

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

MENTOURI-CONSTANTINE

**Faculté des Sciences Exactes
Département de Mathématiques**

École Doctorale de Mathématiques, pôle de Constantine

N° d'ordre :.....

N° de série :.....

MÉMOIRE

Pour l'obtention du grade de

MAGISTER

**SPÉCIALITÉ: MATHÉMATIQUES APPLIQUÉES
OPTION: OPTIMISATION NUMÉRIQUE**

Présenté par : AAID DJAMEL

Intitulé :

**Étude numérique comparative entre des méthodes
de résolution d'un problème de transport à quatre
indices avec capacités**

Soutenu publiquement en 14/02/ 2010 à l'UNIVERSITÉ DE CONSTANTINE

devant le jury composé de

Président :	M. Dekhmouche	M.C.	U .Constantine
Rapporteur :	R. Zitouni	M.C.	U. Sétif
Examineur :	D. Benterki	M.C.	U. Sétif
Examineur :	B. Merikhi	M.C.	U. Sétif

Remerciements

Je tiens à exprimer mes plus vifs remerciements à ALLAH pour m'avoir facilité ce travail.

Je tiens à exprimer tout d'abord ma profonde reconnaissance à mon directeur de recherche, le Docteur Rachid ZITOUNI, qui m'a toujours fait confiance et qui a su m'encourager à surmonter les diverses difficultés. Je tiens aussi à le remercier pour son appui scientifique et pour sa grande disponibilité qui a été essentielle pour la progression de ma recherche. Je remercie également les membres de jury qui ont accepté de juger ce travail, je suis très reconnaissant à leurs remarques et commentaires qui m'ont aidé beaucoup pour mieux présenter ce document.

Je remercie toute l'équipe d'optimisation du laboratoire de mathématiques fondamentales et numériques du département de mathématiques de Sétif pour leurs aides continues et leurs encouragements.

Je remercie aussi mon ami, Monsieur Nabil Ouezen : maître assistant à l'université de Batna, pour son aide, ses conseils et ses encouragements.

Mes remerciements s'adressent aussi à toute l'équipe administrative du département de mathématique de Constantine, en particulier l'équipe de l'école doctorale de mathématique, le comité et le conseil scientifique d'avoir entrepris avec une grande souplesse les démarches nécessaires pour la soutenance.

Enfin, je remercie tous ceux qui m'ont aidé de près ou de loin quelque 'ils soient et d'où qu'ils soient.

Table des matières

Introduction générale	3
1 Généralités sur la programmation mathématique	5
1.1 Préliminaires	6
1.1.1 Convexité	6
1.1.2 Programme Mathématique	7
1.2 Programmation Linéaire	10
1.2.1 Formes usuelles d'un programme linéaire	10
1.2.2 Dual d'un programme linéaire	11
1.2.3 Complexité d'un programme linéaire	12
1.2.4 Méthode de résolution d'un programme linéaire	13
1.2.5 Applications de la programmation linéaire	15
2 Problème de transport à quatre indices avec capacités	16
2.1 Introduction	16
2.2 Position du problème	17
2.3 Définitions et propriétés	18
2.4 Conditions de réalisabilité	21
2.5 Conditions d'optimalité	21
2.6 Méthodes de résolution d'un problème de transport à quatre indices avec capacités	22

2.6.1	Méthode du simplexe	22
2.6.2	Méthode de Ye-Lustig (méthode de points intérieurs)	29
2.6.3	Méthode de R. Zitouni et al. (AL_{PT4C})	35
3	Etude comparative d’algorithmes pour un problème de transport à quatre indices avec capacités	44
3.1	Introduction	44
3.2	Adaptation des méthodes classiques	45
3.3	Résultats numériques :	47
3.4	Comparaison récapitulative	53
3.5	Commentaires	56
3.6	Conclusion	57
	Bibliographie	58

Introduction générale

Après la deuxième guerre mondiale, la méthode du simplexe s'est imposée comme la seule méthode efficace pour la résolution des programmes linéaires. Il a été prouvé que cette méthode est vraiment efficace et a un bon rendement en pratique, mais ce n'est pas le cas en théorie de sorte qu'elle présente une complexité exponentielle. En effet, Klee et Minty ont construit un exemple en 1972, qui montre que l'algorithme du simplexe est de complexité exponentielle. En 1979, le mathématicien soviétique L.G. Kachiyan a mis en oeuvre le premier algorithme polynomial pour la programmation linéaire : l'algorithme des ellipsoïdes. Toutefois, l'algorithme proposé s'est révélé complètement inefficace dans la pratique. En 1984, N. Karmarkar a révolutionné le domaine de la programmation linéaire en mettant en oeuvre un algorithme polynomial basé sur les méthodes de pénalités intérieures.

Celui-ci s'est avéré un compétiteur sérieux de l'algorithme du simplexe.

Depuis, une recherche intense s'est déclenchée, dans ce domaine et a donné comme résultat, une grande variété d'algorithmes de ce type qui peuvent être classés en quatre catégories :

1. méthodes projectives.
2. méthodes affines.
3. méthodes du potentiel.
4. méthodes de trajectoire centrale.

Dans le présent travail, on s'intéresse à une étude comparative entre trois méthodes de résolution d'un problème de transport à quatre indices avec capacités (*PT4C*).

La première est celle du simplexe (Dantzig), la deuxième est une variante du points intérieurs (Ye-Lustig) et la dernière notée AL_{PT4C} , est une variante du simplexe (R. Zitouni et al). Nous présentons leurs principes et leurs algorithmes puis nous analysons les résultats numériques obtenus à travers le traitement de certains exemples de différentes taille.

Ce mémoire est composé de trois chapitres.

Dans le premier, nous rappelons les résultats fondamentaux en programmation linéaire.

Le deuxième chapitre est composé de trois parties dont chacune est consacrée à la description de l'un des trois algorithmes de résolution d'un $PT4C$, cités ci-dessus.

Dans le dernier chapitre, nous présentons l'essentiel de notre travail, qui se résume à une comparaison des trois algorithmes précédents et ceci à travers des différents tests numériques.

Chapitre 1

Généralités sur la programmation mathématique

Introduction

Dans ce chapitre, nous rappelons brièvement quelques notions fondamentales en programmation mathématique. Ensuite, nous citons quelques exemples, d'application de la programmation linéaire.

1.1 Préliminaires

1.1.1 Convexité

La notion de convexité est un outil mathématique important pour l'étude théorique et numérique des problèmes d'optimisation. On présente dans ce paragraphe quelques notions d'analyse convexe d'usage courant [8].

Définition 1

Un sous-ensemble C de \mathbb{R}^n est dit convexe si :

$$\forall x, y \in C : (1 - \lambda)x + \lambda y \in C, \forall \lambda \in [0, 1].$$

Autrement dit si le segment de droite joignant deux points quelconques $x, y \in C$:

$$[x, y] = \{(1 - \lambda)x + \lambda y \in C, 0 \leq \lambda \leq 1\} \text{ est entièrement inclus dans } C.$$

Définition 2

L'intersection de tous les ensembles convexes contenant un sous-ensemble S de \mathbb{R}^n est appelée : enveloppe convexe de S . On la note $\text{conv}(S)$: c'est le plus petit convexe de \mathbb{R}^n contenant S .

Définition 3

L'enveloppe convexe d'un sous-ensemble fini de \mathbb{R}^n , $\text{conv}(\{x_1, \dots, x_{k+1}\})$ est appelée : Polytope. Si de plus, les vecteurs $\{x_i - x_1, i = 2, \dots, k+1\}$ sont linéairement indépendants, alors $\text{conv}(\{x_1, \dots, x_{k+1}\})$ est appelée : un k -simplexe de sommets x_1, \dots, x_{k+1} .

Définition 4

C est dit affine si :

$$\forall x, y \in C : (1 - \lambda)x + \lambda y \in C, \quad \forall \lambda \in \mathbb{R}$$

Définition 5

C est un polyèdre s'il s'écrit comme suit :

$$C = \{x \in \mathbb{R}^n : p_i^t x \leq \alpha_i, \quad i = 1, \dots, m\}$$

où p_i est un vecteur non nul de \mathbb{R}^n et α_i est un scalaire, pour $i = 1, \dots, m$.

Définition 6

Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite convexe sur C si :

$$f[(1 - \lambda)x + \lambda y] \leq (1 - \lambda)f(x) + \lambda f(y), \quad \forall \lambda \in [0, 1]$$

Définition 7

La fonction f est dite affine si :

$$f[(1 - \lambda)x + \lambda y] = (1 - \lambda)f(x) + \lambda f(y), \quad \forall \lambda \in \mathbb{R}.$$

1.1.2 Programme Mathématique

Un programme mathématique est un problème de la forme :

$$(PM) \left\{ \begin{array}{l} \min f(x) \\ f_i(x) \leq 0, \quad i = 1 : m \\ g_j(x) = 0, \quad j = 1 : p \end{array} \right. ,$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction objectif de (PM) .

$D = \{x \in \mathbb{R}^n; f_i(x) \leq 0, g_j(x) = 0, i = 1 : m, j = 1 : p\} \subset \mathbb{R}^n$ représente l'ensemble des solutions réalisables.

- Un point x de D est appelé solution réalisable (ou admissible) de (PM) .
- On appelle solution optimale de (PM) , toute solution réalisable x réalisant le minimum de f sur D , $f(x)$ sera appelée valeur optimale de (PM)

- Si $D = \mathbb{R}^n$, on dit que (PM) est un problème d'optimisation sans contraintes.

Classification d'un programme mathématique

La classification de (PM) est établie à partir de deux propriétés fondamentales des fonctions f et f_i à savoir : la convexité et la différentiabilité. A ce propos (PM) est dit convexe si f et f_i sont convexes et g_i sont affines. Si ces dernières sont toutes différentiables, on dit que (PM) est un programme différentiable.

Notons que le (PM) convexe et différentiable est le modèle le mieux élaboré du domaine. En l'absence de la convexité ou de la différentiabilité, le problème présente des difficultés importantes, surtout sur le plan numérique.

Notons que le cas le plus difficile à traiter est celui des programmes non convexes et non différentiables, par contre, le cas le plus simple est celui de la programmation linéaire où : f, f_i et g_i sont affines.

Principaux résultats d'existence et d'unicité

Théorème 1 *Si f est continue et coercive sur C (i.e. $f(x) \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$ avec $x \in C$), et si C est fermé non vide, alors (PM) admet au moins une solution optimale.*

Théorème 2 *Si f est strictement convexe et si C est convexe, alors la solution optimale de (PM) , si elle existe, est unique.*

Qualification des contraintes

Condition de Slater (1950)

La condition de qualification des contraintes de Slater est comme suit :

C est un polyèdre convexe.

C est convexe et $\text{int}C \neq \emptyset$. (ie : $\exists x^0 \in C$ tel que : $f_i(x^0) < 0, \forall i$).

Condition de Mangasarian-Fromowitz (1967)

La condition de qualification des contraintes de Mangasarian-Fromowitz au point $x^* \in C$, est comme suit :

1- Une contrainte $f_i(x) \leq 0$ est dite active ou saturée en $x^* \in C$ si elle satisfait $f_i(x^*) = 0$. Une contrainte d'égalité est saturée par définition.

2- Un point $x^* \in C$ est dit régulier si les gradients des contraintes saturées en x^* sont linéairement indépendants. On dit également que les contraintes sont qualifiées en x^*

Conditions d'optimalité

Théorème 3 (Karush-Kuhn-Tucher)

Si x^* est une solution optimale locale de (PM) satisfaisant l'une des conditions de qualification précédentes, alors il existe des multiplicateurs $\lambda \in \mathbb{R}_+^m$ et $\mu \in \mathbb{R}^p$ tels que :

$$\left\{ \begin{array}{ll} \nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla f_i(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0 & \text{(condition d'optimalité)} \\ \lambda_i f_i(x^*) = 0, i = 1 : m & \text{(condition de complémentarité)} \\ & g_i(x^*) = 0 \end{array} \right.$$

Si en plus (PM) est convexe, les conditions précédentes sont à la fois nécessaires et suffisantes pour que x^* soit optimum global pour (PM).

$$f_i(x) < 0 \implies \lambda_i = 0.$$

Les multiplicateurs de Kuhn-Tucher généralisent ceux de Lagrange. Autrement dit, si toutes les contraintes sont des égalités on retrouve la condition classique de Lagrange :

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla f_i(x^*) = 0, \lambda_i \in \mathbb{R}.$$

Si les contraintes ne sont pas qualifiées en x^* , les conditions de KKT ne s'appliquent pas (x^* peut être optimal sans vérifier ces conditions).

