, de la direction de recherche "G(X)" de la méthode de Gonzalez et al [14]. C'est l'image inverse de la droite $[Z^r, Z^s]$ du plan des deux critères dans l'espace des décisions R^{n^2} .

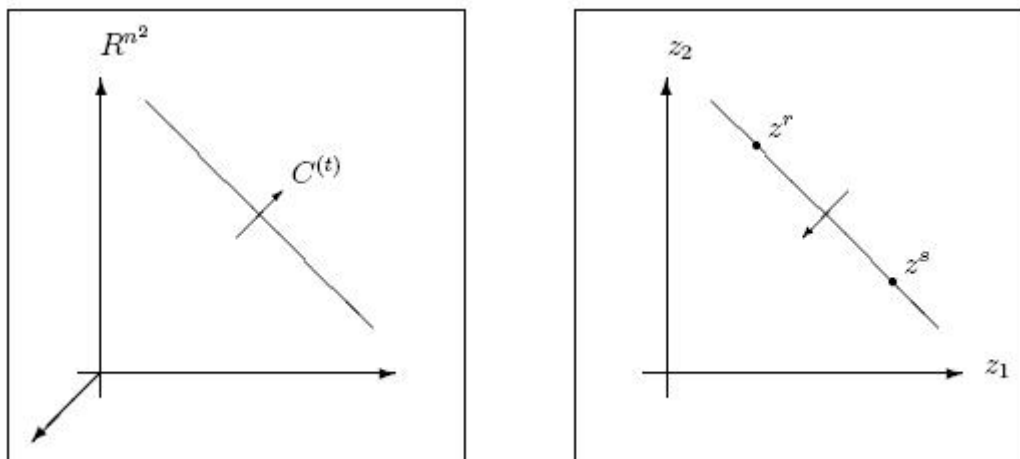


FIG. 3.1 – Direction de recherche et l'Image de la direction de recherche (resp)

Soit $SE(P^{(t)}) = \{X^t, t \in T\}$ l'ensemble de solutions optimales du problème uni-critère $P^{(t)}$ obtenu par la méthode hongroise.

a) Soit $\{X^r, X^s\} \cap SE(P^{(t)}) = \emptyset$; Donc la solution X^t est une nouvelle solution supportée qui appartient à l'ensemble \tilde{S} dont son image constitue un autre point extrême différent de $Z^{(r)}$ et $Z^{(s)}$.

b) Sinon $\{X^r, X^s\} \subset SE(P^{(t)})$.

- Si $SE(P^{(t)}) = \{X^r, X^s\}$, donc il n'y a pas d'autres solutions supportées entre X^r et X^s .
- Dans le cas contraire, les solutions de $SE(P^{(t)}) \setminus \{X^r, X^s\}$ sont des nouvelles solutions supportées où la représentation dans l'espace des critères est situé sur la droite engendrée par les points $Z^{(r)}$ et $Z^{(s)}$. Si de telles solutions existent, elles sont placées dans un ensemble particulier \tilde{S}' .

Cette première phase est poursuivie jusqu'à ce que toutes les paires (X^r, X^s) de l'ensemble \tilde{S} sont examinées. En particulier, dans le cas (a) les paires (X^r, X^t) et (X^t, X^s) doivent être examinées.

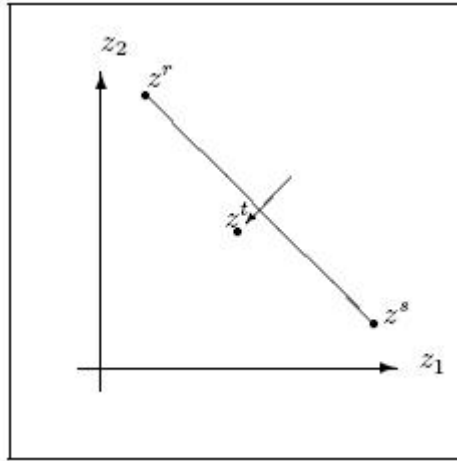


FIG. 3.2 – Nouvelle solution supportée extrême

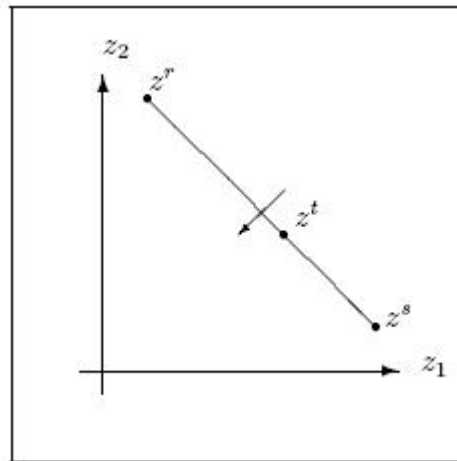


FIG. 3.3 – Nouvelle solution supportée non extrême

A la fin de cette phase, on aura $SE(P) = \tilde{S} \cup \tilde{S}'$.

Après la description de la première phase, nous présenterons par la suite un algorithme qui détermine l'ensemble des solutions efficaces supportées d'un problème d'affectation bi-objectif.

Algorithme 1 (Phase 1 Détermination des solutions supportées)

faire entrer les matrices coûts C^1, C^2 .

Résoudre le problème d'affectation par la méthode hongroise pour la matrice coût C^1 .

Soit \tilde{X}_1 la solution optimale associée.

Résoudre le problème d'affectation par la méthode hongroise pour la matrice coût C^2 .

Soit \tilde{X}_2 la solution optimale associée.

Soit $Z^1 = (Z_1(\tilde{X}_1), Z_2(\tilde{X}_1)) = (\hat{Z}_1, \hat{Z}_2)$

Soit $Z^2 = (Z_1(\tilde{X}_2), Z_2(\tilde{X}_2)) = (\hat{Z}_1, \hat{Z}_2)$

L'ensemble des solutions efficaces supportées initial \tilde{S} est : $\tilde{S} = \{\tilde{X}_1, \tilde{X}_2\}$

La recherche des solutions efficaces supplémentaires entre \tilde{X}_1 et \tilde{X}_2 :

Résolution récursive $(\tilde{X}_1, \tilde{X}_2, \tilde{S})$

faire sortir le résultat \tilde{S} .

Algorithme 2 (Résolution récursive)

faire entrer : $X^{(r)}, X^{(s)} \in \tilde{S}$ avec $Z_1^{(r)} < Z_1^{(s)}$ et $Z_2^{(s)} < Z_2^{(r)}$.

$$\lambda_1^{(t)} := Z_2^{(r)} - Z_2^{(s)}$$

$$\lambda_2^{(t)} := Z_1^{(s)} - Z_1^{(r)}$$

\tilde{S}' c'est l'ensemble des solutions optimales du problème d'affectation avec la matrice coût $\lambda_1^{(t)}C^1 + \lambda_2^{(t)}C^2$.

$$\tilde{S} := \tilde{S} \cup \tilde{S}'.$$

Si $\lambda_1^{(t)} Z_1(X^{(t)}) + \lambda_2^{(t)} Z_2(X^{(t)}) < \lambda_1^{(t)} Z_1^{(r)} + \lambda_2^{(t)} Z_2^{(r)}$ alors $X^{(t)}$ est une nouvelle solution extrême supportée avec $Z^{(t)} = (Z_1^{(t)}, Z_2^{(t)})$.

Résolution récursive($X^{(r)}, X^{(t)}, \tilde{S}$).

Résolution récursive($X^{(t)}, X^{(s)}, \tilde{S}$).

Fin si.

3.2.3 Détermination des solutions non supportées E(P) | SE(P)

Cette phase est la partie la plus originale du travail élaboré par Ulungu et Teghem. L'objectif est de rechercher, pour chaque paire ($X^{(r)}, X^{(s)}$) de l'ensemble de solutions de SE(P), s'il existe des solutions efficaces non supportées X^u vérifiant :

$$\left(\begin{array}{l} Z_1^{(r)} < Z_1^{(u)} < Z_1^{(s)} \\ \text{et} \\ Z_2^{(r)} > Z_2^{(u)} > Z_2^{(s)} \end{array} \right)$$

des solutions pareilles seraient situées² à l'intérieur du triangle $\Delta[YZ^{(s)}Z^{(r)}]$ ³(FIG 3.4).

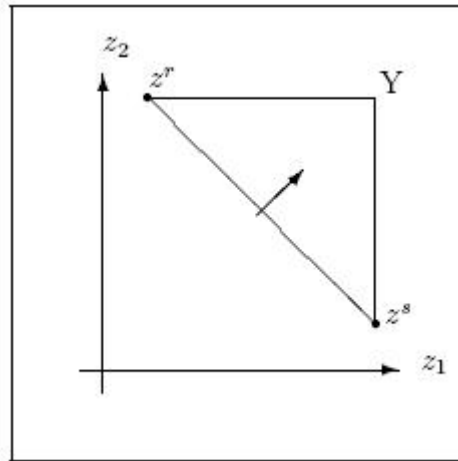


FIG. 3.4 – Zone de détermination des solutions efficaces non-supportées

Pour déterminer l'ensemble de solutions non supportées, la procédure suivante est considérée :

²Dans le cas d'un problème minimiser les objectifs.

