N^0 d'ordre : 22 /2008-M/MT

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

جامعة هواري بومدين
للعلوم و التكنولوجيا
U S T H B

Mémoire

présenté pour l'obtention du diplôme de Magister

En : MATHEMATIQUES

Spécialité : Recherche Opérationnelle (Mathématiques de Gestion)

Par

RAGGAS Nassima

Thème

OPTIMISATION MULTICRITÈRE APPLIQUÉE AU PROBLÈME D'AFFECTATION

Soutenu publiquement le 12/03/2008, devant le jury composé de :

Mr.	**A. BERRACHEDI**	Professeur	U.S.T.H.B	Président.
Mr.	**D. CHAABANE**	Maître de Conférences	U.S.T.H.B.	Directeur de mémoire.
Mr.	**M. MOULAÏ**	Maître de Conférences	U.S.T.H.B.	Examinateur.
Mr.	**M.E-A. CHERGUI**	Chargé de Recherche	U.S.T.H.B.	Examinateur.

Table des matières

Dédicases

★ *A mes très chers parents, pour les sacrifices qu'ils ont fait pour nous.*

★ *A la mémoire de mes grands parents Slimane et Djilali.*

★ *A la mémoire de ma grand mère Yasmine.*

★ *A la mémoire de mon oncle Mohamed, puisse Dieu le tout puissant leurs accorder sa Miséricorde et les accueillir dans son vaste paradis.*

Remerciements

★ Je remercie d'abord le BON DIEU le tout puissant de m'avoir accordé la santé; et le courage d'arriver au terme de ce travail.

★ Je tient à exprimer mes vifs remerciements à mon promoteur Monsieur CHAABANE Djamel qui a été pour moi un guide bien qualifié, ainsi que pour ses précieux conseils et ses orientations au long de ce travail de mémoire.

★ Je remercie également mon enseignant Monsieur le Professeur BERRACHEDI Abdelhafid qui m'a fait l'honneur d'accepter d'être président du jury de mon mémoire.

★ C'est un grand honneur pour moi que d'avoir Messieurs MOULAÏ Mustapha, CHERGUI Mohamed el amine comme membres de mon jury; Je les remercie ici très vivement.

★ Enfin j'adresse une pensée toute particulière à ma famille – mes chers parents, mes frères Réda, Sid Ali, Lamine et Salim, mes sœurs Amina et Imène, ma tente Saïda – pour leur présence et leurs encouragements. Merci à madame Nadia pour son aide précieuse et aux amis pour tous les bons moments passés ensemble. Merci à tous et à toutes qui ont contribué de près ou de loin à la réalisation de ce travail.

Introduction générale

L'objet de la recherche opérationnelle est l'amélioration du fonctionnement des entreprises et des organismes publics par l'application de l'approche scientifique. Reposant sur l'utilisation de méthodes scientifiques et de techniques spécialisées, elle permet d'obtenir une évaluation quantitative des politiques, stratégies et actions possibles dans le cours des opérations d'une organisation ou d'un système. Elle est apparue en tant que telle en Grande-Bretagne durant la seconde guerre mondiale, lorsqu'on décida d'employer des méthodes scientifiques pour étudier divers aspects des opérations militaires (d'ou son nom). Depuis lors, la recherche opérationnelle est devenue un élément important du processus de prise de décision dans de nombreux contextes commerciaux, industriels et gouvernementaux, car elle permet d'appréhender de façon systématique la complexité toujours grandissante des problèmes de gestion auxquels sont confrontés les décideurs.

De nombreux secteurs (télécommunications, transport, etc...) sont concernés par des problèmes complexes de grandes dimensions et multi-critères (qualité de service, etc...) mettant en jeu des coûts très importants et pour lesquels les décisions doivent être prises de façon optimale. Les problèmes d'optimisation rencontrès en pratique sont rarement mono-critère. Il y a généralement plusieurs critères contradictoires à satisfaire simultanément. L'optimisation multi-critères s'intéresse à la résolution de ce type de problèmes, elle possède ses racines au 19ième siècle dans les travaux en économie de Edgeworth et Pareto [6]. Elle a ainsi été initialement utilisée en économie et dans les sciences du management, puis graduellement dans les sciences pour l'ingénieur.

Dans les 30 dernières années, la plupart des travaux ont portés sur la programmation multi-objectifs linéaire en variables continues. Les raisons principales de cet intérêt sont d'une part le développement de la programmation linéaire mono-objectif en recherche opérationnelle, et la facilité relative de traiter de tels problèmes, et d'autre part l'abondance des cas pratiques qui peuvent être formulés sous forme linéaire. Ainsi, un certain nombre de logiciels ont vu le jour depuis le développement de la méthode du simplexe multi-objectifs (Zeleny, 1982 [22]). Le livre de Steuer [15] ainsi que celui de Ehrgott offre une solide introduction aux problèmes d'optimisation multicritère. Concernant l'optimisation multi-objectifs linéaire en variables discrètes et l'optimisation combinatoire multi-objectifs, peu de travaux ont été réalisés avant les années 80-90. Mais depuis, un fort intérêt a été montré pour ce genre de problèmes [20], [7].

Dans la littérature, une attention particulière a portés sur les problèmes à deux critères en utilisant les méthodes exactes telles que le "branch and bound" (Sen et al, 1988), l'algorithme A* (Stewart and White, 1991).

Dans le cadre de ce travail, nous nous somme intéressée à l'étude du problème multicritères d'affectation et spécialement à la détermination des solutions efficaces dans le cas de deux fonctions objectifs d'où l'appellation problème bicritère d'affectation (Bi-critaria Assignment Problem (BAP)). Notons que Charnes et al [20] ont été les premiers a portés attention sur l'extension multi-objectifs du problème d'affectation et depuis plusieurs méthodes ont été établie [18].

Ce mémoire comporte trois chapitres :

Le premier chapitre fournit les éléments de la recherche opérationnelle nécessaires à la compréhension de la suite du mémoire, tels que la notion de base, de dualité...etc. De plus, les caractéristiques principales des problèmes multi-objectifs et de leurs méthodes de résolution sont indiquées.

Le deuxième chapitre est une introduction aux problèmes de la programmation linéaire en nombres entiers. Nous présenterons la structure générale des problèmes (ILP) et (MOILP) et les principales méthodes de résolution existantes dans la littérature, en précisant où réside la difficulté de la résolution de ce type de problèmes.

Le troisième chapitre est consacré au problème bicritère d'affectation, nous formulerons le problème et nous exposerons quelques méthodes existantes dans la littérature notamment celles établie par Malhotra et al [9], et par Ulungu [19], nous présenterons également des exemples illustratifs pour chaque méthode.

Enfin, nous terminerons ce travail par une conclusion générale et perspectives.

Chapitre 1

Optimisation linéaire multi-objectifs

1.1 Introduction

La programmation linéaire, une des branches de la recherche opérationnelle, consiste à établir la théorie et les méthodes de résolution des problèmes d'optimisation sur des ensembles définies par des contraintes (égalités ou inégalités) qui consiste soit à minimiser les coûts ou les dégâts soit à maximiser les gains ou les bénéfices. Dans ce chapitre, nous rappelons les principaux résultats de la programmation linéaire, et les principales méthodes utilisées pour résoudre ces problèmes de manière exacte, ainsi que les notions de bases nécessaires pour l'étude de notre problème.

1.2 Programmation linéaire

1.2.1 Généralités

Notions d'algèbre linéaire

- Un ensemble de vecteurs $V = \{v_i/v_i \in \mathbb{R}^n,\ i = 1, 2, ..., n\}$ est linéairement indépendant si et seulement si $\sum_{i=1}^{k} \lambda_i v_i = 0 \in \mathbb{R}^n \Rightarrow \lambda_1 = \lambda_2 = ... = \lambda_n = 0 \in \mathbb{R}$. Sinon l'ensemble des vecteurs V est linéairement dépendant.

- La dimension de l'ensemble V est le nombre maximum de vecteurs linéairement indépendants.

- Une matrice est dite totalement unimodulaire si et seulement si, tous les déterminants de ses sous-matrices carrées soient égaux à -1, 0 ou 1.

Voisinage

Un voisinage V est une fonction $V : C \longrightarrow P(C)$ qui associe pour chaque $x \in C$ un sous-ensemble $V(x)$ de C des voisins de x

Ensembles convexes et points extrêmes

- Soient $v_1, v_2, ..., v_n$ des éléments dans un espace vectoriel. On appelle combinaison linéaire convexe de ces vecteurs le vecteur :

$V = \lambda_1 v_1, \lambda_2 v_2, ..., \lambda_n v_n$ où les $\lambda_i, (i = 1, 2, ..., n)$ sont des scalaires tels que : $\lambda_i \geq 0$ et $\sum_{i=1}^{n} \lambda_i = 1$.

- Un ensemble $S \subset \mathbb{R}^n$ est *convexe* si et seulement si pour tous x^1, $x^2 \in S$, le point $(\lambda x^1 + (1 - \lambda) x^2) \in S$ pour tout $\lambda \in [0, 1]$. En d'autres termes, un ensemble est dit convexe si pour toutes paires de points (x, y) de cet ensemble, le segment $[xy]$ qui les joints est entièrement inclu dans cet ensemble.

Soit $S \subset \mathbb{R}^n$, un point $\bar{x} \in S$ est un point extrême (sommet) de S si et seulement s'il n'existe pas deux points x^1, $x^2 \in S$, $x^1 \neq x^2$, tel que $\bar{x} = \lambda x^1 + (1 - \lambda) x^2$, $\lambda \in]0, 1[$. Ou encore, un point $\bar{x} \in S$ est défini comme un point extrême s'il ne peut pas être exprimé comme une combinaison linéaire convexe des points de S. Un ensemble convexe peut avoir un nombre fini de points extrêmes, en avoir un nombre infini, ou ne pas avoir de points extrêmes du tout.

- L'ensemble constitué par les combinaisons linéaires convexes d'un ensemble fini de points extrêmes de S est un polyèdre convexe. Un polyèdre convexe est un ensemble convexe borné. Un polyèdre convexe a un nombre fini de points extrêmes.

- Soient $x_1, x_2, ..., x_p$, p points de $\mathbb{R}^n$. On appelle *enveloppe convexe* de ces points l'ensemble $\left\{ x \in \mathbb{R}^n / \sum_{i=1}^{n} \lambda_i x_i, 0 \leq \lambda_i \leq 1 \forall i, \ \ \sum_{i=1}^{n} \lambda_i = 1 \right\}$.

Face

Définition 1.2.1 *Soit $S \subset \mathbb{R}^n$ convexe. Un sous ensemble F de S est dit une face de dimension p si la plus petite variété linéaire L vérifie $L \cap S = F$.*

Les Cônes

Définition 1.2.2 *Soit $v \in V \subset \mathbb{R}^n$, $V \neq \varnothing$, alors V est un cône si et seulement si $\alpha v \in V$ pour tout scalaire $\alpha \geq 0$.*

1.2.2 Formulation des problèmes de Programmation linéaire

On définit un problème linéaire en variables continues comme un problème d'optimisation d'une fonction linéaire (appelée "fonction objectif") en variables (dites "de décision") satisfaisant un ensemble d'équations et/ou d'inéquations linéaires (dites "contraintes").

D'une façon générale, un programme mathématique est un problème d'optimisation dans $\mathbb{R}^n$de la forme :

$$(P) \begin{cases} Minimiser \text{ (ou } Maximiser) \ f(x) \\ \text{sous les conditions :} \\ g_i(x) \leq 0, i = 1, 2, ..., n \\ x \in S \text{ et } S \subset \mathbb{R}^n \end{cases}$$

Tels que

- La fonction $f(x)$ est appelée fonction objectif ou critère.

- L'ensembles des conditions $g_i(x) \leq 0$, $x \in \mathbb{R}^n$ sont les *contraintes* (égalités ou inégalités) du problème.

- Le vecteur $x \in \mathbb{R}^n$ a pour composantes $(x_1, x_{2,\dots,}x_n)^T$ qui sont les inconnues du problème; appeler *variables de décision.*

- S et la région *réalisable* (possible) du problème.

Si les fonctions f et g_i sont linéaires en x, on obtient un problème de programmation linéaire (noté (PL)) généralement écrit sous la forme suivante :

$$(PL)\begin{cases} \min \;\; ou \;\; (\max) \, Z = cx \\ x \in S \end{cases}$$

avec $S = \{x \in \mathbb{R}^n : Ax \leq b, \; x \geq 0\}$; A (matrice des contraintes), c, x et b sont des matrices de dimensions respectives $(m \times n)$, $(1 \times n)$, $(n \times 1)$ et $(m \times 1)$; n est le nombre de variables, m est le nombre de contraintes du système, et Z est la fonction objectif.

1.2.3 Notions fondamentales [12]

Base, solution de base.

Définition 1.2.3 *Un point $x \in S$ est appelé solution réalisable ou admissible ou possible.*

- Toute combinaison linéaire convexe de deux solutions possibles quelconques est aussi une solution possible.

- L'ensemble des solutions possibles ou admissibles est un polyèdre convexe.

Définition 1.2.4 *Une solution de (PL) est appelée solution optimale.*

Optimum global On appelle optimum global (solution optimale) de (PL) une solution qui minimise Z sur l'ensemble de toute les solutions.

Optimum local On dit qu'un vecteur x^0 est un optimum local de (PL) si et seulement s'il existe un voisinage $v(x^0)$ de x^0 tel que x^0 soit un optimum global du problème

$$(PL)\begin{cases} \min Z = Cx \\ \quad s.c \\ x \in S \ \cap v(x^0) \\ S \subset \mathbb{R}^n \end{cases}$$

Définition 1.2.5 *On appelle base de (PL) un ensemble B de m indices pris dans $\{1, 2, ..., n\}$ tel que la sous- matrice A^B formée des m colonnes correspondantes à A soit carrée non singulière (inversible).*

À une base B on associée l'ensemble d'indices complémentaire $N = \{1, 2, ..., n\} \setminus B$. Après une permutation de colonnes on peut écrire :

$$A = \left(A^B A^N\right)$$

On désigne par $\mathcal{I}$ l'ensemble des indices de colonnes correspondantes à la base et $\mathcal{J}$ l'ensemble des indices de colonnes hors base.

Soit B une base de (PL) :

- A^B est dite matrice de base associe à B.

- Si $x \in \mathbb{R}^n$, les composantes x_i avec $i \in B$ sont alors dites de base et notées x_B, celles avec $i \notin B$ sont dites hors base et notées x_N, donc à tout élément $x \in \mathbb{R}^n$ on associée la partition (x_B, x_N).

Définition 1.2.6 *La solution* $\begin{cases} x_B = \left(A^B\right)^{-1} b \\ x_N = 0 \end{cases}$ *est appelée solution de base associée à B.*

- Une base est dite réalisable si la solution de base associée vérifie $x_B \geq 0$.

- Une solution de base associée à une base réalisable est dite solution de base réalisable.

Remarque 1.2.1 *Tout programme linéaire peut s'écrire $A^B x_B + A^N x_N = b$. Où A^B est la sous matrice des colonnes de base, A^N la sous matrice des colonnes hors base, x_B sont les variables de base, x_N les variables hors base.*

Remarque 1.2.2 *À une base correspond une et une seule solution de base; par contre, il peut exister plusieurs bases pour lesquelles une solution donnée est solution de base associée. On dit alors qu'il y a dégénérescence c'est à dire* $x_B = \left(A^B\right)^{-1} b$, *a des composantes nulles .*

1.2.4 Résolution d'un problème linéaire

Bien évidemment, une fois le problème modélisé, se pose la question de la détermination de sa (ou ses) solution(s) optimale(s) et donc de sa résolution. Nous parlerons de résolution exacte d'un problème lorsque l'algorithme utilisé permettra de trouver au moins une solution optimale et de démontrer son optimalité.

Pour les problèmes en variables continues, le nombre de solutions admissibles est infini lorsque leur domaine n'est pas vide. Le domaine des solutions admissibles (ou espace de recherche) d'un programme linéaire en variables continues est soit un polytope convexe (cas d'un domaine non borné), soit un polyèdre convexe (cas d'un domaine borné). De plus, si le domaine est borné et non vide, la solution optimale est un sommet du domaine (ou une face s'il y a plusieurs solutions optimales).

Ainsi, l'algorithme du simplexe permet d'obtenir la solution optimale d'un problème en parcourant la fermeture convexe de l'espace de recherche et ce en passant de sommet en sommet. Malgré une complexité théorique exponentielle, la méthode du simplexe reste largement utilisée dans la pratique.

1.2.5 Méthode du Simplexe

La méthode du simplexe a été développée par G.B.Dantzig (1947) (voir [3]). C'est une procédure itérative qui à travers des opérations répétées permet d'atteindre progressivement la solution optimale.

Forme standard d'un programme linéaire

On dit qu'un programme linéaire est mis sous forme standard si toutes les contraintes sont des égalités.

On peut toujours mettre un programme linéaire quelconque sous forme standard en introduisant des variables supplémentaires appelées variables d'écarts.

Forme canonique d'un programme linéaire

On dit qu'un programme linéaire est mis sous forme canonique relativement à la base des variables $(x_{i1}, x_{i2}, ..., x_{im})$ si :

1. Z est exprimé en fonction des variables hors base $(c_B = 0)$.

2. Les colonnes de la matrice des contraintes correspondantes aux variables de base $\left(A^B\right)$ forment une matrice unité à une permutation près.

Tableau simplexe

L'intérêt du tableau simplexe est de rassembler de façon condensée tous les éléments nécessaires au déroulement de l'algorithme du simplexe.

Considérons le programme linéaire (écrit sous forme standard) suivant :

$$(PL)\begin{cases} \min Z = c_1x_1 + c_2x_2 + ... + c_nx_n \\ s.c \\ a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = b_i, i = 1, 2, ..., m \\ (x_1, x_2, ..., x_n) \geq 0 \end{cases}$$

On peut le représenter par le tableau suivant, appelé tableau simplexe :

x_1	x_2	...	x_n	b
a_{11}	a_{12}		a_{1n}	b_1
a_{21}	a_{22}		a_{2n}	b_2
...	...	...	...	...
a_{m1}	a_{m2}	...	a_{mn}	b_m
c_1	c_2	...	c_n	$Z(x)$

Tableau simplexe

On obtient la forme canonique relative à la base des variables $(x_1, ..., x_m)$

$\begin{matrix} 1 & & 0 \\ & 1 & \\ & & \\ 0 & & 1 \end{matrix}$	$\widetilde{A^N} = \left(A^B\right)^{-1} A^N$	$\widetilde{b} = \left(A^B\right)^{-1} b$
$0 \quad 0 \qquad 0$	$\widetilde{c_N} = c_N - c_B B^{-1} N$	$Z(x) - c_B \left(A^B\right)^{-1} b$

Algorithme de simplexe.

Le principe de l'algorithme consiste à effectuer une série d'opération de pivotage sur la matrice des coefficients de manière à écrire le programme linéaire sous forme canonique par rapport aux bases courantes.

La mise en œuvre de la méthode simplexe peut être divisée en trois (3) étapes.

Étape1(Initialisation) Transformer le programme linéaire sous la forme standard en y introduisant les variables d'écarts, et en suite sous forme canonique par rapport à une base B de façon à avoir une solution de base admissible.

Étape2 Établir le premier tableau du simplexe.

Étape3 Procéder à une série d'itérations.

3.1 Déterminer la colonne pivot.

3.1.1 Si les différentes coefficients canonique $\widetilde{c_N} \geq 0$. Terminer, la solution de base admissible actuelle est optimale.

3.1.2 Choisir s tel que $(\widetilde{c_N})_s = \min\limits_{j \in J} \left\{ (\widetilde{c_N})_j \mid (\widetilde{c_N})_j < 0 \right\}$.

3.2 Déterminer la ligne pivot.

Soit $L = \left\{ l \middle| \left(\widetilde{A}\right)_l^s > 0 \right\}$.

3.2.1 Si $L = \varnothing$. Terminer, (PL) n'admet pas de solution optimale.

3.2.2 Sinon, soit $K = \left\{ k \left| \left(\widetilde{b}\right)_k / \left(\widetilde{A}\right)_k^s = \min\limits_{l \in L} \left[\left(\widetilde{b}\right)_l / \left(\widetilde{A}\right)_l^s \right] \right. \right\}$;choisir $r \in K$.

3.3 Établir le nouveau tableau simplexe.

3.3.1 Pivoter sur l'élément $\left(\widetilde{A}\right)_r^s$, et établir le nouveau tableau simplexe tel que :

l'ensemble des indices de la nouvelle base $J = J \cup \{s\} \setminus \{r\}$.

$a_{rs} = 1$; $a_{rj} = \frac{a_{rj}}{a_{rs}}$ avec $j \neq s$

$a_{ij} \ = a_{ij} \ - a_{is}\frac{a_{rj}}{a_{rs}}$ avec $j \neq s$, $i \neq r$

$\widetilde{b}_i := \widetilde{b}_i - a_{is}\left(\widetilde{b}_r / a_{rs}\right)$ $i \neq r$; $\widetilde{c_N} = \widetilde{c_N} - \left(\widetilde{c_N}\right)_s (a_{rj}/a_{rs})$ .Si la solution est non optimale, répéter l'étape 3. Sinon donner la solution optimale.

Pour résoudre un programme linéaire avec une fonction objectif à maximiser, il suffit de changer, dans l'algorithme :

- Trouver s tel que $\left(\widetilde{c_N}\right)_s = \min\limits_{j \in J} \left\{ \left(\widetilde{c_N}\right)_j \mid \left(\widetilde{c_N}\right)_j < 0 \right\}$

Par

- Trouver s tel que$\left(\widetilde{c_N}\right)_s = \max\limits_{j \in J} \left\{ \left(\widetilde{c_N}\right)_j \mid \left(\widetilde{c_N}\right)_j > 0 \right\}$

Et

- le critère d'arrêt : de $\widetilde{c_N} \geq 0$ en $\widetilde{c_N} \leq 0$.

Finitude de l'algorithme du simplexe.

Théorème 1.2.1 *Si à chaque base rencontrée dans la résolution de (PL) la solution de base est non dégénérée l'algorithme se termine en un nombre fini d'itérations.*

1.2.6 Notion de dualité

Les problèmes de la programmation linéaire existent toujours sous forme de paires, car il est possible de dériver d'un programme linéaire donné son programme dual, selon des relations bien définies. Le programme original est appelé programme primal (P) et le programme obtenu dual (D). Ainsi, est associé à chaque problème de minimisation, un problème de maximisation et inversement.

La notion de dualité est cruciale pour différentes raisons, dans certains cas, résoudre le programme dual s'avère être plus facile que de solutionner le programme primal. Cette alternative permet une économie appréciable de temps de calcul.

Considérons le problème linéaire suivant :

$$(P)\begin{cases} CX = Z\,(Min) \\ AX \leq b \\ x \geq 0 \end{cases}$$

Ecrit sous forme canonique

Définition 1.2.7 *On appelle dual de programme linéaire (P), le programme linéaire (D) écrit sous la forme suivante :* $(D)\begin{cases} Yb = W\,(Max) \\ YA \leq C \\ Y \leq 0 \end{cases}$

Le programme dual (D) est obtenu du programme primal (P) on opérant comme suit:

a). Il y'a autant de variables duales $Y = (y_1, y_2, ..., y_m)$ que de contraintes dans le programme primal.

b). Les coefficients du second membre $b = (b_1, b_2..., b_m)^T$ de (P) figurent dans la fonction objectif du (D) comme coefficients.

c). Les coefficients de la fonction objectif du (P) deviennent les coefficients du second membre du programme dual (D).

d). La matrice transposée des coefficients des contraintes de (P) devient la matrice des coefficients des contraintes de (D).

e). Lorsque, dans le programme primal la fonction objectif est de minimiser, dans le programme dual il s'agit de maximiser.

f). Enfin, pour déterminer le sens des contraintes ($\leq$, $\geq$, $=$), il faut se référer au diagramme de la dualité dans le tableau suivant :

Primal (dual)		Dual (primal)	
Fonction objectif (minimisation)		Fonction objectif (maximisation)	
	$\geq$		≥ 0
$i^{\text{ème}}$ *contrainte*	$=$	$i^{\text{ème}}$ *variable*	$quelconque(\pm 0)$
	$\leq$		≤ 0
	≥ 0		$\leq$
$j^{\text{ème}}$ *variable*	$quelconque(\pm 0)$	$j^{\text{ème}}$ *contrainte*	$=$
	≤ 0		$\geq$

Si (D) est le dual de (P). (P) sera dit primal de (D).