1.2 Programmation Linéaire

La programmation linéaire constitue la pierre angulaire de toute la recherche opérationnelle. Il faut bien sûr éviter de forcer tout modèle à être linéaire. Par contre, un très grand nombre de modèles constituent des extensions de programmes linéaires. Elle peut se définir comme une technique mathématique permettant de résoudre des problèmes de gestion et particulièrement ceux où le gestionnaire doit déterminer, face à différentes possibilités, l'utilisation optimale des ressources de l'entreprise pour atteindre un objectif spécifique comme la maximisation des bénéfices ou la minimisation des coûts.

Un problème de programmation linéaire s'écrit sous la forme :

$$(PL) \begin{cases} \min c^t x \\ Ax = b \text{ ,} \\ x \geq 0 \end{cases} \quad (1.1)$$

où A est une matrice $(m \times n)$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

1.2.1 Formes usuelles d'un programme linéaire

Un programme linéaire s'écrit sous forme canonique si l'ensemble des solutions réalisables est défini par un système d'inéquations linéaires et s'écrit sous la forme suivante :

$$(P) \begin{cases} \min c^t x \\ Ax \geq b \text{ ,} \\ x \geq 0 \end{cases}$$

où $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et $c, x \in \mathbb{R}^n$.

Si l'ensemble des solutions réalisables est défini par un système d'équations linéaires, on dit que le programme est écrit sous forme standard.

On peut toujours mettre un programme linéaire quelconque sous forme standard en introduisant des variables supplémentaires appelées "variables d'écart". Pour cette raison,

on ne considérera dans ce qui suit, que des programmes linéaires sous forme standard :

$$(PL) \begin{cases} \min c^t x = z^* \\ Ax = b \\ x \geq 0 \end{cases},$$

où A est une matrice de type (m, n) supposée de plein rang ($rg(A) = m < n$),

$c \in \mathbb{R}^n$, est le vecteur coût.

$b \in \mathbb{R}^m$, est le second membre.

$c^t x = \sum_{j=1}^n c_j x_j$, est la fonction objectif .

Remarque 1

Il n'est pas restrictif de supposer $x \geq 0$. En effet, si la variable x n'est pas contrainte en signe on pourra remplacer x par la différence $x = x^+ - x^-$; $x^+ \geq 0$; $x^- \geq 0$, en fait $x^+ = \max[0, x]$, $x^- = \max[0, -x]$

1.2.2 Dual d'un programme linéaire

Pour tout problème (PL) , on définit son problème dual de la façon suivante :

$$(D) \begin{cases} \max b^t y \\ A^t y \leq c \end{cases},$$

ce qui est équivalent à :

$$(D) \begin{cases} \max b^t y \\ A^t y + s = c \\ s \geq 0 \end{cases},$$

où s désigne une variable d'écart. Les ensembles des solutions réalisables de (PL) et (D) seront notés respectivement F_p et F_D . Ainsi, on a :

$$F_p = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\},$$

et

$$F_D = \{(y, s) \in \mathbb{R}^m \times \mathbb{R}^m, A^t y + s = c, s \geq 0\}.$$

- Si l'un des problèmes (PL) ou (D) possède une solution optimale finie, il en est de même pour l'autre et leurs valeurs optimales sont égales.
- Si l'un des problèmes (PL) ou (D) est non borné, alors le domaine réalisable de l'autre est vide.

Définition 8

Soient x une solution réalisable pour (PL) et (y, s) une solution réalisable pour (D) . On appelle **marge duale**, la quantité suivante :

$$c^t x - b^t y = x^t c - x^t A^t y = x^t (c - A^t y) = x^t s.$$

Notons que l'optimalité de (PL) et (D) est caractérisée par $x^t s = 0$.

1.2.3 Complexité d'un programme linéaire

Taille du problème

Soit le problème :

$$(PL) \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases} .$$

Deux constantes l et L vont jouer un rôle important dans ce qui suit telles que :

l est la longueur totale des données dans une représentation binaire qui est la sommation des nombres de bits utilisés pour l'entrée des données A , b et c , et $L = 2l + n + 1$.

Comme le déterminant d'une matrice est dominé par le produit des normes de ses colonnes alors, $|\det M| < 2^l$ pour n'importe quelle matrice carrée M de $[A, b]$.

Le lemme suivant montre que 2^{-L} est un nombre très petit [7].

Lemme 1

Soit u^* la valeur optimale de (PL). Si x est un point extrême de F_p , alors on a :

1. Pour tout $i = 1, \dots, n$, $x_i = 0$ ou $x_i > 2^{-L}$.
2. $c^t x = u^*$ ou $c^t x > u^* + 2^{-L}$.
3. Si F_p est borné, alors pour tout $x \in F_p$, on a $c^t x \leq 2^L$

Remarque 2

Ce lemme signifie que si on trouve un point extrême avec un coût plus petit que $u^* + 2^{-L}$, nécessairement ce point est une solution optimale du problème (PL).

1.2.4 Méthode de résolution d'un programme linéaire

1) Méthode du simplexe

Cette méthode a été développée à la fin des années 40 par G. Dantzig. Elle évolue sur la frontière du domaine réalisable de sommet en sommet adjacent, en réduisant la valeur de l'objectif jusqu'à l'optimum. Un critère simple permet de reconnaître le sommet optimal. Le nombre de sommet étant fini, l'algorithme ainsi défini converge en un nombre fini d'itération n'excédant pas le nombre C_n^m , sous l'hypothèse que tous les sommets visités sont non dégénérés. Dans le cas dégénéré l'algorithme risque de cycler, cependant il existe des techniques convenables pour éviter ce phénomène. En général, la méthode du simplexe possède un comportement numérique très satisfaisant confirmé par ses applications multiples dans la résolution d'une large classe de problèmes pratiques. En théorie, la méthode n'a pas autant de succès, elle est plutôt jugée inefficace par sa complexité arithmétique exponentielle de l'ordre $O(2^n)$ opérations.

2) Méthodes de points intérieurs

Les méthodes de points intérieurs sont liées directement aux premiers développements de la programmation non linéaire avec contraintes au milieu des années 50. Elles sont largement utilisées au cours des années 60 en forme de méthodes de fonctions barrières

pour résoudre des programmes non linéaires. Leur utilisation pour la programmation linéaire n'a pas reçu autant d'enthousiasme à cause de la dominance totale de la méthode du simplexe à cette époque. Les méthodes de points intérieurs partent d'un point intérieur au domaine des solutions réalisables, puis au moyen d'une stratégie fixée détermine une valeur approchée de la solution optimale. Les avantages de ces méthodes par rapport à la méthode du simplexe sont nombreux : robustesse, complexité polynomiale est convergence rapide pour les problèmes réels de grandes tailles. Il y a principalement trois catégories de méthodes de points intérieurs : Les méthodes affines, celles de réduction du potentiel et celles de trajectoire centrale.

N. B : Complexité de certains algorithmes

En 1972, Klee et Minty ont donné un exemple pratique qui montre que l'algorithme du simplexe est de complexité exponentielle. En 1979, Kachiyan a publié le premier algorithme polynomial pour la programmation linéaire, sa complexité était de $O(n^{4L})$ opérations arithmétiques avec n la dimension de l'espace et L la longueur des données définie précédemment. Cette borne de complexité a été réduite en 1984 par Karmarkar. L'algorithme de Karmarkar résout le problème de programmation linéaire en $O(nL)$ itérations, chaque itération nécessite la résolution d'un système linéaire dans \mathbb{R} . La complexité de chaque itération est de $O(n^4)$ opérations mais cette borne a été réduite par Karmarkar à $O(n^{3.5})$ opérations en utilisant des techniques d'inversion. En 1986, Renegar a décrit le premier algorithme de $O(\sqrt{n}L)$ itérations. Cet algorithme avait la même complexité totale que celui de Karmarkar, c'est-à-dire $O(n^{3.5})$ opérations. Son approche était basée sur le suivi de la trajectoire centrale. En 1987, Gonzaga et Vaidya ont obtenu simultanément des algorithmes avec une complexité de $O(n^3L)$ opérations, le premier algorithme utilisait la fonction barrière traditionnelle et le deuxième était une extension des résultats de Renegar.

Depuis 1987, avec l'article de Kojima, Mizuno et Yoshise, la scène a été graduellement dominée par les algorithmes primaux-duaux. Ces algorithmes sont efficaces dans la pratique et peuvent s'étendre à d'autres types de problèmes (problèmes quadratiques, pro-

blèmes convexes,...), leur complexité est de l'ordre $O(\sqrt{n}L)$ itérations, cette complexité a été obtenue en premier par Kojima et al. et aussi par Monteiro et Adler. Jusqu'à maintenant la meilleure borne de complexité obtenue pour des problèmes de programmation linéaire est de $O(n^3L)$ opérations arithmétiques.

1.2.5 Applications de la programmation linéaire

La programmation linéaire est de loin la branche de la programmation mathématique la plus utilisée dans les applications pratiques, voici quelques problèmes qui se traitent par la résolution d'un programme linéaire.

- La planification de production, où l'on dispose de plusieurs machines et d'une capacité finie de production à répartir entre plusieurs type de produits
- Si l'on attribue un gain unitaire à chaque produit fabriqué, la programmation linéaire fournira la composition du plan de production idéal (entraînant un bénéfice maximum).
- Le problème de mélange : par exemple, une firme pétrolière dispose de plusieurs sortes de pétrole brut de différentes qualités. Etant données les contraintes de teneurs minimales et maximales en certain types d'hydrocarbures (dépendant de l'utilisation envisagé du pétrole), l'objectif de programme linéaire sera de maximiser le profit (c'est à dire en employant les sortes les moins coûteuses) tout en respectant les contraintes de composition et de disponibilité.
- Le problème de transport : minimiser le coût total de transport de biens entre plusieurs centres de productions (ou dépôts) et plusieurs centres de consommation (ou de vente), sachant que les coûts de transport spécifiques à chaque couple centre de production /centre de consommation (dépendant principalement de la distance entre les deux cites) sont donnés
- Certains problèmes non linéaires peuvent être approchés par un programme linéaire.

Chapitre 2

Problème de transport à quatre indices avec capacités

2.1 Introduction

Dans ce chapitre, on donne une présentation générale du problème de transport à quatre indices avec capacités à savoir : Définitions, propriétés du problème, conditions de réalisabilité, conditions d'optimalité et en fin des méthodes pour le résoudre.

2.2 Position du problème

Le problème de transport à quatre indices avec capacités *PT4C* est formulé comme suit :

$$\begin{aligned}
 \text{Minimiser } Z &= \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q c_{ijkl} x_{ijkl} \\
 &\text{sous les contraintes} \\
 \sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} &= \alpha_i \text{ pour tout } i = 1, \dots, m \\
 \sum_{i=1}^m \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} &= \beta_j \text{ pour tout } j = 1, \dots, n \\
 \sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^q x_{ijkl} &= \gamma_k \text{ pour tout } k = 1, \dots, p \\
 \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p x_{ijkl} &= \delta_l \text{ pour tout } l = 1, \dots, q \\
 0 &\leq x_{ijkl} \leq d_{ijkl} \text{ pour tout } (i, j, k, l)
 \end{aligned}$$

avec $\alpha_i, \beta_j, \gamma_k, \delta_l, d_{ijkl}$ et c_{ijkl} sont donnés tels que pour tout (i, j, k, l) , on ait $\alpha_i > 0, \beta_j > 0, \gamma_k > 0, \delta_l > 0, d_{ijkl} > 0$ et $c_{ijkl} \geq 0$ et $M = m + n + p + q$ et $N = mnpq$.

Le *PT4C* est équivalent à un problème de programmation linéaire à variables bornées : [16]

$$(PL_{vb}) \quad \begin{cases} \min c^t x \\ Ax = b \\ 0 \leq x \leq d \end{cases},$$

où $x = (x_{1111}, \dots, x_{mnpq}) = (x_{ijkl}) \in \mathbb{R}^N$, $c = (c_{1111}, \dots, c_{mnpq}) = (c_{ijkl}) \in \mathbb{R}^N$

$b = (\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_p, \delta_1, \dots, \delta_q) \in \mathbb{R}^M$, $d = (d_{1111}, \dots, d_{mnpq}) = (d_{ijkl}) \in \mathbb{R}^N$,

A est une $M \times N$ matrice des M premières contraintes du *PT4C*.

avec $i = 1, \dots, m, j = 1, \dots, n, k = 1, \dots, p, l = 1, \dots, q$.