³Y représente le point (Z_1^s, Z_2^r) .

le problème $P^{(t)}$ associé aux couple $(X^{(r)}, X^{(s)})$ est optimisé, les affectations (i, j) de coût réduit strictement positif.

$$\text{Soit } L = \left\{ (i, j) \mid \bar{c}_{ij}^{(t)} > 0 \right\}$$

Les affectations sont classées dans l'ordre croissant des valeurs $\bar{c}_{ij}^{(t)}$.

La première imposition de ces affectations conduit à réoptimiser le problème $P^{(t)}$ réduit. Les solutions $X^{(r)}$ et $X^{(s)}$, restent acceptables, dans ce cas une seule itération de la méthode hongroise est nécessaire pour réoptimiser le problème.

Cependant toutes les affectations de L ne sont pas candidates ; certaines peuvent être éliminées dès le début de la liste L, si elles conduiraient à des solutions dominées situées dans les zones hors triangle $\Delta[Y Z^{(s)} Z^{(r)}]$, comme le montre les trois cas ci-dessous.

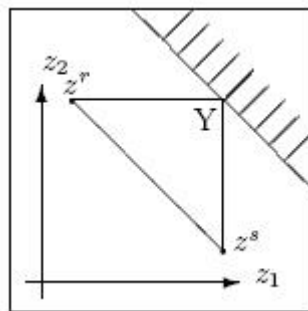


FIG. 3.5 – Cas (1)

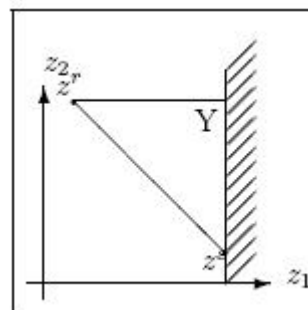


FIG. 3.6 – Cas (2)

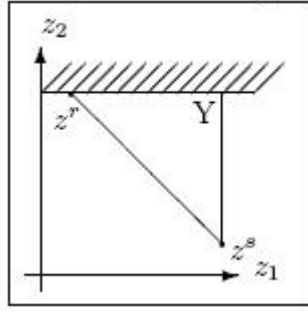


FIG. 3.7 – Cas (3)

L'idée de la détermination d'une affectation pouvant être éliminée de la liste L est de calculer une borne inférieure de l'augmentation, respectivement pour les trois cas :

Le premier cas

du critère $Z^{(t)}$ correspondant la matrice $C^{(t)}$ par rapport à la valeur $\tilde{Z}^{(t)}$ prise par les points $Z^{(r)}$ et $Z^{(s)}$.

Le deuxième cas

du critère Z_1 par rapport à la valeur $Z_1^{(r)}$

Le troisième cas

du critère Z_2 par rapport à la valeur $Z_2^{(s)}$

engendré par l'imposition de cette affectation. Ces bornes inférieures sont calculées à partir des matrices coûts réduits.

Test 1

Test 1 sur le problème $P^{(t)}$

Soit (i^*, j^*) une affectation de L et soit i_r, j_r, i_s, j_s les indices tels que

$$x_{i_r j^*} = x_{i^* j_r} = x_{i_s j^*} = x_{i^* j_s} = 1.$$

Les indices i_r, i_s d'une part et les indices j_r, j_s d'autre part, pouvant désigner le même indice. Par rapport à la solution X^r , l'imposition de (i^*, j^*) va créer au minimum un autre zéro dans la ligne i_r et dans la colonne j_r dont la matrice de coût réduit $\bar{C}^{(t)}$ autre que ceux correspondant aux affectations (i_r, j^*) et (i^*, j_r) .

Si ce nouveau zéro est créer à la position (i_r, j_r) la borne inférieure de l'augmentation de $\tilde{Z}^{(t)}$ sera $\bar{c}_{i_r j_r}^{(t)}$.

Si deux nouveaux zéros sont créer, la borne inférieure de l'augmentation sera :

$$\Gamma_r^{(t)} = \min_{j \neq j^*} \bar{c}_{i_r j}^{(t)} + \min_{i \neq i^*} \bar{c}_{i j_r}^{(t)}. \quad (3.2.3)$$

Le même raisonnement peut être fait en remplaçant X^r par X^s , et la borne inférieure de l'augmentation de $\tilde{Z}^{(t)}$ sera :

$$l_t = \bar{c}_{i^* j^*}^{(t)} + \min \left(\bar{c}_{i_r j_r}^{(t)}, \bar{c}_{i_s j_s}^{(t)}, \Gamma_r^{(t)}, \Gamma_s^{(t)} \right) \quad (3.2.4)$$

L'affectation (i^*, j^*) peut être éliminée de la liste L si $l_t \geq \lambda_1^{(t)} Z_1^s + \lambda_2^{(t)} Z_2^r - \tilde{Z}^{(t)}$;

On a :

$$\tilde{Z}^{(t)} = \lambda_1^{(t)} Z_1^s + \lambda_2^{(t)} Z_2^s = \lambda_1^{(t)} Z_1^r + \lambda_2^{(t)} Z_2^r \quad (3.2.5)$$

On remplaçant $\tilde{Z}^{(t)}$ par sa valeur, on trouve

$$l_t \geq \lambda_1^{(t)} \lambda_2^{(t)} \quad (3.2.6)$$

où l_t représente la borne inférieure de l'augmentation de $\tilde{Z}^{(t)}$.

On peut donc éliminer dans la liste L des affectations qui, une fois imposées et après réoptimisation, fourniront des solutions qui ne sont pas efficaces. Même si l'augmentation réelle de $\tilde{Z}^{(t)}$ est inférieure à $\lambda_1^{(t)} \lambda_2^{(t)}$, les solutions correspondantes peuvent être des solutions qui se situent hors triangle.

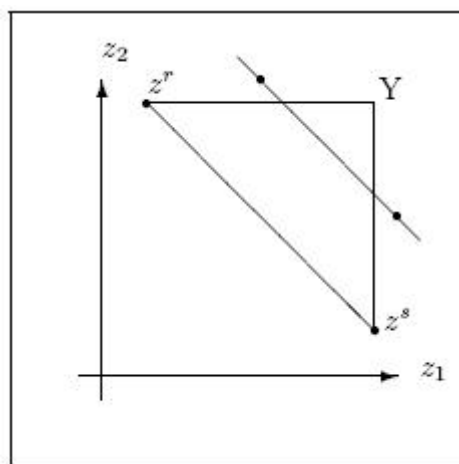


FIG. 3.8 – Solution dominée correspondante à une augmentation de $\tilde{Z}^{(t)}$ inférieure à $\lambda_1^{(t)} \lambda_2^{(t)}$

Les tests 2 et 3 détectent ces cas.

Test 2

Test d'augmentation de $Z_1^{(r)}$

a) $X^r = \tilde{X}_1$

Dans ce cas (comme au test (1)) la matrice de coût réduit de Z_1 relativement à X^r ne contient que des éléments positifs ou nuls. Aussi le même type de raisonnement que celui du test 1 peut être appliqué.

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_1 \geq Z_1^{(s)} - Z_1^{(r)} = \lambda_2^t \quad (3.2.7)$$

l_1 est la borne inférieure de l'augmentation de $Z_1^{(r)}$ engendrée par l'imposition de l'affectation (i^*, j^*) qui vaut :

$$l_1 = \bar{c}_{i^*j^*}^{(1)} + \min \left(\bar{c}_{i_1j_1}^{(1)}, \min_{j \neq j^*} \bar{c}_{i_1j}^{(1)} + \min_{i \neq i^*} \bar{c}_{ij_1}^{(1)} \right) \quad (3.2.8)$$

avec $\tilde{x}_{i^*j_1}^1 = \tilde{x}_{i_1j^*}^1 = 1$ dont les coûts réduits \bar{c}_{ij}^1 de Z_1 relative à X^r .

b) $\underline{X^r} \neq \tilde{X}_1$

Dans ce cas la matrice des coûts réduits de Z_1 relativement X^r contient des éléments négatifs, qui peuvent être utilisés lors d'une réoptimisation.

Aussi la borne inférieure de l'augmentation Z_1^r est calculée sur base des éléments minimaux situés dans les $n - 1$ lignes (colonnes) qui peuvent être réaffecter.

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_1 = \bar{c}_{i^*j^*}^{(1)} + \max \left(\sum_{i \neq i^*} \min_{j \neq j^*} \bar{c}_{ij}^{(1)}, \sum_{j \neq j^*} \min_{i \neq i^*} \bar{c}_{ij}^{(1)} \right) \geq \lambda_2^{(t)} \quad (3.2.9)$$

Ce test (3.2.9) se révélera de moins en moins efficace au fur et à mesure que la valeur de $Z_1^{(r)}$ sera plus grande, c'est-à-dire que la solution X^r est éloigné de \tilde{X}_1 .