Relations entre les valeurs de fonctions objectifs des deux problèmes duaux

Soit (P) un programme linéaire écrit sous forme canonique et soit (D) son dual.

1. Si $(\overline{x}, \overline{y})$ constitue un couple de solutions réalisables de (P, D) alors $C\overline{x} \leq \overline{y}b.$
2. Si $(\overline{x}, \overline{y})$ constitue un couple de solutions réalisables de (P, D) et si de plus $C\overline{x} = \overline{y}b$ alors $(\overline{x}, \overline{y})$ constitue un couple de solutions optimales pour (P, D).

1.3 Programmation Linéaire multi-objectifs

Les problèmes d'optimisation sont généralement optimisés en ne considérant qu'un seul objectif, alors que plusieurs critères contradictoires existent. L'optimisation multi-objectifs s'intéresse à la résolution de ce type de problèmes. Elle cherche à optimiser (maximiser ou minimiser) plusieurs composants d'un vecteur de fonctions coûts. Contrairement à l'optimisation unicritère, la solution d'un problème multicritère n'est pas une solution unique, mais un ensemble de solutions, connu comme l'ensemble des solutions Pareto optimales et c'est au décideur de choisir une parmi elles, selon l'ordre de préférences des critères, et celle qui convient le mieux à son problème. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un composant du vecteur sans dégradation d'au moins un autre composant du vecteur.

1.3.1 Problèmes de programmation mathématique multi-objectifs

Nous définissons un problème multi-objectifs (noté par (MO)) comme un problème de décision qui consiste à optimiser (maximiser ou minimiser) simultanément p ($p \geq 2$) fonctions objectifs notées f_k , $k = 1, ..., p$, sur un ensemble d'actions S.

Ce problème peut être formulé mathématiquement comme suit :

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

où $S = \{x \in \mathbb{R}^n \, | g_j(x) \leq 0 \ , j = 1, 2, ..., m\}$ et $\{f_k\}_{k=1,2,...,p}$ et g_j sont des fonctions à valeurs réelles.

Le symbole “ ” signifie qu'il n'est généralement pas possible de trouver dans S une action qui optimise simultanément les p objectifs.

Définition 1.3.1 *L'espace $\mathbb{R}^n$ dans lequel se situe l'ensemble des actions S $(S \subset \mathbb{R}^n)$ est appelé espace des décisions.*

- Si les objectifs f_p et les fonctions g_j sont linéaires en x, on obtient un problème de *programmation linéaire multicritère* écrit sous la forme suivante :

$$(MOLP) \quad \begin{cases} \text{“}\min\text{”} z_k = c^k x \quad k = 1, 2, ..., p \\ \qquad\qquad x \in S \end{cases}$$

avec $S = \{x \in \mathbb{R}^n : Ax \leq b, \ x \geq 0\}$; A, c^k, x et b sont des matrices des dimensions respectives $(m \times n)$, $(1 \times n)$, $(n \times 1)$ et $(m \times 1)$, où n est le nombre de variables, m est le nombre de contraintes du système, et k est le nombre de fonctions objectifs.

1.3.2 Notions de dominance et d'efficacité

Dominance

Définition 1.3.2 *Soient deux vecteurs critères* Z_1, $Z_2 \in \mathbb{R}^k$. *On dit que* Z_1 *domine* Z_2 *si et seulement si* $z_k^1 \leq z_k^2$ *pour tout* $k \in \{1, ..., p\}$ *et* $z_k^1 < z_k^2$ *pour au moins un indice* $k \in \{1, ..., p\}$.

Autrement dit, Z_1 est au moins aussi bon que Z_2 sur tous les critères, et meilleur que lui sur au moins un critère.

Définition 1.3.3 *Soit* $\overline{x} \in S$ *et* $C^{\geq}$ *le cône semi-positive généré par les gradients des* p *fonctions objectifs où :*

$$C^{\geq} = \{y \in \mathbb{R}^n / Cy \geq 0,\ Cy \neq 0\} \cup \{0 \text{ de } \mathbb{R}^n\}$$

L'ensemble de dominance sur $\overline{x}$ est donné par $D_{\hat{x}} = \{\overline{x}\} \oplus C^{\geq}$. L'ensemble de dominance contient tous les points dont les vecteurs critères dominent le vecteur critère de $\overline{x} \in S$. Une autre façon de décrire l'ensemble de dominance est $D_{\hat{x}} = \{x \in \mathbb{R}^n / x = \overline{x} + y, Cy \geq 0, Cy \neq 0\}$.

Remarque 1.3.1 *L'ensemble addition de deux ensembles* X *et* Y *dans* $\mathbb{R}^n$*(noté* $X \oplus Y$*), est donné par :* $X \oplus Y = \{z \in \mathbb{R}^n /\ z = x + y,\ x \in X,\ y \in Y\}$.

On générale, il existe deux formes de dominance :

Dominance forte

Définition 1.3.4 *Soient deux vecteurs critères* Z_1, $Z_2 \in \mathbb{R}^p$. *On dit que* Z_1 *domine fortement* Z_2 *si et seulement si* $z_k^1 < z_k^2$ *pour tout* $k \in \{1, ..., p\}$. *Si* Z_1 *domine fortement* Z_2*, alors* Z_1 *est meilleur que* Z_2 *sur tous les objectifs.*

Dominance faible

Définition 1.3.5 *Soient deux vecteurs critères* Z_1, $Z_2 \in \mathbb{R}^p$. *Alors* Z_1 *domine faiblement* Z_2 *si et seulement si* $z_k^1 \leq z_k^2$ *et* $z_k^1 \neq z_k^2$*(i.e.*$z_k^1 \leq z_k^2$ *pour tout* $k \in \{1, 2, ..., p\}$ *et* $z_k^1 < z_k^2$ *pour au moins un* k*).*

Efficacité

Définition 1.3.6 *Une solution $\hat{x} \in S$ est efficace si et seulement s'il existe pas une autre solution $x \in S$ et $x \neq \hat{x}$ tel que $z_k(x) \leq z_k(\hat{x})$ et $z_k(x) \neq z_k(\hat{x})$.*

Remarque 1.3.2 *Une solution efficace est l'image réciproque d'un vecteur critère non dominé.*

Théorème 1.3.1 *[15] Soit $D_{\hat{x}}$ l'ensemble de dominance sur $\overline{x} \in S$. Alors, $\overline{x}$ est efficace si et seulement si $D_{\hat{x}} \cap S = \{\overline{x}\}$.*

Efficacité faible

Définition 1.3.7 *Une solution $\hat{x} \in S$ est une solution faiblement efficace s'il n'existe pas une autre solution $x \in S$ et $x \neq \hat{x}$ tel que $z_k(x) < z_k(\hat{x})$ pour tout $k \in \{1, 2, ..., p\}$.*

Une solution est faiblement efficace si son vecteur objectif n'est pas fortement dominé.

Efficacité forte

Définition 1.3.8 *Une solution $\hat{x} \in S$ est une solution fortement efficace s'il n'existe pas de solution $x \in S$ telle que $x \neq \hat{x}$ et $z_k(x) \leq z_k(\hat{x})$.*

Une solution $\hat{x}$ est fortement efficace s'il n'existe pas de solution x telle que le vecteur objectif, qui lui est associé, soit aussi bon que celui de $\hat{x}$. Remarquons que l'efficacité forte implique l'efficacité qui implique à son tour l'efficacité faible

Une solution efficace est aussi appelée solution Pareto optimale ou encore solution non-dominée. Nous ne parlerons donc plus de solution optimale, mais d'un ensemble de solutions Pareto optimales. L'ensemble des solutions efficaces d'un problème est noté $E(.)$. La projection dans l'espace des objectifs de cet ensemble $E(.)$ décrit une frontière communément appelée frontière efficace.

1.3.3 Caractérisation des solutions efficaces continues

Théorème de Geoffrion

Ce théorème concerne la minimisation d'une combinaison convexe des critères. Il s'agit d'une condition nécessaire et suffisante pour la détermination des solutions efficaces dans le cas continu. Soit Λ l'ensemble de tous les vecteurs $\lambda = (\lambda_k)\,, k = 1, ..., p$ définis par :

$$\Lambda = \left\{ \lambda \in \mathbb{R}^p | \sum_{k=1}^{p} \lambda_k = 1, \lambda_k \geq 0;\ k = 1, ..., p \right\}.$$

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

$$(P_\lambda) = \begin{cases} \min\limits_{x \in S} \sum\limits_{k=1}^{p} \lambda_k z_k(x) \\ t.q\ z_k(x) = (z_1(x), z_2(x), ..., z_p(x)) \end{cases}$$

alors z^* est une solution efficace si et seulement si z^* est une solution optimale du problème paramétrique (P_λ).

Ce principe n'est cependant plus valable lorsque le domaine des solutions admissibles n'est pas convexe (qui est le cas pour les problèmes en variables discrètes).

1.3.4 Points particuliers

En vue d'avoir certains points de références, des points particuliers ont été définis dans l'espace des objectifs. Ces points peuvent représenter des solutions réalisables ou non.

1. Le point x^I **idéal** est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré ou encore c'est le point qui optimise tous les critères en même temps :

$$z^I \text{ tel que } \forall i \in \{1, 2, ..., n\},\ z^I = Opt z_i\,(x) = \left(\min_{x \in S} \{z_1\,(x)\}\,, \min_{x \in S} \{z_2\,(x)\}\,, ..., \min_{x \in S} \{z_n\,(x)\} \right)$$

2. De ce point idéal peut être défini le point **utopique** de la façon suivante :

$$z^U = z^I - \epsilon U$$

où $\epsilon > 0$ et U est le vecteur unitaire ($U = (1, ..., 1) \in \Re^n$). Il est clair, de par sa définition, que ce point n'est pas réalisable.

3. Enfin le point **Nadir** qui est défini par :

$$z^N \text{tel que} \, \forall i \in \{1, 2, ..., n\}, z^N = Optz_i(x) = \left(\max_{x \in S} \{z_1(x)\}, \max_{x \in S} \{z_2(x)\}, ..., \max_{x \in S} \{z_n(x)\} \right).$$

Ce point est aussi appelé point **anti-idéal**

1.3.5 Résolution d'un problème multicritère linéaire.

Choix de la méthode d'aide à la décision

La résolution d'un problème multicritère menant à la détermination d'un ensemble de solutions efficaces. Ainsi, avant de se lancer dans la résolution d'un problème multicritère, il faut se poser la question du type de méthode d'optimisation à utiliser. En effet, on peut répartir les méthodes de résolution des problèmes multicritères en trois familles, en fonction du moment où intervient le décideur. Ainsi nous pouvons trouver les familles suivantes :

Les méthodes d'optimisation a priori Dans ce cas, le compromis que l'on désire faire entre les critères (objectifs) a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis. Généralement dans ce type de méthodes le problème est remplacé par un problème unicritère.

Les méthodes d'optimisation progressives ou interactives Dans ce cas, le décideur intervient dans le processus de recherche de solutions en répondant à différentes questions afin d'orienter la recherche. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

Les méthodes d'optimisation a posteriori Dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contre-partie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

Dans ce type de méthode, deux phases importantes sont à considérer : la phase de recherche de l'ensemble des solutions efficaces, que nous appellerons, résolution du problème d'optimisation et la phase de choix parmi ces solutions, qui relève de l'aide à la décision.

Les travaux portés dans ce mémoire, appartiennent à cette dernière catégorie, mais seulement la première phase sera traitée.

Méthodes d'optimisation multicritère [5]

De nombreuses méthodes d'optimisation existent pour les problèmes multicritères linéaire. Elles correspondent à des situations et des problématiques différentes. Toutes ces méthodes peuvent être classées en deux grandes catégories suivant le but recherché :

- Celles visant à obtenir une (ou plusieurs) solution(s) représentant un bon compromis entre les différents critères.
- Celles visant à déterminer l'ensemble de la frontière efficace.

Recherche d'un compromis La première catégorie regroupe des méthodes ramenant la résolution d'un problème multicritère à la résolution d'un (ou de plusieurs) problème(s) unicritère(s). Leur but est de trouver une (ou plusieurs) solution(s) qui correspondra(ont) à un bon compromis entre les différents critères en fonction des préférences exprimées par le décideur à l'aide des paramètres. Ces méthodes peuvent être utilisées soit avec des paramètres fixés a priori, soit de manière interactive en modifiant les paramètres durant la recherche. De nombreuses méthodes peuvent entrer dans cette catégorie, nous en présentons ci-dessous trois couramment utilisées :

Méthode lexicographique Elle consiste à considérer un ordre de priorité (dit lexicographique) entre chacun des objectifs. Le problème sera alors formulé avec l'objectif suivant :

$$lexOpt(z^1, z^2, ..., z^p)$$

Dans ce cas, la résolution pourra s'effectuer en résolvant le problème successivement sur chacun des critères pris par ordre de priorité décroissante. Les valeurs obtenues sur un objectif sont ensuite intégrées comme contraintes pour la résolution sur des objectifs moins prioritaires (cet ajout de contraintes peut casser la structure de la matrice des contraintes). La solution obtenue par une résolution exacte successive de ces problèmes unicritères sera l'une des solutions extrémales de E (une solution située sur l'enveloppe convexe).

Goal programming (programmation par buts). Dans ce type d'approche, le décideur indique une valeur cible (but) et l'objectif est de minimiser l'écart avec cette cible.(c-à-d s'approcher au maximum de valeurs cibles appelées aussi niveaux d'aspiration et notées $\hat{z}^k$, $\forall k = 1, 2, ..., p$) sur chacun des objectifs. Pour cela, des poids sont affectés à chaque objectif (notés λ_k, $\forall k = 1, 2, ..., p$), et la somme pondérée des écarts en excès(notés d_k^+) comme en défaut (notés d_k^-) par rapport aux valeurs cibles doit être minimisée. Le problème sera alors formulé de la manière suivante :

$$\begin{cases} \min z_{goal\ programming} = \sum \lambda^k \left(d_k^+ + d_k^-\right) \\ sc \qquad\qquad\qquad\qquad X \in S \\ \qquad\qquad\qquad\qquad\quad \sum\limits_{i \in I} c_i x_i + d_k^+ + d_k^- = \hat{z}^k, \forall k = 1, 2, ..., p \\ \qquad\qquad\qquad\qquad\quad d_k^+, d_k^- \geq 0 \qquad\qquad , \forall k = 1, 2, ..., p \end{cases}$$

La structure de la matrice des contraintes est alors cassée. La solution obtenue par une résolution exacte de ce problème unicritère ne sera pas obligatoirement l'une des solutions de E.

Méthode "Max-ordering" Une troisième méthode correspond au cas où il faut optimiser non pas tous les objectifs, mais uniquement le moins bon. Dans ce cas, le problème reviendra à optimiser l'objectif suivant pour un problème de maximisation (resp. minimisation) :

$$Max\ z_{\max-Ordering} = \min_k z^k (resp\ Min\ z_{\min-Ordering} = \max_k z^k)$$

La structure de la matrice des contraintes est alors cassée. La solution obtenue par une résolution exacte de ce problème unicritère ne sera pas obligatoirement l'une des solutions de E.

Détermination de la frontière efficace

La deuxième catégorie rassemble les méthodes déterminant l'ensemble complet des solutions efficaces du problème. Ces méthodes permettent au décideur de sélectionner a posteriori la (ou les) solution(s) qui l'intéresse parmi l'ensemble des solutions efficaces. Différentes méthodes peuvent être utilisées, nous présentons ci-dessous deux méthodes courantes :

La méthode du ranking Elle est applicable uniquement aux problèmes bi-critères. Elle consiste à rechercher l'ensemble des solutions situées entre le point Nadir et le point Idéal. Ceux-ci sont obtenus en calculant les points extrêmes de la frontière efficace (par deux résolutions lexicographiques successivement sur chaque critère). Ensuite, en partant de l'une des solutions extrêmes, la deuxième meilleure est cherchée, puis la troisième, . . ., puis la $k^{ièmè}$ jusqu'à atteindre la valeur du point Nadir.

La méthode de la somme pondérée Une autre méthode parfois utilisée, consiste à faire une recherche paramétrique sur la somme pondérée des critères (la structure de la matrice des contraintes est donc conservée). Cependant, cette technique présente l'inconvénient de ne permettre de trouver que les solutions efficaces situées sur l'enveloppe convexe de l'ensemble des solutions efficaces; cette méthode n'est donc valide que pour les problèmes dont toutes les solutions efficaces sont situées sur l'enveloppe convexe de E. Pour les autres problèmes, cette méthode peut cependant être utilisée pour obtenir un sous-ensemble de la frontière efficace.

Chapitre 2

Optimisation multi-objectifs en nombres entiers

2.1 Introduction

De nombreux problèmes d'optimisation se formulent par l'intermédiaire de variables entières, la résolution de ces problèmes est généralement difficile. En effet, le domaine des solutions admissibles n'est plus convexe, ainsi la solution optimale n'est en générale plus un sommet et peut donc être un point quelconque de ce domaine. Toute caractérisation particulière de la solution optimale est perdue, la résolution s'en trouve donc plus difficile. Dans ce chapitre, nous rappelons les principaux résultats de la programmation linéaire unicritère et multicritère en variables entières. Nous abordons les résultats nécessaires pour l'étude et la résolution du problème bicritère d'affectation (BAP) dans le prochain chapitre. Nous formulons les problèmes (ILP) et $(MOILP)$, nous décrivons les méthodes les plus souvent utilisées pour les résoudre. Nous terminons par quelques méthodes de résolutions pour des problèmes qui appartiennent à la même classe que notre problème.

2.2 Programmation unicritère en nombres entiers

2.2.1 Problèmes de programmation linéaire unicritère en nombres entiers

Considérons le problème de programmation linéaire unicritère suivant :

$$(PL)\begin{cases} \min z = cx \\ x \in S \end{cases}$$

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

Si toutes les variables sont entières, on a un problème de programmation linéaire en nombres entiers écrit sous la forme suivante :

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

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

Dans le cas où seulement quelques variables sont entières, on obtient un problème de programmation linéaire mixte donné par :

$$(LPMIX)\begin{cases} \min\ (cx + hy) \\ t.q \quad Ax + Gy = b \\ x \geq 0, y \geq 0 \ et\ entier \end{cases}$$

Si toutes les variables sont restreintes à 0 et 1, on a un problème de programmation en variables binaires écrit comme suit :

$$(LPBIN)\begin{cases} \min cx \\ t.q \quad Ax = b \\ x \in \{0,1\}^n \end{cases}$$

2.2.2 Complexité

Comme pour tout algorithme, une manière de comparer les méthodes de résolutions est de considérer leur complexité mathématique. Ainsi, on parlera d'un algorithme de complexité:

- Polynomiale lorsque son temps de calcul est borné par un polynôme de la taille du problème (c-à-d que la complexité de l'algorithme est en $O(n^p m^q)$ avec p et q constants).

- Exponentielle lorsque son temps de calcul ne peut pas être borné par un polynôme de la taille du problème (c-à-d que la complexité de l'algorithme est par exemple en $O(p^n)$ ou en $O(p^m)$ avec p constant). Par extension, la théorie de la complexité s'intéresse à la complexité des modèles de programmation linéaire. Celle-ci se détermine en fonction de la complexité des algorithmes susceptibles de le résoudre exactement. Ces modèles peuvent ainsi être classés en deux catégories :

– Ceux pour lesquels un algorithme de résolution exacte en temps polynomial est connu. On parlera alors de problèmes faciles ou encore de problèmes de classe $\mathcal{P}$.

– Ceux pour lesquels on ne connait pas d'algorithme de résolution exacte en temps polynomial. On parlera alors de problèmes difficiles. Parmi cette deuxième catégorie, les problèmes pour lesquels il ne peut pas exister d'algorithme de résolution exacte en temps polynomial, à moins que la conjecture $\mathcal{P} \neq \mathcal{NP}$ ne soit fausse, sont appelés $\mathcal{NP}$-difficiles.

Enfin, le choix de la méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. En effet, suivant sa complexité, le problème pourra ou non être résolu de façon optimale. Dans le cas de problèmes classés dans la classe$\mathcal{P}$, un algorithme polynomial a été mis en évidence. Il suffit donc de l'utiliser. Dans le cas de problèmes $\mathcal{NP}$-difficiles, si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé (procédure de séparation et évaluation (Branch & Bound), programmation dynamique...). Malheureusement, ces algorithmes par nature énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles (même si en pratique la taille n'est pas le seul critère limitant).

2.2.3 Optimisation combinatoire

L'optimisation combinatoire regroupe une large classe de problèmes ayant des applications dans de nombreux domaines applicatifs. Un problème d'optimisation combinatoire est défini par un ensemble fini de solutions discrètes D et une fonction objectif f associant à chaque solution une valeur (la plupart du temps, une valeur réelle). Ainsi, un problème d'optimisation combinatoire consiste en l'optimisation (minimisation ou maximisation) d'un certain critère sous différentes contraintes permettant de délimiter l'ensemble des solutions réalisables (ou solutions admissibles). La variété des problèmes d'optimisation combinatoire est en particulier due au large spectre de ses applications. Notons que la plupart des problèmes d'optimisation combinatoire sont $\mathcal{NP}$-difficiles; sauf quelques problèmes qui ont une structure particulière.

Quelques problèmes d'optimisation combinatoire [13]

Problème de sac à dos On dispose de n objets ayant chacun un poids a_j et une valeur $c_j\,(j = 1, 2, ..., n)$. Il faut sélectionner un sous ensemble de ces n objets dont le poids total soit inférieur ou égal à un nombre donné et dont la valeur somme des valeurs des objets sélectionnés, soit maximum.

Le problème doit son nom au scénario qui est souvent utilisé pour l'introduire : un campeur prépare une randonné, les n objets sont ceux qu'il envisage d'emporter. L'objet j ayant un poids a_j et une utilité c_j, le campeur cherche à maximiser l'utilité totale de son chargement tout en limitant son poids (ou son encombrement si les a_j représentent des volumes)

Problème de voyageur de commerce Soient $G = (X, E)$ un graphe complet à n sommets numérotés de 0 à $n-1$ et P une matrice $n \times n$ des poids de chaque arc. Le problème du voyageur de commerce (travelling salesman problem), consiste à trouver dans G un plus court cycle hamiltonien (un sommet représente une ville et le voyageur doit passer une et une seule fois par chaque ville, à partir de la ville 0 pour finalement revenir à son point de départ).

Une manière équivalente de formuler ce problème est de considérer la matrice P dont l'élément p_i^j de la $i^{ème}$ ligne et de la $j^{èmè}$ colonne est définie comme suit :

$$p_i^j = \begin{cases} \text{la distance de la ville } i \text{ à la ville } j \text{ s'il existe un moyen d'aller directement} \\ \qquad\qquad\qquad\qquad \text{de } i \text{ à } j \ \ ((i,j) \in E) \\ \infty \text{ si non} \end{cases}$$

Le problème consiste alors à :

$$\begin{cases} \min \sum_i p_i^{d(i)}, i = 1, 2, ..., n \end{cases}$$

2.2.4 Problème d'affectation ([13], [16])

C'est le problème qui consiste à affecter un nombre $n \in \mathbb{N}$ de sources (ou origines) au même nombre de destinations à un coût minimum. Ainsi, chaque source est associée à une et une seule destination. Cette spécificité implique deux particularités à ce programme linéaire :

- La fonction objectif $C = (c_{ij})_{(i,j=1,\ldots,n)}$ (où $(c_{ij})_{(i,j=1,\ldots,n)}$ est le coût d'affectation de $i^{ème}$ source au $j^{ème}$ destination) correspond à une matrice carrée.
- La solution optimale (ou n'importe quelle solution admissible) est telle qu'il y a une seule affectation dans chaque colonne et chaque ligne.

