ÉTUDE QUALITATIVE DE MODÈLES DE TRANSPORT ET LOCALISATION

Rachid ZITOUNI

Table des matières

Résumé

Dans cette thèse nous nous intéressons à l'étude théorique et à la résolution numérique d'un problème de transport à quatre indices avec capacités. Ce modèle non traité auparavant, est lié à des problèmes pratiques importants, entre autres les problèmes de localisation. Nous avons pu exhiber des conditions et de réalisabilité et d'optimalité, construire un algorithme pour la résolution du problème. L'implantation de l'algorithme avec sa description originelle a donné lieu à un constat encourageant pour le comportement numérique. D'autre part, les aménagements originaux que nous avons effectués sur l'algorithme ont réduit considérablement le volume calculatoire tout en traitant convenablement les problèmes de dégénérescence. Les résultats obtenus ne constituent aucune restriction quant à la généralité. La comparaison de cet algorithme avec celui du simplexe utilisé comme « bench mark », nous a montré son efficacité et sa robustesse.

Abstract

This dissertation thesis addresses the capacitated transportation problem with four subscripts on its theoretical and algorithmical aspects. This model is new and allows the treatment of several important practical problems in particular location problems. We have given feasibility and optimality conditions and designed an algorithm for solving the problem. The numerical implantation of the algorithm under its first form has given encouraging results. Still, we have added modifications which have given very strong improvements in reducing the computational costs and allowing an efficient treatment of degeneracy problems which can occur. The algorithm treats the problem under a quite general form. The comparison with an algorithm based on the simplex used as bench mark for the problem shows the superiority of our algorithm on the computing time and the quality of the results as well.

Introduction

L'analyse de localisation est la composante de l'aide à la décision qui s'intéresse à la localisation d'équipements en répondant aux besoins des clients de manière à optimiser certains critères. Le terme équipements indique des infrastructures : firmes, écoles, centres hospitaliers, etc..., le terme clients représente des dépots, des élèves, des malades, etc.... Un des critères à optimiser est souvent le coût de transport. De même, ce thème couvre une importante classe de problèmes fréquemment rencontrés en pratique : comment répartir des objets, (comptes bancaires, modules de programmes, concentrateurs téléphoniques), comment localiser des équipements ou des installations sur des sites (localisation de centres sociaux, de distribution ou d'autres services), en respectant des contraintes données et en répondant au mieux aux objectifs fixés (minimisation des coûts ou équilibrage des charges entre les sites).

Les problèmes de localisation peuvent être modélisés au moyen de la programmation linéaire, quadratique ou fractionnaire. Les variables peuvent être discrètes ou continues, d'où la nécessité d'établir une méthodologie pour chaque classe. Les aspects abordés sont les suivants : modélisation, existence et propriétés des solutions, conditions d'optimalité, dualité et algorithmes de résolution. Les difficultés peuvent surgir dans tous les aspects avec un degré plus élevé dans le cas discret.

Evidemment pour répondre aux différentes questions soulevées, il est nécessaire d'exploiter tout le savoir faire, et de la recherche opérationnelle, et de la programmation mathématique.

Notre travail consiste à étudier un problème d'importance capitale pour la pratique étroitement lié au contexte de localisation. Il s'agit entre autres du problème de transport à indices multiples avec capacités. Nous étudierons plus spécifiquement le modèle à quatre indices à variables continues. Ce modèle non traité auparavant ne constitue aucune restriction au niveau des résultats établis. On s'intéressera progressivement aux aspects théoriques, algorithmiques et numériques.

Sur le plan théorique, les résultats fondamentaux de la programmation mathématique nous serviront d'appui pour exhiber les conditions d'optimalité pour une éventuelle solution. L'approche simpliciale sera au centre de nos développements algorithmiques. Des aménagements originaux seront au menu donnant lieu à un aspect numérique très encourageant.

Notre travail est réparti en quatre chapitres : le premier contient des rappels de notions fondamentales utiles pour les développements ultérieurs, le second est consacré à la présentation du problème de transport à indices multiples sans capacités qui servira comme source d'inspiration pour l'étude du problème avec capacités. Ce dernier fera l'objet du troisième chapitre à travers une étude théorique et numérique permettant d'établir un constat préliminaire sur le comportement de l'algorithme proposé. Dans le dernier chapitre, on propose des réponses aux questions soulevées dans le chapitre trois, en l'occurrence les difficultés liées à la dégénérescence et au volume calculatoire, ceci est réalisé dans un cadre comparatif appréciable.

Dans notre étude nous avons pu établir des résultats originaux et apporter des contributions intéressantes.

Chapitre 1

Programmation mathématique : généralités et outils de base

1.1 Généralités

Un programme mathématique est un problème de la forme

$$\min\left[f(x) : x \in F\right]$$

où f est la fonction objectif et F est l'ensemble des solutions réalisables.

On distingue deux grandes classes de programmation mathématique selon que la variable x est continue ($x_i \in \Re$), auquel cas, on parle de programmation continue, ou que la variable x est discrète ($x_i \in \{0,1\}$ ou $\{-1,1\}$), on parle alors de programmation discrète.

Evidemment, le cas continu offre plus d'opportunités à cause d'une méthodologie complète, d'une théorie très riche et de propriétés confortables (continuité, convexité, différentiabilité, ...) qui permettent d'étudier les problèmes d'existence, d'unicité des solutions optimales et d'exhiber des conditions d'optimalité conduisant à l'élaboration de tout un arsenal d'algorithmes appropriés bien contrôlés. Par contre dans le cas discret, plus large en applications, on dispose d'une théorie beaucoup moins consistante et peu d'algorithmes, souvent de type heuristique/combinatoire, ce qui entraîne des difficultés tant au niveau théorique qu'algorithmique. Signalons que certaines de ces difficultés peuvent être surmontées en passant du discret au continu moyennant des transformations assez simples.

Exemple

Soit le problème de programmation en nombre entier noté (PLNE) :

$$\text{Trouver un vecteur } x \in \Re^n \text{ tel que}$$

$$Ax \leq b \text{ et } x_i \in \{-1, 1\}, \forall i = 1, \ldots, n.$$

Où A est une $m \times n$ matrice réelle et $b \in \Re^m$.

Les problèmes de programmation en nombre entier qui exigent sur chaque composante du vecteur x de prendre la valeur 0 ou 1, (i. e., $x_i \in \{0, 1\}$), peuvent être facilement transformés à la forme précédente en posant $x_i = \frac{1+\acute{x}_i}{2}$, d'où $\acute{x}_i \in \{-1, 1\}$.

Considérons la relaxation linéaire du problème de programmation en nombre entier notée (PLR) qui permet à chaque variable x_i de prendre des valeurs dans l'intervalle $[-1, 1]$:

Trouver un vecteur $x \in \Re^n$ telle que

$$Ax \leq b \text{ et } x_i \in [-1, 1], \forall i = 1, \ldots, n.$$

Dans [22], Karmarkar a montré que la résolution du problème (PLR) peut être utile dans la résolution du problìme (PLNE). Il a donné l'exemple du problème de programmation quadratique noté (QP), suivant :

$$\max_{x \in \Re^n} \left[\sum_{i=1}^{n} {x_i}^2,\ Ax \leq b,\ -1 \leq x_i \leq 1 \right].$$

Notons que pour toute solution réalisable du problème (QP) on a :
$|x_i| \leq 1 \Rightarrow \sum\limits_{i=1}^{n} {x_i}^2 \leq n$. Donc la valeur maximale de l'objectif est au plus n.

Si x est une solution réalisable du problème (PLNE), alors elle l'est aussi pour le problème (QP) et $x_i \in \{-1, 1\} \Rightarrow \sum\limits_{i=1}^{n} {x_i}^2 = n$, par conséquent x est une solution optimale pour (QP) et la valeur optimale de l'objectif est égale à n.

Inversement, toute solution optimale du problème (QP) qui rend la valeur de l'objectif égale à n, est une solution réalisable du problème (PLNE).

1.1.1 Notions fondamentales d'analyse convexe

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.

• Un sous ensemble C de $\Re^n$ est dit convexe si :

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

Dans tout ce qui suit, on considère une fonction $f : \Re^n \longrightarrow \overline{\Re}$, avec $\overline{\Re} = \Re \cup \{-\infty, +\infty\}$.

• Le domaine effectif de f est l'ensemble

$$dom(f) = \{x \in \Re^n : \ f(x) < +\infty\} .$$

• La fonction f est dite propre si

$$dom(f) \neq \phi \ \text{ et } \ f(x) > -\infty, \forall x \in \Re^n.$$

Dans le cas contraire, on dit que f est impropre.

• Pour $\alpha \in \Re$, l'ensemble :

$$S_\alpha(f) = \{x \in \Re^n : \ f(x) \leq \alpha\}$$

est appelé ensemble de niveau α (ou section) inférieure large de f.

L'ensemble de niveau α inférieure strict de f est

$$S_\alpha(f) = \{x \in \Re^n : f(x) < \alpha\} .$$

- La fonction f est dite convexe si :

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

où, par convention, on prend $\infty - \infty = +\infty$.

On montre que f est convexe si et seulement si pour tout entier $m > 0$ et pour tout $\lambda_i \geq 0$ tels que $\sum\limits_{i=1}^{m} \lambda_i = 1$ on a

$$f(\sum_{i=1}^{m} \lambda_i x_i) \leq \sum_{i=1}^{m} \lambda_i f(x_i), \ \forall x_i \in \Re^n.$$

- La fonction f est dite strictement convexe si :

$$f((1-\lambda)x + \lambda y) < (1-\lambda)f(x) + \lambda f(y), \ \forall x, y \in \Re^n, \ x \neq y, \ \forall \lambda \in]0,1[.$$

- Etant donnée une fonction $g : C \to \Re$ avec $C \subset \Re^n$, on prolonge la fonction g à l'ensemble $\Re^n$ tout entier en prenant

$$h(x) = \begin{cases} g(x), & \text{si } x \in C \\ +\infty & \text{ailleurs.} \end{cases}$$

On a alors, $h : \Re^n \longrightarrow \Re \cup \{+\infty\}$. La fonction h ainsi définie, s'appelle prolongement ou étendue de f sur $\Re^n$.

Si C est convexe alors, la fonction g est convexe sur C si et seulement si son prolongement h est convexe sur $\Re^n$.