Remarque 15 : La présence davantage des coûts réduits négatifs indique que l'imposition de (i^*, j^*) peut conduire à des diminutions de $Z_1^{(r)}$.

Test 3

Test d'augmentation de $Z_2^{(s)}$

Le même raisonnement qu'au paravent.

a) $\underline{X^s} = \tilde{X}_2$

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_2 = \bar{c}_{i^*j^*}^{(2)} + \min \left(\bar{c}_{i_2j_2}^{(2)}, \min_{j \neq j^*} \bar{c}_{i_2j}^{(2)} + \min_{i \neq i^*} \bar{c}_{ij}^{(2)} \right) \geq Z_2^{(r)} - Z_2^{(s)} = \lambda_1^t \quad (3.2.10)$$

avec $\tilde{x}_{i^*j_2}^2 = \tilde{x}_{i_2j^*}^2 = 1$ dont les coûts réduits \bar{c}_{ij}^2 de Z_2 relative à X^s .

b) $\underline{X^s} \neq \tilde{X}_2$

L'affectation (i^*, j^*) peut être éliminée de la liste L si

$$l_2 = \bar{c}_{i^*j^*}^{(2)} + \max \left(\sum_{i \neq i^*} \min_{j \neq j^*} \bar{c}_{ij}^{(2)}, \sum_{j \neq j^*} \min_{i \neq i^*} \bar{c}_{ij}^{(2)} \right) \geq \lambda_1^t \quad (3.2.11)$$

Pour les mêmes raisons, ce test (3.2.11) n'a qu'une chance d'être efficace que pour les faibles valeurs de $Z_2^{(s)}$.

Les affectations restantes de L sont imposées une par une et le même problème $P^{(t)}$ est réoptimisé ; les solutions obtenues sont triées, on maintient que les solutions efficaces.

Les affectations de L sont considérées dans l'ordre des $\bar{c}_{ij}^{(t)}$ croissants, ou dans l'ordre des bornes l_t croissante.

Il n'est toutefois pas nécessaire de considérer toutes les affectations de L pour déterminer si une nouvelle solution X^u se trouve dans le triangle $\Delta[YZ^{(r)}Z^{(s)}]$ est efficace.

En effet, si X^u correspond à la valeur $Z^{(u)}$ pour l'optimisation du critère $P^{(t)}$ et que toutes les bornes inférieurs l_t des autres affectations de L indiquent que leur imposition conduira à une valeur supérieur à $Z^{(u)}$, ceci signifie que X^u est une solution efficace non supportée⁴.

Dès qu'une nouvelle solution efficace X^u (non supportée) est trouvée, le test 1 peut être réappliqué, et le nouveau triangle est considérée (on a pas besoin de tout le triangle(FIG 3.10)).

⁴Ceci revient à explorer le triangle en gardant la même direction de recherche (FIG 3.9).

Le membre l_t est actualisé de manière à éliminer la région hachurée (FIG 3.10) ; sa nouvelle valeur (plus petite que la précédente) vaut :

$$\max(\lambda_1^{(t)} Z_1^{(u)} + \lambda_2^{(t)} Z_2^{(r)} - \tilde{Z}^{(t)}, \lambda_1^{(t)} Z_1^{(s)} + \lambda_2^{(t)} Z_2^{(u)} - \tilde{Z}^{(t)}) = \max(\lambda_1^{(t)} (Z_1^{(u)} - Z_1^{(r)}), \lambda_2^{(t)} (Z_2^{(u)} - Z_2^{(s)})) \quad (3.2.12)$$

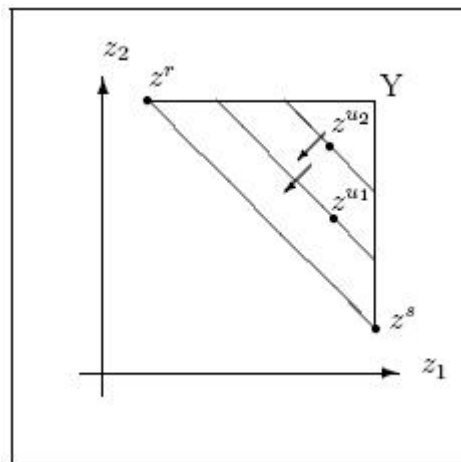


FIG. 3.9 – Exploration dans la même direction

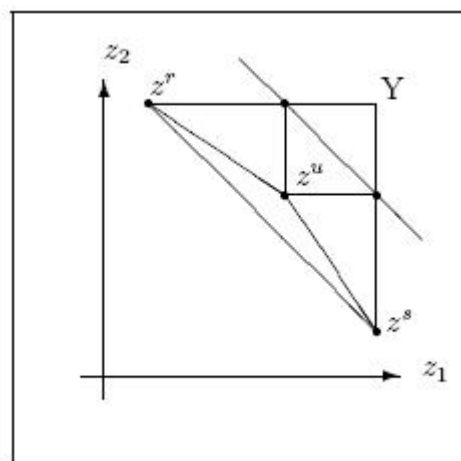


FIG. 3.10 – Actualisation du test 1

Ainsi de nouvelles affectations peuvent être supprimées de L et ne plus être considérées.

Une autre possibilité consiste à, une fois une solution efficace non supportée X^u est déterminée, de considérer les couples (X^r, X^u) et (X^u, X^s) , de modifier la direction de recherche utilisée pour "balayer" le triangle $\Delta[Y Z^r Z^s]$, en la remplaçant par les deux directions de recherche associées respectivement aux droites $Z^r Z^u$ et $Z^u Z^s$ (FIG 3.11).

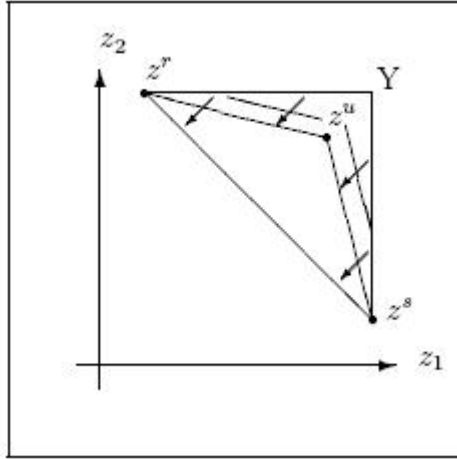


FIG. 3.11 – Exploration dans une direction modifiée

Lorsqu'un nouveau problème $P^{(t)}$ est considéré, il faut cependant réoptimiser (comme dans la phase (1)) avant d'introduire de nouvelles affectations, pour s'assurer que la solution X^u n'est pas dominée par une autre solution du triangle $\Delta[Z^r Z^u Z^s]$. Cette deuxième phase est répétée pour chaque paire $(X^r, X^s) \in \tilde{S}$. Quand on applique la méthode Hongroise pour le problème $P^{(t)}$, si la valeur courante $Z^{(u)}$ vérifie $Z^{(u)} \geq \tilde{Z}^{(t)} + \lambda_1^{(t)} \lambda_2^{(t)}$ alors, le processus d'optimisation s'arrête. Supposons que ce n'est pas le cas, mais la solution X^u est à l'extérieur du triangle $\Delta[Y Z^r Z^s]$ (FIG 3.12). En effet, il se peut que l'imposition de l'affectation (i, j) de L engendre une solution extérieure du triangle $\Delta[Y Z^r Z^s]$ mais néanmoins, par l'imposition simultanée de (i, j) et d'une autre affectation de L conduisent à une solution $X^{u'}$, de valeur $Z^{(u')}$ supérieure, mais elle se situe à l'intérieur du triangle $\Delta[Y Z^r Z^s]$ (FIG 3.12).

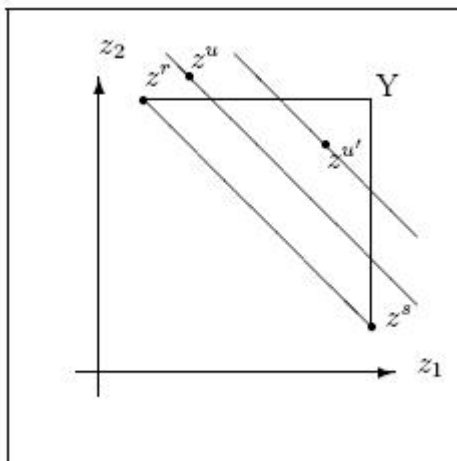


FIG. 3.12 – Solution $Z^{u'}$ est dans le triangle $\Delta[Y Z^r Z^s]$

Il faut donc poursuivre l'optimisation de $P^{(t)}$, en imposant en plus de (i, j) , d'autres affectations de L , jusqu'à ce qu'on trouve une solution à l'intérieur du triangle, ou bien l'inégalité ci-dessus soit vérifiée.

Soit $X^{(i)}, X^{(i+1)} \in S$ défini au paravent.