La formulation mathématique de problème d'affectation unicritère (AP)

Soit un atelier où il y'à n opérations à exécuter par n machines, chacune de ces machines n'effectuant qu'une et une seule tâche. En connaissant les coûts associés à chaque opération effectuée par chaque machine (qui peut être le temps mis par la machine i a l'accomplissement de la tâche j) l'objectif est de déterminer les affectations opération/machine qui minimise le critère. C'est un problème combinatoire avec :

- Choix de variables :

$$x_{ij} = \begin{cases} 1 \ si \text{ l'opération } i \text{ est exécutée par la machine } j \\ 0 \qquad\qquad \text{sinon} \end{cases} \quad i = 1, 2, ..., n \quad j = 1, 2, ..., n$$

- Une opération est exécutée par une et une seule machine :
$\sum\limits_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, ..., n$
- Une machine exécute une et une seule opération :
$\sum\limits_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, ..., n$
Ce qui revient au modèle complet :

$$(AP)\begin{cases} \min \ Z(x) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}x_{ij} \\ \sum\limits_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, ..., n \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, ..., n \\ \quad x_{ij} \in \{0, 1\} \quad \forall (i, j) \end{cases}$$

2.2.5 Solution de problème d'affectation unicritère

Dégénérescence

Vu le caractère fortement dégénéré du problème d'affectation; l'application des méthodes classiques de résolution des problèmes linéaire à ce dernier tel que le simplexe, engendre des calculs très dur et de plus en plus long car au bout de quelques itérations, on retrouve une solution déjà trouver, c'est à dire le problème de cyclage.

Une façon de résoudre le problème d'affectation est d'énumérer les solutions et de choisir celle qui fournit le coût le plus faible. En générale pour une matrice coût a n lignes et n colonnes, il y a $n!$ possibilités d'affectations qui veut dire $n!$ solutions admissibles. Puisque $n!$ s'accroît très vite avec n, l'énumération totale est une tâche presque impossible.

La méthode hongroise

Le mathématicien hongrois Egerváry [12] était le premier à proposer un algorithme (implicite) pour le problème d'affectation, ce que inspire Kuhn pour développer la méthode hongroise (Hungarian method, en hommage), qui est une procédure polynomiale de complexité $O(n^4)$. Plus tard, avec l'introduction des procédures de plus court chemin la complexité a été réduite à $O(n^3)$. Et depuis plusieurs algorithmes pour le (AP) ont été développés.

Dans un survey (bibliographie) présenté par Dell'Amico et Martello [11], plus de 100 papiers sur le problème sont mentionnés.

Principe de la méthode

Soit la matrice coût $C = (c_{ij})_{i,j=1,2,\dots,n}$

$$C = \begin{pmatrix} c_{11} & .. & .. & c_{1n} \\ & . & & \\ & & . & \\ c_{n1} & & & c_{nn} \end{pmatrix}$$

Phase1.

1)- Choisir le minimum de chaque ligne : $m_i = \min\limits_{1\leq j\leq n}\{c_{ij}\}$, $i = 1, \dots, n$.

2)- Pour chaque ligne, on retranche le minimum correspondant et on a $Z \geq \sum\limits_{i=1}^{n} m_i$.

3)- Choisir le minimum de chaque colonne : $l_j = \min\limits_{1\leq i\leq n}\{c'_{ij} = c_{ij} - m_i\}$, $j = 1, \dots, n$.

4)- Pour chaque colonne, on retranche le minimum correspondant et on a:

$$Z \geq \sum_{i=1}^{n} m_i + \sum_{j=1}^{n} l_j.$$

5)- Trouver une solution réalisable en affectant un seul zéro par ligne et par colonne, on obtient le tableau (I).

Si cette solution existe alors c'est la solution optimale du tableau initiale (tableau (0)). Si on a pas (5), aller à la deuxième phase :

Phase2.

a)- Marquer les lignes ayant un zéro non affecté.

b)- Marquer les colonnes ayant un zéro non affecté 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 (II).

Trouver une solution réalisable on affectant un seul zéro par ligne et par colonne en cas d'échec, reprendre la procédure à partir de la deuxième étape. Si une telle solution existe, c'est une solution optimale du (AP).

Justification de la méthode

1. Montrons que si X^* est une solution optimale du problème d'affectation (PA) dans le tableau (I) alors X^* est une solution optimale dans le tableau (0).

1.1. Tableau (0) : $c_{ij},\ \ i,j = 1,2,...,n$

Soit

$$
\begin{aligned}
m_i &= \min_{1\leq j\leq n}\{c_{ij}\}, && i = 1,\ldots,n.\\
c'_{ij} &= c_{ij} - m_i, && i,j = 1,\ldots,n\\
l_j &= \min_{1\leq i\leq n}\{c'_{ij}\}, && j = 1,\ldots,n\\
c''_{ij} &= c'_{ij} - l_j, && i,j = 1,\ldots,n.
\end{aligned}
$$

1.2. Tableau (I) : $c''_{ij},\ \ i,j = 1,2,...,n$

Soit X une affectation, on note par $Z_0(X)$: la valeur de la fonction objectif dans le tableau (0) et $Z_1(X)$ la valeur de la fonction objectif dans le tableau (I).

On a :

$$
\begin{aligned}
Z_1(X) &= \sum_{1\leq i\leq n}\sum_{1\leq j\leq n} c''_{ij}x_{ij}\\
&= \sum_{1\leq i\leq n}\sum_{1\leq j\leq n}\left(c'_{ij} - l_j\right)x_{ij}\\
&= \sum_{1\leq i\leq n}\sum_{1\leq j\leq n} c'_{ij}x_{ij} - \sum_{1\leq i\leq n}\sum_{1\leq j\leq n} l_j x_{ij}
\end{aligned}
$$

D'autre part :

$$
\sum_{1\leq i\leq n}\sum_{1\leq j\leq n} l_j x_{ij} = \sum_{1\leq j\leq n}\sum_{1\leq i\leq n} l_j x_{ij} = \sum_{1\leq j\leq n} l_j \sum_{1\leq i\leq n} x_{ij} = \alpha_1(constante)
$$

D'où

$$
\begin{aligned}
Z_1(X) &= \left(\sum_{1\leq i\leq n}\sum_{1\leq j\leq n}\left(c_{ij} - m_i\right)x_{ij}\right) - \alpha_1\\
&= \left(\sum_{1\leq i\leq n}\sum_{1\leq j\leq n} c_{ij}x_{ij}\right) - \left(\sum_{1\leq i\leq n}\sum_{1\leq j\leq n} m_i x_{ij}\right) - \alpha_1
\end{aligned}
$$

De même:

$\sum\limits_{1\leq i\leq n}\sum\limits_{1\leq j\leq n} m_i x_{ij} = \sum\limits_{1\leq i\leq n} m_i \sum\limits_{1\leq j\leq n} x_{ij} = \alpha_2$

D'où

$Z_1(X) = \left(\sum\limits_{1\leq i\leq n}\sum\limits_{1\leq j\leq n} c_{ij}x_{ij}\right) - \alpha_1 - \alpha_2 = Z_0(X) - (\alpha_1+\alpha_2)$

Posons : $\alpha_1 + \alpha_2 = M$.

D'où :$Z_1(X) = Z_0(X) - M$.

Si X^* est une solution optimale pour le tableau (I) alors $Z_1(X^*) = 0$.

Supposons que X^* ne soit pas optimale pour le tableau (0), donc il existe une autre solution optimale et soit $\overline{X}$ cette solution .

Donc $Z_0(\overline{X}) \leq Z_0(X^*) \Rightarrow Z_1(\overline{X}) + M \leq Z_1(X^*) + M \Rightarrow Z_1(\overline{X}) \leq Z_1(X^*)$ absurde car X^*est solution optimale pour le tableau (I).

D'où X^* est optimale pour le tableau (0), avec $Z_0(X^*) = M$.

2. Montrons que si on obtient une solution réalisable on affectant les zéros du tableau (II) alors c'est une solution optimale de (AP) dans le tableau (0). Après application de la deuxième étape, on obtient le tableau (II) sous la forme suivante :

	$\overline{J}$	J
$\overline{I}$	$d_{ij} = c''_{ij} + \alpha$ *les éléments doublement barrés* $+\alpha$	$d_{ij} = c''_{ij}$ *les éléments barrés*
I	$d_{ij} = c''_{ij}$ *les éléments barrés*	$d_{ij} = c''_{ij} - \alpha$ *les éléments non barrés* $-\alpha$

Où : d_{ij} : les éléments du tableau (II), $\alpha = \min\limits_{\substack{i\in I\\ j\in J}} \left\{c''_{ij}\right\}$

Soit X une affectation :

$$\begin{aligned} Z_2(X) &= \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} \\ &= \sum_{\overline{I}}\sum_{\overline{J}} d_{ij}x_{ij} + \sum_{\overline{I}}\sum_{J} d_{ij}x_{ij} + \sum_{I}\sum_{\overline{J}} d_{ij}x_{ij} + \sum_{I}\sum_{J} d_{ij}x_{ij} \end{aligned}$$

Posons: $\sum\limits_{\overline{I}}\sum\limits_{J} d_{ij}x_{ij} + \sum\limits_{I}\sum\limits_{\overline{J}} d_{ij}x_{ij} = A$

D'où

$$\begin{aligned}
Z_2(X) &= \sum_{\overline{I}}\sum_{\overline{J}} d_{ij}x_{ij} + A + \sum_{I}\sum_{J} d_{ij}x_{ij} \\
&= \left(\sum_{\overline{I}}\sum_{\overline{J}} \left(c^{"}_{ij}+\alpha\right) x_{ij}\right) + A + \left(\sum_{I}\sum_{J} \left(c^{"}_{ij}-\alpha\right) x_{ij}\right) \\
&= \left(\sum_{\overline{I}}\sum_{\overline{J}} c^{"}_{ij}x_{ij}\right) + \left(\sum_{\overline{I}}\sum_{\overline{J}} \alpha x_{ij}\right) + A + \left(\sum_{I}\sum_{J} c^{"}_{ij}x_{ij}\right) - \left(\sum_{I}\sum_{J} \alpha x_{ij}\right) \\
&= \left(\sum_{\overline{I}}\sum_{\overline{J}} c^{"}_{ij}x_{ij} + \sum_{I}\sum_{J} c^{"}_{ij}x_{ij} + A\right) + \alpha\left(\sum_{\overline{I}}\sum_{\overline{J}} x_{ij} - \sum_{I}\sum_{J} x_{ij}\right) \\
&= Z_1(X) + \alpha\left(\sum_{\overline{I}}\sum_{\overline{J}} x_{ij} - \sum_{I}\sum_{J} x_{ij}\right)
\end{aligned}$$

$$\begin{aligned}
\sum_{\overline{I}}\sum_{\overline{J}} x_{ij} - \sum_{I}\sum_{J} x_{ij} &= \sum_{\overline{I}}\sum_{\overline{J}} x_{ij} + \sum_{\overline{I}}\sum_{J} x_{ij} - \sum_{\overline{I}}\sum_{J} x_{ij} - \sum_{I}\sum_{J} x_{ij} \\
&= \left(\sum_{\overline{I}}\sum_{\overline{J}} x_{ij} + \sum_{\overline{I}}\sum_{J} x_{ij}\right) - \left(\sum_{\overline{I}}\sum_{J} x_{ij} + \sum_{I}\sum_{J} x_{ij}\right) \\
&= \sum_{\overline{I}}\left(\sum_{\overline{J}} x_{ij} + \sum_{J} x_{ij}\right) - \sum_{J}\left(\sum_{\overline{I}} x_{ij} + \sum_{I} x_{ij}\right) \\
&= \left|\overline{I}\right| + \left|\overline{J}\right|
\end{aligned}$$

D'ou :

$Z_2(X) = Z_1(X) + \alpha\left(\left|\overline{I}\right| + \left|\overline{J}\right|\right) = Z_1(X) + \beta(constante)$

Si X^*est solution optimale pour le tableau (II), alors $Z_2(X^*) = 0$.

Supposons que X^* ne soit pas optimale pour (PA) dans le tableau (0), et soit $\overline{X}$ une solution optimale pour (PA).

Donc : $Z_0(\overline{X}) \leq Z_0(X^*) \Rightarrow Z_2(\overline{X}) + M - \beta \leq Z_2(X^*) + M - \beta \Rightarrow Z_2(\overline{X}) \leq Z_2(X^*)$ impossible.

D'où X^*est optimale pour le tableau (0).

Remarque 2.2.1 *La solution optimale donnée par la méthode hongroise n'est pas unique.*

2.2.6 Méthodes exactes de résolution

Pour les problèmes en variables discrètes, le nombre de solutions admissibles est fini lorsque leur domaine est borné. Pourtant, le nombre de ces solutions grandissant généralement de manière exponentielle (on parle d'explosion combinatoire), et la propriété de convexité du domaine des solutions n'étant en général plus valable, la résolution de ces problèmes n'est souvent pas aisée. Une exception existe lorsque la matrice des contraintes A est totalement unimodulaire. Tous les sommets du domaine des solutions admissibles de la relaxation linéaire du problème sont alors entiers. Le problème peut donc être résolu comme un problème en variables continues (et donc en temps polynomial). D'une manière générale, la résolution de ces problèmes n'est cependant pas aisée. Beaucoup de ces problèmes sont ainsi $\mathcal{NP}-$ difficiles.

Les méthodes de résolution exactes pour les problèmes en variables discrètes ou mixtes sont nombreuses, cependant trois familles principales peuvent être distinguées :

La programmation dynamique

Consiste à placer le problème dans une famille de problèmes de même nature mais de difficulté décroissante, puis à trouver une relation de récurrence liant les solutions optimales de ces problèmes,

Les méthodes dites par séparation et évaluation (Branch & Bound par exemple)

Consistent à faire une énumération implicite fondée sur un principe de décomposition du problème en sous-problèmes (branchements) [13], et pour chaque sous problème crée, l'évaluation de ceux-ci permet d'avoir une idée de la valeur des solutions qu'il contient, appréciant par défaut la meilleur solution lui appartenant. Ainsi de suite jusqu'à ne plus avoir que des problèmes faciles à résoudre ou qui ne peuvent pas contenir de solutions optimales ou réalisables,

Les méthodes polyhédrales (méthode des coupes)

Consistent à ajouter progressivement des contraintes supplémentaires (ou coupes) afin de ramener la résolution d'un problème linéaire en nombres entiers à celle d'un problème linéaire, et ça en ramenant le domaine des solutions admissibles à un domaine convexe (sans en enlever la ou les solutions optimales bien évidemment). Ces coupes sont appelées inégalités valides lorsqu'elles ne suppriment aucune solution admissible du problème.

De plus, certaines méthodes essayent de combiner les avantages de différentes familles.

Ainsi par exemple, la méthode du Branch & Cut, dérivée du Branch & Bound, intègre une recherche de coupes dans les nœuds de l'arbre de recherche afin d'améliorer la valeur de la relaxation linéaire guidant le processus.

Nous en présentons ci-dessous deux méthodes couramment utilisées :

2.2.7 Procédure par séparation et évaluation *(Branch and Bound)*

La méthode de *Branch and Bound* repose essentiellement sur trois notions clés :

Procédure de séparation

L'ensemble des solutions est séparé (décomposé) successivement en des sous ensembles de tailles de plus en plus réduite en vue d'aboutir à un sous ensemble ne contenant qu'une seule solution ou à un sous ensemble vide ou encore appelé stérile.

Procédure d'évaluation

Pour chaque sous ensemble crée par séparation, l'évaluation permet d'analyser ce sous problème ou encore d'avoir une idée de la valeur des solutions qu'il contient, appréciant par défaut la meilleur solution lui appartenant.

Procédure de cheminement

C'est la règle de sélection du sous ensemble que l'on va séparer à la prochaine itération. Elle indique quels sous-ensembles analysés et dans quel ordre.

Les stratégies classiques de sélection sont : meilleur d'abords, profondeur d'abords et largeur d'abords.

2.2.8 Les coupes de Gomory [13]

Considérons le (ILP) sous forme standard :

$$(ILP)\begin{cases} \min \; z = cx \\ t.q \quad x \in D \end{cases} \tag{(1.1)}$$

La relaxation continue de (1.1) est :

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

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

Description de la méthode

Commençons par résoudre la relaxation linéaire du problème par l'algorithme (primal) du simplexe et considérons le tableau obtenu à l'optimum $\left(\widetilde{A}x = \widetilde{b}\right)$: parmi les variables de base, choisissons une variable $x_i, i \in J$, fractionnaire (s'il n'y en a pas, on est à l'optimum en entier). La contrainte correspondante à x_i se lit directement sur le tableau optimal :

$$x_i + \sum_{J \in \overline{J}} \widetilde{a_{ij}} x_j = \widetilde{b} \tag{(1.2)}$$

Puisque toutes les variables sont positives ou nulles dans (2.1), on a :

$$\sum_{J \in \overline{J}} \lfloor \widetilde{a_{ij}} \rfloor x_j \leq \sum_{J \in \overline{J}} \widetilde{a_{ij}} x_j$$

Donc on a

$$x_i + \sum_{J \in \overline{J}} \lfloor \widetilde{a_{ij}} \rfloor x_j \leq \widetilde{b_i}$$

La patrie gauche de cette inéquation est entière. Le second membre peut donc être remplacé par $\left\lfloor \widetilde{b_i} \right\rfloor$:

$$x_i + \sum_{J \in \overline{J}} \lfloor \widetilde{a_{ij}} \rfloor x_j \leq \left\lfloor \widetilde{b_i} \right\rfloor \tag{(1.3)}$$

$(1.2) - (1.3)$ donne :

$$\sum_{J\in\overline{J}} (\widetilde{a_{ij}} - \lfloor\widetilde{a_{ij}}\rfloor)\, x_j \leq \widetilde{b_i} - \left\lfloor \widetilde{b_i} \right\rfloor$$

Posons :

- $f_{ij} = \widetilde{a_{ij}} - \lfloor\widetilde{a_{ij}}\rfloor$ (partie fractionnaire de $\widetilde{a_{ij}}$, $0 < f_{ij} < 1$)
- $f_i = \widetilde{b_i} - \left\lfloor \widetilde{b_i} \right\rfloor$ (partie fractionnaire de $\widetilde{b_i}$, $0 < f_i < 1$)

Finalement, nous obtenons la contrainte :

$$\sum_{J\in\overline{J}} f_{ij}x_j \leq f_i$$

C'est la coupe de Gomory

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

$$-\sum_{J\in\overline{J}} f_{ij}x_j + s = -f_i \qquad ((1.4))$$

Si la coupe (1.4) est ajoutée au tableau optimal du simplexe d'un programme linéaire, aucun point admissible entier n'est exclu, et le nouveau tableau est écrit sous forme canonique par rapport à la nouvelle base formée de la base optimale précédente à laquelle on ajoute s. Cette nouvelle base n'est pas réalisable.

Remarque 2.2.2 *La base ainsi obtenue vérifie les conditions d'optimalité (coûts réduits positifs ou nuls) et n'est pas primal-admissible : on peut appliquer l'algorithme dual du simplexe pour résoudre le nouveau programme linéaire formé.*

Remarque 2.2.3 *La nouvelle variable s doit être entière comme toutes les autres variables du problème pour pouvoir itérer le processus.*

Par ce procédé, on ajoute ainsi, une par une des coupes jusqu'à obtention d'une solution de base entière.

2.3 Programmation linéaire multicritère en nombres entiers

Un programme linéaire multicritère en nombres entiers est constitué d'un système de contraintes linéaires définissant un domaine discret de solutions réalisables, et d'un ensemble de fonctions linéaires à maximiser ou à minimiser définissant des objectifs conflictuels. Le problème consiste à déterminer toute solution réalisable entière telle qu'il n'existe aucune autre solution réalisable entière qui fournisse des valeurs au moins aussi bonnes que celles sur chaque objectif et même meilleure sur au moins un objectif.

2.3.1 Problèmes de programmation mathématique multicritère en nombres entiers

Considérons le problème de programmation linéaire multicritère suivant :

$$(MOLP)\begin{cases} \text{“}\min\text{”} z_k = c^k x \quad k = 1, 2, ..., p \\ x \in S \end{cases}$$

avec $S = \{x \in \mathbb{R}^n : Ax \leq b,\ x \geq 0\}$; $A \in \mathbb{R}^{(m\times n)}$, $c^k \in \mathbb{R}^{(1\times n)}$, $x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$ $k = 1, 2, ..., p$

Si toutes les variables sont des entiers, nous obtenons un problème de programmation linéaire multicritère en variables entières :

$$(MOILP)\begin{cases} \text{“}\min\text{”} z_k = c^k x \quad k = 1, 2, ..., p \\ x \in D \end{cases}$$

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

2.3.2 Solutions supportées / non supportées

Définition 2.3.1 *Une solution $\hat{x}$ de D est efficace pour le problème $(MOILP)$, s'il n'existe pas une autre solution $x \in D$ telle que $z_k(x) \leq z_k(\hat{x})$, $k = 1, 2, ..., p$ avec au moins une inégalité stricte.*

Pour les problèmes en nombres entiers, deux types de solutions efficaces peuvent être différenciées : les solutions efficaces supportées et les solutions efficaces non supportées. Les premières sont celles situées sur l'enveloppe convexe de l'ensemble des solutions efficaces et peuvent donc être trouvées à l'aide d'une agrégation linéaire des objectifs. Elles sont donc plus simples à obtenir que les solutions non supportées. D'ailleurs, les premiers travaux en optimisation multi-objectifs se sont pour la plupart focalisés sur la recherche de ces solutions supportées en optimisant des combinaisons linéaires des objectifs utilisant différents vecteurs de poids. Alors pourquoi ne pas se satisfaire des solutions supportées?. Tout d'abord parce que ces solutions peuvent ne représenter qu'un petit sous-ensemble des solutions efficaces. De plus, ces solutions supportées ne sont pas forcément bien réparties et ne représentent pas toujours un bon compromis. Donc, si l'on veut obtenir des solutions de bon compromis entre les objectifs, il est nécessaire de considérer les solutions efficaces non supportées.

Notons par $SE(.)$ l'ensemble des solutions efficaces supportées, et par $NSE(.)$ l'ensemble des solutions efficaces non supportées.

2.3.3 Revue des méthodes exactes existantes (voir [4], [6], [10])

Dans la littérature, une attention particulière a portés sur les problèmes à deux critères en utilisant les méthodes exactes. Ces méthodes sont efficaces pour des problèmes de petites tailles. Pour des problèmes à plus de deux critères ou de grandes tailles, il n'existe pas de procédures exactes efficaces, étant donné les difficultés simultanées de la complexité $\mathcal{NP}$-complet, et le cadre multi-critères des problèmes.

Nous présentons ici les principales méthodes exactes développées de façon à obtenir l'ensemble ou une partie de l'ensemble des solutions efficaces.

L'agrégation linéaire

Cette méthode populaire transforme le problème multi-objectifs en un problème mono-objectif en combinant linéairement les différents objectifs. Ainsi, le nouveau problème obtenu, car il s'agit alors d'un problème différent (le problème obtenu conserve la structure de la matrice des contraintes ce qui peut être intéressant lorsque cette structure correspond à un problème caractéristique facile à résoudre), consiste à optimiser $\sum_i \lambda_i z_i$. Le théorème de Geoffrion indique qu'en utilisant différentes valeurs pour le vecteur λ, il est possible d'obtenir toutes les solutions supportées du problème multi-objectifs initial. Par contre, aucune solution non supportées ne peut être trouvée par cette méthode. La méthode d'agrégation linéaire a donc ses limites. Toutefois, elle est intéressante pour des problèmes ayant de nombreux objectifs et/ou un grand nombre de solutions supportées bien réparties. Dans ce contexte, il peut être suffisant de générer les solutions supportées.

Notons, enfin que les résultats obtenus dans la résolution du problème unicritère obtenu dépendent fortement des paramètres choisis pour le vecteur poids λ. Les poids λ_i doivent aussi être choisis en fonction des préférences associées aux critères, ce qui est une tâche délicate. Ainsi, une approche généralement utilisée est de résoudre le problème avec différentes valeurs de λ, d'où le coût associé à cette classe de méthodes.

La recherche dichotomique

La recherche dichotomique offre un schéma d'application de l'agrégation linéaire permettant d'obtenir les solutions non supportées. Cette méthode consiste à explorer de façon dichotomique des intervalles de recherche de plus en plus petits. Tout d'abord les solutions extrêmes sont recherchées. Puis une recherche est menée entre ces solutions r et s suivant une direction perpendiculaire à la droite (r, s). En interdisant de réobtenir les solutions r et s et en éliminant les solutions dominées par ces solutions, cette recherche trouve la meilleure solution efficace relativement à cette direction de recherche, solution qui peut alors être non supportées. Cette nouvelle solution crée deux nouveaux intervalles qu'il faut explorer de la même façon. Cette méthode, très utilisée au bi-objectifs, est intéressante mais nécessite de l'ordre de $2n$ recherches, si n est le nombre de solutions efficaces. La recherche dichotomique est, là encore, applicable aux problèmes bi-objectifs.