Le (PL_{vb}) est équivalent à un problème en forme standard doté d'une structure particulière :

$$(\widehat{PLT}) \quad \begin{cases} \min \widehat{c}^t \widehat{x} \\ \widehat{A} \widehat{x} = \widehat{b} \\ \widehat{x} \geq 0 \end{cases}$$

où $y = (y_1, \dots, y_N) = (y_\tau) \in \mathbb{R}^N$ (y_τ ; variables d'écart $\tau = 1, \dots, N.$), $\widehat{x} = (x, y)^t$,

$\widehat{c} = (c, 0)^t$, 0 est un N -vecteur nul, $\widehat{b} = (b, d)^t$, \widehat{A} est une $(M + N) \times 2N$ matrice correspondant $Ax = b$ et $x_{ijkl} + y_\tau = d_{ijkl}$,

On remarque que :

$$\widehat{A} = \begin{bmatrix} A & 0 \\ I_N & I_N \end{bmatrix},$$

où I_N est la matrice d'identité d'ordre N .

2.3 Définitions et propriétés

Définition 9

Une solution réalisable x d'un PT4C est dite **de base** si les colonnes de la matrice A_x obtenue de A en gardant seulement les colonnes correspondantes aux variables x_{ijkl} vérifiant : $0 < x_{ijkl} < d_{ijkl}$ sont linéairement indépendantes, le 4-uplet (i, j, k, l) associé à une solution réalisable de base est appelé **case intéressante**.

Définition 10

Une solution réalisable de base est dite **non dégénérée** si

$$rg(A_x) = rg(A)$$

Propriétés de la matrice des contraintes,[16], [17]

On suppose que la condition de réalisabilité suivante est vérifiée

$$\sum_{i=1}^m \alpha_i = \sum_{j=1}^n \beta_j = \sum_{k=1}^p \gamma_k = \sum_{l=1}^q \delta_l = H \quad (2.1)$$

d'où

$$\begin{aligned} rg(A) &= M - 3 \\ rg(\widehat{A}) &= M + N - 3 \end{aligned}$$

Contrairement au problème de transport à deux indices, la matrice A n'est pas totalement uni modulaire car elle possède des mineurs qui n'appartiennent pas à l'ensemble $\{-1, 0, 1\}$.

Tableau de transport [16], [17]

Les données du problème en question sont présentées sous forme d'un tableau appelé « tableau du *PT4C* »

d_{1111}	d_{1211}	...	d_{mnpq}	
.	.		.	
c_{1111}	c_{1211}	...	c_{mnpq}	
.	.		.	
x_{1111}	x_{1211}	...	x_{mnpq}	
.	.		.	
1	1	...	0	α_1 .
:	:	...	:	:
0	0	...	1	α_m .
1	0	...	0	β_1 .
0	1	...	0	β_2 .
:	:	...	:	:
0	0	...	1	β_n .
1	1	...	0	γ_1 .
:	:	...	:	:
0	0	...	1	γ_p .
1	1	...	0	δ_1 .
:	:	...	:	:
0	0	...	1	δ_q .

2.4 Conditions de réalisabilité

Théorème 4 [16], [17]

- Une condition nécessaire pour que le problème PT4C possède une solution réalisable est que la condition (2.1) et les conditions suivantes

$$\begin{aligned}\alpha_i &\leq \sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q d_{ijkl} \quad \text{pour tout } i = 1, \dots, m, \\ \beta_j &\leq \sum_{i=1}^m \sum_{k=1}^p \sum_{l=1}^q d_{ijkl} \quad \text{pour tout } j = 1, \dots, n, \\ \gamma_k &\leq \sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^q d_{ijkl} \quad \text{pour tout } k = 1, \dots, p, \\ \delta_l &\leq \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p d_{ijkl} \quad \text{pour tout } l = 1, \dots, q,\end{aligned}$$

soient vérifiées.

- Une condition suffisante pour que le problème PT4C possède une solution réalisable est que la condition (2.1) et la condition suivante

$$\frac{\alpha_i \beta_j \gamma_k \delta_l}{d_{ijkl}} \leq H^3 \quad \text{pour tout } (i, j, k, l)$$

soient vérifiées.

2.5 Conditions d'optimalité

Théorème 5 [16], [17]

Supposons que PT4C est réalisable, alors une solution réalisable x de PT4C est optimale si et seulement s'il existe un vecteur

$$(u_1, \dots, u_m, v_1, \dots, v_n, \dots, w_1, \dots, w_p, \dots, t_1, \dots, t_q)^t \in \mathbb{R}^N,$$

tel que :

$$u_i + v_j + w_k + t_l \leq c_{ijkl} \quad \text{si } x_{ijkl} = 0$$

$$u_i + v_j + w_k + t_l = c_{ijkl} \quad \text{si } 0 < x_{ijkl} < d_{ijkl}$$

et

$$u_i + v_j + w_k + t_l \geq c_{ijkl} \quad \text{si } x_{ijkl} = d_{ijkl}.$$

2.6 Méthodes de résolution d'un problème de transport à quatre indices avec capacités

2.6.1 Méthode du simplexe

Introduction

En 1947 Dantzig a proposé une méthode du simplexe, c'est une technique de résolution très efficace des problèmes de programmation linéaire.

Depuis ce temps la programmation linéaire a suscité un grand intérêt chez les chercheurs qui ont écrit des centaines de livres et ont publié des milliers d'articles sur le sujet. Bien que la complexité de cette méthode soit exponentielle, dans la pratique elle s'est montrée très efficace et d'une grande utilité, surtout dans les domaines de la planification et de l'organisation. Dans cette partie, on commence par une brève description de la méthode du simplexe. Ensuite on étudie le fonctionnement de l'algorithme du simplexe et enfin on donne un résumé de la méthode [13], [10].

Cette méthode permet de résoudre des problèmes avec autant de variables et de contraintes qu'on veut à l'aide de l'ordinateur.

Soit le problème de programmation linéaire suivant :

$$(PL) \quad \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases},$$

où : c, x de \mathbb{R}^n , $A \in \mathbb{R}^{m \times n}$. supposée de plein rang ($rg(A) = m < n$).

On note : $P = \{x; Ax = b, x \geq 0\}$

l'idée de l'algorithme :

Soit x_0 une solution de base admissible.

Pour $k = 0, \dots$ faire trouver x_{k+1} solution de base admissible voisine telle que

$c^t x_{k+1} < c^t x_k$ jusqu'à ce qu'aucune solution de base admissible voisine n'améliore l'objectif. On trouve alors un minimum local, en programmation linéaire

minimum local = minimum globale.

Soit $x \in P$. On va se déplacer le long d'une direction $d \in \mathbb{R}^n$ qui doit nous maintenir dans P .

Définition 11

Soit x un élément d'un polyèdre P . Un vecteur $d \in \mathbb{R}^n$ est appelé *direction admissible* en x s'il existe un scalaire positif θ tel que $(x + \theta d) \in P$.

Soit x une solution de base admissible, soit B la matrice de base associée, alors on peut toujours mettre A sous la forme :

$A = [B \ N]$, où N est la sous matrice de A d'ordre $m \times (n - m)$ formée par les colonnes qui ne figurent pas dans la base. De même on peut partitionner x en $\begin{bmatrix} x_B \\ x_N \end{bmatrix}$,

c en $\begin{bmatrix} c_B \\ c_N \end{bmatrix}$, et d en $\begin{bmatrix} d_B \\ d_N \end{bmatrix}$.

Soient $B(1), \dots, B(m)$ les indices des variables de base, alors $B = [A_{B(1)}, \dots, A_{B(m)}]$;

($A_{B(i)}$ étant la colonne d'ordre $B(i)$ de la matrice A , $i = 1, \dots, m$).

Le système $Ax = b$, s'écrit ainsi : $Bx_B + Nx_N = b$.

La solution x de $Ax = b$, est alors obtenue en faisant $x_N = 0$, elle est déterminée de manière unique par :

$$x_B = B^{-1}b = (x_{B(1)}, \dots, x_{B(m)}).$$

Comment déterminer $(x + \theta d)$?

Choisir une variable x_j hors base (qui vaut 0), et augmenter sa valeur jusqu'à θ , tout en gardant les autres variables hors base à zéro.

Donc :

$$d_j = 1$$

$d_i = 0, i \neq j, i$ indice hors base, alors $d_N = e_j$; ($e_j \in R^{(n-m)}$ le vecteur canonique),

et par conséquent $d = \begin{bmatrix} d_B \\ e_j \end{bmatrix}$.

Il faut rester admissible :

$$A(x + \theta d) = b$$

$$Ax + \theta Ad = b$$

- x est admissible, et donc $Ax = b$.
- Pour que $(x + \theta d)$ soit admissible, il faut que $Ad = 0$.

$$\begin{aligned} Ad &= 0 \\ [B \ N] \begin{bmatrix} d_B \\ e_j \end{bmatrix} &= 0 \\ Bd_B + Ne_j &= 0 \end{aligned}$$

$$Bd_B + A_j = 0; (A_j \text{ étant la colonne d'ordre } j \text{ de la matrice } A)$$

$$Bd_B + A_j = 0.$$

Nous obtenons : $d_B = -B^{-1}A_j$.

La direction d ainsi obtenue est appelée $j^{\text{ième}}$ direction de base, elle garantit que les

contraintes d'égalité seront vérifiées lorsque l'on s'éloigne de x le long de d .

Qu'en est-il des contraintes de non négativité ?

Variables hors-base :

- x_j était nulle et devient positive
- $x_i, i \neq j$, restent à zéro

Variables de base :

Si x est une solution de base admissible non dégénérée, alors $x_B > 0$. Lorsque θ est suffisamment petit $(x_B + \theta d_B) \geq 0$.

Si x est une solution de base admissible dégénérée, alors d n'est pas toujours une direction admissible. C'est le cas lorsque qu'une variable de base $x_i = 0$ et que la composante correspondante d_i de la direction est négative.

Si x est une solution de base admissible non dégénérée, la $j^{\text{ième}}$ direction de base en x est admissible, pour tout j indice de base.

Conditions d'optimalité

Définition 12

Soit x une solution de base, soit B la matrice de base associée, et c_B le vecteur de coût pour les variables de bases. Pour chaque j , le coût réduit est défini par $\bar{c}_j = c_j - c_B^t B^{-1} A_j$, $j = 1, \dots, n$.

Que vaut le coût réduit pour les variables de base ?

Soit $B(i)$ indice d'une variable en base, le coût réduit est $c_{B(i)} - c_B^t B^{-1} A_{B(i)}$

- $B = [A_{B(1)}, \dots, A_{B(m)}]$, $B^{-1} [A_{B(1)}, \dots, A_{B(m)}] = I_m$,

(I_m la matrice identité d'ordre m)

- $B^{-1} A_{B(i)} = e_i$; (e_i est la $i^{\text{ième}}$ colonne de I_m).

- $\bar{c}_{B(i)} = c_{B(i)} - c_B^t B^{-1} A_{B(i)}$

$$= c_{B(i)} - c_{B(i)}^t e_i$$

$$= c_{B(i)} - c_{B(i)}^t$$

$= 0$, le coût réduit des variables de base est nul.

Théorème 6

Considérons une solution de base admissible x , associée à une matrice de base B . Soit \bar{c} le vecteur de coûts réduits correspondant.

1. Si $\bar{c} \geq 0$, alors x est optimal
2. Si x est optimal et non dégénérée, alors $\bar{c} \geq 0$.

Une itération de la méthode du simplexe :

1. Soit une base $B = [A_{B(1)}, \dots, A_{B(m)}]$ et x une solution de base admissible associée à B .

2. Calculer les coûts réduits pour chaque indice j hors base : $\bar{c}_j = c_j - c_B^t B^{-1} A_j$

S'ils sont tous non négatifs, la solution courante est optimale. **STOP.**

3. Si non choisir j tel que $\bar{c}_j < 0$, et calculer $d_B = -B^{-1} A_j$.

Si aucune composante de d_B n'est négative, alors le coût optimal est infini.

STOP.

4 Calculer $\theta^* = \min_{\{i=1, \dots, m; d_{B(i)} < 0\}} \frac{-x_{B(i)}}{d_{B(i)}} = \frac{-x_{B(k)}}{d_{B(k)}}$

5 .Former une nouvelle base en remplaçant $A_{B(k)}$ par A_j .

Si y est la nouvelle solution de base admissible, les valeurs des nouvelles variables de base sont :

- $y_j = \theta^*$
- $y_{B(i)} = x_{B(i)} + \theta^* d_{B(i)}, i \neq k$.

Remarque 3

Etape 3 : Choisir j tel que $\bar{c}_j < 0$, l'algorithme ne spécifie pas quelle variable choisir pour rentrer dans la base, plusieurs règles existent.