Dans la phase (2), l'ensemble P des solutions efficaces potentielles du problème d'affectation avec $Z(X)$ dans le triangle $\Delta[YZ^{(i)}Z^{(i+1)}]$, est maintenu.

Les solutions dans P sont aussi ordonnées dans l'ordre croissant de $Z_1(X)$.

Initialiser, $P = \phi$. Les nouvelles solutions, qui sont des solutions optimales du problème d'affectation, où une ou plusieurs variables dans la liste L ont été fixé égal à 1, ajouter à cette liste. Elles peuvent être rejeter encore une fois si on prouve qu'elles peuvent être dominées.

Enfin, P est l'ensemble de solutions efficaces non-supportées dans le triangle.

Soit $P = \{X^1, \dots, X^s\}$ à quelques étapes de l'algorithme et définissons :

$$\nu = \max_{j=1}^{s-1} \left\{ \lambda_1^{(t)} Z_1^{(j+1)} + \lambda_2^{(t)} Z_2^{(j)} \right\} \quad (3.2.13)$$

Alors chaque affectation faisable avec $Z_1(X) + Z_2(X)$ grande que la borne supérieure (the upper bound)

$$UB = \max \left\{ \nu, \lambda_1^{(t)} Z_1^1 + \lambda_2^{(t)} Z_2^{(i)}, \lambda_1^{(t)} Z_1^{(i+1)} + \lambda_2^{(t)} Z_2^{(s)} \right\} \quad (3.2.14)$$

est dominée.

NB :

Cette borne supérieure est proposée par Tuyttens et al. (2000)[8].

Après la description de la deuxième phase, nous présenterons par la suite un algorithme qui détermine l'ensemble des solutions efficaces non-supportées d'un problème d'affectation bi-objectif.

Algorithme 3 (Phase 2 Détermination des solutions non-supportées)

Faire entrer : S, C^1 et C^2 . $S := S$.

Pour tout $X^{(i)}, X^{(i+1)} \in S$, faire

Trouver une solution \tilde{X} du problème d'affectation avec une matrice coût $C^{(t)} = (Z_2^{(i)} - Z_2^{(i+1)})C^{(1)} + (Z_1^{(i+1)} - Z_1^{(i)})C^{(2)}$.

Soit $\bar{C}^{(t)}$ la matrice coût réduit de $C^{(t)}$ de \tilde{X} .

Soit \bar{C}^1 la matrice coût réduit de C^1 pour $X^{(i)}$.

Soit \bar{C}^2 la matrice coût réduit de C^2 pour $X^{(i+1)}$.

$L := \left\{ (i, j) : \bar{c}_{ij}^{(t)} > 0 \right\}$.

$P = \phi$ (P est l'ensemble de solutions non-supportées qui sont dans le triangle engendrées par les solutions extrêmes $X^{(i)}$ et $X^{(i+1)}$).

Calculer la borne supérieure UB définie dans (3.2.14).

Si $UB \geq \lambda_1^{(t)} Z_1^{(i)} + \lambda_2^{(t)} Z_2^{(i+1)}$ alors

Pour tout $(i, j) \in L$ faire

Si les tests 1, 2, 3 sont vérifiés alors

$L := L \setminus \{(i, j)\}$.

fin pour

Si $L \neq \phi$ alors

Pour tout $(i, j) \in L$ faire

Impose($(i, j), \phi$)

fin pour

fin si

fin si $S = S \cup P$

fin pour
Faire sortir : S.

Algorithme 4 (Impose)

Faire entrer : $(k, l), L'$.

(L' est la liste des variables déjà affectées (phase 1)).

Trouver toutes les solutions du problème d'affectation avec la matrice coût C^λ et $x_{ij} = 1$ pour tout $(i, j) \in L' \cup \{i, j\}$.

$L' = L' \cup \{(k, l)\}$.

Si $UB \leq \lambda_1^{(t)} Z_1(X^{(t)}) + \lambda_2^{(t)} Z_2(X^{(t)})$ alors

Si $Z(X^{(t)})$ est dans le triangle et $Z(X^{(t)})$ est non dominé alors

$P := P \cup \{X^{(t)}\}$.

Enlever les solutions dominées de P.

Calculer la borne supérieure u définie dans (3.2.14).

Fin Si.

Pour tout $(i, j) \in L'$ faire

Si la paire (i, j) est compatible⁵ avec L' alors

Impose($(i, j), L'$).

$L' := L' \setminus \{(k, l)\}$

Fin Si.

Fin Pour.

Fin Si.

3.2.4 Illustration de la méthode sur deux exemples

Exemple 1

$$C^{(1)} = \begin{vmatrix} 7 & 5 & 2 \\ 8 & 3 & 4 \\ 2 & 9 & 5 \end{vmatrix} \quad \text{et} \quad C^{(2)} = \begin{vmatrix} 5 & 2 & 6 \\ 3 & 7 & 8 \\ 9 & 4 & 5 \end{vmatrix}$$

A) Détermination de SE(P)

Par application de la méthode hongroise les tableaux optimaux se présentent :

$$\bar{C}^{(1)} = \begin{vmatrix} 5 & 3 & 0 \\ 5 & 0 & 1 \\ 0 & 7 & 3 \end{vmatrix} \quad \text{et} \quad \bar{C}^{(2)} = \begin{vmatrix} 3 & 0 & 3 \\ 0 & 4 & 4 \\ 5 & 0 & 0 \end{vmatrix}$$

$$\tilde{X}_1 = x_{13} = x_{22} = x_{31} = 1 \quad \text{avec} \quad (\tilde{Z}_1, \hat{Z}_2) = (7, 22)$$

$$\tilde{X}_2 = x_{12} = x_{21} = x_{33} = 1 \quad \text{avec} \quad (\hat{Z}_1, \tilde{Z}_2) = (18, 10)$$

Ce sont les deux solutions efficaces supportées initiales, on déduit les poids de la fonction d'agrégation :

$$\lambda_1^{(t)} = 22 - 10 = 12 \quad \text{et} \quad \lambda_2^{(t)} = 18 - 7 = 11$$

et la matrice

$$C^{(t)} = 12c_{ij}^{(1)} + 11c_{ij}^{(2)}$$

⁵exemple c'est impossible d'imposer x_{13} et x_{15} , $(1, 3)$ et $(1, 5)$ sont incompatible

$$C^{(t)} = \begin{vmatrix} 139 & 82 & 90 \\ 129 & 113 & 136 \\ 123 & 152 & 115 \end{vmatrix}$$

La résolution par la méthode hongroise fournit la matrice des coûts réduits

$$\bar{C}^{(t)} = \begin{vmatrix} 41 & [0] & [0] \\ [0] & 0 & 15 \\ [0] & 45 & [0] \end{vmatrix}$$

Les deux seules affectations optimales sont seules \tilde{X}_1 et \tilde{X}_2 , donc il n'existe pas d'autres solutions supportées. On passe à l'étape de recherche des solutions non supportées.

B) Détermination de E(P) | SE(P)

$$L = \{(2, 3), (1, 1), (3, 2)\}$$

On impose l'affectation (2,3), d'où la solution :

$$X_3 = \{x_{12} = x_{23} = x_{31} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (11, 19)$$

D'après la définition de la solution efficace X_3 est une solution efficace.

On impose ensuite l'affectation (1,1), d'où la solution :

$$X_4 = \{x_{11} = x_{22} = x_{33} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (17, 15)$$

c'est une solution efficace.

Le test (3.2.9) permet d'éliminer l'affectation (3,2) de L

$$t_1 = \bar{c}_{i^*j^*}^{(1)} + \max \left(\sum_{i \neq i^*} \min_{j \neq j^*} \bar{c}_{ij}^{(1)}, \sum_{j \neq j^*} \min_{i \neq i^*} \bar{c}_{ij}^{(1)} \right) \geq \lambda_2^{(t)}$$

donc $7 + 5 > 11$; d'une autre manière quand on impose (3,2), la solution optimale est :

$$X_5 = \{x_{32} = x_{21} = x_{13} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (19, 13)$$

Cette solution est hors triangle, c'est à éliminer.