Méthode ϵ-contrainte

Le principe de la méthode ϵ-contrainte qui consiste, dans le cas bi-objectifs, à borner l'un des objectifs (en général le plus difficile à résoudre) et à optimiser l'autre objectif (optimisation mono-objectif) en tenant compte de cette borne, est intéressant lorsque l'on cherche à énumérer toutes les solutions efficaces. En effet, en utilisant cette méthode itérativement, en repartant à chaque fois de la solution trouvée pour définir la borne suivante, il est possible en utilisant une méthode exacte mono-objectif de générer, l'ensemble des solutions efficace. Par exemple pour le cas bi-objectifs, la solution efficace optimale pour l'objectif z_1 est d'abord recherchée (solution $Optz_1$). Cette solution détermine la borne B_1 sur l'objectif z_2 en dessous de laquelle l'objectif z_2 va devoir être optimisée. Cela nous donne la solution $Opt\ z_2$ qui elle-même détermine la borne B_2, etc. L'inconvénient principal de cette méthode est qu'elle nécessite une résolution mono-objectif pour chacune des solutions. Lorsque ce nombre est élevé, cela peut être vu comme une limite, d'autant plus lorsque la méthode de résolution mono-objectif est coûteuse. De plus, lorsqu'il n'existe pas de méthode mono-objectif efficace, rechercher une solution particulière (respectant une borne, par exemple) est souvent synonyme d'énumération de nombreuses autres solutions dont certaines peuvent être efficaces optimales. Ainsi certaines solutions seront énumérées plusieurs fois sans que la méthode les repère.

Méthode de deux-phases

La méthode de deux phases a initialement été proposée par Gonzalez et al (voir [19]), et par la suite, adaptée par Ulungu et Teghem au problème d'affectation et sac à dos bi-objectifs. Pendant la dernière décennie, cette méthode s'est avérée très efficace pour résoudre même des problèmes qui sont très difficiles, (voir par exemple ([21])).

Comme son nom l'indique, cette méthode est décomposée en deux étapes : la première consiste à trouver toutes les solutions supportées, puis la deuxième phase cherche entre ces solutions les solutions non supportées. Cette méthode travaille donc essentiellement dans l'espace des objectifs et elle est souvent utilisée dans le cas bi-critères.

Première phase L'objectif de la première phase est d'obtenir l'ensemble des solutions efficaces supportées. Comme nous l'avons vu précédemment, ces solutions ont l'avantage d'être relativement faciles à trouver puisqu'elles optimisent une certaine combinaison linéaire des objectifs. Ainsi, durant la première phase de la méthode, les deux solutions extrêmes (solutions optimisant chacun des deux objectifs) sont recherchées. Puis, de façon récursive, dès que deux solutions supportées r et s sont trouvées, la méthode recherche d'éventuelles autres solutions supportées entre r et s, à l'aide de combinaisons linéaires bien choisies des objectifs. A la fin de la première phase l'ensemble des solutions supportées est donc trouvé. Cette première phase rappelle la méthode dichotomique, mais ici seules les solutions supportées sont recherchées. Pour cela, lors de l'exploration entre deux solutions, on s'autorise à retrouver l'une de ces deux solutions, lorsqu'il n'existe pas d'autres solutions supportées dans l'intervalle.

Deuxième phase La deuxième phase consiste alors en la recherche des solutions non supportées. Ces solutions ne peuvent être obtenues par combinaisons d'objectifs. Ulungu et Teghem proposent alors d'utiliser les solutions supportées trouvées pour réduire l'espace de recherche en argumentant que les solutions Pareto non supportées restantes sont forcément dans les triangles basés sur deux solutions supportées consécutives. Ainsi, une recherche de type deuxième phase est exécutée entre chaque couple de solutions supportées adjacentes. La méthode de recherche au sein de ces triangles dépend du problème étudié. A la fin de la deuxième phase, toutes les solutions Pareto sont trouvées.

La méthode en deux-phases présente un schéma de résolution exacte très intéressant car très général et qui ne dépend pas du problème. Son intérêt réside dans une décomposition de l'espace de recherche et l'utilisation de méthodes mono-objectifs pour les différentes résolutions successives (recherche des extrêmes, résolution des agrégations...). Appliquer la méthode en deux-phases pour la résolution d'un problème bi-objectifs nécessite donc d'avoir une méthode mono-objectif efficace, ce qui rend la méthode performante pour ces problèmes là. Cependant, la non existence de méthode exacte efficace pour d'autres problèmes pouvant optimiser chaque objectif séparément peut compromettre l'intérêt de la méthode.

PPM : Parallel Partitionning Method [6]

Cette méthode est décomposée en trois phases :

Phase1 : Recherche des extrêmes et partitionnement de l'espace de recherche. Les deux solutions extrêmes sont recherchées. Ces deux solutions indiquent donc les valeurs minimales et maximales des solutions efficaces pour chacun des objectifs. L'espace contenu entre ces deux solutions est ensuite découpé de façon uniforme suivant l'objectif le plus difficile à résoudre (supposons un découpage suivant z_2).

Phase2 : Recherche d'une solution par partition. À la manière de la méthode ϵ-contrainte, la solution optimisant le second objectif (z_1) est recherchée pour chacune des partitions. À la fin de cette deuxième phase, des solutions, les mieux réparties possible, sont trouvées. Les solutions obtenues sont toutes Pareto optimales (efficaces) mais pas nécessairement supportées.

Phase3 : Recherche des solutions efficaces dans les sous-espaces. Cette dernière phase consiste à rechercher dans chacun des sous-espaces délimités par deux solutions adjacentes obtenues lors de la phase précédente, toutes les solutions efficaces (supportées et non supportées) existantes. Cette méthode a donc l'avantage de découper l'espace de recherche tout en restant plus indépendante de la structure des solutions efficaces. En effet, aucune hypothèse n'est prise concernant la répartition des solutions supportées, puisque la deuxième phase recherche des solutions qui ne sont pas forcément supportées (mais qui sont efficaces).

Remarquons, que pour cette méthode, suivant le type de méthode utilisée pour la troisième phase, il peut être possible de trouver toutes les solutions d'une même partition en une seule exécution.

2.3.4 Abécédaire de méthodes d'optimisation multi-objectifs [1]

Désignons par :

Z : Un vecteur objectif avec $Z = (Z_1, Z_2, ..., Z_k)^T$

Supposons que l'ensemble des solutions réalisables entières est fini.

Méthode de MOMIX(Multi-objectif à variables Mixte)[1]

C'est une méthode introduite par Teghem et Kunch, elle utilise le concept de Branch and Bound interactive. La méthode comporte principalement deux phases.

Phase1 : Initialisation

Déterminer le premier point efficace x^0en résolvant le problème suivant pour $m = 0$.

$$(P^{(m)}) \begin{cases} \min \delta \\ t.q \ \beta_{m,k}(M_{m,k} - c^k x) \leq \delta \quad k = 1, 2..., p \\ x \in S_m \end{cases}$$

où $S_0 = S$; $M_{m,k}$ la valeur optimale de la $k^{ième}$ fonction objectif sur S_m et $\beta_{m,k} = \frac{\alpha_{m,k}}{\sum\limits_{i=1}^{p} \alpha_{m,i}}$ le poids de la $k^{ième}$ fonction objectif .

Phase2.

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

Première étape : Procédure descendante. Soit la $m^{ième}$ itération.

- Soit x^{m-1} le $m^{ième}$ compromis (solution efficace) et $z_k^{(m-1)} = c^k x^{m-1}$ la valeur correspondante pour la $m^{ème}$ fonction objectif.

- Le DM (decision maker) indique le critère $l_m(1) \in \{1, 2, ..., p\}$ à améliorer en priorité.

- Un nouveau compromis est obtenu en résolvant le problème $(P_{(m)})$ avec $S_m = S_{m-1} \cap \left\{ x | z_{l_{m(1)}}(x) > z_{l_{m(1)}}.(x^{m-1}) \right\}$ de façon que le critère $l_m(1)$ sera ainsi amélioré .

Tests d'arrêt. Plusieurs tests d'arrêts ont été définit, on cite parmi :

- $S_m = \varnothing$.

- $M_{m,k} - m_{m,k} \leq \varepsilon_k$ pour toutes les valeurs k (ε_k>0, $fixé$).

- $\hat{z}$, le vecteur des meilleurs valeurs trouvées précédemment, est préféré au vecteur idéal M_m relatif à DM.

- Aucune amélioration n'a été apportée à $\hat{z}$ après un nombre d'itération fixé par le décideur DM.

Deuxième étape : Procédure remontante Le but de cette étape est de scanner la région d'admissibilité négligée à chaque itération de la procédure descendante pour un éventuel meilleur compromis que $\hat{z}$ dans cette région.

- Le nœud correspondant au compromis x^{m-1} est séparé en p branches.

- Soit $l_m(k); k = 1, 2, ..., p$ l'ordre de priorité dans lequel le décideur veut voir améliorer les p critères par rapport à ce compromis.

- Les p nœuds sont obtenus par l'adjonction respective des contraintes suivantes :

$z_{l_m(1)} > z^{m-1}_{l_m(1)}$

$z_{l_m(2)} > z^{m-1}_{l_m(2)}$ et

$z_{l_m(1)} \leq z^{m-1}_{l_m(1)}$

...

$z_{l_m(p)} > z^{m-1}_{l_m(p)}$ et

$z_{l_m(k)} > z^{m-1}_{l_m(k)}; \;\; k = 1, 2, ..., p.$

L'ensemble de solutions admissibles est partitionné et à chaque sous nœud on résous le problème $(P_{(m)})$.

Tests d'arrêt. Les mêmes tests d'arrêts que l'étape précédente sont utilisés

Méthode de J.Sylva et A.Crema. [2]

C'est une méthode développée récemment, elle a été implémentée pour le problème de sac à dos avec trois fonctions objectifs, dix contraintes et jusqu'a 30 variables et aussi pour le problème d'affectation généralisée.

Les auteurs ont considérés le problème $MOILP$ avec paramètre coût entiers i.e.

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

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

Pour justifier la méthode les auteurs ont utilisé la proposition suivante :

Proposition 2.3.1 *Soient $x^1, x^2, ..., x^l$ des solutions efficaces du problème (P) et*

$$\Delta_s = \{x \in \mathbb{Z}^n | Cx \leq Cx^s\} \,((s = 1, ..., l))$$

Soit x^*une solution efficace du problème

$$(P^l) \begin{cases} \text{“}\max\text{”} \; Z = Cx \\ x \in \left(D - \bigcup\limits_{s=1}^{l} \Delta_s\right) \end{cases}$$

Alors x^*une solution efficace du problème (P). En plus, si le problème (P^l) devient impossible (non réalisable) alors $\{Cx^s, s = 1, ..., l\}$ est l'ensemble de tous les points non dominées dans l'espace des critères pour le problème (P).

La méthode est basée sur le corollaire suivant :

Corollaire 2.3.1 *Soient $x^1, x^2, ..., x^l$ des solutions efficaces du problème (P) et*

$$\Delta_s = \{x \in \mathbb{Z}^n | Cx \leq Cx^s\}$$

Si x^* est une solution optimale pour le problème unicritère :

$$(P_\lambda^l) : \max\{\lambda^t Cx | x \in D - \bigcup_{s=1}^{l} \Delta_s\}$$

pour quelques valeurs de $\lambda \in \mathbb{R}^p, \lambda{>}0$. Alors x^* est une solution optimale pour le problème (P).

La méthode

Étape1. Après avoir choisi le paramètre $\lambda{>}0$, on résout en première étape le problème de programmation linéaire unicritère suivant :

$$(P_0) \left\{\max \lambda^t Cx | \; Ax = b, x \geq 0, x \in \mathbb{Z}^n\right\}$$

Si le problème n'admet pas de solution optimale, alors le problème (P) n'a pas de solution efficaces aussi. Sinon, une solution optimale x^1 trouvée est efficace pour le problème (P) (par le corollaire).

Puis, une suite de problèmes unicritères (où à chaque itération on ajoute des contraintes éliminant les solutions déjà trouvées précédemment) est résolue séquentiellement.

après l itérations, si le problème $(P_{(l-1)})$ est irréalisable, alors l'algorithme prend fin. Sinon, une nouvelle solution efficace x^l est générée et un nouveau problème $(P_{(l)})$ est défini en éliminant de l'ensemble d'admissibilité de $(P_{(l-1)})$ toutes les solutions vérifiant $Cx \leq Cx^l$. Mathématiquement, ceci peut être traduit par les contraintes additionnelles suivantes :

$(Cx)_k \geq ((Cx^l)_k + 1)y_s^l - M_k(1 - y_k^l)$. pour $k = 1, 2, ..., p$ et $\sum_{k=1}^{p} y_k^l \geq 1$; $y_k^l \in \{0, 1\}$, pour $k = 1, 2, ..., p$.

où $-M_k$ est la borne inférieure pour toute valeur réalisable de la $k^{ième}$ fonction objectif.

Étapes l. résoudre le problème :

$$(P_l)\begin{cases} \max \lambda^t Cx \\ t.q \\ Ax = b \\ (Cx)_k \geq ((Cx^s)_k + 1)y_k^s - M_k(1 - y_k^s) \\ pour\ \ s = 1, 2, ..., l;\ \ k = 1, 2, ..., p \\ \sum\limits_{k=1}^{p} y_k^s \geq 1;\ \ y_k^s \in \{0, 1\} \\ pour\ \ s = 1, 2, ..., l;\ \ k = 1, 2, ..., p \\ x \geq 0;\ \ x \in \mathbb{Z}^n \end{cases}$$

Pour des problèmes à grandes tailles, l'énumération de toutes les solutions efficaces n'est pas une tache facile, seulement un sous ensemble peut être généré on changeant le problème (P_l) en :

$$(P_l)\begin{cases} \max \lambda^t Cx \\ t.q \\ Ax = b \\ (Cx)_k \geq ((Cx^s)_k + f_k)y_k^s - M_k(1 - y_k^s) \\ pour \;\; s = 1, 2, ..., l; \;\; k = 1, 2, ..., p \\ \sum\limits_{k=1}^{p} y_k^s \geq 1; \;\; y_k^s \in \{0, 1\} \\ pour \;\; s = 1, 2, ..., l; \;\; k = 1, 2, ..., p \\ x \geq 0; \;\; x \in \mathbb{Z}^n \end{cases}$$

où f_k ($f_k \geq 1$, *entier*) représente l'amélioration minimale dans la $k^{ième}$ fonction objectif.

La procédure s'arrête lorsque le problème (P_s) devient impossible à résoudre. À la fin on obtient l'ensemble des solutions efficaces entièrement ou une partie seulement qui intéresse le décideur (selon la valeur de f_k).

2.3.5 Problèmes à variables binaires [1]

Un nombre considérable de problèmes en nombres entiers est formulé comme problèmes à variables qui ne peuvent prendre que deux valeurs 0 et 1 (variables bivalentes), ce type de problèmes est notès $(MOBLP)$ (Multiple Objective Binary Lineair Programming). En plus des difficultés liées aux problèmes en variables entières ou mixtes; dans le cas des problèmes en variables binaires, la principale difficulté provient du grand nombre de variables nécessaires pour modéliser des situations réelles. Cependant certains problèmes présentent des structures particulières. Ils font partie de l'Optimisation Combinatoire. Il est alors possible de tirer profit de l'information liée à cette structure pour faciliter leur résolution. Ces problèmes sont très intéressants et souvent rencontrés dans l'industrie. La formulation de tels problèmes est donné par :

$$(MOBLP)\left\{ \text{“}\max_{x\in S'}\text{”} z_k = c^k x \;;\;\; k = 1, 2, ..., p \right\}$$

où $S' = \{x \in \{0, 1\}^n \mid \;\; Ax \leq b\}$ il est clair que le nombre de solutions peut croître exponentiellement en fonctions de n.

Méthode de Bitran[1]

Cette méthode est spécifique aux problèmes à variables binaires et elle engendre toutes les solutions efficaces.

L'auteur a considéré d'abord le problème suivant :

$(P_{B_1})\left\{ \text{“}\max_{x\in S_1}\text{”} Z_k = c^k x \ ; \ k = 1, 2, ..., p \right\}$; avec $S_1 = B^n\left(\{(0,1)^n\}\right)$ l'ensemble des sommets de l'hypercube unité dans $\mathbb{R}^n$. On a :

$$x \in E(P_{B_1}) \cap S' \Longrightarrow x \in E(MOBLP)$$

Ceci veut dire qu'une solution efficace pour le problème (P_{B_1}) et réalisable pour le problème $(MOBLP)$ est aussi efficace pour ce dernier par contre une solution qui n'est pas efficace pour le problème (P_{B_1}) peut l'être pour le problème $(MOBLP)$.

$$i.e \ \ x \notin E(P_{B_1}) \nRightarrow x \notin E(MOBLP)$$

L'algorithme proposé comporte trois étapes :

Étape1 : Caractériser l'ensemble des solutions efficaces du problème (P_{B_1}).

Étape2 : Parmi toutes ces solutions engendrées, déterminer celles qui sont réalisables pour le problème $(MOBLP)$.

Étape3 : Adjoindre ensuite les autres solutions efficaces de $(MOBLP)$ non engendrées dans l'etape1.

Algorithme

Étape1 : Caractérisation de $E(P_{B_1})$

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

On dit alors que x domine x' dans la direction v^t si et seulement si $x' = x + v^t$. Donc $Cx' \geq Cx$.

Soit $M(v^t)$ l'ensemble des points de S_1 dominés dans la direction v^t par un autre point de S_1.

$$M(v^t) = \left\{x^{t,r} \mid r = 1, 2, ..., R\right\}$$

avec :

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

L'auteur a montré que cet ensemble peut être déterminé par l'ensemble des solutions optimales du problème $\min\limits_{x \in S_1'} v^t x$ où $S_1' = \{x \in \mathbb{R}^n \ \ |0 \leq x_j \leq 1, \ \forall j\}$ est la relaxation linéaire S_1. L'ensemble $E(P_{B_1})$ est déterminé à partir de la résolution successive des problème relaxé, noté (RP_{B_1}) pour différentes directions v^t, $t \in T$. Ce dernier permet de trouver des ensembles $M(v^t)$ puis $E(P_{B_1}) = \overline{\bigcup\limits_{t \in T} M(v^t)}$.

Notons que quelques directions seulement sont examinées.

Remarque 2.3.1 *$\overline{A}$ veut dire le complémentaire de l'ensemble A.*

Étape2 : Détermination de l'ensemble $E_1 = E(P_{B_1}) \cap S$.

Pour cela il suffit de tester l'admissibilité des solutions de $E(P_{B_1})$.

Étape3 : Caractérisation de l'ensemble $E(MOBLP)$.

Pour déterminer l'ensemble des solutions non efficaces dans (P_{B_1}) et efficaces dans $(MOBLP)$ noté par E_2 on utilise la propriété suivante :

$$x \in E_2 \Rightarrow s + v^t \notin S, \forall v^t, t \in T : x \in M(v^t).$$

D'où

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

Méthode de Deckro et Winkofsky[1]

C'est une méthode spécifique aux problèmes à variables binaires et elle est basée sur le concept d'énumération implicite. La procédure est constituée de deux parties.

Première partie : Elle consiste à résoudre successivement et par énumération implicite les p problèmes unicritères suivants :

$$\left\{ \max_{x \in S} c^k x \ \ \forall k \in \{1, 2, ..., p\} \right\}$$

L'ensemble de toutes les solutions réalisables obtenues au cours de la résolution de ces p problèmes sont considérées. Par une comparaison deux à deux, les solutions dominées sont écartées et le reste des solutions forment un ensemble de solutions candidates pour l'efficacité, soit E_1.

Deuxième partie : Dans cette partie on applique simultanément à tous les critères une énumération implicite qui ressemble à la précédente sauf qu'ici le processus de recherche utilise deux concepts : évaluation et direction de préférence.

- Évaluation : Une solution réalisable x est éliminée si elle est dominée par une solution de E_1. Un processus remontant est appliqué dès que toutes les complétions d'une solution partielle non réalisable sont dominées par une solution de E_1.

- Direction de préférence : Considérons une solution partielle x, v est une direction de préférence sur les variables libres de x si et seulement si :

$$c^k(x+v) \geq c^k x,\ k = 1, 2, ..., p \quad \text{et}$$
$$c^k(x+v) > c^k x$$
$$\textit{pour au moins un } k, k \in \{1, 2, ..., p\}$$

- S'il n'existe aucune direction de préférence on commence à remonter. Si non, la solution partielle x est complétée dans la direction de v.

- On actualise l'ensemble E_1 après chaque étape et à la fin on obtient $E_1 = E(MOBLP)$.

2.4 Conclusion

Ces deux premiers chapitres avaient pour objectif de présenter dans un premier temps les principales définitions nécessaires à la présentation des problèmes d'optimisation mono-objectif et multi-objectifs en variables continues et discrètes. Puis différentes problématiques liées aux spécificités du multi-objectifs, comme l'intervention du décideur dans le processus de décision, le choix des méthodes d'optimisation à utiliser ou encore la complexité. De même, les principales méthodes utilisées pour résoudre ces problèmes, de manière exacte, ont été indiquées. En particulier les différences, en terme de difficulté de résolution et de technique employée, découlant du type de variables (continues et discrets), du nombre d'objectifs ou encore de la structure des problèmes, ont été mises en évidence afin de montrer l'importance et la difficulté de la recherche dans le domaine.

Nous nous sommes également astreints à cerner le cadre des études qui seront présentées ci-après, à savoir l'optimisation "A posteriori" (cherchant à générer l'ensemble des solutions efficaces) à l'aide des méthodes exactes.

Chapitre 3

Problème Bicritère d'affectation

3.1 Introduction

Dans certains problèmes d'affectations, plusieurs paramètres et critères sont impliqués dans l'évaluation des divers aspects du problème; par exemple quand il s'agit d'affecter des ouvriers à leurs postes de travail, il est important de considérer la compatibilité et la non compatibilité de l'ouvrier pour un travail, ainsi que la priorité de chaque ouvrier et de chaque poste de travail dans la chaîne de la production; donc plus de critères à gérer et plus de paramètres à prendre en considération, d'où l'extension multi-critères du problème d'affectation. Dans ce chapitre nous formulerons le problème bicritère d'affectation, et nous présenterons quelques méthodes de résolutions existantes dans la littérature.

3.2 Problème multicritère d'affectation

3.2.1 Formulation du Problème.

Soit le problème d'affectation unicritère suivant :

$$(AP)\begin{cases} \min\ z(x) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}x_{ij} \\ \sum\limits_{i=1}^{n} x_{ij} = 1 \quad j = 1,2,...,n \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \quad i = 1,2,...,n \\ \qquad x_{ij} \in \{0,1\} \quad \forall (i,j) \end{cases}$$

Quand il s'agit d'optimiser plusieurs fonctions objectifs en même temps, nous obtenons le problème multicritère d'affectation $(noté\ \ (MOAP))$.

$$(MOAP)\begin{cases} \text{“}\min\text{”} z_k(x) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}^k x_{ij} & k = 1,2,...,n \\ \qquad \sum\limits_{i=1}^{n} x_{ij} = 1 & j = 1,2,...,n \\ \qquad \sum\limits_{j=1}^{n} x_{ij} = 1 & i = 1,2,...,n \\ x_{ij} \in \{0,1\} \quad \forall (i,j) \end{cases}$$

Pour $k = 2$, nous obtenons le problème bicritère d'affectation.

$$(BAP)\begin{cases} \text{“}\min\text{”} z_k(x) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}^k x_{ij} \qquad k = 1,2 \\ \sum\limits_{i=1}^{n} x_{ij} = 1 \quad i = 1,2,...,n \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \quad j = 1,2,...,n \\ \qquad\qquad x_{ij} \in \{0,1\} \quad \forall (i,j) \end{cases}$$

3.2.2 Positionnement du problème

Le problème d'affectation unicritère (AP) est un problème de la programmation linéaire en nombre entier qui peut être résolu comme un programme linéaire dû à l'unimodularité total de la matrice des contraintes; il est considéré comme un problème facile (de la classe $\mathcal{P}$) car des algorithmes permettant de le résoudre de manière exacte en temps polynomial sont connus. Dell'Amico et al [11], donnent des tests comparatifs des différentes implémentations des ces algorithmes. Par contre le problème bicritère ou multicritère d'affectation, comme beaucoup de problème important est un problème $\mathcal{NP}$-difficile (voir [8]). Les premiers papiers sur le problème traitent seulement les solutions efficaces supportées, en utilisant des combinaisons convexes des fonctions objectifs, ou de la programmation par but.