1.1.2 Programmation mathématique

Définitions :

- *Problème d'optimisation*

  Un problème d'optimisation est un problème de la forme :

$$\min\left[f(x) : x \in C\right] \tag{P}$$

où $f : \Re^n \longrightarrow \Re \cup \{+\infty\}$ et $C \subset dom(f)$.

f est appelée fonction objectif et C est appelé ensemble des solutions réalisables. Lorsque $C = dom(f) = \Re^n$, le problème est dit sans contraintes.

- *Programme mathématique*

  Un programme mathématique qu'on note (PM), est un problème d'optimisation, dans lequel l'ensemble C des solutions réalisables est exprimé à l'aide de fonctions contraintes inégalités et/ou égalités, c'est à dire lorsque C est de la forme :

$$C = \{x \in X : \ f_i(x) \leq 0, \, i = 1, \cdots, m, \, g_j(x) = 0, \, j = 1, \cdots, l\},$$

avec $C \subset \cap(dom(f_i) \cap dom(g_j))$.

Classification

a) En théorie :

La classification des problèmes d'optimisation est établie à partir des deux propriétés fondamentales suivantes :

- la convexité de l'ensemble C ;
- la convexité et la différentiabilité de la fonction f.

Pour un programme mathématique (PM), on dit que

- (PM) est convexe si les fonctions f, f_i sont convexes et les fonctions g_j sont affines ;

– (PM) est différentiable si les fonctions f, f_i et g_j sont différentiables.

b) En pratique :

Le traitement numérique exige que l'on distingue le cas linéaire du cas non linéaire. Dans le cas non linéaire, on a :

– Les problèmes sans contraintes ;

– Les problèmes avec contraintes.

Notons que la classe convexe différentiable constitue un modèle de choix pour les développements théoriques et algorithmiques.

• Définitions

– *Solution réalisable* : Un point $x^0 \in C$ (c.à d. vérifiant les contraintes de (P)) est appelé solution réalisable.

– *Solution optimale globale* : Une solution réalisable qui minimise f sur C est appelée solution optimale globale. Nous la noterons x^* ou $\bar{x}$. L'ensemble des solutions optimales globales est noté :

$$arg \min_C f(x)$$

– *Solution optimale locale* : Un point $x^* \in C$ est une solution optimale locale pour (P) s'il existe un voisinage V de x^* tel que :

$$f(x^*) \leq f(x), \quad \forall x \in V.$$

On note par

$$loc \min_C f(x),$$

l'ensemble des solutions optimales locales de (P).

Nous avons toujours

$$arg \min_C f(x) \subseteq loc \min_C f(x).$$

Si le programme (P) est convexe, alors les deux ensembles sont égaux.

Principaux résultats d'existence et d'unicité :

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

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

Conditions d'optimalité

La théorie de la programmation mathématique est principalement orientée vers l'exhibition de conditions nécessaires et suffisantes d'optimalité sur lesquelles sont basées les méthodes numériques de résolution.

Direction admissible :

$d \in \Re^n$ est appelée direction admissible ou réalisable en $\bar{x}$ s'il existe $T > 0$ tel que :

$$(\bar{x} + \alpha d) \in C, \quad \forall \alpha \in [0, T]. \tag{1}$$

Direction de descente : Soit $\bar{x} \in C$ et $d \in \Re^n$, alors :

d est dite direction de descente locale pour f au point $\bar{x}$ s'il existe $T > 0$ tel que :

$$(\bar{x} + \alpha d) \in C \text{ et } f(\bar{x} + \alpha d)) < f(\bar{x}), \quad \forall \alpha \in [0, T]. \tag{2}$$

Une direction de descente est donc nécessairement admissible.

Si f est différentiable en $\bar{x}$, d est une direction admissible et $\nabla f(\bar{x})^t d < 0$, alors d est une direction de descente en $\bar{x}$ pour f. Ainsi, dans le cas sans contraintes ($C = \Re^n$), $d = -\nabla f(\bar{x})$ est une direction de descente, elle est appelée direction de la plus grande pente.

Notons par $\bar{D}$ l'ensemble des directions admissibles en $\bar{x}$.

Conditions d'optimalité :

Soit $\bar{x} \in C$ et f différentiable en $\bar{x}$

– Condition nécessaire du premier ordre :

$$\bar{x} \in loc \min_{C} f(x) \Rightarrow \nabla f(\bar{x})^t d \geq 0, \ \forall d \in \bar{D}. \qquad \textbf{[CN1]}$$

Si C est convexe , la condition s'ecrit :

$$\bar{x} \in loc \min_{C} f(x) \Rightarrow \nabla f(\bar{x})^t (x - \bar{x}) \geq 0, \ \forall x \in C.$$

– Condition nécessaire et suffisante du premier ordre :
Si C est convexe et f est convexe en $\bar{x}$ alors, on a :

$$\bar{x} \in \arg \min_{C} f(x) \Leftrightarrow \nabla f(\bar{x})^t (x - \bar{x}) \geq 0, \ \forall x \in C. \qquad \textbf{[CNS1]}$$

Nous supposons maintenant que C est de la forme

$$C = \{x \in \Re^n : \ f_i(x) \leq 0, \, i = 1, \cdots, m, \, g_j(x) = 0, \, j = 1, \cdots, l\},$$

les fonction f_i et g_j étant définies sur $\Re^n$.
Nous allons donner les conditions d'optimalité en termes des fonctions f_i et g_j. Ces conditions d'optimalité nécessitent des conditions sur les fonctions f_i et g_j déffinissant les contraintes, on les appelle critères de qualification.
Par définition, une contrainte d'inégalité $f_i(x) \leq 0$ est dite active ou saturée en $\bar{x}$ si $f_i(\bar{x}) = 0$. On posera $I(\bar{x}) = \{i \ : \ f_i(\bar{x}) = 0\}$. Une contrainte d'égalité est, par définition, saturée.

Il y a trois critères classiques de qualification de contraintes :

Critère de Karlin (1959) : Les contraintes f_i et g_j sont affines (C est alors un polyèdre convexe), i.e.,

$$f_i(x) = a_i^t x - \alpha_i, \ \forall i \quad \text{et} \quad g_j(x) = b_j^t x - \beta_j, \ \forall j.$$

Critère de Slater (1950) : Les contraintes d'inégalités f_i sont convexes continues, les contraintes d'égalités g_j sont affines et il existe un $\tilde{x}$ tel que

$$f_i(\tilde{x}) < 0 \ \forall i, \quad \text{et} \quad g_j(\tilde{x}) = 0 \ \forall j.$$

Si l'un de ces deux critères ci-dessus, est vérifié alors les contraintes sont qualifiées en tout point réalisable, ce sont donc les critères de qualification globaux. Le critère ci-dessous a un caractère local.

Critère de Mangasarian-Fromovitz (1967) : Soit $\bar{x}$ un point réalisable. Si

$*$ Les vecteurs $\nabla g_j(\bar{x})$ sont linéairement indépendants ;

$*$ Il existe d tel que

$$\langle \nabla g_j(\bar{x}), d \rangle = 0, \quad \forall j$$

et

$$\langle \nabla f_i(\bar{x}), d \rangle < 0, \quad \forall i \in I(\bar{x}).$$

Alors les contraintes sont qualifiées en $\bar{x}$.

On peut maintenant énoncer la condition d'optimalité suivante :

Théorème 3 (Kuhn-Tucker) :

Si la fonction f est différentiable en $\bar{x} \in C$ et si l'un des trois critères précédente de qualification des contraintes est satisfait alors, une condition nécessaire pour que f admette un minimum local en $\bar{x}$, est qu'il existe $\mu_i \in \Re_+, \ \ i = 1, \ldots, m$ et $\lambda_j \in \Re$, $j = 1, \ldots, l$ tels que :

$$\begin{cases} \nabla f(\bar{x}) + \sum\limits_{i=1}^{m} \mu_i \nabla f_i(\bar{x}) + \sum\limits_{j=1}^{l} \lambda_j \nabla g_j(\bar{x}) = 0 \\ \mu_i f_i(\bar{x}) = 0, \quad i = 1, \ldots, m, \\ g_j(\bar{x}) = 0, \quad j = 1, \ldots, l. \end{cases} \qquad [\mathbf{CN1}]$$

Les nombres μ_i et λ_j sont appelés multiplicateurs de Kuhn-Tucker.

Si (PM) est convexe alors l'existence des multiplicateurs de Kuhn-Tucker satisfaisant [**CN1**] est une condition nécessaire et suffisante pour que l'on ait un minimum global en $\bar{x}$.

1.2 Outils de base et applications

1.2.1 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 déteminer, face à différentes possibilités, l'utilisation optimale des ressources de l'entreprise pour atteindre un objectif spécifique comme la maximisation des bénifices ou la minimisation des coûts.

Dans la plupart des cas, les problèmes de l'entreprise pouvant être traités par la programmation linéaire comportent un certain nombre de ressources comme par exemple, main-d'œuvre, matières premières, capitaux, espace, . . ., qui sont disponibles en quantité limitée et qu'on veut répartir d'une façon optimale entre un certain nombre de processus de fabrication. Un programme linéaire (PL) s'exprime de façon générique sous la forme

$$\min\left[c^t x : \, x \geq 0, \, Ax = b\right] \tag{P_0}$$

avec $c, x \in \Re^n$, $b \in \Re^m$ et $A \in \Re^{m \times n}$.

Remarque 1 *La nature d'un programme linéaire ne dépend pas de la forme particulière qu'on lui donne ; par contre, elle dépend de la nature des variables x_i (composantes du vecteur x) qu'il contient.*

- Un (PL) est dit *continu* Si les variables x_i peuvent prendre n'importe quelle valeur réelle. (i. e., s'il est de la forme (P_0)).
- Un (PL) est dit *booléen* Si les variables x_i ne peuvent prendre que les valeurs 0 ou 1 .
- Un (PL) est dit *entier* si les variables x_i ne peuvent prendre que des valeurs entières.
- Un (PL) est dit *mixte-booléen* si certaines variables x_i sont réelles et d'autres sont booléennes.
- Un (PL) est dit *mixte-entier* si certaines variables x_i sont réelles et d'autres entières.

Convention : dans la suite, l'expression « programme linéaire » désignera toujours un programme linéaire continu.

Définitions

Etant donné un programme linéaire de la forme (P_0).