L'ensemble de solutions efficaces est $\{\tilde{X}_1, \tilde{X}_2, X_3, X_4\}$

Exemple 2

$$C^{(1)} = \begin{vmatrix} 5 & 1 & 4 & 7 \\ 6 & 2 & 2 & 6 \\ 2 & 8 & 4 & 4 \\ 3 & 5 & 7 & 1 \end{vmatrix} \quad \text{et} \quad C^{(2)} = \begin{vmatrix} 3 & 6 & 4 & 2 \\ 1 & 3 & 8 & 3 \\ 5 & 2 & 2 & 3 \\ 4 & 2 & 3 & 5 \end{vmatrix}$$

A) Détermination de SE(P)

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(1)} = \begin{vmatrix} 4 & [0] & 3 & 6 \\ 4 & [0] & [0] & 4 \\ [0] & 6 & 2 & 2 \\ 2 & 4 & 6 & [0] \end{vmatrix} \quad \text{et} \quad \bar{C}^{(2)} = \begin{vmatrix} 1 & 4 & 2 & [0] \\ [0] & 2 & 7 & 2 \\ 3 & 0 & [0] & 1 \\ 2 & [0] & 1 & 3 \end{vmatrix}$$

$$\begin{aligned}\tilde{X}_1 &= \{x_{12} = x_{23} = x_{31} = x_{44} = 1\} \quad \text{avec} \quad (\tilde{Z}_1, \tilde{Z}_2) = (6, 24) \\ \tilde{X}_2 &= \{x_{14} = x_{21} = x_{33} = x_{42} = 1\} \quad \text{avec} \quad (\hat{Z}_1, \tilde{Z}_2) = (22, 7)\end{aligned}$$

Ce sont les deux solutions efficaces supportées initiales, on déduit les poids de la fonction d'agrégation :

$$\lambda_1^{(t)} = 24 - 7 = 17 \quad \text{et} \quad \lambda_2^{(t)} = 22 - 6 = 16$$

et la matrice :

$$C^{(t)} = 17c_{ij}^{(1)} + 16c_{ij}^{(2)}$$

$$C^{(t)} = \begin{vmatrix} 133 & 113 & 132 & 151 \\ 118 & 82 & 162 & 150 \\ 114 & 168 & 100 & 116 \\ 115 & 117 & 167 & 97 \end{vmatrix}$$

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(t)} = \begin{vmatrix} [0] & 0 & 13 & 32 \\ 16 & [0] & 74 & 62 \\ 0 & 74 & [0] & 16 \\ 4 & 26 & 70 & [0] \end{vmatrix}$$

La solution qui correspond au tableau $\bar{C}^{(t)}$ est :

$$X_3 = \{x_{11} = x_{22} = x_{33} = x_{44} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (12, 13)$$

On a $\tilde{X}_1 \neq X_3$ et $\tilde{X}_2 \neq X_3$, donc $X_3 \in SE(P)$.

Le processus de recherche des solutions supportées est poursuivie entre d'une part \tilde{X}_1 et X_3 et d'autre part X_3 et \tilde{X}_2 .

La recherche dans l'intervalle (\tilde{X}_1, X_3)

$$\lambda_1^{(t)} = 24 - 13 = 11 \quad \text{et} \quad \lambda_2^{(t)} = 12 - 6 = 6$$

et la matrice

$$C^{(t)} = 11c_{ij}^{(1)} + 6c_{ij}^{(2)}$$

$$C^{(t)} = \begin{vmatrix} 73 & 47 & 68 & 89 \\ 72 & 40 & 70 & 84 \\ 52 & 100 & 56 & 62 \\ 57 & 67 & 95 & 41 \end{vmatrix}$$

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(t)} = \begin{vmatrix} 9 & 0 & [0] & 25 \\ 15 & [0] & 9 & 27 \\ 0 & 74 & [0] & 16 \\ 16 & 43 & 50 & [0] \end{vmatrix}$$

La solution correspond est :

$$X_4 = \{x_{13} = x_{22} = x_{31} = x_{44} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (9, 17)$$

On a $X_4 \neq \tilde{X}_1$ et $X_4 \neq X_3$, donc $X_4 \in SE(P)$.

La recherche des solutions dans les intervalles (\tilde{X}_1, X_4) et (X_4, X_3) est poursuivie.

La recherche dans l'intervalle (\tilde{X}_1, X_4)

$$\lambda_1^{(t)} = 24 - 17 = 7 \quad \text{et} \quad \lambda_2^{(t)} = 9 - 6 = 3$$

et la matrice

$$C^{(t)} = 7c_{ij}^{(1)} + 3c_{ij}^{(2)}$$

$$C^{(t)} = \begin{vmatrix} 44 & 25 & 40 & 55 \\ 45 & 23 & 38 & 51 \\ 29 & 62 & 34 & 37 \\ 33 & 41 & 58 & 22 \end{vmatrix}$$

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(t)} = \begin{vmatrix} 9 & [0] & [0] & 20 \\ 12 & [0] & [0] & 18 \\ [0] & 43 & 0 & 8 \\ 11 & 29 & 31 & [0] \end{vmatrix}$$

$SE(P^{(t)}) = \{\tilde{X}_1, X_4\}$, il n'existe pas des solutions efficaces supportées entre \tilde{X}_1 et X_4 .

La recherche dans l'intervalle (X_4, X_3)

$$\lambda_1^{(t)} = 17 - 13 = 4 \quad \text{et} \quad \lambda_2^{(t)} = 12 - 9 = 3$$

et la matrice :

$$C^{(t)} = 4c_{ij}^{(1)} + 3c_{ij}^{(2)}$$

$$C^{(t)} = \begin{vmatrix} 29 & 22 & 28 & 34 \\ 27 & 17 & 32 & 33 \\ 23 & 38 & 22 & 25 \\ 24 & 26 & 37 & 19 \end{vmatrix}$$

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(t)} = \begin{vmatrix} [0] & 0 & [0] & 6 \\ 3 & [0] & 9 & 10 \\ [0] & 22 & [0] & 3 \\ 4 & 13 & 18 & [0] \end{vmatrix}$$

$SE(P^{(t)}) = \{X_4, X_3\}$, il n'existe pas des solutions efficaces supportées entre X_4 et X_3 .

La recherche dans l'intervalle (X_3, \tilde{X}_2)

$$\lambda_1^{(t)} = 22 - 12 = 10 \quad \text{et} \quad \lambda_2^{(t)} = 13 - 7 = 6$$

et la matrice

$$C^{(t)} = 10c_{ij}^{(1)} + 6c_{ij}^{(2)}$$

$$C^{(t)} = \begin{vmatrix} 60 & 66 & 64 & 62 \\ 46 & 42 & 92 & 66 \\ 62 & 68 & 44 & 54 \\ 58 & 50 & 72 & 56 \end{vmatrix}$$

Par application de la méthode hongroise le tableau optimal se présente :

$$\bar{C}^{(t)} = \begin{vmatrix} [0] & 10 & 4 & [0] \\ [0] & [0] & 46 & 18 \\ 18 & 28 & [0] & 8 \\ 4 & [0] & 18 & [0] \end{vmatrix}$$

$SE(P^{(t)}) = \{X_3, \tilde{X}_2\}$, il n'existe pas des solutions efficaces supportées entre \tilde{X}_2 et X_3 .

A la fin l'ensemble de solutions efficaces supportées est $SE(P) = \{\tilde{X}_1, X_3, X_4, \tilde{X}_2\}$.

B) Détermination de $E(P)|SE(P)$

Dans la Phase (2) on trouve trois triangles défini par quatres solutions efficaces supportées.

1) Dans le triangle $\tilde{X}_1 X_4$

On part du tableau optimal suivant :

$$\bar{C}^{(t)} = \begin{vmatrix} 9 & [0] & [0] & 20 \\ 12 & [0] & [0] & 18 \\ [0] & 43 & [0] & 8 \\ 11 & 29 & 31 & [0] \end{vmatrix} \quad \text{avec} \quad \lambda_1^{(t)} = 7 \quad \text{et} \quad \lambda_2^{(t)} = 3$$

$L = \{(3, 4), (1, 1), (4, 1), (2, 1), (2, 4), (1, 4), (4, 2), (4, 3), (3, 2)\}$

Le test (3.2.6) permet d'éliminer les affectations (2,4), (1,4), (4,2), (4,3) et (3,2) car pour chacune d'elles, la borne inférieure de l'augmentation de la valeur de \tilde{Z}^t est supérieure ou égale à $\lambda_1^{(t)} \cdot \lambda_2^{(t)} = 21$; à titre d'exemple, pour (2,4) cette borne l_t vaut $18 + 11 = 29$.(tableau $\bar{C}^{(t)}$)

Le test (3.2.9) permet d'éliminer les quatres affectations restantes de L car pour chacune d'elles, la borne inférieure de l'augmentation de la valeur de \tilde{Z}_1 est supérieure ou égale à $\lambda_2^{(t)} = 3$; à titre d'exemple, pour (3,4) cette borne vaut $2 + 2 = 4$ (tableau $\bar{C}^{(1)}$)

On en déduit donc qu'il n'y a pas de solutions efficaces non supportées dans le triangle associé à $\tilde{X}_1 X_3$.

2) Dans le triangle $X_4 X_3$

On part du tableau optimal

$$\bar{C}^{(t)} = \begin{vmatrix} [0] & 0 & [0] & 6 \\ 3 & [0] & 9 & 10 \\ [0] & 22 & [0] & 3 \\ 4 & 13 & 18 & [0] \end{vmatrix} \quad \text{avec} \quad \lambda_1^{(t)} = 4 \quad \text{et} \quad \lambda_2^{(t)} = 3$$

$L = \{(2, 1), (3, 4), (4, 1), (1, 4), (2, 3), (2, 4), (4, 2), (4, 3), (3, 2)\}$

Le test (3.2.6) permet d'éliminer les affectations (2,4), (4,2), (4,3) et (3,2), car pour chacune d'elles, la borne inférieure de l'augmentation de la valeur de $\tilde{Z}^{(t)}$ est supérieur ou égale à $\lambda_1^{(t)} \cdot \lambda_2^{(t)} = 12$; à titre d'exemple, pour (2,4) cette borne vaut $10 + 4 = 14$ (tableau $\bar{C}^{(t)}$).