3.2.3 Importance du problème

L'importance du problème d'affectation peut être résumée en deux points essentiels :

- Les applications pratiques du problème d'affectation qui sont nombreuses et inclurent affecter les professeurs aux classes d'école, les locataires aux appartements, les travaux aux machines, etc....

- Le problème d'affectation (AP) surgit comme un sous-problème dans des systèmes de décision, plus complexes (transport, voyageur de commerce, etc...).

3.3 Solution du problème bicritère d'affectation

Contrairement au problème d'affectation unicritère, qui est un problème facile (de la classe $\mathcal{P}$) car on connait un algorithme polynomial pour le résoudre, à savoir la méthode hongroise, pour le problème multicritère d'affectation, comme beaucoup de problèmes de même natures ($\mathcal{NP}$-difficile), il n'existe pas d'algorithmes efficaces pour le résoudre de manière exacte. Résoudre le problème bicritère d'affectation revient à trouver un ensemble de solutions efficaces. Dans la littérature, on trouve quelques algorithmes pour déterminer l'ensemble de toutes les solutions efficaces [19], [21] ou une partie de cet ensemble [9]. Notons que la plupart de méthodes sont de type "primal-dual".

3.3.1 La méthode de Malhotra et al ([9]).

C'est une méthode de type "primal-dual", basée sur le principe d'énumération dans l'espace des critères pour générer l'ensemble des paires non dominées (z_1, z_2) dans l'ordre croissant du premier critère z_1, et décroissant du deuxième z_2, on partant de la solution optimale du premier critère z_1. Pour cela ils proposent de diviser le problème bi-critère en deux problèmes unicritères (P_1) et (P_2) de matrices coûts C^1, C^2 respectivement et de travailler simultanément sur les deux problèmes duaux $(D_1), (D_2)$. À chaque itération la réalisabilité et la non réalisabilité des contraintes duales est utilisée pour déterminer l'arête admissible incidente à la base courante et parmi ces nouvelles bases celle qui fournit la plus petite valeur de z_1 est retenue pour l'itération suivantes.

Notation :

Soit le problème (BAP) suivant :

$$(BAP)\begin{cases} \text{“}\min\text{”} z_k(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^k x_{ij} & k = 1,2 \\ \sum_{i=1}^{n} x_{ij} = 1 & i = 1,2,...,n \\ \sum_{j=1}^{n} x_{ij} = 1 & j = 1,2,...,n \\ x_{ij} = 0,1 & i,j = 1,2,...,n \end{cases}$$

Et les deux problèmes unicritères (P_1), (P_2):

$$(P_1)\begin{cases} \text{“}\min\text{”} z_1(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^1 x_{ij} & \\ \sum_{i=1}^{n} x_{ij} = 1 & i = 1,2,...,n \\ \sum_{j=1}^{n} x_{ij} = 1 & j = 1,2,...,n \\ x_{ij} = 0,1 & i,j = 1,2,...,n \end{cases} \text{ et } (P_2)\begin{cases} \text{“}\min\text{”} z_2(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^2 x_{ij} & \\ \sum_{i=1}^{n} x_{ij} = 1 & i = 1,2,...,n \\ \sum_{j=1}^{n} x_{ij} = 1 & j = 1,2,...,n \\ x_{ij} = 0,1 & i,j = 1,2,...,n \end{cases}$$

Et qu'on associée a leurs relaxations linéaires les deux problèmes duaux $(D1)$, $(D2)$ respectivement suivants :

$$(D_1)\begin{cases} \text{“}\max\text{”} \sum_{i=1}^{n} u_i^1 + \sum_{j=1}^{n} v_j^1 \\ \textit{sous les contraintes} \\ u_i^1 + v_j^1 \leq c_{ij}^1 \quad i,j = 1,2,...,n \\ \textit{avec } u_i^1, v_j^1 \textit{ sans restriction de signes} \forall i, \forall j \end{cases}$$

$$(D_2)\begin{cases} \text{“}\max\text{”} \sum\limits_{i=1}^{n} u_i^2 + \sum\limits_{j=1}^{n} v_j^2 \\ \textit{sous les contraintes} \\ u_i^2 + v_j^2 \leq c_{ij}^2 \quad i,j = 1,2,...,n \\ \textit{avec } u_i^2, v_j^2 \textit{ sans restriction de signes} \forall i, \forall j \end{cases}$$

Notations

On note par D_{ij} ($i,j = 1,2,...,n$) les contraintes de (D_1) tels que :

- $D_{ij} = 0$ si la contrainte D_{ij} est satisfaite comme égalité.
- $D_{ij} = -1$ si la contrainte D_{ij} est satisfaite comme inégalité.
- $D_{ij} = 1$ si la contrainte D_{ij} est violée (non satisfaite).

De même on note D'_{ij} ($i,j = 1,2,...,n$) les contraintes de (D_2) tels que :

- $D'_{ij} = 0$ si la contrainte D'_{ij} est satisfaite comme égalité.
- $D'_{ij} = -1$ si la contrainte D'_{ij} est satisfaite comme inégalité.
- $D'_{ij} = 1$ si la contrainte D'_{ij} est violée.

Soit $X_l^{(p)}$ $(p \in \mathbb{N}, p \geq 1)$ une affectation. On note par :

- $B_l^{(p)}$ les bases de l'affectation $X_l^{(p)}$.
- $S_l^{(p)}$ l'ensemble des indices (i,j) hors base tel que $D_{ij} = -1$ et $D'_{ij} = 1$:

$$S_l^{(p)} = \left\{ (i,j) \notin B_l^{(p)} |\ D_{ij} = -1 \text{ et } D'_{ij} = 1 \right\}$$

Définition 3.3.1 *Soit $X_l^{(p)}$une affectation, et $B_l^{(p)}$une base de $X_l^{(p)}$ tel que $\forall (i,j) \notin B_l^{(p)}$, $D_{ij} \neq 1$.On appelle arête admissible incidente à $X_l^{(p)}$, l'ensemble des solution $\hat{X}_{l(ij)}^{(p)}$ où $(i,j) \notin B_l^{(p)}$ tel que $D_{ij} = -1$ et $D'_{ij} \neq 1$.*

- $\overline{F}^{(p)} = \left\{ (z_1, z_2) \;\middle|\; z_1 = \sum_{i=1}^{n}\sum_{i=1}^{n} c^1_{ij}\hat{x}^{(p)}_{l(ij)},\ z_2 = \sum_{i=1}^{n}\sum_{i=1}^{n} c^2_{ij}\hat{x}^{(p)}_{l(ij)}, (i,j) \in S^{(p)}_l, l = 1, 2, ..., q_p \right\}$, où $\hat{X}^{(p)}_l = (\hat{x}^{(p)}_{l(ij)})$ les affectations appartenant à l'arête admissible incidente à $X^{(p)}_l$.
- $\hat{F}^{(p)} = \overline{F}^{(p)} \cup F^{(p-1)} - \left\{ (z^{(p)}_1, z^{(p)}_2) \right\}$ (*avec* $p \in \mathbb{N}$, $p \geq 2$).
- $F^{(p)} = \left\{ (z_1, z_2) \;\middle|\; (z_1, z_2) \text{ est non dominées dans } \hat{F}^{(p)} \right\}$

La procédure

Initialisation Déterminer la solution optimale du premier critère, en cas de non unicité de la solution (c-à-d il existe plusieurs solutions optimales pour (P_1)) ; celle qui fournit la plus petite valeur de z_2 est prise.

Soit $X^{(1)}_l$ cette solution.

$X^{(1)}_l$ est la première solution efficace générée, et soit $(z^{(1)}_1, z^{(1)}_2)$ les valeurs prise par cette solution.

Itération1

- Déterminer $B^{(1)}_l$ la base correspondante à $X^{(1)}_l$.
- Déterminer l'ensemble $S^{(1)}_l, \overline{F}^{(1)}, F^{(1)}$.
- Identifier la deuxième paire efficace $(z^{(2)}_1, z^{(2)}_2)$ parmi les éléments de $F^{(1)}$, celle correspondante à la plus petite valeur de z_1 et l'affectation correspondante $X^{(2)}_l$, $l = 1, 2, ..., q_p$. Cette solution est retenue pour l'itération suivante.

(p-1)^{ième} **Itération**

- Déterminer $B^{(p-1)}_l$ la base correspondante à l'affectation $X^{(p-1)}_l$, $p \geq 2$ sélectionnée l'itération précédente (i.e. l'affectation qui fournit la plus petite valeur de z_1 et en cas de non unicité celle qui donne la plus petite valeur de z_2).
- Déterminer l'ensemble $S^{(p-1)}_l$, $\overline{F}^{(p-1)}, F^{(p-1)}$.

- Identifier la $p^{ième}$ paire efficace $(z_1^{(p)}, z_2^{(p)})$ parmi les éléments de $F^{(p-1)}$ (celle correspondante à la plus petite valeur de z_1) et l'affectation correspondante X_l^p, $l = 1, 2, ..., q_p$. Cette solution est retenue pour l'itération suivante.

p^{ième} Itération

- Déterminer $B_l^{(p)}$ la base correspondante à $X_l^{(p)}, p \geq 2$.
- Déterminer l'ensemble $S_l^{(p)}$, $\overline{F}^{(p)}, F^{(p)}$.
- Identifier la $(p+1)^{ième}$ paire efficace $(z_1^{(p+1)}, z_2^{(p+1)})$ parmi les éléments de $F^{(p)}$ (celle correspondante à la plus petite valeur de z_1) et l'affectation correspondante $X_l^{(p+1)}$, $l = 1, 2, ..., q_p$.

Test d'arrêt Le processus d'énumération s'arrête dès que $F^{(N)} = \varnothing$ ($N \in \mathbb{N}$).

L'ensemble constitué par les paires engendrées à chaque itération, est l'ensemble des solutions efficaces du problème bi-critères d'affectation $E(BAP)$.

Notons enfin, que les auteurs ne précisent pas quelle méthode ont utilisé pour déterminer les variables duales (u_i, v_j) , $i, j = 1, ..., n$.

3.3.2 Exemple illustratif de Malhotra et al [9]

Soit le problème bi-critères d'affectation définit par les matrices coûts suivantes :

$$C^1 \begin{pmatrix} 5 & 2 & 6 & 3 \\ 2 & 4 & 3 & 1 \\ 3 & 5 & 4 & 8 \\ 6 & 7 & 4 & 5 \end{pmatrix} \quad et\ C^2 \begin{pmatrix} 4 & 2 & 3 & 1 \\ 2 & 5 & 3 & 4 \\ 5 & 3 & 4 & 5 \\ 3 & 4 & 2 & 3 \end{pmatrix}$$

$\left(z_1^{(1)}, z_2^{(1)}\right) = (10, 13)$, $X_1^{(1)} \equiv \{x_{12} = x_{24} = x_{31} = x_{43} = 1, \text{ les autres } x_{ij} = 0\}$.
$\overline{F}^{(1)} = \{(13, 11)\}$.
$F^{(1)} = \overline{F}^{(1)} = \{(13, 11)\}$.
$\left(z_1^{(2)}, z_2^{(2)}\right) = (13, 11)\ et\ X_1^{(2)} \equiv \{x_{12} = x_{21} = x_{33} = x_{44} = 1, \text{ les autres } x_{ij} = 0\}$

$B_1^{(2)} = \{(1,2), (1,4), (2,1), (2,4), (3,3), (4,3), (4,4)\}.$

$S_1^{(2)} = \{(3,2)\}.$

$\overline{F}^{(2)} = \{(14,8)\}.$

$$\begin{aligned} \hat{F}^{(2)} &= \overline{F}^{(2)} \cup \left\{ F^{(1)} - \left\{ \left(z_1^{(2)}, z_2^{(2)} \right) \right\} \right\} = \overline{F}^{(2)} \cup \varnothing. \\ &= \overline{F}^{(2)}. \end{aligned}$$

$F^{(2)} = \{(14,8)\}.$

$\left(z_1^{(3)}, z_2^{(3)} \right) = (14,8)$, et $X_1^{(3)} \equiv \{x_{14} = x_{21} = x_{32} = x_{43} = 1, \text{ les autres } x_{ij} = 0\}.$

$B_1^{(3)} = \{(1,2), (1,4), (2,1), (2,4), (3,2), (4,3), (4,4)\}.$

$S_1^{(3)} = \varnothing.$

$\overline{F}^{(3)} = \varnothing.$

$\hat{F}^{(3)} = \overline{F}^{(3)} \cup \left\{ F^{(2)} - \left\{ \left(z_1^{(3)}, z_2^{(3)} \right) \right\} \right\} = \overline{F}^{(3)} \cup \varnothing = \varnothing.$

$F^{(3)} = \varnothing$

Fin de procédure.

$$A = \{(10,13),\ (13,11),\ (14,8)\}$$

3.3.3 Adaptation de la méthode hongroise [19]

Cette méthode est une concrétisation du principe décrit par Malhotra et al [9] à l'aide de la méthode hongroise pour le problème d'affectation unicritère. Cette adaptation a permis aux auteurs dans [19] de montrer que la méthode de Malhotra donne effectivement l'ensemble complet des solutions efficaces pour quelques exemples, mais elle échoue pour d'autres exemples. Ils donneront un contre exemple.

Comme vu ci-dessus, la méthode [9], propose de déterminer à chaque itération la base admissible correspondante à la solution optimale courante. L'adaptation de la méthode hongroise, nécessite donc, une procédure pour déterminer une base duale admissible correspondante à la solution optimale déterminée par la méthode hongroise, pour cela, [19] propose une procédure pour construire une telle base.

Construction d'une base duale admissible correspondante à une solution optimale obtenue par la méthode hongroise [19].

Initialisation Soit $\tilde{X}$ la solution optimale obtenue par la méthode hongroise et soit :

$$I_1 = \{(i_p, p)|\ p = 1, ..., n\} \text{ les affectations correspondantes.}$$

Rappelons que dans un problème d'affectation, il y'a :

- $2n$ contraintes, $(2n-1)$ contraintes linéairement indépendantes et donc $(2n-1)$ variables de base.
- n^2 variables : $\begin{cases} (2n-1) \text{ variables de base} \\ n^2-(2n-1) \text{ variables hors base.} \end{cases}$
- $(2n-1)$ variables de base : $\begin{cases} n \text{ variables (celles correspondantes aux } n \text{ affectations) sont} \\ \qquad\qquad \text{strictement positives et égales à 1} \\ n-1 \text{ variables égale à 0} \end{cases}$

Déterminer ces $(2n-1)$ variables de base pour la solution optimale $\tilde{X}$ revient à fixer $(n-1)$ variables supplémentaires d'indices (i,j) qui seront de base, en plus des n paires d'indices correspondantes aux n affectations de I_1. Ces de $(2n-1)$ variables doivent déterminées, en plus une base duale admissible étant donné que la solution est optimale, c'est à dire une solution duale $\{u_i, v_j; i, j = 1, 2, ..., n\}$ et donc $2n$ variables duales (u_i, v_j), $i, j = 1, 2, ..., n$, vérifiants (par le théorème des écarts complémentaires) :

$$(1) \begin{cases} (2n-1) \text{ égalités} & u_i + v_j = c_{ij} & \forall (i,j) \in I_B \quad (1,1) \\ \text{avec } |I_B| = 2n-1 & et & I_1 \subset\ I_B \\ n^2-(2n-1) \text{ inégalités} & u_i + v_j \leq c_{ij} & \forall (i,j) \in I_{HB} \quad (2,2) \\ \text{avec } I_{HB} \text{ complémentaire de } I_B \text{ par rapport aux } n^2 \text{paires } (i,j) \end{cases}$$

La méthode hongroise fournit les n paires d'indices I_1, et une solution duale admissible (non nécessairement de base) vérifiant :

$$(2) \begin{cases} n \text{ égalités} \ \ u_i + v_j = c_{ij} & \forall (i,j) \in I_1 & (2.1) \\ n^2 - n \text{ inégalités } u_i + v_j \leq c_{ij} & \forall (i,j) \in I_2 = \{(i,j)|\ \tilde{x}_{ij} = 0\} & (2.2) \end{cases}$$

Procédure de construction

Itération 1 Fixons une variable duale arbitrairement choisit soit v_n à une valeur arbitraire soit 0.

Pour le couple $(i_n, n) \in I_1$ $(i.e\ u_{i_n} + v_n = c_{i_n n})$

$$v_n = 0 \overset{(2.1)}{\Rightarrow} u_{i_n} = c_{i_n n} - v_n$$

(v_n, u_{i_n}) étant fixées, portées dans les contraintes $(2, 2)$ où figurent ces deux variables, on déduit des bornes supérieurs pour les $(2n - 1)$ variables restantes (car on a fixée deux variables parmi $2n$ variables) .

D'où les bornes supérieures suivantes :

$$\begin{cases} v_n\ fix\acute{e}e \rightarrow u_i \leq \overline{u_i} \qquad \forall i \neq i_n & (3.1) \\ u_{i_n} fix\acute{e}e \rightarrow v_j \leq \overline{v_j} \qquad \forall j \neq n & (3.2) \end{cases}$$

Pour déterminer les $2(n - 1)$ variables restantes, on procède comme suit :

- Fixer une variable parmi ces $2(n-1)$ variables arbitrairement, soit $v_{(n-1)}$, et soit α l'écart type positif par rapport à sa borne supérieure, telle que :

$$\begin{cases} v_{(n-1)} = \overline{v}_{(n-1)} - \alpha & (4.1) \\ avec \qquad \alpha \geq 0 & (6.0) \end{cases}$$

La contrainte (2.1) dans laquelle cette variable intervienne fournit la valeur de l'autre variable présente dans cette contrainte :

soit :

$u_{i_{(n-1)}} + v_{(n-1)} = c_{i_{(n-1)}(n-1)} \Rightarrow u_{i_{(n-1)}} = c_{i_{(n-1)}(n-1)} - v_{(n-1)}, avec\ v_{(n-1)} = \overline{v}_{(n-1)} - \alpha$.

D'où

$$u_{i_{(n-1)}} = c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \qquad ((4,2))$$

- Pour que la contrainte (3.1) soit vérifiée par cette variable : $u_{i_{(n-1)}} \leq \ \overline{u}_{i_{(n-1)}}$, il faut que

$$c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \leq \overline{u}_{i_{(n-1)}} \qquad ((5,1))$$

Ce qui nous fournit une borne supérieur pour le paramètre α :

$$\alpha \leq \overline{u}_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} \tag{(6,1)}$$

En portant la valeur (4.1) dans les $(n-2)$ contraintes (2.2) faisant intervenir simultanément les $(n-2)$ variables $u_i(i \neq i_n, i_{(n-1)})$ et la variable $v_{(n-1)}$,

on obtient :

$$u_i \leq c_{i(n-1)} - \overline{v}_{(n-1)} + \alpha \tag{(5,2)}$$

Ces conditions (5.2) portées dans les $(n-2)$ contraintes (2.1) faisant intervenir simultanément $u_{i_J}(i_j \neq i_n, i_{(n-1)})$ et $v_j(j \neq n, n-1)$: $u_{i_j} + v_j = c_{i_j j}$ fournissent une condition sur les variables v_j.

$$v_j \geq c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \alpha \tag{(5.3)}$$

Pour que la contrainte (3.2) soit vérifiée par cette variable il faut que : $c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \alpha \ \leq \overline{v}_j, j \neq n, n-1$, ce qui nous fournit des bornes inférieures sur la valeur du paramètre α

$$\alpha \geq c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \overline{v}_j \tag{(6,2)}$$

De même pour la valeur (4.2) :

En portant cette valeur dans les $(n-2)$ contraintes (2.2) faisant intervenir la variable $u_{i_{(n-1)}}$ et les $(n-2)$ variables $v_j(j \neq n, n-1)$, on aura :

$$v_j \leq c_{i_{(n-1)}j} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)} - \alpha \tag{(5,4)}$$

Ces conditions $(5,4)$ portées dans les $(n-2)$ contraintes (2.1) fournissent une condition sur les variables $u_{i_j}(j \neq n, n-1)$:

$$u_{i_j} \geq c_{i_j j} - c_{i_{(n-1)}j} + c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \qquad ((5,5))$$

Le respect de la contrainte (3.1) :

$u_{i_j} \leq \overline{u}_{i_j} \Rightarrow c_{i_j j} - c_{i_{(n-1)}j} + c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \leq \overline{u}_{i_j}$, fournit une borne supérieur de la valeur du paramètre α :

$$\alpha \leq \overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)}j} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)} \qquad ((6,3))$$

En regroupant les $2(n-1)$ conditions (6) sur α :

$$\begin{cases} \alpha \geq 0 & (6.0) \\ \alpha \geq c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \overline{v}_j & (6.2) \end{cases}$$

et

$$\begin{cases} \alpha \leq \overline{u}_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} & (6.1) \\ \alpha \leq \overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)}j} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} & (6.3) \end{cases}$$

On aura

$$\underline{\alpha} \leq \alpha \leq \overline{\alpha}$$

Avec

$$\begin{cases} \underline{\alpha} = \max(0, c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \overline{v}_j) \qquad \text{\textit{où} } j \neq n, n+1 \ \text{\textit{dans les deux cas}} \\ \overline{\alpha} = \min(\overline{u}_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)}; \ \ \overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)}j} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)}) \end{cases} \qquad (8.2)$$

La fixation du paramètre α à une de ces bornes $\underline{\alpha}$ et $\overline{\alpha}$ entraînera deux conséquences :

1. Les variables $u_{i_{(n-1)}}$ et $v_{(n-1)}$ sont fixées par les relations $(4.1), (4.2)$.

2.

a) Soit une contrainte (2.2) sera fixée à l'égalité. C'est le cas dans les deux situations suivantes :

- $\alpha = \underline{\alpha} = 0$:

  - Par la relation (4.1) : $v_{(n-1)} = \overline{v}_{(n-1)}$ ce qui entraine par la relation (3.2) :

  $\overline{v}_{(n-1)} = \ c_{i_n(n-1)} - u_{i_n}$ que $v_{(n-1)} = \ c_{i_n(n-1)} - u_{i_n}$.

  - D'où une contrainte (2.2) serrée : $u_{i_n} + v_{(n-1)} = c_{i_n(n-1)}$.

- $\alpha = \overline{\alpha} = \overline{u}_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)}$:

  - Par la relation (4.2) $u_{i_{(n-1)}} = c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \ \rightarrow \alpha = u_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)} \rightarrow \ \overline{u}_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} = u_{i_{(n-1)}} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)}$

    D'où $\overline{u}_{i_{(n-1)}} = u_{i_{(n-1)}}$.

    Ce qui entraine par la relation (3.1) $\overline{u}_{i_{(n-1)}} = c_{i_{(n-1)}n} - v_n$ que $u_{i_{(n-1)}} = c_{i_{(n-1)}n} - v_n$; d'où une contrainte(2.2) serrée : $u_{i_{(n-1)}} + v_n = c_{i_{(n-1)}n}$.

b) Soit deux autres variables seront fixées et deux contraintes de (2.2) seront serrées. C'est le cas dans les deux situations suivantes :

- $\alpha = \underline{\alpha} = c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \overline{v}_j$:

  - Par la relation $(5,3)$ *on a que* $v_j \geq c_{i_j j} - c_{i_j(n-1)} + \overline{v}_{(n-1)} - \alpha$ ce qui implique $v_j \geq \overline{v}_j$ alors que par (3.2) on a $v_j \leq \overline{v}_j, d'où\ v_j = \overline{v}_j$.

    $v_j = \overline{v}_j$ entraine par la relation (3.2)($\overline{v_j} = c_{i_n j} - u_{i_n}$): $v_j = c_{i_n j} - u_{i_n}$; d'où une contrainte (2.2) serrée : $u_{i_n} + v_j = c_{i_n j}$.