- on appelle *solution* tout vecteur x tel que $Ax = b$ et *solution réalisable* (*admissible*) tout vecteur x tel que $Ax = b$ et $x \geq 0$.
- Supposons que $rang(A) = m$, on appelle *base* B, toute sous-matrice carrée d'odre m et régulière extraite de A.

- On associe à toute base B, les décompositions $A = [B \mid N]$ et $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$. Si l'on fixe le vecteur x_N à zéro, alors le vecteur $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ vérifiant $Ax = b$ est dit *solution de base* .
- Une solution de base est dite *réalisable* si $x_B \geq 0$.
- Si B est la matrice unité, alors la solution de base correspondante est dite *explicitée.*
- Les composantes du vecteur x_B sont appelées *variables de base* et Les composantes du vecteur x_N sont appelées *variables hors-base.*
- Une solution de base est dite *dégénérée* si certaines de ses variables de base sont nulles.
- Une solution de base est dite *optimale* si elle rend la fonction objectif (économique) $c^t x$ optimale.

Résultats fondamentaux

Etant donné un ensemble $\mathbf{P} = \{x : Ax \leq b, x \geq 0\}$. Alors :

- $\mathbf{P}$ est un polyèdre convexe, il peut être vide, borné ou non borné. Un polyèdre convexe borné est un polytope.
- Lorsque $\mathbf{P}$ est un polyèdre convexe, l'ensemble des solutions optimales du programme linéaire $\min [c^t x : x \in \mathbf{P}]$ contient au moins un sommet de $\mathbf{P}$. (un point d'un polyèdre est un sommet s'il n'est pas une combinaison linéaire convexe de deux autres points du polyèdre).

  N. B. : Les théorèmes précédents restent évidemment valables si $\mathbf{P} = \{x : Ax = b, x \geq 0\}$.
- Si $rang(A) = m$, alors tout sommet de $\mathbf{P} = \{x : Ax = b, x \geq 0\}$ est une solution de base réalisable. La réciproque est vraie.

Dualité

La dualité est un concept fondamental en programmation linéaire, elle conduit à un résultat de grande portée théorique et pratique : le théorème de dualité.

Définition 1 *Le dual du programme linéaire* (P_0) *est le programme linéaire suivant*

$$\max\left[b^t y : \ A^t y \leq c, \ y \in \Re^m\right] \qquad (D_0)$$

Le programme (P_0) *est dit primal.*

Propriétés fondamentales de la dualité

Etant donnés deux programmes duaux sous les formes (P_0) et (D_0), on a :

Lemme 1 *Si* x *est une solution réalisables de* (P_0) *et* y *est une solution réalisables de* (D_0)*, alors*

$$c^t x \geq b^t y.$$

Corollaire 1 *Soient* $\bar{x}$ *une solution réalisable de* (P_0) *et* $\bar{y}$ *une solution réalisable de* (D_0) *telles que*

$$c^t \bar{x} = b^t \bar{y},$$

alors $\bar{x}$ *est une solution optimale de* (P_0) *et* $\bar{y}$ *est une solution optimale de* (D_0).

Théorème 4 (**théorème de la dualité**) *: Une condition nécessaire et suffisante pour qu'une solution réalisable* x^* *(ou* y^**) de l'un des programmes duaux, soit optimale est qu'il existe une solution réalisable* y^* *(ou* x^**) de l'autre programme telle que*

$$c^t x^* = b^t y^*,$$

cette solution y^* *(ou* x^**) du deuxième programme est aussi optimale.*

Théorème 5 (**théorème faible des écarts complémentaires**) *: Si* x *et* y *sont respectivement deux solutions réalisables du primal et du dual, alors elles sont optimales si et seulement si*

$$(Ax - b)^t y = 0 \text{ et } (c - A^t y)^t x = 0.$$

Résolution d'un programme linéaire

Beaucoup de problèmes réels de recherche opérationnelle peuvent être exprimés comme un problème de (PL). Pour cette raison un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires. Le terme programmation linéaire suppose que les solutions à trouver doivent être représentées en variables réelles. S'il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, on parle alors de programmation linéaire en nombres entiers $(PLNE)$. Il est important de savoir que ces derniers sont nettement plus difficiles à résoudre que les (PL) à variables continues.

Du point de vue théorique, une solution optimale d'un programme linéaire est généralement atteinte en un sommet du polyèdre convexe des solutions de base réalisables qui sont en nombre fini.

Du point de vue pratique, une bonne méthode de résolution ne doit pas examiner tous les sommets du polyèdre pour arriver à l'optimum.

Algorithmes

Le fameux algorithme du simplexe introduit par G. B. Dantzig en 1947 permet de résoudre les problèmes de programmation linéaire en construisant tout d'abord une solution réalisable de base qui est un sommet du polyèdre convexe puis en se déplaçant d'un sommet en sommet adjacent en améliorànt la valeur de l'objectif jusqu'à l'optimum. Bien que cet algorithme soit efficace en pratique et qu'il soit assuré de trouver l'optimum, son comportement dans le pire des cas peut être mauvais. Il est ainsi possible de construire un (PL) pour lequel la méthode du simplexe requiert un nombre d'étapes exponentiel en la taille du problème [23]. Le premier algorithme polynomial pour la (PL) a été proposé par L. Khachiyan en 1979. Il est basé sur la méthode de l'ellipsoïde en optimisation non linéaire précédemment proposée par N. Shor. Cette méthode est elle-même une généralisation de la méthode de l'ellipsoïde en optimisation convexe due à A. Nemirovski, et à D. Yudin. Cependant, l'efficacité pratique de l'algorithme de L. Khachiyan est décevante : l'algorithme du simplexe est pratiquement toujours plus performant. En revanche, ce résultat a encouragé la recherche dans les méthodes de points intérieurs. Par opposition à l'algorithme du simplexe qui considère uniquement la frontière du polyèdre convexe définie par les contraintes, les méthodes de points intérieurs évoluent dans l'intérieur du polyèdre convexe. En 1984, N. Karmarkar propose une méthode projective. C'est le premier algorithme efficace à la fois en théorie et en pratique. Sa complexité dans le pire des cas est polynomiale et les expérimentations sur des problèmes pratiques montrent qu'il est plus performant que l'algorithme du simplexe pour les problèmes de très grande taille à matrices creuses, l'algorithme du simplexe restant compétitif pour les problèmes de taille plus modeste. Dès lors, plusieurs méthodes de point intérieur ont été proposées et étudiées. Une des méthodes les plus célèbres est la méthode prédictive/corrective qui fonctionne très bien en pratique même si son étude théorique est encore imparfaite. Pour la résolution pratique de problèmes de (PL) ordinaires, il est actuellement commun de considérer comme équivalentes les (bons) codes basés sur les méthodes dérivées du simplexe ou de points intérieurs. De plus, pour la résolution de problèmes de grande taille, une technique comme la génération de colonnes peut se révéler extrêmement efficace. Des techniques basées sur la (PL) sont de plus en plus utilisées pour l'optimisation de divers problèmes industriels tels que l'optimisation des flux de transports ou la planification de la production. Toutefois, les modèles de (PL) se révèlent insuffisants pour représenter de nombreux problèmes, la programmation linéaire en nombres entiers permet alors de modéliser un grand nombre de problèmes supplémentaires.

Applications

La programmation linéaire est essentiellement appliquée pour résoudre des problèmes d'optimisation (problèmes stratégiques et tactiques, dans le vocabulaire de la recherche opérationnelle). Les domaines d'application de ces problèmes sont très nombreux aussi bien dans la nature des problèmes abordés (planification et contrôle de la production, distribution dans des réseaux) que dans les secteurs d'industrie : industrie manufacturière, énergie (pétrole, gaz, électricité, nucléaire), transports (aériens, routiers et ferroviers), télécommunications, industrie forestière, finance, ...

Exemple

Parmi les modèles d'application de la programmation linéaire très connus dans la littérature , on trouve le problème de transport à deux indices (souvent appelé problème de Hitchcock), présenté comme suit :

Un ensemble de m sources (ou centres de production) $s_1, \ldots, s_m$ offrent respectivement des quantités $a_1, \ldots, a_m$ d'un certain bien, qui est demandé en quantités $b_1, \ldots, b_n$ par n puits (ou centres de consommation) $d_1, \ldots, d_n$. Les frais de transport d'une unité de bien de s_i vers $d_j (i = 1, \ldots, m; j = 1, \ldots, n)$ sont donnés par c_{ij}. Le problème est de déterminer les quantités x_{ij} à transporter de s_i vers d_j de manière à respecter les contraintes de demandes et d'approvisionnements et à minimiser le coût total de transport. Il est mathématiquement formulé par le programme linéaire suivant

$$\begin{array}{l}
\text{Minimiser } Z = \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} c_{ij} x_{ij} \\
\qquad \text{sous les contraintes} \\
\sum\limits_{j=1}^{n} x_{ij} = a_i, \quad i = 1, \ldots, m, \\
\sum\limits_{i=1}^{m} x_{ij} = b_j, \quad j = 1, \ldots, n, \\
\quad x_{ij} \geq 0, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n,
\end{array}$$

avec $a_i \geq 0$, $b_j \geq 0$ et $c_{ij} \geq 0$.

1.2.2 Prolèmes de localisation

L'analyse de localisation traite le problème de décision concernant la localisation des équipements (ou des installations) en répondant aux besoins des clients de sorte qu'un certain critère (minimisation d'un coût, maximisation d'un profit,) soit optimisé. Le terme "équipements" indique : usines, écoles, centres sociaux, , tandis que le terme " clients " indique : dépts, élèves, nécessiteux Trois classes fondamentales, peuvent être identifiées dans l'analyse de localisation : localisation continue, localisations de réseaux et localisation discrète. La différence entre ces trois domaines revient à la structure de l'ensemble des solutions possibles pour localiser des équipements. Donc, localiser des équipements dans un espace continu correspond au modèle de localisation continue, tandis que : localiser des équipements sur des sommets ou des arrêtes d'un réseau, correspond au modèle de localisation de réseaux. Finalement, si l'ensemble des localisations possibles est fini, alors on a un modèle de localisation discrète.