Une matrice des coûts réduits du critère Z_1 pour la solution X_3 vaut :

$$\bar{C}^{(t)} = \begin{vmatrix} 1 & -3 & [0] & 4 \\ 3 & [0] & 0 & 4 \\ [0] & 6 & 2 & 2 \\ 4 & 4 & 6 & [0] \end{vmatrix}$$

Le test (3.2.9) permet d'éliminer les affectations (2,1) et (1,4); à titre d'exemple, pour l'affectation (2,1), la borne inférieure de l'augmentation de la valeur de Z_1 vaut : $4 + \max(-3 + 2 + 0, -3 + 0 + 0) = 3$ égale à $\lambda_2^{(t)}$.

Aussi les trois affectations restantes sont (3,4), (4,1) et (2,3) doivent être examiner.

Les trois affectations restantes sont (3,4), (4,1) et (2,3), elles apparaissent dans la solution X_5 qui empêche donc de les éliminer sur base de l'augmentation de Z_1 .

L'imposition dans le problème $P^{(t)}$ des deux premières affectations engendre la même solution

$$X_5 = \{x_{13} = x_{22} = x_{34} = x_{41} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (13, 14)$$

Cette solution est à l'extérieur du triangle (l'augmentation de Z_1 est supérieure à $a_2^{(t)}$).

L'imposition de l'affectation (2,3) dans $P^{(t)}$ donne la solution

$$\tilde{X}_1 = \{x_{12} = x_{23} = x_{31} = x_{44} = 1\}$$

Il n'y a pas de solutions non supportées dans le triangle engendré par X_4 et X_3 .

3) Dans le triangle $X_3\tilde{X}_2$

On part du tableau optimal suivant :

$$\bar{C}^{(t)} = \begin{vmatrix} [0] & 10 & 4 & [0] \\ [0] & [0] & 46 & 18 \\ [0] & 22 & [0] & 3 \\ 4 & [0] & 18 & [0] \end{vmatrix} \quad \text{avec} \quad \lambda_1^{(t)} = 6 \quad \text{et} \quad \lambda_2^{(t)} = 10$$

$L = \{(1, 3), (4, 1), (3, 4), (1, 2), (2, 4), (3, 1), (4, 3), (3, 2), (2, 3)\}$ Le test (3.2.6) ne permet d'éliminer aucune affectation de L.

Le test (3.2.10) permet d'éliminer les paires (1,2), (3,1) et (2,3) de la liste L car pour chacune d'elles la valeur de \tilde{Z}_2 est supérieur à $a_1 = 6$; à titre d'exemple, pour (1,2) la borne l_2 vaut $4 + 1 + 1 = 6 \geq 6$ (tableau $\bar{C}^{(2)}$)

L'affectation (4,1) est imposée en premier lieu car elle correspond à la plus petite valeur de l_t ($l_t = 4$); la réoptimisation de $P^{(t)}$ fournit la solution

$$X_6 = \{x_{14} = x_{22} = x_{33} = x_{41} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (16, 11)$$

C'est une solution efficace non supportée car les valeurs des bornes inférieurs l_t de l'augmentation de $Z^{(t)}$ correspondant à l'imposition des autres affectations de L sont toutes supérieurs $Z^u - Z^t = 4$; à titre d'exemple, pour (1,3), $l_t = 4 + 8 = 12$.

Le test (3.2.6) peut dès lors être actualisé.

En imposant (1,2), l'affectation de L correspondant la plus petite valeur de l_t ($l_t = 10$), on obtient la solution

$$X_7 = \{x_{12} = x_{21} = x_{33} = x_{44} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (12, 14)$$

Cette solution est une solution dominée par X_3 .

En imposant (1,3) ou (3,4) correspondant la valeur $l_t = 12$, on obtient la solution

$$X_8 = \{x_{13} = x_{21} = x_{34} = x_{42} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (19, 10)$$

c'est une solution efficace non-supportée, d'une part elle n'est dominé par aucune solution efficace déjà trouvée et d'autre part elle correspond à une augmentation de 12 de $Z^{(t)}$, alors que les bornes l_t des trois affectations restantes de L sont toutes supérieures à 12 :

$$\begin{cases} l_t = 18 & \text{pour l'affectation (2, 4)} \\ l_t = 26 & \text{pour l'affectation (4, 3)} \\ l_t = 32 & \text{pour l'affectation (3, 2)} \end{cases} \quad (3.2.15)$$

L'actualisation du test 1 fournit un nouveau second membre de (1) égal à $\max(6.4; 7.1 + 3.3; 3.10) = 30$, ce qui permet d'éliminer l'affectation (3,2) de L.

L'imposition de (2,4) fournit la solution

$$X_9 = \{x_{11} = x_{24} = x_{33} = x_{42} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (20, 10)$$

qui est une solution dominée par X_8 ; tandis que l'imposition de (4, 3) fournit la solution

$$X_{10} = \{x_{11} = x_{22} = x_{34} = x_{43} = 1\} \quad \text{avec} \quad (Z_1, Z_2) = (18, 12)$$

qui est une solution dominée par X_6 .

En conclusion, les six solutions efficaces sont les solutions :

$$\begin{cases} \text{supportées} \{ \tilde{X}_1, X_4, X_3, \tilde{X}_2 \} \\ \text{non supportées} \{ X_6, X_8 \} \end{cases} \quad (3.2.16)$$

3.3 La méthode de résolution proposée pour le problème bi-KP

3.3.1 Formulation mathématique du problème bi-KP

Modélisation

Le problème de sac à dos bi-objectif (bi-KP) :

$$(bi - KP) \begin{cases} \text{"max"} Z_k(X) = \sum_{i=1}^n C_i^k x_i & k = 1, 2 \\ \sum_{i=1}^n a_i x_i \leq b \\ x_i \in \{0, 1\} \end{cases} \quad (3.3.1)$$

3.3.2 La première méthode proposée

Notation

Cette méthode résout le problème de sac à dos bi-objectif, elle consiste à :

- 1) Chercher le nombre d'indices de variables qui peuvent entrer dans la base pour le premier et le deuxième objectif après avoir trier les a_i :

$$n_1 = \min \left\{ k \in \mathbb{N} \mid \sum_{i=1}^k a_i \leq b \right\}$$

$$n_2 = \max \left\{ k \in \mathbb{N} \mid \sum_{i=1}^k a_i \leq b \right\}$$

$$B = \{n \in \mathbb{N} \mid n_1 \leq n \leq n_2\}$$

- 2) Une fois B est déterminé.

Le nombre de possibilités d'avoir $X_{n_1} = 1$ est $C_N^{n_1}$.

Soit

$$L_{n_1} = \{(Z_1(X_i), Z_2(X_i)) \mid X_i \text{ contient } n_1 \text{ composantes égal à } 1\}$$

Déterminons l'ensemble de solutions dans l'espace des critères $(Z_1(X_i), Z_2(X_i))$

On poursuit la procédure jusqu'à L_{n_2} tel que :

$$L_{n_2} = \{(Z_1(X_i), Z_2(X_i)) \mid X_i \text{ contient } n_2 \text{ composantes égal à } 1\}$$

Ensuite, on aura un ensemble de solutions réalisables. Le décideur choisit celles qui seraient efficaces d'après sa structure de préférences (ses demandes).

N.B :

On doit vérifier à chaque fois que la solution trouvée est réalisable, et par la définition de la solution optimale au sens de Pareto, on trouve toutes les solutions efficaces.

Algorithme

Debut

Etape initiale :

Trier les a_i .

Etape 2 :

Chercher le nombre d'indices possibles de variables qui peuvent entrer dans la base.

$$B = \{i \in \mathbb{N} \mid n_1 \leq i \leq n_2\}$$

$$n_1 = \min \left\{ k \in \mathbb{N} \mid \sum_{i=1}^k a_i \leq b \right\}$$

$$n_2 = \max \left\{ k \in \mathbb{N} \mid \sum_{i=1}^k a_i \leq b \right\}$$

Etape 3 :

Les fonctions objectifs associées $L_i = \{(Z_1(X_i), Z_2(X_i)) \mid X \text{ contient } i \text{ composantes égal à } 1\}$

Etape finale :

$$L = \bigcup_{n_1}^{n_2} L_i \xrightarrow{\text{Pareto}} L^*$$

Fin**Exemple**

Soit le problème de sac à dos suivant :

$$(bi - KP) \begin{cases} \max Z_1(X) = 11x_1 + 5x_2 + 7x_3 + 13x_4 + 3x_5 \\ \max Z_2(X) = 9x_1 + 2x_2 + 16x_3 + 5x_4 + 4x_5 \\ 4x_1 + 2x_2 + 8x_3 + 7x_4 + 5x_5 \leq 16 \\ x_i \in \{0, 1\} \quad i = \overline{1, 5} \end{cases} \quad (3.3.2)$$

Les étapes de la méthode :

$$B = \{1, 2, 3\}$$

On a C_5^1 possibilités d'avoir une variable (d'indice 1) entrante dans la base.