Retenons la règle de Bland : Parmi les j tels que $\bar{c}_j < 0$, choisir l'indice le plus petit

Détermination d'une solution initiale de base

Pour initialiser l'algorithme du simplexe, il faut disposer d'une solution de base réalisable initiale, or une telle solution de base initiale n'est pas nécessairement disponible. De plus, en général, nous ne savons même pas si le domaine réalisable n'est pas vide.

Soit à résoudre le problème de programmation linéaire suivant :

$$(PL) \quad \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases},$$

où : c, x de \mathbb{R}^n , $A \in \mathbb{R}^{m \times n}$. supposée de plein rang ($rg(A) = m < n$).

Introduire les variables artificielles t_1, \dots, t_m , et appliquer la méthode du simplexe au problème auxiliaire

$$(PA) \quad \begin{cases} \min \sum_{i=1}^m t_i = w \\ Ax + t = b \\ x, t \geq 0 \end{cases},$$

si la valeur optimale w est positive (i.e., $w > 0$), alors le domaine réalisable du problème original est vide (i.e., le problème original n'est pas réalisable), si la valeur optimale w est nulle (i.e., $w = 0$), alors le domaine réalisable du problème original n'est pas vide (i.e., le problème original est réalisable). Dans ce cas, les valeurs que prennent les variables x_i constituent une solution pour le problème original où toutes les variables artificielles t_i sont égales à 0.

Tableau du simplexe et implémentation

Maintenir en permanence : $B^{-1}[A, b] = [B^{-1}A_1 \dots B^{-1}A_n, B^{-1}b]$, cette matrice s'appelle le tableau du simplexe.

- La dernière colonne contient les valeurs des variables en base.
- Les autres colonnes permettent de calculer \bar{c}_j et d_B .

Définition 13

Lors d'une itération du simplexe : Si j est l'indice de la variable qui entre en base, la colonne $B^{-1}A_j$ est appelée colonne du pivot.

Si la variable de base $B(k)$ sort de la base, la ligne k du tableau est appelée ligne du pivot.

L'élément qui se trouve sur la ligne du pivot et la colonne du pivot est appelé **le pivot**.

On augmente le tableau avec une ligne relative aux coûts : $[c^t - c_B^t B^{-1}A, -c_B^t B^{-1}b]$,

et par conséquent on a ;

- $-c_B^t B^{-1}b = -c_B^t x_B = -c^t x = -\text{fonction objectif}$,
- $c^t - c_B^t B^{-1}A$ coûts réduits.

Le tableau complet sera donc

$B^{-1}A$	$B^{-1}b$
$c^t - c_B^t B^{-1}A$	$-c_B^t B^{-1}b$

en posant : $B^{-1}A = (\bar{a}_{ij})$,

$B^{-1}b = (\bar{b}_i)$ sont les solutions

$\bar{c}_j = c^t - c_B^t B^{-1}A$ sont les coûts réduits

$-c_B^t B^{-1}b$ est la valeur de l'objectif

, avec $i = 1, \dots, m$ et $j = 1, \dots, n$.

Algorithme 1

1. Soit une matrice de base B , une solution de base admissible x et le tableau du simplexe associés.

2. Examiner les coûts réduits dans la dernière ligne du tableau.

S'ils sont tous **non négatifs**, la solution de base admissible courante est optimale.

STOP.

Si non, choisir j tel que $\bar{c}_j < 0$. qui devient \bar{c}_s ; (s indice de la colonne du pivot)

3. Si tous les $\bar{a}_{is} < 0$ le coût optimal est $-\infty$ **STOP.**

4. Pour chaque i tel que $(\bar{a}_{is} > 0)$, calculer $\left(\frac{\bar{b}_i}{\bar{a}_{is}}\right)$.

Soit r l'indice de la ligne correspondant au rapport le plus petit $\left(\frac{\bar{b}_r}{\bar{a}_{rs}}\right)$.
 (r l'indice de la ligne du pivot), la variable x_r sort de la base. la variable x_s rentre en base.

5. Pivoter autour du pivot (\bar{a}_{rs}) . Aller à l'étape (2).

Fin Algorithme.

2.6.2 Méthode de Ye-Lustig (méthode de points intérieurs)

Introduction

Malgré sa complexité exponentielle, l'algorithme du Simplexe est resté longtemps l'algorithme de référence pour la programmation linéaire. La méthode est demeurée sans compétiteurs sérieux jusqu'en 1984 où un nouvel algorithme polynomial, l'algorithme des méthodes projectives a été proposé par Narendra Karmarkar aux laboratoires de AT&T Bell aux Etas Unis. Mais la mise en place et les transformations nécessaires pour résoudre un problème standard de programmation linéaire par la méthode de Karmarkar sont demeurées secrètes jusqu'en 1985. Cet algorithme a été essentiellement différent de la méthode du simplexe par le fait que, dans cette dernière, on se déplace en suivant la frontière du domaine réalisable, alors que pour la méthode de Karmarkar, on progresse tout en restant strictement à l'intérieur du domaine réalisable, d'où la connotation "intérieur", et c'était grâce à lui qu'une recherche intense dans les méthodes du points intérieurs s'est déclenchée et a donné comme résultat une grande variété d'algorithmes de ce type. L'algorithme de Karmarkar avait un grand intérêt théorique puisqu'il est polynomial; en effet, il exige un $o(n^3)$ opérations arithmétiques, d'où une complexité totale de $o(n^4L)$ opérations. Karmarkar a même été capable de la réduire à $o(n^{3.5}L)$ en utilisant un processus de mise à jour partiel.

Ces méthodes projectives n'ont pas été présentées seulement par Karmarkar mais aussi par Todd et Burell, Anstreicher, Gay et Ye et Kojima[15]. Ces chercheurs ont suggéré des modifications de l'algorithme original de Karmarkar afin d'aboutir à un algorithme très efficace et plus utile dans la pratique. Ces méthodes ont été mise en ouvre hors

des laboratoires Bell par Adler et al, Gill et al, Monma et Morton et Mcshane et al[12], ces différentes mise en oeuvre ont donné de bons résultats prouvant leur supériorité par rapport aux autres méthodes (simplexe, ellipsoïde ...).

Dans cette partie on étudie l'approche de Ye-Lustig.

L'approche de Ye-Lustig [11], [4], [2].

Soit le programme linéaire sous forme standard :

$$(PL) \quad \begin{cases} \min c^t x = z^* \\ Ax = b \\ x \geq 0 \end{cases},$$

où : c, x de \mathbb{R}^n , $A \in \mathbb{R}^{m \times n}$.

avec l'hypothèse suivante :

H1) La matrice des contraintes A est de plein rang ($rg(A) = m < n$)

Dans cette méthode, à chaque itération k , on utilise une transformation projective qui ramène la région admissible polyédrique $\{Ax = b, x \geq 0\}$ à un simplexe

$$S_{n+1} = \left\{ x \in \mathbb{R}_+^{n+1}, \sum_{i=1}^{n+1} x_i = 1 \right\}.$$

Cette transformation est définie par :

$$T_a : \mathbb{R}_+^n \longrightarrow S_{n+1}$$

$$T_a(x^k) = y = \begin{cases} y_i = \frac{\frac{x_i^k}{a_i}}{1 + \sum_{i=1}^n \frac{x_i^k}{a_i}}, \quad i = 1, \dots, n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i, \quad a \in \mathbb{R}_+^n \end{cases}.$$

Le problème (PL) devient alors :

$$P_t \begin{cases} \min(D_k c, -z^*)^t y \\ A_k y = 0 \\ y \in S_{n+1} = \left\{ y \in \mathbb{R}^{n+1}, y \geq 0, \sum_{i=1}^{n+1} y_i = 1 \right\} \end{cases},$$

où $A_k = [AD_k \ -b] \in \mathbb{R}^{(n+1) \times m}$; $D_k = \text{diag}(x^k)$, matrice diagonale.

Comme le calcul d'une solution optimale d'un programme linéaire, sur une sphère est évident, on introduit à chaque itération la plus grande sphère inscrite dans le simplexe S_{n+1} .

On obtient le sous problème de P_t suivant

$$P_s \begin{cases} \min(D_k c, -z^*)^t y \\ A_k y = 0 \\ \|y - e_{n+1}\|^2 \leq r^2 < 1 \end{cases},$$

où $r = \frac{1}{\sqrt{n(n+1)}}$, est le rayon de la sphère.

$\frac{e_{n+1}}{n+1} = \left(\frac{1}{n+1}, \dots, \frac{1}{n+1}\right)^t \in \mathbb{R}^{n+1}$, est le centre de la sphère.

Comme la valeur de z^* est généralement inconnue, Ye-Lustig approxime z^* à chaque itération par des bornes supérieures $z^k = c^t x^k > z^*$.

Le problème P_s est remplacé par :

$$(P_k) \begin{cases} \min(D_k c, -c^t x^k)^t y \\ A_k y = 0 \\ \|y - e_{n+1}\|^2 \leq r^2 < 1 \end{cases}.$$

En utilisant les conditions d'optimalités de KKT relatives au problème (P_k), la so-

lution optimale est donnée par :

$$y_k^* = \frac{e_{n+1}}{n+1} - \alpha^k r d^k,$$

où α^k est le pas de déplacement ; $0 < \alpha_k < 1$.

$d^k = \frac{p^k}{\|p^k\|}$, p^k est la projection du vecteur coût $(D_k c, -c^t x^k)^t$ sur le noyau de la matrice A_k ,

$$p^k = (I - A_k^t (A_k A_k^t)^{-1} A_k) (D_k c, -c^t x^k)^t$$

La vitesse de convergence de cette méthode dépend du calcul de la projection p^k , et aussi du pas de déplacement.

Calcul de la projection

Nous avons

$$p^k = (I - A_k^t (A_k A_k^t)^{-1} A_k) (D_k c, -c^t x^k)^t,$$

Posons

$$u_k = (A_k A_k^t)^{-1} A_k (D_k c, -c^t x^k)^t,$$

donc

$$p^k = (D_k c, -c^t x^k) - A_k^t u_k,$$

le calcul de p^k dépend ainsi de la manière par laquelle, on résout le système linéaire :

$$A_k A_k^t u_k = A_k (D_k c, -c^t x^k)^t,$$

dont la matrice $A_k A_k^t$ est symétrique définie positive.

Algorithme 2

Initialisation

$$x^0 > 0, k = 0$$

Tan que $\frac{\|d^k\|}{|c^t x^0|} > \varepsilon$ **faire**

1. Construire $D_k = \text{diag}(x^k)$, $A_k = [AD_k, -b]$, $B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$.

2. Calcul de la projection p_k

$$p_k = [I - B_k^t (B_k B_k^t)^{-1} B_k] (D_k c, -c^t x^k)^t$$

3. Normaliser la projection p_k

$$d^k = \frac{p_k}{\|p_k\|}, r = \frac{1}{\sqrt{n(n+1)}}$$

4. Calculer l'itéré suivant

$$y^k = \frac{e_{n+1}}{n+1} - \alpha^k r d^k, \alpha^k \text{ le pas de déplacement}$$

5. Revenir au problème original par T_k^{-1}

$$x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k [n]}{y_{n+1}^k}$$

$$k = k + 1$$

Fin tant que

Fin algorithme.

Calcul d'une solution initiale strictement réalisable

Un point strictement réalisable de (PL) est une solution du problème :

$$(PF) \begin{cases} Ax = b \\ x > 0 \end{cases},$$

En utilisant la technique de la variable artificielle, le problème (PF) est équivalent au problème d'optimisation suivant :

$$(P_\lambda) \begin{cases} \min \lambda \\ Ax + \lambda q = b \\ x > 0, \lambda \geq 0 \end{cases},$$

tel que : $q = b - Aa$ et $a \in \mathbb{R}_+^n$

(P_λ) est équivalent au programme linéaire :

$$(\tilde{P}\tilde{L}) \begin{cases} \min \tilde{c}^t \tilde{x} = w^* = 0 \\ \tilde{A}\tilde{x} = b \\ \tilde{x} \geq 0 \end{cases},$$

où $\tilde{x} = (x, \lambda)^t$, $\tilde{c} = (0, 0, 0, \dots, 1)^t \in \mathbb{R}^{n+1}$, $\tilde{A} = [A, b - Aa] \in \mathbb{R}^{m \times (n+1)}$.