Parmi les modèles de localisation discrète les plus connus dans la littérature, on trouve le problème suivant :

Problème de localisation pris en compte de coûts fixes

Supposons qu'un industriel envisage la construction de dépôts de marchandises (équipements). Il a prospecté m sites possibles et connaît, pour chacun de ceux-ci la capacité correspondante de stockage a_i, $i = 1, \ldots, m$ et le coût d'investissement ρ_i que nécessite la création du dit dépôt, plus précisément les ρ_i représentent les sommes à payer chaque période pour amortir l'investissement (remboursement des emprunts + intérêts). L'industriel connaît également les n points de demande (clients) ainsi que la quantité de marchandise b_j demandée chaque période en chaque point j, $j = 1, \ldots, n$. En fin le coût de transport du dépôt i au point de demande j est c_{ij}.

Le problème, qui consiste à déterminer les localisations des dépôts à construire de manière à minimiser le coût total (coût de transport + coût d'investissement) tout en assurant la capacité de stockage nécessaire, peut se formuler comme un programme linéaire dont certaines variables sont entières.

En posant :

$$y_i \text{ (variable de décision)} = \begin{cases} 1, & \text{si le } i^{eme}\text{dépôt est construit (ouvert)} \\ 0 & \text{sinon} \end{cases}$$

et x_{ij} = quantité de marchandise transportée du dépôt i au client j,

ce problème se formule comme suit :

$$\text{Minimiser} Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} \rho_i y_i$$

sous les contraintes

$$\begin{array}{ll} \sum_{j=1}^{n} x_{ij} \leq a_i y_i, & i = 1, \ldots, m, \\ \sum_{i=1}^{m} x_{ij} \geq b_j, & j = 1, \ldots, n, \\ x_{ij} \geq 0, & i = 1, \ldots, m,\ j = 1, \ldots, n, \\ y_i \in \{0, 1\}, & i = 1, \ldots, m, \end{array}$$

avec $a_i \geq 0$, $b_j \geq 0$, $\rho_i \geq 0$ et $c_{ij} \geq 0$.

Notons que le premier groupe de contraintes exprime que chaque dépôt ne peut emmètre plus de sa capacité de stockage. la difficulté essentiel de ce problème provient de l'existence de coûts liés à la décision d'investissement.

En supposant que chaque équipement (ici dépôt) peut satisfaire les demandes de tous les clients (i. e. ayant des capacités de stockage illimités), le problème formulé ci-dessus, est appelé « problème de localisation sans capacité ». Dans le cas contraire, il est est appelé « problème de localisation avec capacités ».

Chapitre 2

Problème de transport : classification et état de l'art

2.1 Introduction

Le problème de transport, souvent observé dans différents domaines de l'industrie, est introduit pour la première fois par Hitchcock en 1941, traité et étudié en détail par Koopmans en 1947, L. V. Kantorovich et M. K. Savourine en 1949 et puis G. B. Dantzig en 1951. Le problème de transport à deux indices largement étudié dans la litérature, 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 demandés pour n destinations (noeuds des demandes). Ensuite, l'étude est prolongée aux problèmes à un nombre d'indices supérieur à deux. Depuis les années soixantes, 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 ($PT\ell$) n'a pas été considérablement étudié pour $\ell \geq 3$, parmi les anciennes références, nous citons [8], [9], [17] et [39]. D'autre références récentes concernant ces problèmes sont [4], [25], [31], [36], [40] et [42]. En utilisant la même notation que dans la référence [31], le problèmes de transport à ℓ indices sans capacités est formulé comme suit :

Etant donné $\Gamma = \{1, \ldots, \ell\}$ et pour $r \in \Gamma$, l'ensemble $E_r = \{1, \ldots, n_r\}$. On note par E le produit des ensembles E_r, i.e., $E = E_1 \times E_2 \times \ldots \times E_\ell$. Un élément $a \in E$ est un vecteur de ℓ composantes, i.e., $a = (a_1, a_2, \ldots, a_\ell)$, $a_r \in E_r$ avec $1 \leq r \leq \ell$. On note par a_r, la $r-$ième composante du vecteur a. A tout vecteur $a \in E$ on associe un coût unitaire c_a, en outre, à tout $r \in \Gamma$ et à tout $s \in E_r$, on définit un sous ensemble E_{rs} de E par $E_{rs} = \{a \in E : a_r = s\}$. A tout sous ensemble E_{rs}, on associe un réel non négatif e_{rs} qu'on appelle demande. Le problème de transport à indices multiples, noté par $(PT\ell)$ est de la forme suivante :

$$\begin{gathered}\text{Minimiser} \sum_{a \in E} c_a x_a \\ \text{sous les contraintes} \\ \sum_{a \in E_{rs}} x_a = e_{rs} \quad \text{pour} \quad r \in \Gamma \text{ et } \quad s \in E_r, \\ x_a \geq 0, \quad \text{pour} \quad a \in E.\end{gathered}$$

Ce problème consiste à minimiser le coût total de l'affectation x_a, $a \in E$, sous les contraintes de positivité

$$x_a \geq 0, \quad \text{pour} \quad a \in E,$$

et les contraintes saturées

$$\sum_{a \in E_{rs}} x_a = e_{rs} \quad \text{pour} \quad r \in \Gamma \text{ et } \quad s \in E_r.$$

Un problème étroitement lié au $(PT\ell)$ est le problème d'affectation axiale à ℓ indices. Ce problème est obtenu lorsque les ensembles E_r ayant le même cardinal, tous les nombres e_{rs} sont égaux à 1 et $x_a \in \{0,1\}$. Les problèmes d'affectation à indices multiples sont souvent rencontrés dans différents domaines d'application. Dans le cas « sans capacités », on cite l'exemple d'un établissement scolaire (lycée, collège), où il s'agit d'affecter à une classe, une salle de classe (toutes les salles pouvant accueillir le même nombre d'élèves), un créneau du temps et un enseignant. Ici, nous avons un problème d'affectation à quatre indices. Dans une usine, il s'agit d'affecter une tache de production, un site, une machine, un technicien, un ouvrier, un temps,..., il s'agit d'un problème d'affectation à six indices au moins. Une autre application intéressante d'un problème lié au problème d'affectation à ℓ indices peut être rencontrée dans le domaine de la biologie moléculaire.

2.2 Problème de transport à quatre indices sans capacités

2.2.1 Position du problème

Prenons $\ell = 4$ dans la formulation du problème $(PT\ell)$ présenté ci-dessus. Soient
$E_1 = \{i : i = 1, ..., m\}$, $E_2 = \{j : j = 1, ..., n\}$, $E_3 = \{k : k = 1, ..., p\}$, $E_4 = \{l : l = 1, ..., q\}$, $E = \{a : a = (i,j,k,l) \in E_1 \times E_2 \times E_3 \times E_4\}$, $e_{1i} = \alpha_i$, $e_{2j} = \beta_j$, $e_{3k} = \gamma_k$ et $e_{4l} = \delta_l$.

Alors en prenant $a = ijkl$ (lorsque a est un indice) au lieu de $a = (i,j,k,l)$, le problème de transport axial « de somme axiale » à quatre indices sans capacités $(PT4)$ est formulé comme suit :

$$\text{Minimiser } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \sum_{l=1}^{q} c_{ijkl} x_{ijkl} \tag{2.1}$$

sous les contraintes

$$\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \alpha_i \qquad \text{pour tout} \quad i = 1, ..., m, \tag{2.2}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \beta_j \qquad \text{pour tout} \quad j = 1, ..., n, \tag{2.3}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} x_{ijkl} = \gamma_k \qquad \text{pour tout} \quad k = 1, ..., p, \tag{2.4}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijkl} = \delta_l \qquad \text{pour tout} \quad l = 1, ..., q, \tag{2.5}$$

$$x_{ijkl} \geq 0 \qquad \text{pour tout} \quad (i, j, k, l). \tag{2.6}$$

Dans ce problème, α_i, β_j, γ_k, δ_l, d_{ijkl} et c_{ijkl} sont donnés et tels que pour tout (i, j, k, l), on a $\alpha_i > 0$, $\beta_j > 0$, $\gamma_k > 0$, $\delta_l > 0$, et $c_{ijkl} \geq 0$.

Cette formulation est équivalente au programme linéaire suivant

$$\begin{cases} \min Z = c^t x \\ \text{s.c.} \\ Ax = b \\ x \geq 0 \end{cases} \tag{2.7}$$

avec :

- $x = (x_{1111}, ..., x_{mnpq})^t \in \Re^N$,
- $c = (c_{1111}, ..., c_{mnpq})^t \in \Re^N$,
- $d = (d_{1111}, ..., d_{mnpq})^t \in \Re^N$,
- $b = (\alpha_1, .., \alpha_m, \beta_1, .., \beta_n, \gamma_1, .., \gamma_p, \delta_1, .., \delta_q)^t \in \Re^M$,
- A est une $M \times N$ matrice,
- $M = m + n + p + q$ and $N = mnpq$.

2.2.2 Préliminaires

Matrice des coefficients

Dans cette représentation $x = (x_{1111}, x_{1211}, ..., x_{mnpq})$, on associe à chaque $(i, j, k, l) \in \{1, .., m\} \times \{1, .., n\} \times \{1, .., p\} \times \{1, .., q\}$ un vecteur $P_{ijkl} \in \Re^M$.

Seulement quatre composantes du vecteur P_{ijkl} sont non nulles, elles sont situées dans les lignes i, $m+j$, $m+n+k$ et $m+n+p+l$, et ayant 1 comme valeur commune. La matrice A des coefficients est de la forme

$$A = \begin{array}{c}
\begin{array}{ccccccccccccc}
P_{1111} & P_{1211} & \dots & P_{1npq} & P_{2111} & P_{2211} & \dots & P_{2npq} & \dots & P_{m111} & P_{m211} & \dots & P_{mnpq}
\end{array} \\
\begin{pmatrix}
1 & 1 & \dots & 1 & & & \dots & & \dots & & & & \\
 & & \dots & & 1 & 1 & \dots & 1 & \dots & & & & \\
 & & \dots & & & & \dots & & \dots & 1 & 1 & \dots & 1 \\
1 & & \dots & & 1 & & \dots & & \dots & 1 & & \dots & \\
 & 1 & \dots & & & 1 & \dots & & \dots & & 1 & \dots & \\
 & & \dots & 1 & & & \dots & 1 & \dots & & & \dots & 1 \\
1 & 1 & \dots & & 1 & 1 & \dots & & \dots & 1 & 1 & \dots & \\
 & & \dots & 1 & & & \dots & 1 & \dots & & & \dots & 1 \\
1 & 1 & \dots & & 1 & 1 & \dots & & \dots & 1 & 1 & \dots & \\
 & & \dots & 1 & & & \dots & 1 & \dots & & & \dots & 1
\end{pmatrix}
\end{array}$$

Les autres éléments de A sont tous nuls.