X_1 veut dire que $x_i = 0 \quad i = 2, \dots, 5$ (exemple $X_1 = (1, 0, 0, 0, 0)$)

ou X_2 ou X_3 ou X_4 ou X_5 .

L'ensemble de solutions d'indices 1 est $\{X_1, X_2, X_3, X_4, X_5 \quad tq \quad X_1 = X_2 = X_3 = X_4 = X_5 = 1\}$

Soit L_1 l'ensemble de solutions dans l'espace des critères :

$$L_1 = \{(11, 9), (5, 2), (7, 16), (13, 5), (3, 4)\}$$

On a C_5^2 possibilités d'avoir la variable entrante d'indice 2.

X_{12} veut dire que $x_i = 0 \quad i = 3, \dots, 5$ (exemple $X_{12} = (1, 1, 0, 0, 0)$)

L'ensemble de solutions d'indices 2 est $\{X_{12}, X_{13}, X_{14}, X_{15}, X_{23}, X_{24}, X_{25}, X_{34}, X_{35}, X_{45}\} \quad tq$

$$X_{12} = X_{23} = X_{34} = X_{45} = \dots = X_{15} = 1$$

Soit L_2 l'ensemble de solutions dans l'espace des critères :

$$L_2 = \{(16, 11), (18, 25), (24, 14), (14, 13), (12, 18), (18, 7), (8, 6), (20, 21), (10, 20), (16, 9)\}$$

On a C_5^3 possibilités d'avoir la variable entrante d'indice 3.

X_{123} veut dire que $x_i = 0 \quad i = 4, 5$ (exemple $X_{123} = (1, 1, 1, 0, 0)$)

L'ensemble de solutions d'indices 3 est $\{X_{123}, X_{124}, X_{125}, X_{134}, X_{135}, X_{145}, X_{234}, X_{235}, X_{245}, X_{345}\}$

$$tq \quad X_{123} = X_{234} = \dots = X_{145} = 1$$

On remarque que X_{135} et X_{134} et X_{234} et X_{345} sont inadmissibles.

Soit L_3 l'ensemble de solutions dans l'espace des critères :

$$L_3 = \{(23, 27), (29, 16), (19, 15), (27, 18), (15, 22), (21, 11)\}$$

L'ensemble de solutions globale dans l'espace des critères :

$L_1 \cup L_2 \cup L_3$, ensuite nous détectons par la définition de la solution optimale au sens de Pareto

l'ensemble de solutions efficaces qui est :

$$\{(29, 16), (27, 18), (23, 27)\}.$$

3.3.3 La deuxième méthode proposée

on propose une autre méthode inspirée de la deuxième phase de la méthode en deux phases ainsi que cette dernière méthode :

La méthode est en deux phases, elle consiste à :

- 1) Déterminer les solutions efficaces extrêmes en maximisant (ou en minimisant) chaque objectif à part, en maximisant (ou en minimisant) le problème paramétrique (agrégation) :
- 2) A partir des solutions supportées, déterminer le reste des solutions supportées et les solutions non-supportées, en imposant aux composantes qui sont nulles d'être non nulles.

Reprenons le même exemple.

Exemple

Soit le problème de sac à dos suivant :

$$(bi - KP) \begin{cases} \max Z_1(X) = 11x_1 + 5x_2 + 7x_3 + 13x_4 + 3x_5 \\ \max Z_2(X) = 9x_1 + 2x_2 + 16x_3 + 5x_4 + 4x_5 \\ s|c \quad 4x_1 + 2x_2 + 8x_3 + 7x_4 + 5x_5 \leq 16 \\ x_i \in \{0, 1\} \quad i = \overline{1, 5} \end{cases} \quad (3.3.3)$$

Détermination des solutions efficaces

Par la méthode de décomposition⁶ ou la méthode séparation et évaluation, on trouve les solutions supportées extrêmes :

$$\begin{aligned} \tilde{X}_1 &= (1, 1, 0, 1, 0) \quad \text{avec} \quad (\tilde{Z}_1, \hat{Z}_2) = (29, 16) \\ \tilde{X}_2 &= (1, 1, 1, 0, 0) \quad \text{avec} \quad (\tilde{Z}_1, \hat{Z}_2) = (23, 27) \end{aligned}$$

1) On commence par la solution optimale du 1^{er} objectif $\tilde{X}_1 = (1, 1, 0, 1, 0)$.

On remarque que les composantes x_3 et x_5 de \tilde{X}_1 sont égales à 0.

Imposant pour $x_3 = 1$, on aura les possibilités suivantes :

$$X_{123} = (1, 1, 1, 0, 0), X_{134} = (1, 0, 1, 1, 0), X_{135} = (1, 0, 1, 0, 1), X_{234} = (0, 1, 1, 1, 0); \\ X_{235} = (0, 1, 1, 0, 1), X_{345} = (0, 0, 1, 1, 1).$$

Les solutions $X_{134} = (1, 0, 1, 1, 0)$, $X_{135} = (1, 0, 1, 0, 1)$, $X_{234} = (0, 1, 1, 1, 0)$, $X_{345} = (0, 0, 1, 1, 1)$ sont irréalisables.

Les solutions restantes sont $X_{123} = (1, 1, 1, 0, 0)$, $X_{235} = (0, 1, 1, 0, 1)$.

On a

$$\begin{aligned} X_{123} &= (1, 1, 1, 0, 0) \quad \text{avec} \quad (Z_1, Z_2) = (23, 27) \\ X_{235} &= (0, 1, 1, 0, 1) \quad \text{avec} \quad (Z_1, Z_2) = (15, 22) \end{aligned}$$

Imposant pour $x_5 = 1$, on aura les possibilités suivantes :

$$X_{125} = (1, 1, 0, 0, 1), X_{135} = (1, 0, 1, 0, 1), X_{145} = (1, 0, 0, 1, 1), X_{235} = (0, 1, 1, 0, 1), \\ X_{245} = (0, 1, 0, 1, 1), X_{345} = (0, 0, 1, 1, 1).$$

Les solutions $X_{135} = (1, 0, 1, 0, 1)$, $X_{435} = (0, 0, 1, 1, 1)$ sont irréalisables.

Les solutions restantes sont $X_{125} = (1, 1, 0, 0, 1)$, $X_{235} = (0, 1, 1, 0, 1)$, $X_{245} = (0, 1, 0, 1, 1)$, $X_{154} = (1, 0, 0, 1, 1)$.

On a

$$\begin{aligned} X_{145} &= (1, 0, 0, 1, 1) \quad \text{avec} \quad (Z_1, Z_2) = (27, 18) \\ X_{254} &= (0, 1, 0, 1, 1) \quad \text{avec} \quad (Z_1, Z_2) = (21, 11) \end{aligned}$$

⁶Voir l'exemple de la section 1.6.

$$X_{125} = (1, 1, 0, 0, 1) \quad \text{avec} \quad (Z_1, Z_2) = (19, 15)$$

$$X_{235} = (0, 1, 1, 0, 1) \quad \text{avec} \quad (Z_1, Z_2) = (15, 22)$$

2) On termine par la solution du 2ème objectif $\tilde{X}_2 = (1, 1, 1, 0, 0)$.

On remarque que les composantes x_4 et x_5 de \tilde{X}_2 sont égales à 0.

Pour le cas où $x_5 = 0$, c'est déjà étudié.

Imposant pour $x_4 = 1$, on aura les possibilités suivantes :

$$X_{124} = (1, 1, 0, 1, 0), X_{134} = (1, 0, 1, 1, 0), X_{145} = (1, 0, 0, 1, 1), X_{234} = (0, 1, 1, 1, 0),$$

$$X_{245} = (0, 1, 0, 1, 1), X_{345} = (0, 0, 1, 1, 1).$$

Les solutions $X_{134} = (1, 0, 1, 1, 0)$, $X_{234} = (0, 1, 1, 1, 0)$, $X_{345} = (0, 0, 1, 1, 1)$ sont irréalisables.

Les solutions restantes sont $X_{124} = (1, 1, 0, 1, 0)$, $X_{145} = (1, 0, 0, 1, 1)$, $X_{245} = (0, 1, 0, 1, 1)$.

On a :

$$X_{124} = (1, 0, 0, 1, 1) \quad \text{avec} \quad (Z_1, Z_2) = (29, 16)$$

$$X_{145} = (1, 0, 0, 1, 1) \quad \text{avec} \quad (Z_1, Z_2) = (27, 18)$$

$$X_{245} = (1, 0, 0, 1, 1) \quad \text{avec} \quad (Z_1, Z_2) = (21, 11)$$

Toutes les solutions possibles ont été détectés, on les rassemble dans un ensemble, notons K :
 $K = \{(23, 27), (15, 22), (27, 18), (21, 11), (19, 15), (15, 22), (29, 16)\}$.