Etant donné que $(a; 1)$ est une solution strictement réalisable de $(\tilde{P}\tilde{L})$ pour tout $a > 0$. Alors le calcul de la solution optimale de $(\tilde{P}\tilde{L})$ est donné par la phase 2 de l'algorithme de Ye-Lustig. L'algorithme correspondant se présente comme suit :

Algorithme d'initialisation

Initialisation $x^0 = a$, $\lambda^0 = 1$, $\tilde{x} = (x^0, \lambda^0)$ et $k = 0$

1) **Si** $\|Ax^0 - b\| \leq \varepsilon$ **Stop**; x^k est une solution approximative de (PF) .

Sinon aller à l'étape (2)

2) **Si** $\lambda^k \leq \varepsilon$ **Stop**; x^k est une solution approximative de (PF) .

Sinon aller à l'étape (3).

3) Poser $D_k = \text{diag}(\tilde{x}^k)$, $B = [A, b - Aa]$ et $r = \frac{1}{\sqrt{(n+1)(n+2)}}$

Calculer $p_k = [I - B_k^t (B_k B_k^t)^{-1} B_k] (D_k \hat{c}, -\tilde{c}^t \tilde{x}^k)^t$

Tel que $B_k = [BD_k, -b]$ et $\tilde{c} = (0, 0, 0, \dots, 1)^t \in \mathbb{R}^{n+1}$

Calculer $y^{k+1} = \frac{e_{n+2}}{n+2} - \alpha^k r \frac{p_k}{\|p_k\|}$, $\tilde{x}^{k+1} = (y_{n+2}^{k+1})^{-1} D_k y^{k+1} [n+1]$.

4) **Faire** $k = k + 1$ et retourner à l'étape (2).

Fin

2.6.3 Méthode de R. Zitouni et al. (AL_{PT4C})

Introduction

Le problème de transport est introduit pour la première fois par Hitchcock en 1941, traité et étudié en détail par Koopmans en 1947, L.V Kantrovitch et M.K.Savourine en 1949 et puis G.B.Dantzig en 1951. Le problème de transport à deux indices largement étudié dans la littérature, est un modèle générique pour les problèmes d'affectation et peut être formulé comme un programme linéaire avec une structure spéciale des contraintes. Dans sa forme classique, le problème de transport consiste à minimiser le coût de transport des marchandises disponibles en m sources (noeuds des disponibilités) et demandes pour n destinations (noeuds des demandes). Ensuite, l'étude est prolongée aux problèmes à un nombre d'indices supérieur à deux. Depuis les années soixante, plusieurs études ont été publiées sur le problème de transport à trois indices et plus général, sur celui à indices multiples sans capacités. Le problème de transport axial (de somme axiale) à l indices (PTl) n'a pas été considérablement étudié pour $l \geq 3$, le problème de transport à quatre indices avec capacités. Ce cas n'a pas été traité auparavant que ce soit dans ses aspects théoriques, algorithmiques ou numériques. En 2003, R. Zitouni et A. Keraghel ont introduit pour la première fois, une méthode de résolution d'un problème de transport à quatre indices avec capacités [17], [18].

Cet algorithme est une variante de l'algorithme du simplexe, adapté pour le $PT4C$ [16], qu'on le note AL_{PT4C} , Il est décrit comme suit :

Algorithme 3

Initialisation :

Pour tout (i, j, k, l) , $b_{ijkl} = 0$ (b_{ijkl} est une variable booléenne qui vaut 1 si x_{ijkl} est déjà déterminée et 0 dans le cas contraire),

$$E = \{(i, j, k, l), \text{ tel que } b_{ijkl} = 0\}.$$

Itération :

Tant que $E \neq \Phi$ faire :

- Choisir un 4-uplet $(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \in E$, tel que $c_{\bar{i}\bar{j}\bar{k}\bar{l}} = \min_{(i,j,k,l) \in E} c_{ijkl}$,

- Prendre $x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \min(\hat{\alpha}_{\bar{i}}, \hat{\beta}_{\bar{j}}, \hat{\gamma}_{\bar{k}}, \hat{\delta}_{\bar{l}}, d_{\bar{i}\bar{j}\bar{k}\bar{l}})$, et $b_{\bar{i}\bar{j}\bar{k}\bar{l}} = 1$,

(i.e., $x_{\bar{i}\bar{j}\bar{k}\bar{l}}$ est déterminée),

- Actualiser $\alpha_{\bar{i}}$, $\beta_{\bar{j}}$, $\gamma_{\bar{k}}$, et $\delta_{\bar{l}}$ comme suit :

$$1) \hat{\alpha}_{\bar{i}} = \alpha_{\bar{i}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}},$$

si $\hat{\alpha}_{\bar{i}} = 0$ alors prendre $x_{ijkl} = 0$ pour tout $(j, k, l) \neq (\bar{j}, \bar{k}, \bar{l})$ et $b_{ijkl} = 1$ pour tout (j, k, l) ,

$$2) \hat{\beta}_{\bar{j}} = \beta_{\bar{j}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}},$$

si $\hat{\beta}_{\bar{j}} = 0$ alors prendre $x_{ijkl} = 0$ pour tout $(i, k, l) \neq (\bar{i}, \bar{k}, \bar{l})$ et $b_{ijkl} = 1$ pour tout (i, k, l) ,

$$3) \hat{\gamma}_{\bar{k}} = \gamma_{\bar{k}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}},$$

si $\hat{\gamma}_{\bar{k}} = 0$ alors prendre $x_{ij\bar{k}l} = 0$ pour tout $(i, j, l) \neq (\bar{i}, \bar{j}, \bar{l})$ et $b_{ij\bar{k}l} = 1$ pour tout (i, j, l) ,

$$4) \hat{\delta}_{\bar{l}} = \delta_{\bar{l}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}}.$$

si $\hat{\delta}_{\bar{l}} = 0$ alors prendre $x_{ij\bar{k}\bar{l}} = 0$ pour tout $(i, j, k) \neq (\bar{i}, \bar{j}, \bar{k})$ et $b_{ij\bar{k}\bar{l}} = 1$ pour tout (i, j, k) .

- Calculer ε avec la procédure décrite en (P1) ci-dessous,

Si $\varepsilon = 0$ Aller en Phase 2.

Si non faire appeler la procédure (P2).

Phase 2 : Amélioration d'une solution de base d'un PT4C

Au début de la **Phase 2**, on connaît une solution initiale de base.

- a) Déterminer l'ensemble $I^{(r)}$ des cases (i, j, k, l) intéressantes
- b) Pour tout $(i, j, k, l) \in I^{(r)}$, résoudre le système linéaire

$$u_i^{(r)} + v_j^{(r)} + w_k^{(r)} + t_l^{(r)} = c_{ijkl}.$$

- c) Pour tout $(i, j, k, l) \notin I^{(r)}$ déterminer

$$\Delta_{ijkl}^{(r)} = c_{ijkl} - (u_i^{(r)} + v_j^{(r)} + w_k^{(r)} + t_l^{(r)})$$

Si la condition d'optimalité suivante est vérifiée

$$\begin{aligned} \Delta_{ijkl}^{(r)} &\geq 0 \text{ pour tout } \Delta_{ijkl}^{(r)} \in \Gamma_0^{(r)} \\ \Delta_{ijkl}^{(r)} &\leq 0 \text{ pour tout } \Delta_{ijkl}^{(r)} \in \Gamma_d^{(r)}, \end{aligned}$$

où $\Gamma_0^{(r)}$ et $\Gamma_d^{(r)}$ sont deux tableaux tels que

$$\begin{aligned} \Gamma_0^{(r)} &= \left\{ \Delta_{ijkl}^{(r)} \text{ tel que } x_{ijkl}^{(r)} = 0 \right\}, \\ \Gamma_d^{(r)} &= \left\{ \Delta_{ijkl}^{(r)} \text{ tel que } x_{ijkl}^{(r)} = d_{ijkl} \right\}, \end{aligned}$$

alors, la solution $x^{(r)}$ est optimale. **Fin.**

- d) Déterminer

$$\begin{aligned} \Delta_{i_0 j_0 k_0 l_0}^{(r)} &= \min_{ijkl} \left[\Delta_{ijkl}^{0(r)}, -\Delta_{ijkl}^{d(r)} \right] \text{ tel que :} \\ \Delta_{ijkl}^{0(r)} &\in \Gamma_0^{(r)} \text{ avec } \Delta_{ijkl}^{0(r)} < 0, \\ \Delta_{ijkl}^{d(r)} &\in \Gamma_d^{(r)} \text{ avec } \Delta_{ijkl}^{d(r)} > 0, \end{aligned}$$

et spécifier si $\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_0^{(r)}$ où $\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_d^{(r)}$

e) Construire par la procédure décrite en (P5) ci-dessous,
un cycle $\mu^{(r)}$ contenant quelques cases intéressantes (i, j, k, l)
et la case non intéressante (i_0, j_0, k_0, l_0) correspondante à $\Delta_{i_0 j_0 k_0 l_0}^{(r)}$.

Prendre

$\sigma^{(r)} = \{(i, j, k, l) \text{ tel que } (i, j, k, l) : \text{case formant le cycle}\}$

$\sigma^{(r)-} = \{(i, j, k, l) \text{ tel que } (i, j, k, l) \in \sigma^{(r)}, \text{ avec } \alpha_{ijkl} < 0\}$

$\sigma^{(r)+} = \{(i, j, k, l) \text{ tel que } (i, j, k, l) \in \sigma^{(r)}, \text{ avec } \alpha_{ijkl} > 0\}$

Si $\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_0^{(r)}$, déterminer le nouvel itéré par la procédure (P3) ci-dessous :

Sinon ($\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_d^{(r)}$), déterminer le nouvel itéré par la procédure (P4) ci-dessous :

f) Répéter **a)**, . . . , **e)** jusqu'à ce que la condition d'optimalité soit vérifiée.

Fin algorithme.

L'algorithme Al_{PT4C} fait appel aux procédures suivantes :

(P1) **Calcul de ε** :

$$\varepsilon = \sum_{i=1}^m a_i = \sum_{j=1}^n b_j = \sum_{k=1}^p e_k = \sum_{l=1}^q f_l$$

tel que :

$$\begin{aligned}
a_i &= \alpha_i - \sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} & \text{avec } i &= 1, \dots, m, \\
b_j &= \beta_j - \sum_{i=1}^m \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} & \text{avec } j &= 1, \dots, n, \\
e_k &= \gamma_k - \sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^q x_{ijkl} & \text{avec } k &= 1, \dots, p, \\
f_l &= \delta_l - \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p x_{ijkl} & \text{avec } l &= 1, \dots, q.
\end{aligned}$$

(P2) Construire un problème ($PT4C(\tilde{M})$), voir remarque 3, et trouver une solution réalisable initiale de base $\bar{x}^{(0)}$ pour le problème ($PT4C(\tilde{M})$), par la même procédure qu'en étape 1.

Si $\bar{x}^{(0)}$ est optimale alors le problème ($PT4C$) est non réalisable. **Fin.**

Amélioration d'une solution réalisable de base pour ($PT4C(\tilde{M})$)

Initialisation : $\varepsilon > 0$ connu, $r = 1$.

- 1) Déterminer $\bar{x}^{(r)}$, (Phase 2)
- 2) Si $x_{m+1n+1p+1q+1} = \varepsilon$, alors $x^{(r)} = (x_{ijkl}^{(r)})$ avec $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, p$, et $l = 1, \dots, q$, est une solution initiale de base pour le problème $PT4C$, **Aller en Phase 2.**
- 3) Si $x^{(r)}$ est optimale (Phase 2), alors le problème ($PT4C$) est non réalisable. **Fin**
- 4) Prendre $r = r + 1$ et répéter 1), jusqu'à 3).