    Mais par (2.1) on a que $u_{i_j} + v_j = c_{i_j j} \rightarrow u_{i_j} + \overline{v}_j = c_{i_j j} \rightarrow u_{i_j} = c_{i_j j} - \overline{v}_j$; et par (4.1) : $v_{(n-1)} = \overline{v}_{(n-1)} - \alpha \rightarrow v_{(n-1)} = \overline{v}_{(n-1)} - c_{i_j j} + c_{i_j(n-1)} - \overline{v}_{(n-1)} + \overline{v}_j \rightarrow v_{(n-1)} = -c_{i_j j} + c_{i_j(n-1)} + \overline{v}_j$ et dés lors $u_{i_j} + v_{(n-1)} = (c_{i_j j} - \overline{v}_j) + (-c_{i_j j} + c_{i_j(n-1)} + \overline{v}_j) \rightarrow u_{i_j} + v_{(n-1)} = c_{i_j(n-1)}$.d'où une contrainte (2.2) serrée : $u_{i_j} + v_{(n-1)} = c_{i_j(n-1)}$.

- $\alpha = \overline{\alpha} = \overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)}j} - c_{i_{(n-1)}(n-1)} + \overline{v}_{(n-1)}$:

- Par la relation (5.5) on a que $u_{i_j} \geq c_{i_j j} - c_{i_{(n-1)} j} + c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \;\rightarrow u_{i_j} \geq \overline{u}_{i_j}$ alors que par (3.1) on a $u_{i_j} \leq \overline{u}_{i_j}$, *d'où* $u_{i_j} = \overline{u}_{i_j}$.

  $u_{i_j} = \overline{u}_{i_j}$ entraine par la relation (3.1) ($\overline{u}_{i_j} = c_{i_j n} - v_n$) : $u_{i_j} = c_{i_j n} - v_n$; d'où une contrainte (2.2) serrée : $u_{i_j} + v_n = c_{i_j n}$.

  Mais par (2.1) on a que $u_{i_j} + v_j = c_{i_j j} \rightarrow v_j = c_{i_j j} - u_{i_j} \rightarrow v_j = c_{i_j j} - \overline{u}_{i_j}$; et par (4.1) : $u_{i_{(n-1)}} = c_{i_{(n-1)}(n-1)} - \overline{v}_{(n-1)} + \alpha \quad \rightarrow u_{i_{(n-1)}} = \overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)} j}$, et dés lors $u_{i_{(n-1)}} + v_j = (\overline{u}_{i_j} - c_{i_j j} + c_{i_{(n-1)} j}) + (c_{i_j j} - \overline{u}_{i_j}) \rightarrow u_{i_{(n-1)}} + v_j = c_{i_{(n-1)} j}$; d'où une contrainte (2.2) serrée : $u_{i_{(n-1)}} + v_j = c_{i_{(n-1)} j}$.

Notons à la fin que la base obtenue par cette méthode n'est pas unique, à chaque fois qu'on change le choix des variables v_j et des paramètres α, on trouve une base différente mais toujours duale admissible étant donné que le problème d'affectation est fortement dégénéré.

Soit le problème d'affectation unicritère définit par la matrice coût suivante :

$$C = \begin{pmatrix} 5 & 1 & 4 & 7 \\ 6 & 2 & 2 & 6 \\ 2 & 8 & 4 & 4 \\ 3 & 5 & 7 & 1 \end{pmatrix}$$

Initialisation Par application de la méthode hongroise, nous obtenons le tableau optimal suivant :

$$\overline{C} = \begin{pmatrix} 4 & (0) & 3 & 6 \\ 4 & 0 & (0) & 4 \\ (0) & 6 & 2 & 2 \\ 2 & 4 & 6 & (0) \end{pmatrix}$$

Telle que $\widetilde{X} \equiv x_{12} = x_{23} = x_{31} = x_{44} = 1$. Avec $I_1 = \{(1,2), (2,3), (3,1), (4,4)\}$,
$I_2 = \{(1,1), (1,3), (1,4), (2,1), (2,2), (2,4), (3,2), (3,3), (3,4), (4,1), (4,2), (4,3)\}$.
D'où le système suivant :

$$\begin{cases} u_1 + v_2 = 1 \\ u_2 + v_3 = 2 \\ u_3 + v_1 = 2 \\ u_4 + v_4 = 1 \end{cases} \qquad (2.1)$$

$$\begin{cases} u_1 + v_1 \leq 5 \\ u_1 + v_3 \leq 4 \\ u_1 + v_4 \leq 7 \\ u_2 + v_1 \leq 6 \\ u_2 + v_2 \leq 2 \\ u_2 + v_4 \leq 6 \\ u_3 + v_2 \leq 8 \\ u_3 + v_3 \leq 4 \\ u_3 + v_4 \leq 4 \\ u_4 + v_1 \leq 3 \\ u_4 + v_2 \leq 5 \\ u_4 + v_3 \leq 7 \end{cases} \qquad (2.2)$$

Construction d'une base primale duale admissible:

Fixons $v_4 = 0 \Rightarrow u_4 = 1$

$$v_4 = 0 \Rightarrow \begin{cases} u_1 \leq 7 \Rightarrow \overline{u_1} = 7 \\ u_2 \leq 6 \Rightarrow \overline{u_2} = 6 \\ u_3 \leq 4 \Rightarrow \overline{u_3} = 4 \end{cases} \qquad (3.1)$$

$$u_4 = 1 \Rightarrow \begin{cases} v_1 \leq 2 \Rightarrow \overline{v_1} = 2 \\ v_2 \leq 4 \Rightarrow \overline{v_2} = 4 \\ v_3 \leq 6 \Rightarrow \overline{v_3} = 6 \end{cases} \qquad (3.2)$$

Il reste 6 variables à déterminer avec le système suivant :

$$\begin{cases} u_1 + v_2 = 1 \\ u_2 + v_3 = 2 \\ u_3 + v_1 = 2 \end{cases} \qquad (2.1)$$

$$\begin{cases} u_1 + v_1 \leq 5 \\ u_1 + v_3 \leq 4 \\ u_2 + v_1 \leq 6 \\ u_2 + v_2 \leq 2 \\ u_3 + v_2 \leq 8 \\ u_3 + v_3 \leq 4 \end{cases} \qquad (2.2)$$

Soit $v_3 = \overline{v_3} - \alpha = 6 - \alpha \rightarrow v_3 = 6 - \alpha$ (4.1), avec $\alpha \geq 0$ (6.0)

$u_2 + v_3 = 2 \Rightarrow u_2 = 2 - v_3 \Rightarrow u_2 = 2 - (6 - \alpha)$, d'où $u_2 = -4 + \alpha$ (4.2)

$u_2 \leq 6 \Rightarrow -4 + \alpha \leq 6 \Rightarrow \alpha \leq 6 + 4 \rightarrow \alpha \leq 10$ (6.1)

$v_3 = 6 - \alpha$:

de (2.2) : $\begin{cases} u_1 + v_3 \leq 4 \\ u_3 + v_3 \leq 4 \end{cases} \Rightarrow \begin{cases} u_1 \leq 4 - (6 - \alpha) \\ u_3 \leq 4 - (6 - \alpha) \end{cases} \Rightarrow \begin{cases} u_1 \leq -2 + \alpha \\ u_3 \leq -2 + \alpha \end{cases}$ (5.2)

de (2.1) : $\begin{cases} u_1 = 1 - v_2 \\ u_3 = 2 - v_1 \end{cases} \Rightarrow \begin{cases} 1 - v_2 \leq -2 + \alpha \\ 2 - v_1 \leq -2 + \alpha \end{cases} \Rightarrow \begin{cases} v_2 \geq 3 - \alpha \\ v_1 \geq 4 - \alpha \end{cases}$ (5.3)

$\begin{cases} v_2 \leq 4 \\ v_1 \leq 2 \end{cases} \Rightarrow \begin{cases} 4 \geq 3 - \alpha \\ 2 \geq 4 - \alpha \end{cases} \Rightarrow \begin{cases} \alpha \geq -1 \\ \alpha \geq 2 \end{cases}$ (6.2)

$u_2 = -4 + \alpha$:

De (2.2) : $\begin{cases} v_1 \leq 6 - (-4 + \alpha) \\ v_2 \leq 2 - (-4 + \alpha) \end{cases} \Rightarrow \begin{cases} v_1 \leq 10 - \alpha \\ v_2 \leq 6 - \alpha \end{cases}$ (5.4)

De (2.1) : $\begin{cases} v_1 = 2 - u_3 \\ v_2 = 1 - u_1 \end{cases} \Rightarrow \begin{cases} 2 - u_3 \leq 10 - \alpha \\ 1 - u_1 \leq 6 - \alpha \end{cases} \Rightarrow \begin{cases} u_3 \geq -8 + \alpha \\ u_1 \geq -5 + \alpha \end{cases}$ (5.5)

$\begin{cases} u_3 \leq 4 \\ u_1 \leq 7 \end{cases} \Rightarrow \begin{cases} 4 \geq -8 + \alpha \\ 7 \geq -5 + \alpha \end{cases} \Rightarrow \begin{cases} \alpha \leq 12 \\ \alpha \leq 12 \end{cases}$ (6.3)

On obtient les systèmes suivants :

$$\left[\left\{\begin{array}{l}\alpha \leq 10\\ \alpha \leq 12\\ \alpha \leq 12\end{array}\right. et \left\{\begin{array}{l}\alpha \geq 0\\ \alpha \geq -1\\ \alpha \geq 2\end{array}\right.\right] \Rightarrow \left\{\begin{array}{l}\underline{\alpha} = \max(0,-1,2) = 2\\ \overline{\alpha} = \min(10,12,12) = 10\end{array}\right. \Rightarrow 2 \leq \alpha \leq 10$$

Posons $\alpha = 10 \overset{(4.1)}{\Rightarrow} v_3 = -4 \; et \; u_2 = 6$; et la contrainte $u_2 + v_4 \leq 6$ serrée.

Les borne actualisées : $\left\{\begin{array}{l}u_1 \leq 7\\ u_3 \leq 4\end{array}\right. et \left\{\begin{array}{l}v_1 \leq 0\\ v_2 \leq -4\end{array}\right.$

Il reste quatre(4) variables à déterminer avec le système suivant :

$$\left\{\begin{array}{l}u_1 + v_2 = 1\\ u_3 + v_1 = 2\end{array}\right. \qquad (2.1)$$

$$\left\{\begin{array}{l}u_1 + v_1 \leq 5\\ u_3 + v_2 \leq 8\end{array}\right. \qquad (2.2)$$

Posons $v_1 = -\alpha$, avec $\alpha \geq 0$

$v_1 = -\alpha \Rightarrow u_3 = 2 + \alpha \Rightarrow \alpha \leq 2.$

$v_1 = -\alpha \Rightarrow u_1 \leq 5 + \alpha \Rightarrow v_2 \geq -4 - \alpha \Rightarrow \alpha \geq 0.$

$u_3 = 2 + \alpha \Rightarrow v_2 \leq 6 - \alpha \Rightarrow u_1 \geq -5 + \alpha \Rightarrow \alpha \leq 12.$

D'où :

$$0 \leq \alpha \leq 2$$

Soit $\alpha = 2 \Rightarrow v_1 = -2 \; et \; u_3 = 4$ et la contrainte $u_3 + v_4 \leq 4$ serrée.

Il reste, v_2 et u_1 avec les bornes actualisées $u_1 \leq 7 \; et \; v_2 \leq -4$. Fixons v_2 à sa borne supérieure: $i.e \; v_2 = -4. \overset{(2.1)}{\Rightarrow} u_1 = 5$, d'où la contrainte $u_2 + v_2 \leq 2$ serrée.

Nous avons déterminé la solution duale admissible de base suivante :

$$\left(\begin{array}{llll}u_1 = 5 & , u_2 = 6 & , u_3 = 4 & , u_4 = 1\\ v_1 = -2 & , v_2 = -4 & , v_3 = -4 & , v_4 = 0\end{array}\right)$$

Telle que $I_B = I_1 \cup \{(2,2), (2,4), (3,4)\}$.

3.3.4 Application du principe de Malhotra et al [9]

Une itération consiste à déterminer la solution efficace correspondante à la plus faible augmentation de z_1 alors que z_2 doit diminuer, et cela en partant de la solution optimale de premier critère.

Etant donné la dégénérescence de la base, on doit distinguer entre deux types de changements de base classiques :

a) Ceux dégénérés, ne modifiant pas la solution, appelés de $type(a)$.

b) Ceux non dégénérés, produisent une nouvelle solution, appelés de $type(b)$.

Procédure

Initialisation

1- Déterminer la solution optimale $\widetilde{X_1}$ du premier critère z_1 par la méthode hongroise tel que $\tilde{X}_1 \equiv \{\tilde{x}_{ij} = 1 \ si \ (i,j) \in I, \ \tilde{x}_{ij} = 0 \ si \ non\}$, où I est l'ensemble des affectations correspondantes. Et soit $(\tilde{z}_1 = z_1^{(1)}, \tilde{z}_2 = z_2^{(1)})$ les valeurs correspondantes des critères.

Remarque 3.3.1 *Si la solution optimale du premier critère n'est pas unique, celle qui fournit la plus petite valeur de z_2 est choisit.*

2- Déterminer l'ensemble des $(2n-1)$ indices de base optimale associée, ainsi que la solution optimale duale correspondante $\left\{u_i^{(1)}, v_j^{(1)}; i,j = 1,2,...,n\right\}$.

3- Construire la matrice des coûts réduits $\overline{C}^{(1)} = (\overline{c_{ij}}^{(1)})_{i,j=1,2,\ldots,n}$:

$$\text{Où} \begin{cases} \bar{c}_{ij}^{(1)} = c_{ij}^{(1)} - u_i^{(1)} - v_j^{(1)} = 0 & \forall (i,j) \in I_B \\ \bar{c}_{ij}^{(1)} = c_{ij}^{(1)} - u_i^{(1)} - v_j^{(1)} \geq 0 & \forall (i,j) \in I_{HB} \end{cases}$$

Où I_B est l'ensemble des $(2n-1)$ indices de base, et I_{HB} l'ensemble des $n^2 - (2n-1)$ indices hors base.

4- Pour la même base, on détermine la solution optimale duale correspondante au deuxième critère $\left\{u_i^{(2)}, v_j^{(2)}; i,j = 1,2,...,n\right\}$, et on construit la matrice des coûts réduits

$\overline{C}^{(2)} = (\overline{c}_{ij}^{(2)})_{i,j=1,2,\ldots,n} : \left\{ \overline{c}_{ij}^{(2)} = c_{ij}^{(2)} - u_i^{(2)} - v_j^{(2)} = 0 \quad \forall (i,j) \in I_B \right. .$

Pour les indices $(i,j) \in I_{HB}$, il existe des indices pour lesquels $c_{ij}^{(2)} \leq 0$, étant donné que la solution de base n'est pas duale admissible pour le deuxième critère.

Á chaque itération

Déterminer l'ensemble suivant : $(\overline{I} \equiv S_1^{(P)})$

$$\overline{I} = \{(i,j) \in I_{HB} | \overline{c}_{ij}^{(1)} > 0, \overline{c}_{ij}^{(2)} < 0\}$$

Tester tous les changements de base possibles. Ne garder que les changements de base de *type* (b).

Parmi toutes les solutions engendrées, seules les solutions non dominées sont retenues. Celle correspondante à la plus petite valeur de z_1 est prise pour l'itération suivante.

Le même principe et réappliqué à partir de cette nouvelle solution efficace, et ainsi de suite jusqu'aucune autre solution ne puisse être générée.

Cette adaptation, a permis de montrer que le principe, tel qu'il est présenté dans [9] ne fonctionne pas toujours, et de le rectifier par la suite.

Un contre exemple a été construit pour montrer la défaillance du principe.

3.3.5 Contre exemple [19]

$$C^{(1)} \begin{pmatrix} 5 & 1 & 4 & 7 \\ 6 & 2 & 2 & 6 \\ 2 & 8 & 4 & 4 \\ 3 & 5 & 7 & 1 \end{pmatrix} \quad et \ C^{(2)} \begin{pmatrix} 3 & 6 & 4 & 2 \\ 1 & 3 & 8 & 3 \\ 5 & 2 & 2 & 3 \\ 4 & 2 & 3 & 5 \end{pmatrix}$$

$\left(z_1^{(1)}, z_2^{(1)}\right) = (6, 24)\,, X_1^{(1)} = \{x_{12} = x_{23} = x_{31} = x_{44} = 1, \text{les autres } x_{ij} = 0\}\,.$

Les tableaux des coûts réduits :

$$\overline{C}^{(1)} = \begin{pmatrix} 2 & (0) & 3 & 2 \\ 2 & 0 & (0) & 0 \\ (0) & 8 & 4 & 0 \\ 4 & 8 & 10 & (0) \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} -5 & (0) & -7 & -4 \\ -4 & 0 & (0) & 0 \\ (0) & -1 & -6 & 0 \\ -3 & -3 & -7 & (0) \end{pmatrix}$$

$I_B = \{(1,2), (2,2), (2,3), (2,4), (3,1), (3,4), (4,4)\}$.

$I_{HB} = \{(1,1)\,(1,3), (1,4), (2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$.

$\overline{I} = \{(1,1), (1,3), (1,4), (2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$.

Seules les affectations $(1,3), (4,1), (4,3)$ permettent des changements de base de *type* (b); elles conduisent respectivement aux solutions

$X_3 : x_{13} = x_{22} = x_{31} = x_{44} = 1$ de valeur $(9,17)$.

$X_4 : x_{12} = x_{23} = x_{34} = x_{41} = 1$ de valeur $(10,21)$.

$X_6 : x_{12} = x_{24} = x_{31} = x_{43} = 1$ de valeur $(16,17)$.

Seule la solution X_3 de valeur $(9,17)$ est non dominée.

$$\overline{C}^{(1)} = \begin{pmatrix} -1 & -3 & (0) & -1 \\ 2 & (0) & 0 & 0 \\ (0) & 8 & 4 & 0 \\ 4 & 8 & 10 & (0) \end{pmatrix}, \overline{C}^{(2)} = \begin{pmatrix} 2 & 7 & (0) & 3 \\ -4 & (0) & 0 & 0 \\ (0) & -1 & -6 & 0 \\ -3 & -3 & -7 & (0) \end{pmatrix}$$

$I_B = \{(1,3), (2,2), (2,3), (2,4), (3,1), (3,4), (4,4)\}$.

$I_{HB} = \{(1,1)\,(1,2), (1,4), (2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$.

$\overline{I} = \{(2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$

Seules les affectations $(4,1), (4,2)$ permettent des changements de base de *type* (b); elles conduisent respectivement aux solutions

$X_5 : x_{13} = x_{22} = x_{34} = x_{41} = 1$ de valeur $(13,14)$.

$X_7 : x_{13} = x_{24} = x_{31} = x_{42} = 1$ de valeur $(17,14)$.

Ces deux solutions ne sont retenues puisqu'elles sont dominées.

3.3.6 Alternative proposée par Ulungu [19]

Contrairement à ce qui est fait par Malhotra et al [9], cette méthode propose de tester tous les changements de bases même ceux avec des variables hors base qui, si elles rentrent dans la base ne modifie pas la solution; en faisant rentrer deux variables simultanément dans la base y compris celles qui diminuent la valeur de z_1 et augmente la valeur de z_2.

Une liste L conservera les solutions engendrées et à chaque itération seuls les changements de base qui conduisent à des solutions non dominées par les solutions de L sont acceptés et garder pour l'itération suivante.

La procédure

Soit le problème (BAP).

$$(BAP)\begin{cases} \text{“}\min\text{”} z_k(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^k x_{ij} & k=1,2 \\ \sum_{i=1}^{n} x_{ij} = 1 & i=1,2,...,n \\ \sum_{j=1}^{n} x_{ij} = 1 & j=1,2,...,n \\ x_{ij} = 0,1 & i,j=1,2,...,n \end{cases}$$

qu'on associée à sa relaxation linéaire les deux problèmes duaux suivants :

$$\left(\begin{cases} \text{“}\max\text{”}(\sum_{i=1}^{n} u_i^k + \sum_{j=1}^{n} v_j^k) \\ u_i^k + v_j^k \leq c_{ij}^k \\ u_i^k, v_j^k \ \textit{sans restriction de signe} \forall i \ \textit{et} \ \forall j \end{cases}\right) \quad k=1,2$$

Initialisation

1- Déterminer la solution optimale $\tilde{X}_1$ du premier critère z_1 par la méthode hongroise tel que $\tilde{X}_1 \equiv \{\tilde{x}_{ij} = 1 \ si \ (i,j) \in I, \ \tilde{x}_{ij} = 0 \ si \ non\}$, où I est l'ensemble des affectations correspondantes. Et soit $(\tilde{z}_1 = z_1^{(1)}, \tilde{z}_2 = z_2^{(1)})$ les valeurs correspondantes des critères.

Remarque 3.3.2 *Si la solution optimale du premier critère n'est pas unique, celle qui fournit la plus petite valeur de z_2 est choisit.*

2- Déterminer l'ensemble des $(2n-1)$ indices de base optimale associée, ainsi que la solution optimale duale correspondante $\left\{u_i^{(1)}, v_j^{(1)}; i,j = 1,2,...,n\right\}$.

3- Construire la matrice des coûts réduits $\overline{C}^{(1)} = (\overline{c_{ij}}^{(1)})_{i,j=1,2,...,n}$:

$$\text{Où} \begin{cases} \overline{c_{ij}}^{(1)} = c_{ij}^{(1)} - u_i^{(1)} - v_j^{(1)} = 0 \ \ \forall (i,j) \in I_B \\ \overline{c_{ij}}^{(1)} = c_{ij}^{(1)} - u_i^{(1)} - v_j^{(1)} \geq 0 \ \ \forall (i,j) \in I_{HB} \end{cases}$$

Où I_B est l'ensemble des $(2n-1)$ indices de base, et I_{HB} l'ensemble des $n^2 - (2n-1)$ indices hors base.

4- Pour la même base, on détermine la solution optimale duale correspondante au deuxième critère$\left\{u_i^{(2)}, v_j^{(2)}; i,j = 1,2,...,n\right\}$, et on construit la matrice des coûts réduits

$\overline{C}^{(2)} = (\overline{c_{ij}}^{(2)})_{i,j=1,2,\ldots,n} : \left\{\overline{c_{ij}}^{(2)} = c_{ij}^{(2)} - u_i^{(2)} - v_j^{(2)} = 0 \quad \forall (i,j) \in I_B\right.$. Pour les indices $(i,j) \in I_{HB}$, il existe des indices pour lesquels $\overline{c_{ij}}^{(2)} \leq 0$, étant donné que la solution de base n'est pas duale admissible pour le deuxième critère.

Itération1 Soit $(\tilde{z}_1, \tilde{z}_2)$ les valeurs prises par la solution courante $(\widetilde{X_1})$.

Étape1. Déterminer les deux ensembles suivants :

$$\overline{I} = \{(i,j) \in I_{HB} | \overline{c_{ij}}^{(1)} > 0,\ \overline{c_{ij}}^{(2)} < 0\}.$$

$$\overline{I^*} = \{(i,j) \in I_{HB} | \overline{c_{ij}}^{(1)} < 0,\ \overline{c_{ij}}^{(2)} > 0\}.$$

Étape2. Tester tous les changements de base possibles. Pour cela appelons :

$$\begin{cases} I_a(I_a^*) \text{ les affectations de} \overline{I}(\overline{I^*}) \text{ qui conduisent à un changement de base de } type(a). \\ \qquad\qquad\qquad\qquad \text{et} \\ I_b(I_b^*) \text{ celles de qui produisent un changement de } type(b). \end{cases}$$

a) Pour les changements de base de $(i,j) \in I_b \cup I_b^*$, les valeurs des critères deviennent :

$$\begin{cases} z_1 = \tilde{z}_1 + \overline{c_{ij}}^{(1)} \\ z_2 = \tilde{z}_2 + \overline{c_{ij}}^{(2)} \end{cases} si (i,j) \in I_b \cup I_b^*$$

Et soit (i_b, j_b) l'affectation de $I_b \cup I_b^*$ fournissant le couple (z_1, z_2) non dominée par les solution de L.

b) Pour les changements de base de $(i,j) \in I_a \cup I_a^*$, on distingue deux cas :

b.1) $(i,j) \in I_a$:

L'imposition de $(i,j) \in I_a$ ne modifie pas la valeur de $(\tilde{z}_1, \tilde{z}_2)$, pour cela, on est obligé de faire rentrer une deuxième variable permettant d'effectuer en rentrant dans la base avec (i,j) un changement de $type(b)$. Mais on peut pas tester toutes les variables, pour cela on applique la procédure du choix suivante :

Procédure du choix. Appelons $(l(i,j), p(i,j))$ la variable candidate à rentrer simultanément dans la base avec (i,j) :

Choix de $(l(i,j), p(i,j))$

On note par :

- I_{ij} le sous ensemble d'affectation de $I_{HB} \setminus \{(i,j)\}$ permettant d'effectuer, en rentrant dans la base avec (i,j) un changement de $type(b)$.
- $\overline{I_{ij}} = \left\{(l,p) \in I_{ij} \,\middle|\, \overline{c}_{ij}^{(1)} + \overline{c}_{lp}^{(1)} \geq 0,\ \overline{c}_{ij}^{(2)} + \overline{c}_{lp}^{(2)} \leq 0\right\}$, avec $\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}_{ij}^{(1)} + \overline{c}_{lp}^{(1)} \\ z_2 = \tilde{z}_2 + \overline{c}_{ij}^{(2)} + \overline{c}_{lp}^{(2)} \end{pmatrix}$.
- Parmi toutes les affectations $(l,p) \in \overline{I_{ij}}$, celle qui réalise le $\min\limits_{(l,p)\in\overline{I_{ij}}} (\overline{c}_{ij}^{(1)} + \overline{c}_{lp}^{(1)})$ est choisit pour rentrer dans la base avec (i,j), avec $\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}_{ij}^{(1)} + \overline{c}_{lp}^{(1)} \\ z_2 = \tilde{z}_2 + \overline{c}_{ij}^{(2)} + \overline{c}_{lp}^{(2)} \end{pmatrix}$. En cas d'ex aequo(égalité), sélectionner celle de plus petite valeur de $(\overline{c}_{ij}^{(2)} + \overline{c}_{lp}^{(2)})$.