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

Notons que

$$rang(A) = M - 3.$$

Tableau de transport

Il est utile de présenter les données du problème grâce à un tableau qu'on appelle tableau de transport. C'est un tableau de M lignes et N colonnes, deux lignes marginales et aussi une colonne marginale. les cases de ces N colonnes de la première et la deuxième ligne marginale sont réservées aux valeurs des quantités, c_{ijkl}, et x_{ijkl} respectivement. Les cases de la colonne marginale sont réservées aux valeurs des quantités α_i, β_j, γ_k, et δ_l respectivement. Finalement, l'entrée de la case du tableau située à la ligne correspondante à $\alpha_{i'}$ et à la colonne P_{ijkl} est 1 si $i = i'$ et 0 ailleurs. Même chose pour $\beta_{j'}$, $\gamma_{k'}$, et $\delta_{l'}$. On donne un exemple ci-dessous.

c_{1111} .	c_{1211} .	$\ldots$	c_{mnpq} .	
x_{1111} .	x_{1211} .	$\ldots$	x_{mnpq} .	
1	1	$\ldots$	0	α_1 .
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
0	0	$\ldots$	1	α_m .
1	0	$\ldots$	0	β_1 .
0	1	$\ldots$	0	β_2 .
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
0	0	$\ldots$	1	β_n .
1	1	$\ldots$	0	γ_1 .
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
0	0	$\ldots$	1	γ_p .
1	1	$\ldots$	0	δ_1 .
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
0	0	$\ldots$	1	δ_q .

Cycle

On appelle cycle, tout ensemble μ contenant un nombre $\omega > 1$ de cases (i, j, k, l) dont les vecteurs P_{ijkl} correspondants sont liés, mais toute sous famille de $(\omega - 1)$ éléments de cette famille de vecteurs est libre.

Les vecteurs P_{ijkl} correspondant à un cycle μ vérifient la relation

$$\sum_{(i,j,k,l)} \in \mu \alpha_{ijkl} P_{ijkl} = 0, \quad \text{avec } \alpha_{ijkl} \neq 0.$$

Les nombres α_{ijkl} sont appelés coefficients du cycle.

2.2.3 Définitions

- Une solution réalisable x de $(PT4)$ est dite *de base* si les colonnes de la matrice A_x obtenue de A en gardant seulement les colonnes correspondant aux variables $x_{ijkl} > 0$ sont linéairement indépendantes.
- Une solution réalisable de base est dite *non dégénérée* si

$$rang(A_x) = rang(A).$$

- Soit x une solution réalisable de base, le 4-uplet (i, j, k, l) tel que

$$x_{ijkl} > 0$$

est appellé *case intéressante*. Dans le cas contraire, (i, j, k, l) est dite *non intéressante*.

2.2.4 Résolution du problème

Théorème 6 (Condition de réalisabilité)
Le problème $(TP4)$ admet une solution réalisable si et seulement si

$$\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j = \sum_{k=1}^{p} \gamma_k = \sum_{l=1}^{q} \delta_l.$$

Preuve

- Supposons que le problème $(TP4)$ ait une solution réalisable $x = x_{ijkl}$, alors il est clair que

$$\sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \sum_{j=1}^{n} \beta_j = \sum_{k=1}^{p} \gamma_k = \sum_{l=1}^{q} \delta_l.$$

- Supposons que la condition

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

est vérifiée.

Alors si on pose $\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j = \sum_{k=1}^{p} \gamma_k = \sum_{l=1}^{q} \delta_l = H$ et on prend un vecteur x tel que $x_{ijkl} = \frac{\alpha_i \beta_j \gamma_k \delta_l}{H^3}$ pour tout (i, j, k, l), on peut facilement vérifier que x est une solution réalisable pour le problème $(TP4)$.

Théorème 7 (Conditions d'optimalité)

Soit x une solution réalisale pour le problème $(TP4)$, alors x est optimale si et seulement s'il existe un vecteur

$$(u_1, .., u_m, v_1, .., v_n, w_1, .., w_p, t_1, .., t_q)^t \in \Re,^M$$

tel que :

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

$$\textit{et}$$

$$u_i + v_j + w_k + t_l = c_{ijkl} \quad \textit{pour } x_{ijkl} > 0.$$

Preuve

Considérons la formulation suivante du problème $(PT4)$:

$$\min \left[c^t x : \ x \geq 0, \ Ax = b \right].$$

Alors, son dual qu'on note $(DTP4)$ est formulé comme suit :

$$\max \left[b^t y : \ A^t y \leq c, \ y \in \Re^M \right]$$

Soit $y = (u_i, v_j, w_k, t_l)^t \in \Re^M$ avec $i = 1, ..., m$, $j = 1, ..., n$, $k = 1, ..., p$ et $l = 1, ..., q$, une solution réalisable du problème $(DTP4)$, alors une solution réalisable $x = x_{ijkl}$ du problème $(TP4)$ est optimale si et seulement si la condition de complémentarité suivante est vérifiée

$$(c - A^t y)^t x = 0,$$

i. e.,

$$(c_{ijkl} - u_i + v_j + w_k + t_l)^t x_{ijkl} = 0 \qquad (*)$$

Donc :

1. Si $x_{ijkl} = 0$, alors il vient des formulations des deux problèmes duaux que
$$u_i + v_j + w_k + t_l \leq c_{ijkl},$$
car on a $(c - A^t y)^t x \geq 0$.
2. Si $x_{ijkl} > 0$, alors il vient de la condition $(*)$ que
$$u_i + v_j + w_k + t_l = c_{ijkl}.$$

$\square$

Méthode de résolution

La méthode de résolution permettant d'obtenir une solution réalisable initiale de base pour un problème de transport à deux indices peut être généralisée pour un problème de transport à quatre indices. Elle se compose de deux phases.

Phase 1 : (*Détemination d'une solution réalisable initiale de base*)

Elle procède comme suit :

1. Donner à une variable quelconque $x_{\bar{i}\bar{j}\bar{k}\bar{l}}$ la valeur
$$x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \min(\alpha_{\bar{i}}, \beta_{\bar{j}}, \gamma_{\bar{k}}, \delta_{\bar{l}})$$
(voir **Remarque 2** ci-dessous)
2. Remplacer les quantités $\alpha_{\bar{i}}$, $\beta_{\bar{j}}$, $\gamma_{\bar{k}}$ et $\delta_{\bar{l}}$ par $\alpha_{\bar{i}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}}$, $\beta_{\bar{j}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}}$, $\gamma_{\bar{k}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}}$ et $\delta_{\bar{l}} - x_{\bar{i}\bar{j}\bar{k}\bar{l}}$ (respectivement).

3. Si $x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \alpha_{\bar{i}}$, alors prendre $x_{\bar{i}jkl} = 0, \forall (j,k,l) \neq (\bar{j},\bar{k},\bar{l})$, i. e., la ligne $\bar{i}$ dans le tableau de transport est supprimée. De la même façon, supprimer la ligne $\bar{j}$, $\bar{k}$ ou $\bar{l}$ dans le tableau de transport, si $x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \beta_{\bar{j}}$, $x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \gamma_{\bar{k}}$ ou $x_{\bar{i}\bar{j}\bar{k}\bar{l}} = \delta_{\bar{l}}$ (respectivement).
4. Répéter les trois opérations précédentes jusqu'à ce que toutes les valeurs des variables x_{ijkl} soient déterminées, (voir **Remarque 3** ci-dessous).

Phase 2 : (*amélioration d'une solution réalisable de base*)

Soit $x^{(0)}$ une solution initiale de base. Prendre $r = 0$.

a) Déterminer l'ensemble $I^{(r)}$ des cases (i,j,k,l) intéressantes,

(voir **Remarque 4**, ci-dessous).

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

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

Si la condition d'optimalité suivante est vérifiée

$$\triangle_{ijkl}^{(r)} \geq 0, \qquad \forall (i,j,k,l) \notin I^{(r)}$$

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

d) Déterminer

$$\triangle_{i_0j_0k_0l_0}^{(r)} = \min_{(i,j,k,l)} \triangle_{ijkl}^{(r)}, \quad \text{tel que } \triangle_{ijkl}^{(r)} < 0.$$

e) Détermier 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 $\triangle_{i_0j_0k_0l_0}^{(r)}$.

Prendre

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

$$\sigma^{(r)-} = \left\{(i,j,k,l) : (i,j,k,l) \in \sigma^{(r)} \text{ et } \alpha_{ijkl} < 0\right\}$$

déterminer

$$\theta^{(r)} = \min_{(i,j,k,l)\in\sigma^{(r)-}} \left(x^{(r)}_{ijkl} \big/ - \alpha_{ijkl} \right),$$

Ensuite, établir

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

f) Prendre $r = r+1$ et répéter **a), à e)** jusqu'à ce que la condition d'optimalité soit vérifiée.

Remarque 2 *La phase 1 de cette méthode, procède, en général, selon l'une des deux façons suivantes :*

- **Méthode de gauche à droite (G. D) :** *Elle consiste à choisir à chaque stade de calcul, la variable x_{ijkl} située à gauche de chaque tableau de transport réduit. Cette procédure remplace celle du coin nord-ouest utilisée dans la résolution des problèmes de transport à deux indices.*

- **Méthode du coût minimal (H. S. Houthakker) :** *Elle consiste à choisir à chaque stade de calcul, la variable $x_{\bar{i}\bar{j}\bar{k}\bar{l}}$ correspondant à l'élément $c_{\bar{i}\bar{j}\bar{k}\bar{l}} = \min\limits_{(i,j,k,l)} c_{ijkl}$. Cette procédure permet d'obtenir une solution réalisable, en général, plus proche de la solution optimale que celle donnée par la méthode* (**G. D**)

Remarque 3 *On doit obtenir au plus $(m+n+p+q-3)$ variables non nulles, car à chaque pas on supprime une ligne dans le tableau de transport sauf au dernier où l'on supprime à la fois quatre lignes.*

Remarque 4 *Si la solution x est dégénérée (i.e., le nombre de colonnes de A_x est strictement inférieur au rang(A)), on complète A_x en ajoutant des colonnes P_{ijkl} correspondant aux variables nulles, de façon que la matrice obtenue contenant rang(A) colonnes linéairement indépendantes. Par suite, l'ensemble $I^{(r)}$ est déterminé.*

Lemme 2 *Le vecteur $x = (x_{ijkl})$ obtenu par la phase 1 de cette méthode de résolution, est une solution réalisable de base du problème $(PT4)$.*

Preuve

– Démontrons que x est réalisable :

(Par récurrence sur le nombre de lignes du tableau de transport qui est $M = m + n + p + q$)

1) Supposons qu'on a un problème de transport $(PT4)$ tel que $m = n = p = q = 1$, alors si on applique la méthode (**G. D**), on obtient

$$x_{1111} = min(\alpha_1, \beta_1, \gamma_1, \delta_1) = \alpha_1 = \beta_1 = \gamma_1 = \delta_1.$$

il est clair que $x = (x_{1111})$ est une solution réalisable pour $(PT4)$, donc ce lemme est vrai pour $M = 4$.