(P3) **Détermination d'un nouvel itéré le cas où $\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_0^{(r)}$**

$$\begin{aligned}
\theta_1^{(r)} &= \min_{(ijkl) \in \sigma^{(r)-}} \left(\frac{x_{ijkl}^{(r)}}{-\alpha_{ijkl}} \right) \\
\theta_2^{(r)} &= \min_{(ijkl) \in \sigma^{(r)+}} \left(\frac{d_{ijkl} - x_{ijkl}^{(r)}}{\alpha_{ijkl}} \right) \\
\theta^{(r)} &= \min(\theta_1^{(r)}, \theta_2^{(r)}).
\end{aligned}$$

Ensuite, prendre

$$x^{(r+1)} = \left\{ x_{ijkl}^{(r)} + \alpha_{ijkl} \theta^{(r)}, (i, j, k, l) \in \sigma^{(r)} \right\} \cup \left\{ x_{ijkl}^{(r)}, (i, j, k, l) \notin \sigma^{(r)} \right\}$$

(P4) **Détermination d'un nouvel itéré le cas où** $\Delta_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_d^{(r)}$

$$\begin{aligned} \theta_1^{(r)} &= \min_{(ijkl) \in \sigma^{(r)+}} \left(\frac{x_{ijkl}^{(r)}}{\alpha_{ijkl}} \right) \\ \theta_2^{(r)} &= \min_{(ijkl) \in \sigma^{(r)-}} \left(\frac{d_{ijkl} - x_{ijkl}^{(r)}}{-\alpha_{ijkl}} \right) \\ \theta^{(r)} &= \min(\theta_1^{(r)}, \theta_2^{(r)}). \end{aligned}$$

Ensuite, prendre

$$x^{(r+1)} = \left\{ x_{ijkl}^{(r)} - \alpha_{ijkl} \theta^{(r)}, (i, j, k, l) \in \sigma^{(r)} \right\} \cup \left\{ x_{ijkl}^{(r)}, (i, j, k, l) \notin \sigma^{(r)} \right\}$$

(P5) **Détermination d'un cycle :**

Un cycle $\mu^{(r)}$ est déterminé par la résolution du système, dont la matrice est inversible puisque les colonnes de cette dernière constituent une base,

$$\sum_{(ijkl) \in I^{(r)}} \alpha_{ijkl} P_{ijkl} = -P_{i_0 j_0 k_0 l_0}.$$

Les solutions non nulles α_{ijkl} , sont les coefficients du cycle $\mu^{(r)}$.

Remarque 4

Le problème $PT4C(\tilde{M})$ est obtenu à partir du problème $PT4C$ en ajoutant quatre points fictifs d'indices $m+1, n+1, p+1$ et $q+1$ tels que :

$c_{m+1,n+1,p+1,q+1} = 0$, $c_{m+1,jkl} = c_{i,n+1,kl} = c_{ij,p+1,l} = c_{ijk,q+1} = \tilde{M}$ (où \tilde{M} est un nombre suffisamment grand) et les capacités des chemins reliant un point fictif sont infinies.

Problème de dégénérescence

Si la solution x est dégénérée (i.e., le nombre de colonnes de A_x est strictement inférieur au $rg(A)$), on complète A_x en ajoutant des colonnes de façon que la matrice obtenue contenant $rg(A)$. Par suite, l'ensemble $I^{(r)}$ est déterminé.

(PD) **le traitement de la dégénérescence** : Soient :

E_b : l'ensemble des vecteurs correspondant aux variables $x_{ijkl}^{(r)}$ vérifiant

$$0 < x_{ijkl}^{(r)} < d_{ijkl}$$

N_b : le nombre des éléments de E_b .

E_h : l'ensemble des vecteurs correspondant aux variables $x_{ijkl}^{(r)}$ vérifiant

$$x_{ijkl}^{(r)} = 0 \text{ ou } x_{ijkl}^{(r)} = d_{ijkl}$$

N_h : le nombre des éléments de E_h

E_s : Un sous ensemble quelconque de s éléments de E_h , soit l'ensemble des s premiers éléments dans E_h , en allant de gauche à droite, tel que

$$s = rg(A) - N_b$$

Au début de cette procédure, on connaît un sous ensemble E_s de E_h .

Si l'ensemble $(E_s \cup E_b)$ est non libre, remplacer (le deuxième, le troisième, ..., ou le s^{ieme}) élément dans E_s par le $(s + 1)^{ieme}$ élément dans E_h , et répéter cette opération jusqu'à ce qu'une base $I^{(r)}$ soit déterminée.

Convergence de l'algorithme Al_{PT4C} [16]

La convergence de l'algorithme repose sur les deux lemmes suivants :

Lemme 2

Supposons qu'on a $\varepsilon > 0$ à la fin de l'étape 1 de l'algorithme Al_{PT4C} , alors :

1. Si $\bar{x}^{(r)}$ est une solution réalisable de base du problème $(PT4C(\widetilde{M}))$, telle que

$$x_{m+1,n+1,p+1,q+1}^{(r)} = \varepsilon$$

alors $x^{(r)} = (x_{ijkl}^{(r)})$ avec $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, p$ et $l = 1, \dots, q$, est une solution réalisable de base initiale pour le problème $PT4C$

2. Si $\bar{x}^{(r)}$ est une solution optimale du problème $(PT4C(\widetilde{M}))$ telle que.

$$x_{m+1,n+1,p+1,q+1}^{(r)} < \varepsilon$$

alors le problème $PT4C$ n'est pas réalisable..

Lemme 3

Soient $x^{(r)}$ et $x^{(r+1)}$ deux solutions réalisables consécutives non dégénérées dont les valeurs de l'objectif correspondantes sont $Z^{(r)}$ et $Z^{(r+1)}$ respectivement, alors :

$$Z^{(r+1)} = Z^{(r)} - (-1)^n \theta^{(r)} \Delta_{i_0 j_0 k_0 l_0}^{(r+1)}$$

avec

$$\eta = \begin{cases} 1 & \text{si } x_{i_0 j_0 k_0 l_0}^{(r)} = 0, \\ 0 & \text{si } x_{i_0 j_0 k_0 l_0}^{(r)} = d_{i_0 j_0 k_0 l_0} \end{cases}$$

D'où $Z^{(r+1)} < Z^{(r)}$.

Théorème 7

Supposons que le problème est non dégénéré, alors l'algorithme Al_{PT4C} converge en un nombre fini d'itérations.

Preuve.

Le lemme précédent montre que l'algorithme Al_{PT4C} garantit que la même base ne peut jamais apparaître dans deux itérations distinctes, et comme le nombre de sommets visités est nécessairement fini, l'algorithme Al_{PT4C} converge et sa convergence est finie. ■

Chapitre 3

Etude comparative d'algorithmes pour un problème de transport à quatre indices avec capacités

3.1 Introduction

Le but de ce travail est d'élargir l'étude comparative de certaines méthodes de résolution d'un problème de transport à quatre indices avec capacités, dans l'intention d'assembler les avantages de chaque méthode, et de découvrir leurs inconvénients, pour pouvoir innover des nouvelles techniques qui améliorent certaines méthodes existantes, afin d'aboutir à des résultats numériques plus précis et moins coûteux.

3.2 Adaptation des méthodes classiques

Soit le problème de transport à quatre indices avec capacités suivant :

$$\text{Minimiser } Z = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q c_{ijkl} x_{ijkl}$$

sous les contraintes

$$\sum_{j=1}^n \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} = \alpha_i \text{ pour tout } i = 1, \dots, m$$

$$\sum_{i=1}^m \sum_{k=1}^p \sum_{l=1}^q x_{ijkl} = \beta_j \text{ pour tout } j = 1, \dots, n$$

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^q x_{ijkl} = \gamma_k \text{ pour tout } k = 1, \dots, p$$

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p x_{ijkl} = \delta_l \text{ pour tout } l = 1, \dots, q$$

$$0 \leq x_{ijkl} \leq d_{ijkl} \text{ pour tout } (i, j, k, l)$$

Dans ce problème, $\alpha_i, \beta_j, \gamma_k, \delta_l, d_{ijkl}$ et c_{ijkl} sont donnés tels que pour tout i, j, k, l , on ait $\alpha_i > 0, \beta_j > 0, \gamma_k > 0, \delta_l > 0, d_{ijkl} > 0$ et $c_{ijkl} \geq 0$.

et $M = m + n + p + q, N = mnpq$.

Pour résoudre ce problème par l'une des méthodes classiques décrites précédemment, il est nécessaire de réécrire le modèle précédent sous forme d'un programme linéaire standard défini comme suit :

$$(PL) \quad \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases} \quad (3.1)$$

où : c, x vecteurs de \mathbb{R}^n , $b \in \mathbb{R}^m$ et la matrice $A \in \mathbb{R}^{m \times n}$.

avec l' hypothèse suivante :

H1) La matrice des contraintes A est de plein rang ($rg(A) = m < n$).

Le modèle $PT4C$ est équivalent au problème de programmation linéaire doté d'une structure particulière :

$$(\widehat{PLT}) \quad \begin{cases} \min \widehat{c}^t \widehat{x} \\ \widehat{A} \widehat{x} = \widehat{b} , \\ \widehat{x} \geq 0 \end{cases}$$

où $x = (x_{1111}, \dots, x_{mnpq}) = (x_{ijkl}) \in \mathbb{R}^N$, $c = (c_{1111}, \dots, c_{mnpq}) = (c_{ijkl}) \in \mathbb{R}^N$

$b = (\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_p, \delta_1, \dots, \delta_q) \in \mathbb{R}^M$,

$d = (d_{1111}, \dots, d_{mnpq}) = (d_{ijkl}) \in \mathbb{R}^N$,

A est une $M \times N$ matrice correspondant au M premières contraintes de $PT4C$.

avec $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, p$, $l = 1, \dots, q$ $y = (y_1, \dots, y_N) \in \mathbb{R}^N$.

$(y_\tau$; variables d'écart $\tau = 1, \dots, N$), $\widehat{x} = (x, y)^t$, $\widehat{c} = (c, 0)^t$,

0 est un N -vecteur nul, $\widehat{b} = (b, d)^t$, \widehat{A} est une $(M + N) \times 2N$ matrice correspondant $Ax = b$ et $x_{ijkl} + y_\tau = d_{ijkl}$,

mais on sait que $rg(\widehat{A}) = M + N - 3$, (**H1**) du modèle (3.1) n'est pas vérifiée.

N.B : Pour que l'hypothèse (**H1**) du modèle (3.1) soit vérifiée, nous proposons l'adaptation suivante :

On considère la matrice augmentée

$$\left[\widehat{A}, \widehat{b} \right] \in \mathbb{R}^{M+N} \times \mathbb{R}^{2N+1}.$$

En utilisant une méthode d'élimination de Gauss afin de débarrasser des trois lignes à éliminer, et par conséquent nous obtiendrons ce qui suit :

$$\begin{aligned} \widehat{A}_0 &\in \mathbb{R}^{M+N-3} \times \mathbb{R}^{2N}, \\ \widehat{b}_0 &\in \mathbb{R}^{M+N-3}, \end{aligned}$$

où \widehat{A}_0 est une matrice de plein rang ($rg(\widehat{A}_0) = M + N - 3$).

En fin, on obtient le modèle ci-dessous qui est équivalent à (3.1) et vérifie l'hypothèse **(H1)**.

$$(\widehat{PLT}_0) \begin{cases} \min \widehat{c}^T \widehat{x} \\ \widehat{A}_0 \widehat{x} = \widehat{b}_0, \\ \widehat{x} \geq 0 \end{cases}$$

qui peut être résolu par l'une des méthodes classiques précédentes.

3.3 Résultats numériques :

Comparaison et sensibilité

Dans le but de tester la performance et l'efficacité de la méthode AL_{PT4C} , nous allons la comparer par les deux méthodes classiques (du simplexe et celle de Ye-Lustig) à travers quelques exemples variés et de différentes tailles dont les dimensions sont donnés au tableau comparatif citons un exemple illustratif :

Un exemple illustratif :

Soient $m = 4, p = n = q = 3$

C'est un problème de dimension (121×216)

Affichage des disponibilités

Disponibilité [1]=6.75, Disponibilité [2]=6.75, Disponibilité [3]=6.75, Disponibilité [4]=6.75