L'introduction du couple de variables $((i,j), (l(i,j), p(i,j)))$ induit les valeurs :

$$\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}_{ij}^{(1)} + \overline{c}_{l(i,j),p(i,j)}^{(1)} \\ z_2 = \tilde{z}_2 + \overline{c}_{ij}^{(2)} + \overline{c}_{l(i,j),p(i,j)}^{(2)} \end{pmatrix}.$$

b.2) $(i,j) \in I_a^*$: le même raisonnement

Procédure du choix Appelons $(l^*(i,j), p^*(i,j))$ la variable candidate à rentrer simultanément dans la base avec (i,j).

Choix de $(l^*(i,j), p^*(i,j))$

On note par :

- I^*_{ij} le sous ensemble d'affectations de $I_{HB} \setminus \{(i,j)\}$ permettant d'effectuer, en rentrant dans la base avec (i,j) un changement de type(b).

- $\overline{I^*_{ij}} = \left\{(l^*,p^*) \in I^*_{ij} \mid \overline{c}^{(1)}_{ij} + \overline{c}^{(1)}_{l^*p^*} \leq 0,\ \overline{c}^{(2)}_{ij} + \overline{c}^{(2)}_{l^*p^*} \geq 0\right\}$, avec $\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}^{(1)}_{ij} + \overline{c}^{(1)}_{l^*p^*} \\ z_2 = \tilde{z}_2 + \overline{c}^{(2)}_{ij} + \overline{c}^{(2)}_{l^*p^*} \end{pmatrix}$.

- Parmi toutes les affectations $(l^*,p^*) \in \overline{I^*_{ij}}$, celle qui réalise le $\min\limits_{(l^*,p^*)\in\overline{I^*_{ij}}} (\overline{c_{ij}}^{(1)} + \overline{c}^{(1)}_{l^*p^*})$ est choisit pour rentrer dans la base avec (i,j), avec $\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}^{(1)}_{ij} + \overline{c}^{(1)}_{l^*p^*} \\ z_2 = \tilde{z}_2 + \overline{c}^{(2)}_{ij} + \overline{c}^{(2)}_{l^*p^*} \end{pmatrix}$. En cas d'ex aequo, sélectionner celle de plus petite valeur de $(\overline{c}^{(2)}_{ij} + \overline{c}^{(2)}_{l^*p^*})$.

L'introduction du couple de variables $((i,j),(l^*(i,j),p^*(i,j)))$ induit les valeurs :

$$\begin{pmatrix} z_1 = \tilde{z}_1 + \overline{c}^{(1)}_{ij} + \overline{c}^{(1)}_{l^*(i,j),p^*(i,j)} \\ z_2 = \tilde{z}_2 + \overline{c}^{(2)}_{ij} + \overline{c}^{(2)}_{l^*(i,j),p^*(i,j)} \end{pmatrix}.$$

Parmi toutes les nouvelles valeurs de (z_1, z_2) engendrées, soit par **a)**, soit par **b),** est sélectionné la paire correspondante à la plus petite valeur de z_1. Le changement de base correspondant conduit à une nouvelle solution introduite dans L. Cette même solution est retenue pour l'itération suivante.

Et ainsi de suite à chaque itération on détermine les ensembles $\overline{I}$, $\overline{I^*}$, et les affectations I_a (I^*_a), I_b (I^*_b). Déterminer l'affectation (i_b, j_b), les sous ensembles $I_{ij}, (I^*_{ij})$, $\overline{I_{ij}}(\overline{I^*_{ij}})$ et choisir les affectations $(l(i,j),p(i,j))$, $((l^*(i,j),p^*(i,j)))$, et parmi toutes les nouvelles valeurs engendrées, choisir celle qui corresponde à la plus faible valeur de z_1.

Test d'arrêt Le processus s'arrête pour l'une des possibilités suivantes :

- Soit on trouve une solution dominée par les solutions de L.

- Soit l'ensemble I_b (I^*_b) est vide et pour toutes les affectations $(i,j) \in I_a \cup I^*_a$, les sous ensembles I^*_{ij} et $\overline{I^*_{ij}}$ sont vides.

La liste L ainsi construite, constitue l'ensemble des solutions efficaces du problème bicritères d'affectation $E(BAP)$.

Exemple 3.3.1

$$C^{(1)}\begin{pmatrix} 5 & 1 & 4 & 7 \\ 6 & 2 & 2 & 6 \\ 2 & 8 & 4 & 4 \\ 3 & 5 & 7 & 1 \end{pmatrix} \quad et \ C^{(2)}\begin{pmatrix} 3 & 6 & 4 & 2 \\ 1 & 3 & 8 & 3 \\ 5 & 2 & 2 & 3 \\ 4 & 2 & 3 & 5 \end{pmatrix}$$

Initialisation

- $\begin{pmatrix} u_1 = 5, \ v_1 = -2 \\ u_2 = 6, \ v_2 = -4 \\ u_3 = 4, \ v_3 = -4 \\ u_4 = 1, \ v_4 = 0 \end{pmatrix}$

$\tilde{X}_1 \equiv x_{12} = x_{23} = x_{31} = x_{44} = 1$, avec $(z_1 = 6, \ z_2 = 24)$.

$I_B = \{(1,2), (2,2), (2,3), (2,4), (3,1), (3,4), (4,4)\}$.

$I_{HB} = \{(1,1)(1,3), (1,4), (2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$.

Tableaux des coûts réduits:

$\overline{C}^{(1)}$:

$\overline{c}_{12}^{(1)} = \overline{c}_{22}^{(1)} = \overline{c}_{23}^{(1)} = \overline{c}_{24}^{(1)} = \overline{c}_{31}^{(1)} = \overline{c}_{34}^{(1)} = \overline{c}_{44}^{(1)} = 0$

$\overline{c}_{11}^{(1)} = c_{11}^{(1)} - u_1^{(1)} - v_1^{(1)} = 5 - 5 + 2 = 2$, $\overline{c}_{13}^{(1)} = c_{13}^{(1)} - u_1^{(1)} - v_3^{(1)} = 4 - 5 + 4 = 3$

$\overline{c}_{14}^{(1)} = c_{14}^{(1)} - u_1^{(1)} - v_4^{(1)} = 7 - 5 - 0 = 2$, $\overline{c}_{21}^{(1)} = c_{21}^{(1)} - u_2^{(1)} - v_1^{(1)} = 6 - 6 + 2 = 2$

$\overline{c}_{32}^{(1)} = c_{32}^{(1)} - u_3^{(1)} - v_2^{(1)} = 8 - 4 + 4 = 8$, $\overline{c}_{33}^{(1)} = c_{33}^{(1)} - u_3^{(1)} - v_3^{(1)} = 4 - 4 + 4 = 4$

$\overline{c}_{41}^{(1)} = c_{41}^{(1)} - u_4^{(1)} - v_1^{(1)} = 3 - 1 + 2 = 4$, $\overline{c}_{42}^{(1)} = c_{42}^{(1)} - u_4^{(1)} - v_2^{(1)} = 5 - 1 + 4 = 8$

$$\overline{c}_{43}^{(1)} = c_{43}^{(1)} - u_4^{(1)} - v_3^{(1)} = 7 - 1 + 4 = 10$$

$\overline{C}^{(2)}$:

$\overline{c}_{12}^{(2)} = \overline{c}_{22}^{(2)} = \overline{c}_{23}^{(2)} = \overline{c}_{24}^{(2)} = \overline{c}_{31}^{(2)} = \overline{c}_{34}^{(2)} = \overline{c}_{44}^{(2)} = 0$

$\overline{c}_{11}^{(2)} = c_{11}^{(2)} - u_1^{(2)} - v_1^{(2)} = 3 - 6 - 2 = -5, \overline{c}_{13}^{(2)} = c_{13}^{(2)} - u_1^{(2)} - v_3^{(2)} = 4 - 6 - 5 = -7$

$\overline{c}_{14}^{(2)} = c_{14}^{(2)} - u_1^{(2)} - v_4^{(2)} = 2 - 6 - 0 = -4, \overline{c}_{21}^{(2)} = c_{21}^{(2)} - u_2^{(2)} - v_1^{(2)} = 1 - 3 - 2 = -4$

$\overline{c}_{32}^{(2)} = c_{32}^{(2)} - u_3^{(2)} - v_2^{(2)} = 2 - 3 - 0 = -1, \overline{c}_{33}^{(2)} = c_{33}^{(2)} - u_3^{(2)} - v_3^{(2)} = 2 - 3 - 5 = -6$

$\overline{c}_{41}^{(2)} = c_{41}^{(2)} - u_4^{(2)} - v_1^{(2)} = 4 - 5 - 2 = -3, \overline{c}_{42}^{(2)} = c_{42}^{(2)} - u_4^{(2)} - v_2^{(2)} = 2 - 5 - 0 = -3$

$$\overline{c}_{43}^{(2)} = c_{43}^{(2)} - u_4^{(2)} - v_3^{(2)} = 3 - 5 - 5 = -7$$

D'où

$\tilde{X}_1 \equiv x_{12} = x_{23} = x_{31} = x_{44} = 1$, avec $(z_1 = 6,\ z_2 = 24)$.

$$\overline{C}^{(1)} = \begin{pmatrix} 2 & (0) & 3 & 2 \\ 2 & 0 & (0) & 0 \\ (0) & 8 & 4 & 0 \\ 4 & 8 & 10 & (0) \end{pmatrix},\ \overline{C}^{(2)} = \begin{pmatrix} -5 & (0) & -7 & -4 \\ -4 & 0 & (0) & 0 \\ (0) & -1 & -6 & 0 \\ -3 & -3 & -7 & (0) \end{pmatrix}$$

Itération 1.

- $\overline{I} = \{(1,1),(1,3),(1,4),(2,1),(3,2),(3,3),(4,1),(4,2),(4,3)\}$.
- $\overline{I^*} = \varnothing \Rightarrow I_b^* = I_a^* = \varnothing$.
- $I_a = \{(1,1),(1,4),(2,1),(3,2),(3,3),(4,2)\}$.
- $I_b = \{(1,3),(4,1),(4,3)\}$.

1.a) $(i,j) \in I_b^* \cup I_b$:

- $$\left.\begin{array}{l} (i,j) \equiv (1,3) \Rightarrow \begin{cases} z_1 = 6+3 = 9 \\ z_2 = 24-7 = 17 \end{cases} \\ (i,j) \equiv (4,1) \Rightarrow \begin{cases} z_1 = 6+4 = 10 \\ z_2 = 24-3 = 21 \end{cases} \\ (i,j) \equiv (4,3) \Rightarrow \begin{cases} z_1 = 6+10 = 16 \\ z_2 = 24-7 = 17 \end{cases} \end{array}\right\} \Rightarrow (i_b, j_b) \equiv (1,3)\ \mathit{avec}\ (z_1, z_2) = (9,17).$$

1.b) $(i,j) \in I_a(I_a^* = \varnothing)$

- $(i,j) \equiv (1,1) \Rightarrow I_{ij} = \{(2,3),(3,3)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (3,3) \Rightarrow \begin{cases} z_1 = 6+2+4 = 12 \\ z_2 = 24-5-6 = 13 \end{cases}$$

- $(i,j) \equiv (1,4) \Rightarrow I_{ij} = \{(4,2),(4,3)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (4,2) \Rightarrow \begin{cases} z_1 = 6+2+8 = 16 \\ z_2 = 24-4-3 = 17 \end{cases}$$

- $(i,j) \equiv (2,1) \Rightarrow I_{ij} = \{(2,3), (4,3)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (2,3) \Rightarrow \begin{cases} z_1 = 6+2+4 = 12 \\ z_2 = 24-4-6 = 14 \end{cases}$$

- $(i,j) \equiv (3,2) \Rightarrow I_{ij} = \{(1,1)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (1,1) \Rightarrow \begin{cases} z_1 = 6+8+2 = 16 \\ z_2 = 24-1-5 = 18 \end{cases}$$

- $(i,j) \equiv (3,3) \Rightarrow I_{ij} = \{(1,1), (2,1)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (1,1) \Rightarrow \begin{cases} z_1 = 6+4+2 = 12 \\ z_2 = 24-6-5 = 13 \end{cases}$$

- $(i,j) \equiv (4,2) \Rightarrow I_{ij} = \{(1,1), (1,3), (1,4)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (1,1) \Rightarrow \begin{cases} z_1 = 6+8+2 = 16 \\ z_2 = 24-3-5 = 16 \end{cases}$$

Par comparaison des valeurs de (z_1, z_2) ; la variable x_{13} rentre seule dans la base.

$X_3 \equiv x_{13} = x_{22} = x_{31} = x_{44} = 1$, $(z_1, z_2) = (9, 17)$.

$$L = \{(6,24), (9,17)\}$$

Itération 2.

$$\overline{C}^{(1)} = \begin{pmatrix} -1 & -3 & (0) & -1 \\ 2 & 0 & (0) & 0 \\ (0) & 8 & 4 & 0 \\ 4 & 8 & 10 & (0) \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} 2 & 7 & (0) & 3 \\ -4 & 0 & (0) & 0 \\ (0) & -1 & -6 & 0 \\ -3 & -3 & -7 & (0) \end{pmatrix}$$

- $\overline{I} = \{(2,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}.$
- $\overline{I^*} = \{(1,1), (1,2), (1,4)\}$
- $I_a = \{(2,1), (3,2), (3,3), (4,3)\}.$
- $I_b = \{(4,1), (4,2)\}.$
- $I_a^* = \{(1,1), (1,4)\}$
- $I_b^* = \{(1,2)\}$

2.a) $(i,j) \in I_b^* \cup I_b$:

- $$\left.\begin{array}{l} (i,j) \equiv (4,1) \Rightarrow \begin{cases} z_1 = 9+4 = 13 \\ z_2 = 17-3 = 14 \end{cases} \\ (i,j) \equiv (4,2) \Rightarrow \begin{cases} z_1 = 9+8 = 17 \\ z_2 = 17-3 = 14 \end{cases} \\ (i,j) \equiv (1,2) \Rightarrow \begin{cases} z_1 = 9-3 = 6 \\ z_2 = 17+7 = 24 \end{cases} \end{array}\right\} \Rightarrow (i_b, j_b) \equiv (4,1) \ avec \ (z_1, z_2) = (13, 14).$$

2.b) $(i,j) \in I_a \cup I_a^*$:

2.b.1. $(i,j) \in I_a$

- $(i,j) \equiv (2,1) \Rightarrow I_{ij} = \{(3,2)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (3,2) \Rightarrow \begin{cases} z_1 = 9+2+8 = 19 \\ z_2 = 17-4-1 = 12 \end{cases}$$

- $(i, j) \equiv (3, 2) \Rightarrow I_{ij} = \{(2, 1)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (2,1) \Rightarrow \begin{cases} z_1 = 9 + 8 + 2 = 19 \\ z_2 = 17 - 4 - 1 = 12 \end{cases}$$

- $(i, j) \equiv (3, 3) \Rightarrow I_{ij} = \{(1, 1)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (1,1) \Rightarrow \begin{cases} z_1 = 9 + 4 - 1 = 12 \\ z_2 = 17 - 6 + 2 = 13 \end{cases}$$

- $(i, j) \equiv (4, 3) \Rightarrow I_{ij} = \{(1, 1), (1, 2), (1, 4)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (1,2) \Rightarrow \begin{cases} z_1 = 9 + 10 - 3 = 16 \\ z_2 = 17 - 7 + 7 = 17 \end{cases}$$

2.b.2. $(i, j) \in I_a^*$

- $(i, j) \equiv (1, 1) \Rightarrow I_{ij}^* = \{(3, 2), (3, 3)\} \Rightarrow \overline{I_{ij}^*} = \varnothing.$

- $(i, j) = (1, 4) \Rightarrow I_{ij}^* = \{(4, 2), (4, 3)\} \Rightarrow \overline{I_{ij}^*} = \varnothing.$

Par comparaison des valeurs de (z_1, z_2) ; les variables x_{33} et x_{11} rentrent simultanément dans la base.

$X_4 \equiv x_{11} = x_{22} = x_{33} = x_{44} = 1, (z_1, z_2) = (12, 13)$.

$$L = \{(6, 24), (9, 17), (12, 13)\}$$

Itération 3.

$$\overline{C}^{(1)} = \begin{pmatrix} (0) & -3 & 0 & -1 \\ 3 & (0) & 0 & 0 \\ -3 & 4 & (0) & -4 \\ 5 & 8 & 10 & (0) \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} (0) & 7 & 0 & 3 \\ -6 & (0) & 0 & 0 \\ 4 & 5 & (0) & 6 \\ -5 & -3 & -7 & (0) \end{pmatrix}$$

- $\overline{I} = \{(2,1), (4,1), (4,2), (4,3)\}$.
- $\overline{I^*} = \{(1,2), (1,4), (3,1), (3,4)\}$.
- $I_a = \{(2,1), (4,1), (4,3)\}$.
- $I_b = \{(4,2)\}$.
- $I_a^* = \{(1,2), (1,4), (4,3)\}$.
- $I_b^* = \{(3,1)\}$.

3.a) $(i,j) \in I_b^* \cup I_b$:

- $$\left.\begin{array}{l} (i,j) \equiv (4,2) \Rightarrow \left\{\begin{array}{l} z_1 = 12 + 8 = 20 \\ z_2 = 13 - 3 = 10 \end{array}\right. \\ (i,j) \equiv (3,1) \Rightarrow \left\{\begin{array}{l} z_1 = 12 - 4 = 9 \\ z_2 = 13 + 3 = 17 \end{array}\right. \end{array}\right\} \Rightarrow (i_b, j_b) \equiv (3,1) \text{ solution dominée}$$

3.b) $(i,j) \in I_a \cup I_a^*$:

3.b.1. $(i,j) \in I_a$

- $(i,j) \equiv (2,1) \Rightarrow I_{ij} = \{(1,2), (3,2)\} \Rightarrow \overline{I_{ij}} = \varnothing$.
- $(i,j) \equiv (4,1) \Rightarrow I_{ij} = \{(1,2), (1,4)\} \Rightarrow \overline{I_{ij}} = \{(1,4)\}$.

$$(l(i,j), p(i,j)) \equiv (1,4) \Rightarrow \left\{\begin{array}{l} z_1 = 12 + 5 - 1 = 16 \\ z_2 = 13 - 5 + 3 = 11 \end{array}\right.$$

- $(i,j) \equiv (4,3) \Rightarrow I_{ij} = \{(3,1), (3,4)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (3,4) \Rightarrow \left\{\begin{array}{l} z_1 = 12 + 10 - 4 = 18 \\ z_2 = 13 - 7 + 6 = 12 \end{array}\right.$$

3.b.2 $(i,j) \in I_a^*$

- $(i,j) \equiv (1,2) \Rightarrow I_{ij}^* = \{(2,1),(3,1),(4,1)\} \Rightarrow \overline{I_{ij}^*} = \{(2,1),(3,1)\}$.

$$(l^*(i,j), p^*(i,j)) \equiv (3,1) \Rightarrow \begin{cases} z_1 = 12 + -3 - 3 = 6 \\ z_2 = 13 + 7 + 4 = 24 \end{cases}$$

- $(i,j) \equiv (1,4) \Rightarrow I_{ij}^* = \{(4,1)\} \Rightarrow \overline{I_{ij}^*} = \varnothing$.
- $(i,j) \equiv (3,4) \Rightarrow I_{ij}^* = \{(4,1),(4,2),(4,3)\} \Rightarrow \overline{I_{ij}^*} = \varnothing$.

Par comparaison des valeurs de (z_1, z_2) ; x_{41} et x_{14} rentrent simultanément dans la base.
$X_4 \equiv x_{14} = x_{22} = x_{33} = x_{41} = 1$, $(z_1, z_2) = (16, 11)$.

$$L = \{(6,24),\ (9,17),\ (12,13),\ (16,11)\}$$

Itération 4.

$$\overline{C}^{(1)} = \begin{pmatrix} 0 & -3 & 0 & (0) \\ 3 & (0) & 0 & 1 \\ -3 & 4 & (0) & -3 \\ (0) & 3 & 5 & -4 \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} 0 & 7 & 0 & (0) \\ -6 & (0) & 0 & -3 \\ 4 & 5 & (0) & 3 \\ (0) & 2 & -2 & 2 \end{pmatrix}$$

- $\overline{I} = \{(2,1),(2,4),(4,3)\}$.
- $\overline{I^*} = \{(1,2),(3,1),(3,4),(4,4)\}$.
- $I_a = \{(2,1),(2,4),(4,3)\}$.
- $I_b = \varnothing$.
- $I_a^* = \{(1,2),(3,1)\}$.
- $I_b^* = \{(3,4),(4,4)\}$.

4.a) $(i,j) \in I_b^* \cup I_b$:

- $$(i,j) \equiv (3,4) \Rightarrow \begin{cases} z_1 = 16 - 3 = 13 \\ z_2 = 11 + 3 = 14 \end{cases} \text{Solution dominée}$$
  $$(i,j) \equiv (4,4) \Rightarrow \begin{cases} z_1 = 16 - 4 = 12 \\ z_2 = 11 + 2 = 13 \end{cases} \text{Solution dominée}$$

4.b) $(i,j) \in I_a \cup I_a^*$:

4.b.1. $(i,j) \in I_a$:

- $(i,j) \equiv (2,1) \Rightarrow I_{ij} = \{(4,2)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (4,2) \Rightarrow \begin{cases} z_1 = 16 + 3 + 3 = 22 \\ z_2 = 11 - 6 + 2 = 7 \end{cases}$$

- $(i,j) \equiv (2,4) \Rightarrow I_{ij} = \{(1,2)\} \Rightarrow \overline{I_{ij}} = \varnothing.$
- $(i,j) \equiv (4,3) \Rightarrow I_{ij} = \{(3,1), (3,4)\} \equiv \overline{I_{ij}}$

$$(l(i,j), p(i,j)) \equiv (3,4) \Rightarrow \begin{cases} z_1 = 12 + 10 - 4 = 18 \\ z_2 = 13 - 7 + 6 = 12 \end{cases}$$

4.b2. $(i,j) \in I_a^*$:

- $(i,j) \equiv (1,2) \Rightarrow I_{ij}^* = \{(2,4)\} \equiv \overline{I_{ij}^*}$

$$(l^*(i,j), p^*(i,j)) \equiv (2,4) \Rightarrow \begin{cases} z_1 = 16 - 3 + 1 = 14 \\ z_2 = 11 + 7 - 3 = 15 \end{cases} \text{Solution dominée}$$

- $(i,j) \equiv (3,1) \Rightarrow I_{ij}^* = \{(4,2), (4,3), (4,4)\} \Rightarrow \overline{I_{ij}^*} = \{(4,2), (4,4)\}$

$$(l^*(i,j), p^*(i,j)) \equiv (4,4) \Rightarrow \begin{cases} z_1 = 16 - 3 - 4 = 9 \\ z_2 = 11 + 4 + 2 = 17 \end{cases} \text{Solution dominée}$$

Par comparaison des valeurs de (z_1, z_2) ; x_{21} et x_{42} rentrent simultanément dans la base.