2) Supposons que ce lemme est vrai pour un problème $(PT4)$ tel que $M = \eta - 1$, $\eta \geq 4$ et prouvons qu'il est vrai aussi pour un problème tel que $M = \eta$.

Soit un problème $(PT4)$ tel que $M = \eta$. Effectuons un pas par la méthode (**G. D**) :

$$x_{1111} = min(\alpha_1, \beta_1, \gamma_1, \delta_1) = \alpha_1 = \beta_1 = \gamma_1 = \delta_1.$$

Soit $x_{1111} = \alpha_1$, alors $x_{1jkl} = 0 \ \forall (j, k, l) \neq (1, 1, 1)$, (i. e., la ligne $i = 1$ dans le tableau de transport est supprimée). Le tableau réduit correspond à un problème $(PT4)$ tel que $M = \eta - 1$ et comme le lemme est vrai pour $M = \eta - 1$, alors on a

$$\sum_{j=1}^{n} \sum_{k=1}^{p} \sum_{l=1}^{q} x_{ijkl} = \alpha_i \qquad \text{pour tout} \quad i = 2, ..., m,$$

$$\sum_{i=2}^{m} \sum_{k=1}^{p} \sum_{l=1}^{q} x_{ijkl} = \beta_j \qquad \text{pour tout} \quad j = 2, ..., n,$$

$$\sum_{i=2}^{m} \sum_{j=1}^{n} \sum_{l=1}^{q} x_{ijkl} = \gamma_k \qquad \text{pour tout} \quad k = 2, ..., p,$$

$$\sum_{i=2}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijkl} = \delta_l \qquad \text{pour tout} \quad l = 2, ..., q.$$

On a aussi,

$$\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{1jkl} = \alpha_1,$$

$$\sum_{i=2}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{i1kl} = \beta_1 - \alpha_1,$$

$$\sum_{i=2}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} x_{ij1l} = \gamma_1 - \alpha_1,$$

$$\sum_{i=2}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijk1} = \delta_l - \alpha_1,$$

$$x_{ijk1} \geq 0, \quad \forall i = 2, ..., m,\ j = 1, ..., n,\ k = 1, ..., p \ \text{ et } \ l = 1, ..., q.$$

Et comme $x_{ijkl} = 0,\ \forall (j,k,l) \neq (1,1,1),$ on en déduit que

$$\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{i1kl} = \sum_{k=1}^{p}\sum_{l=1}^{q} x_{11kl} + \sum_{i=2}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{i1kl} = \beta_1.$$

De même, on trouve que

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} x_{ij1l} = \gamma_1,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijk1} = \delta_l.$$

Donc, il résulte de ce qui précède que $x = (x_{ijkl})$ est une solution réalisable pour le problème $(PT4)$.

- Démontrons que x est de base :
  Supposons que cette solution réalisable x est non dégénérée (i. e., le nombre de variables $x_{ijkl} > 0$ est $\vartheta = M - 3$). Nous allons prouver par récurrence sur ϑ que les vecteurs P_{ijkl} correspondant aux variables $x_{ijkl} > 0$ sont linéairement indépendants.
  1) Supposons qu'on a un problème de transport $(PT4)$ tel que $m = n = p = q = 1$, c. à. d. $\vartheta = 1$, alors la solution réalisable $x = (x_{1111})$ est de base car P_{1111} est linéairement indépendant. Donc le lemme est vrai pour $\vartheta = 1$.

2) Supposons que le lemme est vrai pour un problème $(PT4)$ tel que $\vartheta = \rho - 1$ c. à. d. $M = \rho + 2$ et prouvons qu'il est vrai aussi pour un problème tel que $\vartheta = \rho,\ \rho \geq 1$.
Effectuons un pas par la méthode (**G. D**) :

$$x_{1111} = min(\alpha_1, \beta_1, \gamma_1, \delta_1) = \alpha_1 = \beta_1 = \gamma_1 = \delta_1.$$

Soit $x_{1111} = \alpha_1$, alors $x_{1jkl} = 0\ \forall (j,k,l) \neq (1,1,1),$.
Posons $I = \{(i,j,k,l) : i \neq 1 \text{et} x_{ijkl} > 0.\}$, le cardinal de I est $\vartheta = \rho - 1$, et prenons

$$\sum_{(i,j,k,l)} \lambda_{ijkl} P_{ijkl} = 0 : \quad x_{ijkl} > 0. \qquad (S1)$$

Alors

$$(S1) \Longrightarrow \sum_{(i,j,k,l)\in I} \lambda_{ijkl} P_{ijkl} + \lambda_{1111} P_{1111} = 0$$

$$\Longrightarrow \begin{cases} \sum_{(i,j,k,l)\in I} \lambda_{ijkl} P_{ijkl} = 0 & (S2) \\ \text{et} \\ \lambda_{1111} = 0 \end{cases}$$

$\lambda_{1111} = 0$ vient du fait que la première ligne du tableau de transport dans ce cas est composée par des 0 sauf un 1 correspondant au vecteur P_{1111}.
Comme le lemme est vrai pour $\vartheta = \rho - 1$, alors

$$(S2) \Longrightarrow \lambda_{ijkl} = 0 : \quad \text{pour tout } (i,j,k,l) \in I.$$

Par conséquent, tous les vecteurs P_{ijkl} correspondant aux variables $x_{ijkl} > 0$ sont linéairement indépendants et la solution x est de base. □

Chapitre 3

Problème de transport à quatre indices avec capacités : Etude théorique et numérique

3.1 Introduction

Nous traitons dans ce chapitre 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.

Dans [42], nous avons introduit pour la première fois, une méthode de résolution d'un problème de transport à quatre indices avec capacités. Ce type de problèmes permet d'obtenir une idée assez précise sur la résolution d'un problème à un nombre d'indices $\ell > 4$ (à indices multiples) tout en évitant une charge de calcul fictif. L'étude numérique qu'on a réalisée sur cette méthode montre qu'elle est stable, efficace et pourra donc être considérée comme une véritable concurrente de celle du simplexe pour la résolution de ce genre de problèmes.

3.2 Position du problème

Le problème de transport (de somme axiale) à quatre indices avec capacités ($PT4C$) est formulé comme suit :

$$\text{Minimiser } Z = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} c_{ijkl}x_{ijkl} \tag{3.1}$$

sous les contraintes :

$$\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \alpha_i \qquad \text{pour tout} \quad i = 1, ..., m, \tag{3.2}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ijkl} = \beta_j \qquad \text{pour tout} \quad j = 1, ..., n, \tag{3.3}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} x_{ijkl} = \gamma_k \qquad \text{pour tout} \quad k = 1, ..., p, \tag{3.4}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijkl} = \delta_l \qquad \text{pour tout} \quad l = 1, ..., q, \tag{3.5}$$

$$0 \leq x_{ijkl} \leq d_{ijkl} \qquad \text{pour tout} \quad (i, j, k, l). \tag{3.6}$$

Dans ce problème, α_i, β_j, γ_k, δ_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 $c_{ijkl} \geq 0$.

Cette formulation est équivalente au programme linéaire suivant

$$\begin{cases} \min Z = c^t x \\ \text{s.c.} \\ Ax = b \\ 0 \leq x \leq d \end{cases}, \tag{3.7}$$

avec :

- $x = (x_{1111}, ..., x_{mnpq})^t \in \Re^N$,
- $c = (c_{1111}, ..., c_{mnpq})^t \in \Re^N$,
- $d = (d_{1111}, ..., d_{mnpq})^t \in \Re^N$,
- $b = (\alpha_1, .., \alpha_m, \beta_1, .., \beta_n, \gamma_1, .., \gamma_p, \delta_1, .., \delta_q)^t \in \Re^M$,

- A est une $M \times N$ matrice,
- $M = m + n + p + q$ et $N = mnpq$.

3.3 Définitions

Une solution réalisable x de $(PT4C)$ 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.

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

$$rang(A_x) = rang(A).$$

Soit x une solution réalisable de base, on appelle *case intéressante*, le 4-uple (i, j, k, l) tel que

$$0 < x_{ijkl} < d_{ijkl}.$$

On suppose que la condition de réalisabilité suivante est vérifiée

$$\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j = \sum_{k=1}^{p} \gamma_k = \sum_{l=1}^{q} \delta_l = H \tag{3.8}$$

d'où

$$Rang(A) = M - 3.$$

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

Tableau de transport

le tableau de transport associé au problème $(PT4C)$ ne diffère de celui associé au problème $(PT4)$ que par une ligne marginale supplémentaire dont les cases sont réservées aux valeurs des capacités d_{ijkl}. On donne un exemple ci-dessous.

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

3.4 Résolution du problème

3.4.1 Conditions de réalisabilité

Théorème 8 **1)** *Une condition nécessaire pour que le problème* $(PT4C)$ *possède une solution réalisabile est que la condition (3.8) et les conditions* *suivantes*

$$\begin{cases} \alpha_i \leq \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{p} \sum\limits_{l=1}^{q} d_{ijkl} & \textit{pour} \quad i = 1, ..., m, \\ \beta_j \leq \sum\limits_{i=1}^{m} \sum\limits_{k=1}^{p} \sum\limits_{l=1}^{q} d_{ijkl} & \textit{pour} \quad j = 1, ..., n, \\ \gamma_k \leq \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} \sum\limits_{l=1}^{q} d_{ijkl} & \textit{pour} \quad k = 1, ..., p, \\ \delta_l \leq \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{p} d_{ijkl} & \textit{pour} \quad l = 1, ..., q. \end{cases} \tag{3.9}$$

soient vérifiées.