Finalement, l'ensemble de solutions efficaces est :

$$\{(23, 27), (27, 18), (29, 16)\}$$

3.3.4 Conclusion

Nous avons proposé deux méthodes de résolution pour le problème de Sac à dos bi-objectif, la première consiste à chercher toutes les solutions réalisables par une énumération des indices des variables qui peuvent entrer dans la base, ensuite à l'aide de la définition de la solution optimale au sens de Pareto on détecte toutes les solutions efficaces.

La deuxième consiste à chercher l'ensemble de toutes les solutions efficaces en deux phases, la première phase détermine les deux solutions extrêmes, tandis que la deuxième détermine toutes les solutions réalisables restantes en imposant aux variables hors base, d'être dans la base. Enfin, on détecte les solutions efficaces. L'avantage des deux méthodes est qu'elles sont faciles à comprendre et moins difficiles à mettre en oeuvre par rapport à la méthode en deux phases de Ulungu et Teghem. Leur inconvénient est l'inconvénient de tous les problèmes combinatoires (explosion combinatoire à partir d'une certaine taille), mais elles restent des méthodes différentes de celles qui existent déjà.

Chapitre 4

Conclusion générale et perspectives

Les problèmes d'optimisation multi-critères occupent une place éminente dans la recherche opérationnelle, et correspondent à des situations de décision très difficiles. L'objectif principal de notre mémoire a été de faire une étude sur quelques problèmes d'optimisation multi-critères en mettant le point sur le problème de sac à dos et le problème d'affectation bi-objectifs. Le travail présenté est basé sur deux principaux objectifs :

Au cours de cette étude, on s'est croisé par une méthode dite méthode de décomposition développée pour les problèmes d'optimisation à grande taille dont la matrice des contraintes est creuse, on n'a pas pu s'en passer sans réaliser un premier objectif : une adaptation de cette méthode de résolution au problème de sac à dos unicritère¹ (ce cas reste une source d'inspiration pour le développement des méthodes de résolution des problèmes multi-objectifs) qui a fait l'objet de nombreuses études, parmi ces études, la méthode séparation et évaluation (Branch and Bound) et la programmation dynamique. Nous avons essayé de faire entrer quelques modifications au problème de sac à dos uni-critère sans perdre aucun point de l'ensemble de solutions. Nous avons remarqué que l'entrée d'une seule variable dans la base ne modifie pas celle qui entrera prochainement dans la base, d'où vient l'idée de faire entrer simultanément des variables dans la base. Cette démarche a fait un deuxième point dans cette adaptation. Cette tâche a été accomplie avec deux exemples illustratifs.

Nous avons constaté que le problème de sac à dos peut être résolu efficacement par cette méthode sous certaines conditions².

Enfin une suggestion pour faire entrer les coupes de Gomory simultanément dans la base au cas d'une solution non entière.

Notre deuxième objectif (intérêt) s'est ensuite porté sur les problèmes multi-objectifs, nous avons présenté un panorama des méthodes de résolution des problèmes multi-objectifs en nombres continus et en nombres entiers en donnant quelques critiques pour certaines méthodes établies dans le cas continu.

Notre étude est basée sur le problème de sac à dos et le problème d'affectation bi-objectifs. En premier lieu, nous avons décrit la méthode en deux phases de Ulungu et Teghem pour le problème d'affectation bi-objectifs. En second lieu, deux méthodes de résolution pour le problème de sac à dos bi-objectifs ont été proposées faciles à comprendre et moins difficiles à mettre en oeuvre, la première basée sur une énumération, tandis que la deuxième inspirée de celle de

¹Le problème de dégénérescence a lieu dans ce problème

²Les conditions sont décrites dans le chapitre (1)

Ulungu et Teghem. A la fin de chaque méthodes un exemple didactique est présenté.

Ce travail ouvre des perspectives de recherche :

Les résultats obtenus de ces méthodes sont encourageants malgré qu'on prévoit un temps d'exécution assez grand pour les problèmes à grande taille.

- C'est pourquoi notre premier effort devra se porter sur ce point que nous espérons programmer par des méthodes qui soit performantes, par une meilleure gestion des temps de calculs.
- Par la suite, nous essayerons de s'intéresser à utiliser le Logiciel Cplex pour la résolution des problèmes multi-objectifs.
- Une étude comparative sur la méthode proposée et celle développée par Ulungu et Teghem peut être considérée.

Résoudre exactement des problèmes d'optimisation combinatoire NP-difficile est très ardu, en effet pour de tels problèmes, les méthodes exactes exigent un temps de résolution qui croit rapidement, et un effort calculatoire qui croit exponentiellement avec la taille des instances du problème (explosion combinatoire). Et rapidement, les heuristiques, et les métaheuristiques deviennent l'unique moyen d'obtenir une bonne solution en un temps raisonnable. Malgré que pour les problèmes d'optimisation combinatoire les plus difficiles, les méthodes exactes demeurent très efficaces pour de petites tailles d'instances, la solution étant trouvée en un temps très court.

- En conclusion, nous avons aussi constaté que pour les problèmes d'optimisation combinatoire bi-objectifs l'ensemble des solutions optimales au sens de Pareto est divisé en deux parties : l'ensemble de solutions supportées et l'ensemble de solutions non supportées. La première partie est facile à avoir tandis que la deuxième partie est très difficile à avoir. Nous proposerons pour l'avenir de résoudre les problèmes bi-objectifs par une combinaison entre la méthode exacte et la méthode métaheuristique tel que la première partie peut être résolu par la méthode exacte et la deuxième partie peut être résolu par la méthode métaheuristique en prenant compte l'ensemble de solutions de la première partie comme un voisinage. Le temps de calcul diminuera certainement par cette combinaison de méthodes.
- Adapter l'optimisation combinatoire multi-objectif pour les réseaux de Neurone.

Bibliographie

- [1] J.Acher, J.Gardelle : *Programmation linéaire*, Bordas, Paris, 1978.
- [2] Antony Przybylski, Xavier Gandibleux, Matthias Ehrgott : *Une Méthode de Résolution Exacte pour le Problème d'Affectation Bi-objectifs*(2006).
- [3] Bouchez François-Xavier, D.Tuytens : *Résolution de problèmes d'optimisation combinatoire multi-objectif à l'aide du logiciel d'optimisation CPLEX*(2005-2006).
- [4] D. Chaabane : *Optimisation Multicritère en nombres entiers. Thèse de doctorat, U.S.T.H.B, Faculté de Mathématique*(2005).
- [5] D. Chaabane : *Contribution à l'Optimisation Multicritère en Variables Discrètes : Dissertation soumise à la Faculté Polytechnique de Mons en vue l'obtention du titre de Docteur en Sciences Appliquées*(2006-2007).
- [6] Diagne Coumbé Kébé : *Rapport technique de Doctorat Troisième Cycle ; Université Cheikh Anta Diop, Faculté des sciences et Techniques, Département de Mathématiques et Informatique, Sénégal*,2004.
- [7] F.Droesbeke, M.Halin, C.Lefevre : *Programmation Linéaire*, Edition Marketing,1986.
- [8] M.Ehrgott : *Multicriteria Optimisation* 2nd edition,2005.
- [9] Éric D. Taillard : *Principes d'implémentation des métaheuristiques*(Chapitre 2).
- [10] Fabien Degotin et Xavier Gandibleux : *Un retour d'expériences sur la résolution de problèmes combinatoires bi-objectifs* ; Journée Programmation Mathématiques Multiobjectifs,2002.
- [11] Mady Ouedraogo : *Programmation linéaire : Implémentation d'une nouvelle méthode de résolution des problèmes de grande dimension à variables bornées intégrant l'entrée simultanée de plusieurs variables dans la base ;(Mémoire de DEA)*,2004-2005
- [12] Michel Salomon : *Techniques d'optimisation "Les métaheuristiques"*,2004-2005.
- [13] Rita Malhotra, H.L.Bhatia, M.C.Puri : *Bi-criteria Assignment Problem* ;(Vol.19, No.2,1982).
- [14] E.L. Ulungu : *Optimisation Combinatoire multicritère : Détermination de l'ensemble des solutions efficaces et méthodes interactives* ; thèse de doctorat, Université de Mons-Hainault, Faculté des Sciences,1993.
- [15] Ulungu E.L. et Teghem J : *The two phases method : An efficient procedure to solve bi-objective combinatorial optimisation problems* : Journal Foundations of Computing and Decision Sciences,1995.
- [16] Michel Minoux et Michel Gondran : *Graphes et algorithmes*, Editions Eyrolles, 1985.
- [17] Michel Sakarovitch : *Optimisation Combinatoire (Méthodes mathématiques et algorithmiques)*,1984.