$X_5 \equiv x_{14} = x_{21} = x_{33} = x_{42} = 1$, $(z_1, z_2) = (22, 7)$.

$$L = \{(6,24), (9,17), (12,13), (16,11), (22,7)\}$$

Itération 5.

$$\overline{C}^{(1)} = \begin{pmatrix} 0 & -6 & 0 & (0) \\ (0) & -6 & -3 & -2 \\ -3 & 1 & (0) & -3 \\ 0 & (0) & 5 & -4 \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} 0 & 5 & 0 & (0) \\ (0) & 4 & 6 & 3 \\ 4 & 3 & (0) & 3 \\ 0 & (0) & -2 & 2 \end{pmatrix}$$

- $\overline{I} = \{(4,3)\}$.
- $\overline{I^*} = \{(1,2),(2,2),(2,3),(2,4),(3,1),(3,4),(4,4)\}$.
- $I_a = \{(4,3)\}$.
- $I_b = \varnothing$.
- $I_a^* = \{(1,2),(2,3),(3,1),(4,4)\}$.
- $I_b^* = \{(2,2),(2,4),(3,4)\}$.

5.a) $(i,j) \in I_b^* \cup I_b$:

$$\left.\begin{array}{l} (i,j) \equiv (2,2) \Rightarrow \begin{cases} z_1 = 22 - 6 = 16 \\ z_2 = 7 + 4 = 11 \end{cases} \textit{Solution dominée} \\ (i,j) \equiv (2,4) \Rightarrow \begin{cases} z_1 = 22 - 2 = 20 \\ z_2 = 7 + 3 = 10 \end{cases} \\ (i,j) \equiv (3,4) \Rightarrow \begin{cases} z_1 = 22 - 3 = 19 \\ z_2 = 7 + 3 = 10 \end{cases} \end{array}\right\} \Rightarrow (i_b, j_b) \equiv (3,4) \text{avec}$$

$$(z_1, z_2) = (19, 10)$$

5.b) $(i,j) \in I_a \cup I_a^*$:

5.b.1. $(i,j) \in I_a$:

$$(i,j) \equiv (4,3) \Rightarrow I_{ij} = \{(3,2)\} \Rightarrow \overline{I_{ij}} = \varnothing.$$

5.b.2. $(i,j) \in I_a^*$:

$(i,j) \equiv (1,2) \Rightarrow I_{ij}^* = \{(4,4)\} \equiv \overline{I_{ij}^*}$

$$(l^*(i,j), p^*(i,j)) \equiv (4,4) \Rightarrow \begin{cases} z_1 = 22 - 6 - 4 = 12 \\ z_2 = 7 + 5 + 2 = 14 \end{cases} \text{Solution dominée}$$

$(i,j) \equiv (2,3) \Rightarrow I_{ij}^* = \{(3,1), (3,2), (3,4)\} \equiv \overline{I_{ij}^*}$

$$(l^*(i,j), p^*(i,j)) \equiv (3,4) \Rightarrow \begin{cases} z_1 = 22 - 3 - 3 = 16 \\ z_2 = 7 + 6 + 3 = 16 \end{cases} \text{Solution dominée}$$

$(i,j) \equiv (3,1) \Rightarrow I_{ij}^* = \{(2,4), (2,3)\} \equiv \overline{I_{ij}^*}$

$$(l^*(i,j), p^*(i,j)) \equiv (2,3) \Rightarrow \begin{cases} z_1 = 22 - 3 - 3 = 16 \\ z_2 = 7 + 4 + 6 = 17 \end{cases} \text{Solution dominée}$$

$(i,j) \equiv (4,4) \Rightarrow I_{ij}^* = \{(1,2)\} \equiv \overline{I_{ij}^*}$

$$(l^*(i,j), p^*(i,j)) \equiv (1,2) \Rightarrow \begin{cases} z_1 = 22 - 4 - 6 = 12 \\ z_2 = 7 + 2 + 5 = 14 \end{cases} \text{Solution dominée}$$

Par comparaison des valeurs de (z_1, z_2) ; x_{34} rentre seule dans la base.

$X_6 \equiv x_{13} = x_{21} = x_{34} = x_{42} = 1, (z_1, z_2) = (19, 10)$.

$$L = \{(6,24), (9,17), (12,13), (16,11), (22,7), (19,10)\}$$

Itération 6.

$$\overline{C}^{(1)} = \begin{pmatrix} 0 & -6 & (0) & 0 \\ (0) & -6 & -3 & -2 \\ 0 & 4 & 3 & (0) \\ 0 & (0) & 5 & -4 \end{pmatrix}, \ \overline{C}^{(2)} = \begin{pmatrix} 0 & 5 & (0) & 0 \\ (0) & 4 & 6 & 3 \\ 1 & 0 & -3 & (0) \\ 0 & (0) & -2 & 2 \end{pmatrix}$$

- $\overline{I} = \{(3,3), (4,3)\}$.
- $\overline{I^*} = \{(1,2), (2,2), (2,3), (2,4), , (4,4)\}$.

- $I_a = \{(4,3)\}$.
- $I_b = \{(3,3)\}$.
- $I_a^* = \{(1,2),(4,4)\}$.
- $I_b^* = \{(2,2),(2,3),(2,4)\}$.

6.a) $(i,j) \in I_b^* \cup I_b$:

$$(i,j) = (3,3) \Rightarrow \begin{cases} z_1 = 19 + 3 = 22 \\ z_2 = 10 - 3 = 7 \end{cases} \textit{Solution dominée}$$

$$(i,j) = (2,3) \Rightarrow \begin{cases} z_1 = 19 - 3 = 16 \\ z_2 = 10 + 6 = 16 \end{cases} \textit{Solution dominée}$$

$$(i,j) = (2,4) \Rightarrow \begin{cases} z_1 = 19 - 2 = 17 \\ z_2 = 10 + 3 = 13 \end{cases} \textit{Solution dominée}$$

$$(i,j) = (2,2) \Rightarrow \begin{cases} z_1 = 19 - 6 = 13 \\ z_2 = 10 + 4 = 14 \end{cases} \textit{Solution dominée}$$

6.b) $(i,j) \in I_a \cup I_a^*$:

6.b.1. $(i,j) \in I_a$:

$$(i,j) \equiv (4,3) \Rightarrow I_{ij} = \{(1,2)\} \Rightarrow \overline{I_{ij}} = \varnothing.$$

6.b.2. $(i,j) \in I_a^*$:

$$(i,j) \equiv (1,2) \Rightarrow I_{ij}^* = \{(3,4)\} \equiv \overline{I_{ij}^*}$$

$$(l^*(i,j), p^*(i,j)) \equiv (3,4) \Rightarrow \begin{cases} z_1 = 19 - 6 + 5 = 18 \\ z_2 = 10 + 5 - 2 = 13 \end{cases} \text{Solution dominée}$$

$$(i,j) \equiv (4,4) \Rightarrow I_{ij}^* = \{(3,2)\} \equiv \overline{I_{ij}^*}$$

$$(l^*(i,j), p^*(i,j)) \equiv (3,2) \Rightarrow \begin{cases} z_1 = 19 - 4 + 4 = 19 \\ z_2 = 10 + 2 - 0 = 12 \end{cases} \text{Solution dominée}$$

Aucune nouvelle solution. Toutes les solutions engendrées sont dominées. Fin d'itération.

$$L = \{(6,24), (9,17), (12,13), (16,11), (22,7), (19,10)\} = E(BAP).$$

La méthode décrite ci-dessus a été programmée à l'aide du logiciel Delphi 5. Ce programme peut déterminer l'ensemble de toutes les solutions efficaces du problème bi-critères d'affectation (s'il en existent), ou alors un sous-ensemble de solutions efficaces .

Les tests effectués sur une variété de problèmes (BAP) ont révélé que l'efficacité de la méthode dépend de la base duale admissible trouvée, car pour certaines bases, le programme donne l'ensemble de toutes les solutions efficaces, pour d'autres bases correspondantes à la même solution, seulement un sous ensemble est trouvé. Par exemple pour l'exemple illustratif, il suffit de prendre pour la dernière itération $v_2 = \overline{v_2} - \alpha$ au lieu de $v_1 = \overline{v_1} - \alpha$ pour que la méthode ne donne pas toutes les solutions efficaces malgré que la base obtenue est duale admissible.

Nous avons aussi remarquer que, pour d'autres bases des solutions qui ne sont pas efficaces ont été engendrées.

3.3.7 Méthode de deux phases [19], [21]

Contrairement à la méthode précédente, celle-ci travaille sur un problème d'affectation uni-critère dont la fonction objectif est une agrégation linéaire des deux autres. Comme vu ci-dessous, la première phase va déterminer toutes les solutions efficaces supportées, tandis que la seconde détermine les solutions efficaces non supportées.

Phase (I) : détermination de $SE(P)$

Cette phase est identique à celle de Aneja et nair [19] pour le problème de transport.

Soit S la liste des solution efficaces supportées; cette liste est initialisée par les deux solutions optimales et efficaces en cas de non unicité $\tilde{X}_1$ et $\tilde{X}_2$ de valeurs respectives $(\tilde{z}_1^{(1)}, \tilde{z}_2^{(1)})$ et $(\tilde{z}_1^{(2)}, \tilde{z}_2^{(2)})$ des deux critères. Les solutions de S sont rangées dans l'ordre croissant de premier critère z_1.

Une agrégation est réalisée à l'aide des poids :

$$a_1^{(t)} = z_2^{(r)} - z_2^{(s)} \quad et \quad a_2^{(t)} = z_1^{(s)} - z_1^{(r)}$$

Où $(z_1^{(r)} = \tilde{z}_1^{(1)}, z_2^{(r)} = \tilde{z}_2^{(1)})$ et $(z_1^{(s)} = \tilde{z}_1^{(2)}, z_2^{(s)} = \tilde{z}_2^{(2)})$ sont les valeurs données par les deux premières solutions efficaces et supportées consécutives X^r, X^s de S.

On obtient le problème d'affectation unicritère $P^{(t)}$ de matrice coût $C^{(t)}$

$$C^{(t)} = (a_1^{(t)} c_{ij}^{(1)} + a_2^{(t)} c_{ij}^{(2)})$$

Soit $\{X^{(t)}, t \in T\}$ l'ensemble des solutions optimales de ce critère obtenues par la méthode hongroise.

Deux cas se manifestes :

Premier cas : $\{X^r, X^s\} \cap \left\{X^{(t)}, t \in T\ \right\} = \varnothing$. Alors les solutions $X^{(t)} t \in T$ sont des nouvelles solutions efficaces supportées inclues dans S.

Deuxième cas : $\{X^r, X^s\} \subset \left\{X^{(t)}, t \in T\ \right\}$.

- Si X^r et X^s sont les deux seules solutions $X^{(t)}$ $\left(\{X^r, X^s\} \cap \left\{X^{(t)}, t \in T\ \right\} = \{X^r, X^s\}\right)$, donc il n'y a pas d'autres solutions supportées entre X^r et X^s

- Dans le cas contraire, c.à.d qu'il existe d'autres solutions $X^{(t)}$, alors ces dernières sont de nouvelles solutions supportées. Ces nouvelles solutions, si elles existent, elles sont placées dans un ensembles particulier note S'.

On continu la procédure tant que les autres paires n'ont pas été examinées, en particulier les paires (X^r, X^t) et (X^t, X^s) .

À la fin de cette phase, on obtient :

$$SE(BAP) = S \cup S^{'}$$

Phase (II) : Détermination de $E(P)\backslash SE(P)$

L'objectif de cette phase est de rechercher, pour chaque paire de solution (X^r, X^s) de $SE(P)$, s'il existe des solutions efficaces non supportées X^u : $(z_1^{(u)}, z_2^{(u)})$ vérifiant :

$$\begin{pmatrix} z_1^{(r)} < z_1^{(u)} < z_1^{(s)} \\ z_1^{(r)} > z_2^{(u)} > z_2^{(s)} \end{pmatrix}$$

La procédure

- Soit $P^{(t)}$ le problème associé au couple (X^r, X^s) est optimisé.
- Soit $L = \left\{(i,j) \,\middle|\, \overline{c}_{ij}^{(t)} > 0\right\}$

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

  Les affectations de L ne sont pas toutes candidate à être imposées, certaines d'elles peuvent être éliminées de L; se sont celles qui conduisent à des solutions dominées. Pour se faire cette phase propose de calculer une borne inférieure de l'augmentation pour les trois cas suivants :

- Du critère correspondant à la matrice $C^{(t)}$ par rapport à la valeur prise par Z^r, Z^s.
- Du critère z_1 par rapport à la valeur z_1^r.
- Du critère z_2 par rapport à la valeur z_2^s.

Test1 : test sur le problème $P^{(t)}$ Soit (i^*, j^*) une affectation de L.

Et soit i_r, i_s, j_r, j_s des indices tel que : $x_{i_r j^*} = x_{i^* j_r} = x_{i_s j^*} = x_{i^* j_s} = 1$.

1.a) Par rapport à X^r

L'imposition de (i^*, j^*) impose au minimum de créer un nouveau zéro dans la ligne i_r et dans la colonne j_r.

- Si ce nouveau zéro est créé en (i_r, j_r) la borne inférieur de l'augmentation de $z^{(t)}$ sera $\overline{c}_{i_r j_r}^{(t)}$.
- Si deux nouveaux zéros sont créés, la borne inférieure de l'augmentation $z^{(t)}$ sera :

$$\Psi_s^{(t)} = \min_{j \neq j^*} \overline{c}_{i_s j}^{(t)} + \min_{i \neq i^*} \overline{c}_{i j_s}^{(t)}$$

Alors la borne inférieur de l'augmentation de $z^{(t)}$ pour ce test sera :

$$l_t = \overline{c}_{i^* j^*}^{(t)} + \min\left(\overline{c}_{i_r j_r}^{(t)}, \overline{c}_{i_s j_s}^{(t)}, \Psi_r^{(t)}, \Psi_s^{(t)}\right)$$

Une affectation peut être éliminée de la liste L si :

$$l_t \geq a_1^{(t)} \cdot a_2^{(t)}$$

Test2 : test d'augmentation de $z_1^{(r)}$

a) $X^r = \tilde{X}_1$

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_1 \geq a_2^{(t)}$$

Où l_1 est la borne inférieur de l'augmentation de $z_1^{(r)}$ engendrée par l'imposition de l'affectation (i^*, j^*) qui vaut :

$$l_1 = \overline{c}_{i^* j^*}^{(1)} + \min\left(\overline{c}_{i_1 j_1}^{(1)}; \min_{j \neq j^*} \overline{c}_{i_1 j}^{(1)}; \min_{i \neq i^*} \overline{c}_{i j_1}^{(1)}\right)$$

Avec i_1, j_1, des indices tel que: $x_{i_1 j^*} =.x_{i^* j_1} = 1$

b) $X^r \neq \tilde{X}_1$

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_1' = \overline{c}_{i^* j^*}^{(1)} + \max\left(\sum_{i \neq i^*} \min_{j \neq j^*} \overline{c}_{ij}^{(1)}; \sum_{j \neq j^*} \min_{i \neq i^*} \overline{c}_{ij}^{(1)}\right) \geq a_2^{(t)}$$

Test3 : test d'augmentation de $z_2^{(s)}$

a) $X^s = \tilde{X}_2$

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l_2 = \overline{c}^{(2)}_{i^*j^*} + \min\left(\overline{c}^{(2)}_{i_2j_2}; \min_{j\neq j^*} \overline{c}^{(2)}_{i_2j}; \min_{i\neq i^*} \overline{c}^{(2)}_{ij_2}\right) \geq a_1^{(t)}$$

Avec i_2, j_2 indices tel que$x_{i_2j^*} =.x_{i^*j_2} = 1$. Les coûts réduits $\overline{c_{ij}}^{(2)}$ sont ceux du critère z_2 relative à la solution X^s.

b) $X^s \neq \tilde{X}_2$

L'affectation (i^*, j^*) peut être éliminée de la liste L si :

$$l'_2 = \overline{c}^{(2)}_{i^*j^*} + \max\left(\sum_{i\neq i^*} \min_{j\neq j^*} \overline{c}^{(2)}_{ij}; \sum_{j\neq j^*} \min_{i\neq i^*} \overline{c}^{(2)}_{ij}\right) \geq a_1^{(t)}$$

Pour le reste (i.e. les affectations qui ne sont éliminées par aucun des trois tests) deux variantes existent pour la procédure :

- les affectations restantes de L sont imposées tour à tour et le même problème est ré-optimisé; les solutions obtenues sont triées afin de ne conserver que les solutions efficaces et les affectations sont considérées dans l'ordre de bornes d'augmentation croissante. Dès qu'une nouvelle solution efficaces X^u est trouvée, le test1 peut être réappliqué est la borne inférieur de l'augmentation de $z^{(t)}$sera :

$$l'_t = \max(a_1^{(t)}(z_1^{(u)} - z_1^{(r)}), a_2^{(t)}(z_1^{(u)} - z_1^{(s)}))$$

- Pour chaque paire (X^r, X^s), il faut poursuivre l'optimisation de $P^{(t)}$, en imposant en plus de $(i,j) \in L$, d'autres affectations de L, jusqu'à ce que :

  a) Soit on trouve une solution X^u telle que :

$$\begin{pmatrix} z_1^{(r)} < z_1^{(u)} < z_1^{(s)} \\ z_2^{(r)} > z_2^{(u)} > z_2^{(s)} \end{pmatrix}$$

b)Soit $z^{(u)} \geq z^{(t)} + a_1^{(t)} a_2^{(t)}$

Cette méthode a été implémentée afin d'engendrer des exemples numériques pour tester son efficacité. Cependant, ce procédé prend beaucoup du temps (voir [17]). En conséquence, des méthodes d'approximation ont été proposées pour calculer les solutions efficaces avec une durée de calcul raisonnable (voir [14], [8]), ces mèthodes combinent entre méthodes exactes et méthodes approchées.

Chapitre 4

Conclusion générale et perspectives

L'optimisation multi-objectifs (multicritère) est sans doute un axe de recherche primordial pour les scientifiques et les ingénieurs, non seulement à cause de la nature multi-critères de la plupart des problèmes réels, mais aussi parce que de nombreuses questions restent ouvertes dans ce domaine. Elle a pour caractéristique principale de fournir un ensemble de solutions qui représentent l'ensemble des solutions efficaces. Dans le cas continu (où l'ensemble d'admissibilité est convexe), cette tache est relativement facile car l'ensemble des solutions efficaces peut être trouvé par combinaison convexe des critères ce qui est impossible dans le cas entier (la propriété de convexité n'est pas conservée) où seulement un sous ensemble est généré sauf pour quelques problèmes représentant une structure particulière.

Dans le cadre de ce travail nous avons réalisé une synthèse sur les méthodes de résolution des problèmes unicritères et multicritères dans le cas continu et discret en précisant la difficulté relative à la résolution de chaque type de problème. Ces méthodes ont été classées selon l'intervention du décideur dans le processus de résolution et le but recherché.

Pour les méthodes exactes pour l'optimisation multi-objectifs, différentes perspectives sont envisageables. Tout d'abord, l'amélioration des performances de ces méthodes notamment l'implémentation de ces dernières ce qui permettrait de résoudre des problèmes de tailles plus grandes. Une autre perspective concerne le développement de nouvelles méthodes exactes ou l'adaptation des méthodes existantes pour des problèmes à plus de deux objectifs ou encore la coopération entre méthodes exactes et approchées afin de réduire le temps de calcul.

Dans la classe des problèmes à variables binaires, en plus des difficultés liées aux problèmes en variables discrètes, ces problèmes souffrent de l'explosion combinatoire ce qui rend nécessaire pour toute méthode de résolution de considérer le temps de calcul. L'implémentation de la méthode de l'adaptation de la méthode hongroise pour le problème bi-critères d'affectation proposée par Ulungu dans [19] présenté dans ce mémoire, nous a permis de constater que pour des instances de moyennes tailles la méthode ne donne pas toutes les solutions efficaces et que pour des bases correspondantes à la même solution optimale, elle donne toutes les solutions efficaces et pour d'autres bases seulement un sous ensemble est généré et pour d'autres bases elle génère des solutions qui ne sont pas efficaces.

Pour le problème d'affectation nos perspectives sont nombreuses on site parmi :

- L'amélioration des méthodes déjà existantes en diminuant le nombre d'itérations et par conséquent le temps de calcul.
- L'adaptation des méthodes existantes pour des problèmes à plus de deux objectifs.
- L'adaptation de nouvelles méthodes pour le problème d'affectation tel que PPM.

Ce qui est sûr qu'il reste beaucoup à faire pour le problème d'affectation multicritère.

Bibliographie

[1] Dj. Chaabane. Optimisation Multicritère en nombres entiers. Thèse de doctorat d'état, Faculté de Mathématique, U.S.T.H.B, 16-05-2005.

[2] A.Crema, J.Sylva. A method for finding the set of non- dominated vectors of Multiple Objective Integer Linear Programs, European Journal of Operational Research, in press 2003.

[3] G. B. Dantzig, Linear Programming and Extensions. Princeton university press, 1963.

[4] F. Degoutin, Problèmes bi-objectifs en optimisation combinatoire et capacité d'infrastructures ferroviaires. Mémoire de recherche préparé dans le cadre du DEA AISIH, Université de Valenciennes, 2002.

[5] X. Delorme. Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires. Thèse de doctorat, université de Valenciennes et du Hainaut-Cambrésis, Discipline : Informatique, 2003.

[6] C. Dhaenens-Flipo"Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances. Habilitation de diriger des Recherches de l'U.S.T.L. Discipline : Informatique, 2005.

[7] M. Ehrgott and X. Gandibleux. A survey and Annotated Bibliography of Multiobjective combinatorial optimization. OR. Spektrum, Springer-Verlag, 22: 425-460, 2000.

[8] X. Gandibleux. École d'Automne de Recherche Opérationnelle. LAMIH - Recherche Opérationnelle et Informatique, université de Valenciennes et du Hainaut-Cambrésis 2003.

[9] R. Malhotra, H.L. Bhatia, and M.C. Puri. Bi-criteria assignment problem. Opsearch. Journal of the Operational Research Society of India, 19(2): 84–96, 1982.

[10] H. Meunier. Algorithmes évolutionnaire paralléles pour l'optimisation multi-objectif de réseaux de télécommunication mobiles. Thèse de doctorat, de l'U.S.T.L, dicipline : Informatique, 2002.

[11] C.R. Pedersen, Multicritaria descrete optimization-and related topics. Phd thesis, Department of Operations Research, University of Aarhus, 2006.

[12] M. Sakarovitch. Graphe et programmation linéaire. Tome1; Paris.

[13] M. Sakarovitch, 1983. 'Techniques mathématique de la recherche opérationnelle, tome IV Optimisation combinatoire', ENSIMAG, Université Scientifique et Médicale, Institut National Polytechnique de Grenoble.

[14] A. Przybylski, X. Gandibleux, and M. Ehrgott. Seek and cut algorithm computing minimal and maximal complete efficient solution sets for the biobjective assignment problem. In 6th Int. Multi-Objective Programming and Goal Programming conf (MOPGP'04), 2004.

[15] R.E. Steuer. Multiple Criteria Optimization: Theory, Computation, and Application.Wiley séries in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, New York, 1986.

[16] J. Teghem. Programmation linéaire Ellipses, 1996.

[17] D. Tuyttens, J. Teghem, Ph. Fortemps, and K. Van Nieuwenhuyse. Performance of the mosa method for the bicriteria assignment problem. Journal of Heuristics, 6:295–310, 2000.).

[18] E.L.Ulungu. J.Teghem. Multicritaria assignment problem: a new method, Technical report,.Faculté Polytechnique de Monis (1992).

[19] E.L. Ulungu. Optimisation Combinatoire multicritère: Détermination de l'ensemble des solutions efficaces et mèthodes intéractives. thèse de doctorat, Université de Mons-Hainault, Faculté des Sciences, 1993.

[20] E.L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: A survey. Journal of Multi-Criteria Decision Analysis, 3:83–104,1994.

[21] E.L. Ulungu and J. Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. Foundations of Computing and Decision Sciences, 20:149–165, 1995.

[22] M.Zeleny, Multiple criteria problem solving. McGraw-Hill, New York, (1982).