2) *Une condition suffisante pour que le problème* $(PT4C)$ *possède une solution réalisabile est que la condition (3.8) et la condition suivante*

$$\frac{\alpha_i \beta_j \gamma_k \delta_l}{H^3} \leq d_{ijkl}, \qquad \textit{pour tout} \ \ (i, j, k, l) \tag{3.10}$$

soient vérifiées.

Preuve

1) On vérifie facilement que si $x = (x_{ijkl})$ est une solution réalisable pour le problème $(PT4C)$, alors les conditions (3.8) et (3.9) sont satisfaites.

2) Soit $x = (x_{ijkl})$ un vecteur de $\Re^N$ tel que

$$x_{ijkl} = \frac{\alpha_i \beta_j \gamma_k \delta_l}{d_{ijkl}}, \ \ \text{pour tout} \ \ (i, j, k, l).$$

On vérifie facilement que x est une solution réalisable pour le problème $(PT4C)$.

Si l'ensemble des solutions réalisables du problème est non vide alors il est un polyèdre convexe borné, et comme la fonction objectif est continue, alors le problème admet au moins une solution optimale.

Dans le paragraphe suivant, on introduit une condition pour qu'un point réalisable devienne optimal.

3.4.2 Conditions d'optimalité

Théorème 9 *Supposons que* $(PT4C)$ *est réalisable, alors une solution réalisable* x *de* $(PT4C)$ *est optimale si et seulement s'il existe un vecteur*

$$(u_1, .., u_m, v_1, .., v_n, w_1, .., w_p, t_1, .., t_q)^t \in \Re,^M$$

tel que :

$$\begin{array}{ll} u_i + v_j + w_k + t_l \leq c_{ijkl} & si \quad x_{ijkl} = 0, \\ u_i + v_j + w_k + t_l = c_{ijkl} & si \quad 0 < x_{ijkl} < d_{ijkl}, \\ & et \\ u_i + v_j + w_k + t_l \geq c_{ijkl} & si \quad x_{ijkl} = d_{ijkl}. \end{array}$$

Preuve

Considérons la formulation suivante du problème $(PT4C)$:

$$\min_{x\geq 0}[< c, x >: \ Ax = b, -x \geq -d], \tag{3.11}$$

son dual est

$$\max_{z\geq 0}[< b, y > - < d, z >: \ A^t y - z \leq c]. \tag{3.12}$$

Soit (y, z) une solution optimale du problème (3.12), alors une solution réalisable x du problème (3.11) est optimale si et seulement si les deux conditions de complémentarité suivantes sont vérifiées

$$(A^t y - z - c)_{ijkl} x_{ijkl} = 0, \tag{3.13}$$

et

$$(d - x)_{ijkl} z_{ijkl} = 0. \tag{3.14}$$

1. Si $x_{ijkl} = 0$, alors $x_{ijkl} < d_{ijkl}$ et (3.14) impliquent $z_{ijkl} = 0$. Donc $(A^t y)_{ijkl} \leq c_{ijkl}$.

2. Si $0 < x_{ijkl} < d_{ijkl}$ alors (3.14) implique $z_{ijkl} = 0$ et (3.13) implique $(A^t y - z - c)_{ijkl} = 0$. Donc $(A^t y)_{ijkl} = c_{ijkl}$.

3. Si $x_{ijkl} = d_{ijkl}$ alors $x_{ijkl} > 0$ et (3.13) impliquent $(A^t y - z - c)_{ijkl} = 0$, ce qui implique $(A^t y - c)_{ijkl} = z_{ijkl}$. Donc $(A^t y)_{ijkl} \geq c_{ijkl}$. □

En tenant compte des particularités du problème $(PT4C)$, on propose dans le prochain paragraphe un algorithme inspiré de celui du simplexe pour résoudre ce problème. On le note « Algorithme Al_{PT4C} ».

3.4.3 Algorithme Al_{PT4C}

Cet algorithme, partage avec l'algorithme du simplexe et celui des potentiels une structure composée de deux phases, la convergence finie et l'utilisation du principe de pivot. Il est décrit comme suit :

Phase 1 : *elle permet de déterminer une solution réalisable de base ou confirmer que* $(PT4C)$ *est non réalisable*

Etape1 :

Initialisation : pour tout (i,j,k,l), $\hat{\alpha}_i = \alpha_i$, $\hat{\beta}_j = \beta_j$, $\hat{\gamma}_k = \gamma_k$, $\hat{\delta}_l = \delta_l$ et $b_{ijkl} = 0$, (b_{ijkl} est une variable booléene 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 $c_{\bar{i}\bar{j}\bar{k}\bar{l}} = \min\limits_{(i,j,k,l)\in E} c_{ijkl}$,

(voir **Remarque 5,** ci-dessous)

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

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

- Actualiser $\hat{\alpha}_{\bar{i}}, \hat{\beta}_{\bar{j}}, \hat{\gamma}_{\bar{k}}$, et $\hat{\delta}_{\bar{l}}$ comme suit :

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

2) $\hat{\beta}_{\bar{\jmath}} = \hat{\beta}_{\bar{\jmath}} - x_{\bar{\imath}\bar{\jmath}\bar{k}\bar{l}}$,
si $\hat{\beta}_{\bar{\jmath}} = 0$ alors prendre $x_{i\bar{\jmath}kl} = 0$ pour tout $(i, k, l) \neq (\bar{\imath}, \bar{k}, \bar{l})$ et $b_{i\bar{\jmath}kl} = 1$ pour tout (i, k, l),

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

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

Etape 2 :
a) Prendre

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

i) Si $\varepsilon = 0$, alors $x = (x_{ijkl})$ est une **solution initiale de base** pour le problème $(PT4C)$, on la note $x^{(0)}$. **Aller en Phase 2.**

ii) Construire un problème $(PT4C(\tilde{M}))$ avec la procédure décrite en (P1) ci-dessous, et trouver une solution réalisable initiale de base $\overline{x}^{(0)}$ pour le problème $(PT4C(\tilde{M}))$, de même qu'en étape 1.

Alors, d'après la **Remarque 5**, $x^{(0)}_{m+1,n+1,p+1,q+1} = 0$
($\overline{x}^{(0)} = (x_{ijkl})$, avec $i = 1, ..., m+1$, $j = 1, ..., n+1$, $k = 1, ..., p+1$ et $l = 1, ..., q+1$).

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

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

Initialisation : $r = 1$, $\varepsilon > \mathbf{0}$ connu,

1) Déterminer $\overline{x}^{(r)}$, (Phase 2).

2) Si $x^{(r)}_{m+1,n+1,p+1,q+1} = \varepsilon$, alors $x^{(r)} = (x^{(r)}_{ijkl})$ avec $i = 1, .., m$, $j = 1, .., n$, $k = 1, .., p$, et $l = 1, .., q$, est une **solution initiale de base** pour le problème $(PT4C)$. **Aller en Phase 2.**

3) Si $\overline{x}^{(r)}$ est optimal (Phase 2), alors **le problème** $(PT4C)$ est **non réalisable. Fin.**

4) Prendre $r = r + 1$ et répéter 1), jusqu'à 3).

Ensuite, on décrit la seconde phase.

Phase 2 : *Recherche d'une solution optimale pour* $(PT4C)$

Au début de la Phase 2, on connaît une solution initiale de base $x^{(0)}$. Prendre $r = 0$.

a) Déterminer l'ensemble $I^{(r)}$ des cases (i, j, k, l) intéressantes, (voir **Remarque 6**, ci-dessous).

b) Pour tout $(i, j, k, l) \in I^{(r)}$, résoudre le système linéaire

$$u^{(r)}_i + v^{(r)}_j + w^{(r)}_k + t^{(r)}_l = c_{ijkl}.$$

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

$$\triangle^{(r)}_{ijkl} = c_{ijkl} - (u^{(r)}_i + v^{(r)}_j + w^{(r)}_k + t^{(r)}_l),$$

et

$$\Gamma^{(r)}_0 = \left\{\triangle^{(r)}_{ijkl} \quad \text{tel que} \quad x^{(r)}_{ijkl} = 0\right\},$$
$$\Gamma^{(r)}_d = \left\{\triangle^{(r)}_{ijkl} \quad \text{tel que} \quad x^{(r)}_{ijkl} = d_{ijkl}\right\}.$$

Si la condition d'optimalité suivante est vérifiée

$$\begin{gathered}\triangle_{ijkl}^{(r)} \geq 0 \quad \text{pour tout} \quad \triangle_{ijkl}^{(r)} \in \Gamma_0^{(r)} \\ \text{et} \\ \triangle_{ijkl}^{(r)} \leq 0 \quad \text{pour tout} \quad \triangle_{ijkl}^{(r)} \in \Gamma_d^{(r)},\end{gathered}$$

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

d) Déterminer

$$\triangle_{i_0j_0k_0l_0}^{(r)} = \min_{(i,j,k,l)} \left[\triangle_{ijkl}^{0(r)}, -\triangle_{ijkl}^{d(r)}\right] \text{ tel que :}$$

$$\begin{gathered}\triangle_{ijkl}^{0(r)} \in \Gamma_0^{(r)}, \quad \text{avec} \quad \triangle_{ijkl}^{0(r)} < 0 \\ \text{et} \\ \triangle_{ijkl}^{d(r)} \in \Gamma_d^{(r)}, \quad \text{avec} \quad \triangle_{ijkl}^{d(r)} > 0,\end{gathered}$$

et spécifier si $\triangle_{i_0j_0k_0l_0}^{(r)} \in \Gamma_0^{(r)}$ (ou $\in \Gamma_d^{(r)}$).

e) Construire par la procédure décrite en (P2) 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 $\triangle_{i_0j_0k_0l_0}^{(r)}$.