Affichage des demandes

Demande [1]=9, Demande [2]=9, Demande [3]=9

Affichage des charges

Charge [1]=4.5, charge [2]=4.5, charge [3]=18

Affichage des qualités

Qualité [1]=6.75, Qualité [2]=11.25, Qualité [3]=9

Affichage des Coûts

$c[1][1][1][1] = 2.25$

$c[1][1][1][2] = 4.5$

$c[1][1][1][3] = 6.75$

$c[1][1][2][1] = 4.5$	$c[1][1][2][2] = 6.75$	$c[1][1][2][3] = 9$
$c[1][1][3][1] = 6.75$	$c[1][1][3][2] = 9$	$c[1][1][3][3] = 11.25$
$c[1][2][1][1] = 4.5$	$c[1][2][1][2] = 6.75$	$c[1][2][1][3] = 9$
$c[1][2][2][1] = 6.75$	$c[1][2][2][2] = 9$	$c[1][2][2][3] = 11.25$
$c[1][2][3][1] = 9$	$c[1][2][3][2] = 11.25$	$c[1][2][3][3] = 13.5$
$c[1][3][1][1] = 6.75$	$c[1][3][1][2] = 9$	$c[1][3][1][3] = 11.25$
$c[1][3][2][1] = 9$	$c[1][3][2][2] = 11.25$	$c[1][3][2][3] = 13.5$
$c[1][3][3][1] = 11.25$	$c[1][3][3][2] = 13.5$	$c[1][3][3][3] = 15.75$
$c[2][1][1][1] = 4.5$	$c[2][1][1][2] = 6.75$	$c[2][1][1][3] = 9$
$c[2][1][2][1] = 6.75$	$c[2][1][2][2] = 9$	$c[2][1][2][3] = 11.25$
$c[2][1][3][1] = 9$	$c[2][1][3][2] = 11.25$	$c[2][1][3][3] = 13.5$
$c[2][2][1][1] = 6.75$	$c[2][2][1][2] = 9$	$c[2][2][1][3] = 11.25$
$c[2][2][2][1] = 9$	$c[2][2][2][2] = 11.25$	$c[2][2][2][3] = 13.5$
$c[2][2][3][1] = 11.25$	$c[2][2][3][2] = 13.5$	$c[2][2][3][3] = 15.75$
$c[2][3][1][1] = 9$	$c[2][3][1][2] = 11.25$	$c[2][3][1][3] = 13.5$
$c[2][3][2][1] = 11.25$	$c[2][3][2][2] = 13.5$	$c[2][3][2][3] = 15.75$
$c[2][3][3][1] = 13.5$	$c[2][3][3][2] = 15.75$	$c[2][3][3][3] = 18$
$c[3][1][1][1] = 6.75$	$c[3][1][1][2] = 9$	$c[3][1][1][3] = 11.25$
$c[3][1][2][1] = 9$	$c[3][1][2][2] = 11.25$	$c[3][1][2][3] = 13.5$
$c[3][1][3][1] = 11.25$	$c[3][1][3][2] = 13.5$	$c[3][1][3][3] = 15.75$
$c[3][2][1][1] = 9$	$c[3][2][1][2] = 11.25$	$c[3][2][1][3] = 13.5$
$c[3][2][2][1] = 11.25$	$c[3][2][2][2] = 13.5$	$c[3][2][2][3] = 15.75$
$c[3][2][3][1] = 13.5$	$c[3][2][3][2] = 15.75$	$c[3][2][3][3] = 18$
$c[3][3][1][1] = 11.25$	$c[3][3][1][2] = 13.5$	$c[3][3][1][3] = 15.75$
$c[3][3][2][1] = 13.5$	$c[3][3][2][2] = 15.75$	$c[3][3][2][3] = 18$
$c[3][3][3][1] = 15.75$	$c[3][3][3][2] = 18$	$c[3][3][3][3] = 20.25$
$c[4][1][1][1] = 9$	$c[4][1][1][2] = 11.25$	$c[4][1][1][3] = 13.5$
$c[4][1][2][1] = 11.25$	$c[4][1][2][2] = 13.5$	$c[4][1][2][3] = 15.75$

$$\begin{aligned}
c[4][1][3][1] &= 13.5 \\
c[4][2][1][1] &= 11.25 \\
c[4][2][2][1] &= 13.5 \\
c[4][2][3][1] &= 15.75 \\
c[4][3][1][1] &= 13.5 \\
c[4][3][2][1] &= 15.75 \\
c[4][3][3][1] &= 18
\end{aligned}$$

$$\begin{aligned}
c[4][1][3][2] &= 15.75 \\
c[4][2][1][2] &= 13.5 \\
c[4][2][2][2] &= 15.75 \\
c[4][2][3][2] &= 18 \\
c[4][3][1][2] &= 15.75 \\
c[4][3][2][2] &= 18 \\
c[4][3][3][2] &= 20.25
\end{aligned}$$

$$\begin{aligned}
c[4][1][3][3] &= 18 \\
c[4][2][1][3] &= 15.75 \\
c[4][2][2][3] &= 18 \\
c[4][2][3][3] &= 20.25 \\
c[4][3][1][3] &= 18 \\
c[4][3][2][3] &= 20.25 \\
c[4][3][3][3] &= 22.5
\end{aligned}$$

Affichage des Capacités

$$\begin{aligned}
d[1][1][1][1] &= 11.25 \\
d[1][1][2][1] &= 13.5 \\
d[1][1][3][1] &= 15.75 \\
d[1][2][1][1] &= 13.5 \\
d[1][2][2][1] &= 15.75 \\
d[1][2][3][1] &= 18 \\
d[1][3][1][1] &= 15.75 \\
d[1][3][2][1] &= 18 \\
d[1][3][3][1] &= 20.25 \\
d[2][1][1][1] &= 13.5 \\
d[2][1][2][1] &= 15.75 \\
d[2][1][3][1] &= 18 \\
d[2][2][1][1] &= 15.75 \\
d[2][2][2][1] &= 18 \\
d[2][2][3][1] &= 20.25 \\
d[2][3][1][1] &= 18 \\
d[2][3][2][1] &= 20.25 \\
d[2][3][3][1] &= 22.5 \\
d[3][1][1][1] &= 15.75 \\
d[3][1][2][1] &= 18
\end{aligned}$$

$$\begin{aligned}
d[1][1][1][2] &= 13.5 \\
d[1][1][2][2] &= 15.75 \\
d[1][1][3][2] &= 18 \\
d[1][2][1][2] &= 15.75 \\
d[1][2][2][2] &= 18 \\
d[1][2][3][2] &= 20.25 \\
d[1][3][1][2] &= 18 \\
d[1][3][2][2] &= 20.25 \\
d[1][3][3][2] &= 22.5 \\
d[2][1][1][2] &= 15.75 \\
d[2][1][2][2] &= 18 \\
d[2][1][3][2] &= 20.25 \\
d[2][2][1][2] &= 18 \\
d[2][2][2][2] &= 20.25 \\
d[2][2][3][2] &= 22.5 \\
d[2][3][1][2] &= 20.25 \\
d[2][3][2][2] &= 22.5 \\
d[2][3][3][2] &= 24.75 \\
d[3][1][1][2] &= 18 \\
d[3][1][2][2] &= 20.25
\end{aligned}$$

$$\begin{aligned}
d[1][1][1][3] &= 15.75 \\
d[1][1][2][3] &= 18 \\
d[1][1][3][3] &= 20.25 \\
d[1][2][1][3] &= 18 \\
d[1][2][2][3] &= 20.25 \\
d[1][2][3][3] &= 22.5 \\
d[1][3][1][3] &= 20.25 \\
d[1][3][2][3] &= 22.5 \\
d[1][3][3][3] &= 24.75 \\
d[2][1][1][3] &= 18 \\
d[2][1][2][3] &= 20.25 \\
d[2][1][3][3] &= 22.5 \\
d[2][2][1][3] &= 20.25 \\
d[2][2][2][3] &= 22.5 \\
d[2][2][3][3] &= 24.75 \\
d[2][3][1][3] &= 22.5 \\
d[2][3][2][3] &= 24.75 \\
d[2][3][3][3] &= 27 \\
d[3][1][1][3] &= 20.25 \\
d[3][1][2][3] &= 22.5
\end{aligned}$$

$d[3][1][3][1] = 20.25$	$d[3][1][3][2] = 22.5$	$d[3][1][3][3] = 24.75$
$d[3][2][1][1] = 18$	$d[3][2][1][2] = 20.25$	$d[3][2][1][3] = 22.5$
$d[3][2][2][1] = 20.25$	$d[3][2][2][2] = 22.5$	$d[3][2][2][3] = 24.75$
$d[3][2][3][1] = 22.5$	$d[3][2][3][2] = 24.75$	$d[3][2][3][3] = 27$
$d[3][3][1][1] = 20.25$	$d[3][3][1][2] = 22.5$	$d[3][3][1][3] = 24.75$
$d[3][3][2][1] = 22.5$	$d[3][3][2][2] = 24.75$	$d[3][3][2][3] = 27$
$d[3][3][3][1] = 24.75$	$d[3][3][3][2] = 27$	$d[3][3][3][3] = 29.25$
$d[4][1][1][1] = 18$	$d[4][1][1][2] = 20.25$	$d[4][1][1][3] = 22.5$
$d[4][1][2][1] = 20.25$	$d[4][1][2][2] = 22.5$	$d[4][1][2][3] = 24.75$
$d[4][1][3][1] = 22.5$	$d[4][1][3][2] = 24.75$	$d[4][1][3][3] = 27$
$d[4][2][1][1] = 20.25$	$d[4][2][1][2] = 22.5$	$d[4][2][1][3] = 24.75$
$d[4][2][2][1] = 22.5$	$d[4][2][2][2] = 24.75$	$d[4][2][2][3] = 27$
$d[4][2][3][1] = 24.75$	$d[4][2][3][2] = 27$	$d[4][2][3][3] = 29.25$
$d[4][3][1][1] = 22.5$	$d[4][3][1][2] = 24.75$	$d[4][3][1][3] = 27$
$d[4][3][2][1] = 24.75$	$d[4][3][2][2] = 27$	$d[4][3][2][3] = 29.25$
$d[4][3][3][1] = 27$	$d[4][3][3][2] = 29.25$	$d[4][3][3][3] = 31.5$

1-Résolution par le Simplexe

* La valeur optimale : $Z^* = 369,536$

* Le nombre d'itérations est : $k = 118$

* Le temps d'exécution est : $t = 0,172$ seconde

La solution optimale est :

$X[1][3][2][3]=1.9999755$	$X[1][1][3][3]=2.24999325$
$X[1][2][2][2]=2.24999755$	$X[2][1][1][3]=4.75002$
$X[2][1][3][2]=1.9999755$	$X[3][2][3][1]=6.75$
$X[3][1][3][2]=13.526685e-007$	$X[1][3][3][2]=0.2500335$
$X[4][3][3][2]=6.75$	$X[1][1][2][1]=0.2500245$

2-Résolution par Ye-Lustig

* La valeur optimale : $Z^* = 369,479$

* Le nombre d'itérations est : $k = 11$

* Le temps d'exécution est : $t = 0,797$ second

X[1][1][1][1]=0.10351	X[1][1][1][2]=0.137348
X[1][1][1][3]=0.12148	X[1][1][2][1]=0.103047
X[1][1][2][2]=0.137398	X[1][1][2][3]=0.121873
X[1][1][3][1]=0.34584	X[1][1][3][2]=0.658537
X[1][1][3][3]=0.507213	X[1][2][1][1]=0.10372
X[1][2][1][2]=0.138709	X[1][2][1][3]=0.123064
X[1][2][2][1]=0.103657	X[1][2][2][2]=0.139098
X[1][2][2][3]=0.123584	X[1][2][3][1]=0.344981
X[1][2][3][2]=0.659786	X[1][2][3][3]=0.515767
X[1][3][1][1]=0.102378	X[1][3][1][2]=0.137103
X[1][3][1][3]=0.12169	X[1][3][2][1]=0.104456
X[1][3][2][2]=0.137654	X[1][3][2][3]=0.122493
X[1][3][3][1]=0.349044	X[1][3][3][2]=0.657757
X[1][3][3][3]=0.50978	X[2][1][1][1]=0.103182
X[2][1][1][2]=0.136428	X[2][1][1][3]=0.12088
X[2][1][2][1]=0.100641	X[2][1][2][2]=0.136832
X[2][1][2][3]=0.121537	X[2][1][3][1]=0.327771
X[2][1][3][2]=0.695529	X[2][1][3][3]=0.503894
X[2][2][1][1]=0.102548	X[2][2][1][2]=0.138095
X[2][2][1][3]=0.122686	X[2][2][2][1]=0.104075
X[2][2][2][2]=0.138632	X[2][2][2][3]=0.123416
X[2][2][3][1]=0.357979	X[2][2][3][2]=0.653679
X[2][2][3][3]=0.504154	X[2][3][1][1]=0.103362
X[2][3][1][2]=0.136656	X[2][3][1][3]=0.121535

$X[2][3][2][1]=0.103985$	$X[2][3][2][2]=0.137358$
$X[2][3][2][3]=0.122169$	$X[2][3][3][1]=0.344182$
$X[2][3][3][2]=0.647695$	$X[2][3][3][3]=0.517082$
$X[3][1][1][1]=0.0927731$	$X[3][1][1][2]=0.13693$
$X[3][1][1][3]=0.121527$	$X[3][1][2][1]=0.105378$
$X[3][1][2][2]=0.137619$	$X[3][1][2][3]=0.122401$
$X[3][1][3][1]=0.379997$	$X[3][1][3][2]=0.636415$
$X[3][1][3][3]=0.468252$	$X[3][2][1][1]=0.102469$
$X[3][2][1][2]=0.138869$	$X[3][2][1][3]=0.123507$
$X[3][2][2][1]=0.10503$	$X[3][2][2][2]=0.139563$
$X[3][2][2][3]=0.124268$	$X[3][2][3][1]=0.3308$
$X[3][2][3][2]=0.709367$	$X[3][2][3][3]=0.507179$
$X[3][3][1][1]=0.105314$	$X[3][3][1][2]=0.137568$
$X[3][3][1][3]=0.12238$	$X[3][3][2][1]=0.0936289$
$X[3][3][2][2]=0.13833$	$X[3][3][2][3]=0.123143$
$X[3][3][3][1]=0.344381$	$X[3][3][3][2]=0.676486$
$X[3][3][3][3]=0.504048$	$X[4][1][1][1]=0.103842$
$X[4][1][1][2]=0.136168$	$X[4][1][1][3]=0.120293$
$X[4][1][2][1]=0.106061$	$X[4][1][2][2]=0.13686$
$X[4][1][2][3]=0.121545$	$X[4][1][3][1]=0.341354$
$X[4][1][3][2]=0.728363$	$X[4][1][3][3]=0.487648$
$X[4][2][1][1]=0.103168$	$X[4][2][1][2]=0.138239$
$X[4][2][1][3]=0.122749$	$X[4][2][2][1]=0.0956801$
$X[4][2][2][2]=0.139103$	$X[4][2][2][3]=0.123626$
$X[4][2][3][1]=0.358672$	$X[4][2][3][2]=0.606238$
$X[4][2][3][3]=0.506768$	$X[4][3][1][1]=0.106423$
$X[4][3][1][2]=0.136962$	$X[4][3][1][3]=0.121561$
$X[4][3][2][1]=0.0993138$	$X[4][3][2][2]=0.137869$

$$X[4][3][2][3]=0.122327$$

$$X[4][3][3][1]=0.360197$$

$$X[4][3][3][2]=0.662408$$

$$X[4][3][3][3]=0.503326$$

3-Résolution par l' AL_{PT4C}

* La valeur optimale : $Z^* = 369.563$

* Le nombre d'itérations est : $k = 10$

* Le temps d'exécution est : $t = 0.078$ seconde

la solution optimale est :

$$X[1][1][1][1]=4.5$$

$$X[1][1][2][1]=2.25$$

$$X[2][1][2][2]=2.25$$

$$X[2][2][3][2]=4.5$$

$$X[3][2][3][2]=4.5$$

$$X[3][3][3][3]=2.25$$

$$X[4][3][3][3]=6.75$$

3.4 Comparaison récapitulative

Observons que :

1) La méthode du simplexe traite le problème de dégénérescence en utilisant la règle du plus petit indice dite de Bland, cependant l' AL_{PT4C} utilise un procédé qui consiste à permuter les colonnes, et répéter cette opération jusqu'à trouver une base, et par conséquent le problème de dégénérescence sera remédié, mais cette opération représente l'étape la plus coûteuse de l' AL_{PT4C} , tandis que la méthode de Ye-Lustig ne souffre pas du problème de dégénérescence, c'est un avantage remarquable.

2) L' AL_{PT4C} attaque le problème de transport telle qu'il est, sans lui augmenter la taille. Cependant l'utilisation des deux algorithmes classiques, nécessite l'augmentation de la taille du problème traité, c'est un désavantage qu'il faut signaler.

3) Efficacité théorique : il est possible de prouver que les méthodes de points

intérieurs s'exécutent en temps polynomial (ce qui n'est pas le cas de l'algorithme du simplexe)

4) Efficacité pratique : Des recherches dans la littérature montrent que le temps d'exécution concernant les méthodes de points intérieurs sont compétitifs, [3].

- Pour des problèmes de petite taille ($n + m < 2000$), la méthode du simplexe semble imbattable
- Pour des problèmes de taille moyenne ($n + m < 10000$), aucune méthode ne présente d'avantage clair. Le résultat dépendra du problème, en particulier de la structure des éléments non-nuls de la matrice A (qui influe de façon notable sur l'efficacité des méthodes de points intérieurs)
- Pour des problèmes de très grande taille ($n + m > 10000$), les méthodes de points intérieurs semblent prendre le dessus sur leur concurrent.

Tableau comparatif

Problème		Valeur optimale			Nombre d'itérations			Temps d'exécution		
Pb	Taile	Alg1	Alg 2	Alg 3	Alg1	Alg 2	Alg 3	Alg 1	Alg 2	Alg 3
01	121×216	369, 536	369, 479	369, 536	118	11	10	0, 172	0, 797	0, 078
02	149×270	22, 25	22, 2856	22, 25	146	0 5	11	0, 282	0, 750	0, 125
03	158×288	4, 375	4, 5422	4, 375	155	04	11	0, 344	0, 734	0, 125
04	167×300	40	40, 0754	40	164	52	16	0, 375	11, 453	0, 172
05	198×360	10, 25	10, 296	10, 25	195	10	17	0, 609	4, 031	0, 266
06	216×400	4, 5	4, 34705	4, 5	213	5	11	0, 797	2, 953	0, 360
07	257×480	38, 75	38, 2094	38, 75	254	10	15	1, 282	11, 110	0, 329
08	287×540	39, 75	39, 3243	39, 75	284	05	14	1, 703	8.297	0, 437
09	318×600	10, 4375	11, 7481	10, 4375	315	10	16	2, 281	23, 000	0, 485
10	378×720	95, 0625	94, 4135	95, 0625	375	24	15	3, 609	97, 203 (1, 62 m)	0, 687
11	469×900	11, 3125	10, 0207	11, 3125	466	41	16	6, 437	380.451 (6, 34 m)	1, 000
12	499×960	45	44, 835	45	496	39	15	7, 609	373, 485 (6, 23 m)	4, 187
13	560×1080	46, 25	45.7914	46, 25	557	51	17	10, 438	692, 922 (11, 54 m)	1, 406
14	619×1200	11, 9375	12, 49	11, 9375	617	49	16	14, 141	1167, 69 (19, 46 m)	6, 187
15	741×1440	48, 75	47, 9445	48, 75	738	34	17	22, 890	1086, 38 (18, 10 m)	8, 672
16	771×1500	12, 1875	13, 1581	12, 1875	768	42	17	25, 125	1508, 59 (25, 14 m)	2, 766

3.5 Commentaires

A travers les tests numériques effectués, nous constatons que la méthode AL_{PT4C} semble mieux se comporter que les deux autres du simplexe et de Ye-Lustig. En effet, nous observons qu'elle donne des meilleurs écarts tant en temps d'exécution qu'en nombre d'itération, notamment pour les problèmes de tailles importantes. Ainsi, la méthode AL_{PT4C} est conçue essentiellement pour traiter un $PT4C$ tel qu'il est sans lui augmenter la taille, cependant la résolution d'un $PT4C$ par les deux autres méthodes, nécessite une autre reformulation qui conduit à introduire de nombreuses variables d'écart entraînant une augmentation de la taille du problème en question et par conséquent du coût de résolution, ce qui conduit à un mauvais comportement numérique. En tenant compte aux résultats obtenus, nous pouvons noter que la méthode AL_{PT4C} est la plus efficace face aux deux autres méthodes.

3.6 Conclusion

Dans ce travail, nous avons présenté et mis en oeuvre trois méthodes de résolution d'un problème de transport à quatre indices avec capacités à savoir : la méthode de points intérieurs (Ye-Lustig), la méthodes du Simplexe (Dantzig) et la méthode AL_{PT4C} (R. Zitouni). L'étude numérique comparative que nous avons effectuée sur les trois méthodes en question, nous a donné des résultats importants. Ces derniers que nous les avons récapitulés dans le paragraphe 3.4, montrent les similarités et les différences entre les méthodes étudiées et permettant d'évoluer des nouvelles techniques numériques en exploitant les avantages de chacune de ces méthodes tout en évitant leurs inconvénients.

Bibliographie

- [1] L. Adler, N. Karmarkar, M. Resende, and G. Veiga, *An implementation of Karmarkar's algorithm for linear programming*, Math. Program, 44, pp. 997–335, (1989).
- [2] D. Benterki, Merikhi, *A modified algorithm for the strict feasibility*, Rairo – Operation Research, pp 395-399, (2001).
- [3] F. Glineur, *Interior-point methods for linear programming : a guided tour*, JORBEL, Belgian Journal of Operations Research, Statistics and Computer Science, vol. 38 (1), pp. 3–30, (1998).
- [4] D. M. Gay, *variant of Karmarkar's linear programming algorithm for problems in standard form*, Math. Program, 37, pp. 81–90, (1987).
- [5] C.C. Gonzaga, *An algorithm for solving linear programming problems in $O(n^3L)$ operations*, In N. Meggido, editor, Progress in Math, Program : Interior Point and Related Methods, pp. 1–28. Springer Verlag New York, (1987).
- [6] N. Karmarkar, *new polynomial-time algorithm for linear programming*, In Proceedzng of the 16th Annual ACM Symposium on Weory of computing, pp. 303–311, (1984).
- [7] A. Kallele, *Une synthèse sur les méthodes de points interieurs*, Mémoire présenté au Département de mathématiques et d'informatique en vue de l'obtention du grade de maîtrise en sciences (MSc.) Faculté des sciences Université de Sherbrooke, (1998).
- [8] A. Keraghel, *Analyse convexe : Théorie fondamentale et exercices*, Editions Dar El'Houda, Ain Mlila, Algérie, (2001).

- [9] L. G. Khachiyan, *A polynomial algorithm for linear programming*, Soviet Mathematics Doklady, 20, pp. 191–194, (1979).
- [10] V. Klee and G. J. Minty, *How Good is the Simplex Algorithm ?*, In O. Shisha, editor, Inequalities, III Academic Press, pp. 159–175, New York, (1972).
- [11] A. Leulmi, *Une procédure améliorante d'une méthode projective en programmation linéaire*, Thèse de magister, institut de Mathématique, Université de Skikda, (2007).
- [12] K.A. McShane, C.L. Monma, and D.F. Shanno, *An implementation of a primaldual-interior point method for linear programming*. ORSA journal on computing.1 :70-83, (1989).
- [13] M. Minoux, *Programmation Mathématique : Théorie et algorithmes*, T 1 et T 2, Dunod, (1983).
- [14] M. J. Todd and B. P. Burrell, *An extension of Karmarkar's algorithm for linear programming using dual variables*, Algorithmica, 1, pp. 409–424, (1986).
- [15] Y. Ye and M. Kojima, *Recovering optimal basis is Karmarkar's polynomial algorithm for linear programming*. Mathematical Programming, 39 :305-317, (1987).
- [16] R. Zitouni, *Etude qualitative de modèle de transport et localisation*, Thèse de Doctorat de Mathématiques appliquées, Université Farhat Abbas, Sétif, (2007).
- [17] R. Zitouni and A. Keraghel, *Resolution of a capacitated transportation problem with four subscripts*, Kybernetes, 32, 9/10, pp. 1450–1463, (2003).
- [18] R. Zitouni and A. Keraghel, *A note on the algorithm of resolution of a capacitated transportation problem with four subscripts*, Far East J. Math. Sci. (FJMS), 26/3, pp. 769–778, (2007).

Résumé

Dans ce mémoire, nous nous sommes intéressés à une étude numérique comparative entre trois méthodes de résolution d'un problème de transport à quatre indices avec capacités. Les méthodes choisies pour cette étude, sont celles du simplexe, de Ye-Lustig et d' Al_{PT4C} (introduite récemment par R. Zitouni). Nous avons établi avec succès, l'implantation des trois algorithmes tout en traitant convenablement certaines difficultés rencontrées. A travers des simulations numériques différentes, nous avons pu obtenir des résultats satisfaisants. Ces derniers, montrent que la méthode Al_{PT4C} est la plus robuste vis-à-vis aux deux autres. Ainsi, ils pourraient nous aider à exploiter les avantages des méthodes concernées, afin d'innover des techniques améliorant le comportement numérique d' Al_{PT4C} .

Mots clés : Programmation linéaire, Problème de transport à quatre indices avec capacités, Méthode du simplexe, Méthodes de points intérieurs.

Abstract

In this dissertation, we are interested in a comparative numerical study of three methods of resolution of a capacitated transportation problem with four subscripts. The methods selected for this study are those for the simplex, the Ye-Lustig and the Al_{PT4C} (introduced recently by R. Zitouni). We have successfully established the implantation of the three algorithms with suitable treating some difficulties found. Through some various numerical simulations, we have obtained satisfied results. These last ones show that the Al_{PT4C} method is the most robust against than the two others. Thus, they could help us to exploit the advantages of the concerned methods, in order to innovate some techniques improving the Al_{PT4C} 's numerical behavior.

Key words: Linear programming, Capacitated transportation problem with four subscripts, Simplex method, Interior point methods.