Prendre

$$\begin{aligned}\sigma^{(r)} &= \left\{(i,j,k,l) \text{ tel que } (i,j,k,l) : \text{case formant le cycle } \mu^{(r)}\right\} \\ \sigma^{(r)-} &= \left\{(i,j,k,l) \text{ tel que } (i,j,k,l) \in \sigma^{(r)}, \quad \text{avec } \alpha_{ijkl} < 0\right\} \\ \sigma^{(r)+} &= \left\{(i,j,k,l) \text{ tel que } (i,j,k,l) \in \sigma^{(r)}, \quad \text{avec } \alpha_{ijkl} > 0\right\}\end{aligned}$$

Si $\triangle_{i_0j_0k_0l_0}^{(r)} \in \Gamma_0^{(r)}$, déterminer

$$\begin{gathered}\theta_1^{(r)} = \min_{(i,j,k,l)\in\sigma^{(r)-}} \left(x_{ijkl}^{(r)} \big/ -\alpha_{ijkl}\right), \\ \theta_2^{(r)} = \min_{(i,j,k,l)\in\sigma^{(r)+}} \left(\left(d_{ijkl} - x_{ijkl}^{(r)}\right) \big/ \alpha_{ijkl}\right), \\ \theta^{(r)} = \min(\theta_1^{(r)}, \theta_2^{(r)}).\end{gathered}$$

Ensuite, prendre

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

Sinon ($\triangle_{i_0j_0k_0l_0}^{(r)} \in \Gamma_d^{(r)}$)**,** déterminer

$$\theta_1^{(r)} = \min_{(i,j,k,l)\in\sigma^{(r)+}} \left(x_{ijkl}^{(r)} \big/ \alpha_{ijkl} \right),$$
$$\theta_2^{(r)} = \min_{(i,j,k,l)\in\sigma^{(r)-}} \left(\left(d_{ijkl} - x_{ijkl}^{(r)} \right) \big/ -\alpha_{ijkl} \right),$$
$$\theta^{(r)} = \min(\theta_1^{(r)}, \theta_2^{(r)}).$$

Ensuite, prendre

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

f) Prendre $r = r + 1$ et répéter **a),** ..., **e)** jusqu'à ce que la condition d'optimalité soit vérifiée.

Dans la description de l'algorithme Al_{PT4C}, on a fait référence aux deux remarques suivantes :

Remarque 5 *S'il y a plusieurs éléments correspondant au minimum de c_{ijkl}, on choisit un, par exemple le premier trouvé dans le tableau de transport en allant de gauche à droite.*

Remarque 6 *Si la solution x est dégénérée (i.e., le nombre de colonnes de A_x est strictement inférieur au rang(A)), on complète A_x en ajoutant des colonnes de façon que la matrice obtenue contenant rang(A) colonnes linéairement indépendantes. Par suite, l'ensemble $I^{(r)}$ est déterminé.*

Aussi, l'algorithme Al_{PT4C} fait appel aux deux procédures suivantes :

(P1)– **Construction du problème** $(PT4C(\tilde{M}))$:

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 :

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

(P2)– **Détermination d'un cycle :**

Un cycle $\mu^{(r)}$ est déterminé par la résolution du système
$\sum\limits_{(i,j,k,l)\in I^{(r)}} \alpha_{ijkl} P_{ijkl} = -P_{i_0 j_0 k_0 l_0}$.
Les solutions non nulles α_{ijkl}, sont les coefficients du cycle $\mu^{(r)}$.

Convergence de l'algorithme Al_{PT4C}

On commence en premier lieu, par établir les deux lemmes suivants :

Lemme 3 *Supposons qu'on a $\varepsilon > 0$ à la fin de l'étape 1 de l'algorithme Al_{PT4C}, alors :*

1. *Si $\overline{x}^{(r)}$ est une solution réalisable de base du problème $(PT4C(\tilde{M}))$, telle que*
$$x^{(r)}_{m+1,n+1,p+1,q+1} = \varepsilon,$$
alors $x^{(r)} = (x^{(r)}_{ijkl})$ avec $i = 1,..,m,\ j = 1,..,n,\ k = 1,..,p,$ et $l = 1,..,q$, est une solution réalisable de base initiale pour le problème $(PT4C)$.
2. *Si $\overline{x}^{(r)}$ est une solution optimale du problème $(PT4C(\tilde{M}))$ telle que*
$$x^{(r)}_{m+1,n+1,p+1,q+1} < \varepsilon,$$
alors le problème $(PT4C)$ n'est pas réalisable.

preuve

1. Supposons que le problème $(PT4C(\tilde{M}))$ ait une solution réalisable de base $\overline{x}^{(r)}$ telle que
$$x^{(r)}_{m+1,n+1,p+1,q+1} = \varepsilon.$$

• Prouver que $x^{(r)} = (x^{(r)}_{ijkl})$ avec $i = 1,..,m,\ j = 1,..,n,\ k = 1,..,p,$ et $l = 1,..,q$, est une solution réalisable pour le problème $(PT4C)$, revient à prouver que $a_i = b_j = e_k = f_l = 0, \quad \forall i = 1,..,m,\ j = 1,..,n,\ k = 1,..,p,$ et $l = 1,..,q$, avec

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

Soients

$$
\begin{aligned}
I &= \{i : i = 1, ..., m\}, & \overline{I} &= \{i : i = 1, ..., m+1\},\\
J &= \{j : j = 1, ..., n\}, & \overline{J} &= \{j : j = 1, ..., n+1\},\\
K &= \{k : k = 1, ..., p\}, & \overline{K} &= \{k : k = 1, ..., p+1\},\\
L &= \{l : \ l = 1, ..., q\}, & \overline{L} &= \{l : \ l = 1, ..., q+1\}.
\end{aligned}
$$

Comme

$$
\varepsilon = \alpha_{m+1} = \sum_{j=1}^{n+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q+1} x_{m+1,jkl}^{(r)}
$$

$$
= \sum_{j=1}^{n+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q} x_{m+1,jkl}^{(r)} + \sum_{j=1}^{n+1}\sum_{k=1}^{p} x_{m+1,jk,q+1}^{(r)} + \sum_{j=1}^{n} x_{m+1,j,p+1,q+1}^{(r)}
$$

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

alors, on obtient

$$
\sum_{j=1}^{n+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q} x_{m+1,jkl}^{(r)} + \sum_{j=1}^{n+1}\sum_{k=1}^{p} x_{m+1,jk,q+1}^{(r)} + \sum_{j=1}^{n} x_{m+1,j,p+1,q+1}^{(r)} = 0 \qquad (1)
$$

De même, comme

$$
\varepsilon = \beta_{n+1} = \sum_{i=1}^{m+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q+1} x_{i,n+1,k,l}^{(r)}
$$

$$
= \delta_{p+1} = \sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{l=1}^{q+1} x_{i,j,p+1,l}^{(r)}
$$

$$
= \gamma_{q+1} = \sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{k=1}^{p+1} x_{i,j,k,q+1}^{(r)}
$$

on obtient les trois équations suivantes

$$\sum_{i=1}^{m+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q} x^{(r)}_{i,n+1,kl} + \sum_{i=1}^{m+1}\sum_{k=1}^{p} x^{(r)}_{i,n+1,k,q+1} + \sum_{i=1}^{m} x^{(r)}_{i,n+1,p+1,q+1} = 0 \qquad (2)$$

$$\sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{l=1}^{q} x^{(r)}_{ij,p+1,l} + \sum_{i=1}^{m+1}\sum_{j=1}^{n} x^{(r)}_{ij,p+1,q+1} + \sum_{i=1}^{m} x^{(r)}_{i,n+1,p+1,q+1} = 0 \qquad (3)$$

$$\sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{k=1}^{p} x^{(r)}_{ijk,q+1} + \sum_{i=1}^{m+1}\sum_{j=1}^{n} x^{(r)}_{ij,p+1,q+1} + \sum_{i=1}^{m} x^{(r)}_{i,n+1,p+1,q+1} = 0 \qquad (4)$$

On a aussi

$$\forall i \in I, \qquad \sum_{j=1}^{n+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q+1} x^{(r)}_{ijkl} = \alpha_i.$$

i. e., $\forall\, i \in I$,

$$\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x^{(r)}_{ijkl} + \sum_{k=1}^{p}\sum_{l=1}^{q} x^{(r)}_{i,n+1,kl} + \sum_{j=1}^{n+1}\sum_{l=1}^{q} x^{(r)}_{ij,p+1,l} + \sum_{j=1}^{n+1}\sum_{k=1}^{p+1} x^{(r)}_{ijk,q+1} = \alpha_i,$$

d'où

$$a_i = \sum_{k=1}^{p}\sum_{l=1}^{q} x^{(r)}_{i,n+1,kl} + \sum_{j=1}^{n+1}\sum_{l=1}^{q} x^{(r)}_{ij,p+1,l} + \sum_{j=1}^{n+1}\sum_{k=1}^{p+1} x^{(r)}_{ijk,q+1}, \forall\, i \in I.$$

Il s'en suit que :

$$(2) \Rightarrow \sum_{k=1}^{p}\sum_{l=1}^{q} x^{(r)}_{i,n+1,kl} = 0, \quad \forall\, i \in I,$$

$$(3) \Rightarrow \sum_{j=1}^{n+1}\sum_{l=1}^{q} x^{(r)}_{ij,p+1,l} = 0, \quad \forall\, i \in I,$$

et

$$(4) \Rightarrow \sum_{j=1}^{n+1}\sum_{k=1}^{p} x^{(r)}_{ijk,q+1} = \sum_{j=1}^{n} x^{(r)}_{ij,p+1,q+1} = x^{(r)}_{i,n+1,p+1,q+1} = 0, \forall\, i \in I.$$

Par conséquent

$$a_i = 0, \quad \forall\, i \in I.$$

De même, on trouve

$$\begin{aligned}
(1),(3) \text{ et } (4) &\Rightarrow b_j = 0, \quad \forall\, j \in J,\\
(1),(2) \text{ et } (4) &\Rightarrow e_k = 0, \quad \forall\, k \in K,\\
(1),(2) \text{ et } (3) &\Rightarrow f_l = 0, \quad \forall\, l \in L.
\end{aligned}$$

- Reste à prouver que $x^{(r)}$ est de base.

Ecrivons $\overline{x}^{(r)}$ sous la forme

$$\begin{aligned}
\overline{x}^{(r)} = &\left\{x^{(r)}_{ijkl} : (i,j,k,l) \in I \times J \times K \times L\right\}\\
&\cup \left\{x^{(r)}_{ijkl}(\text{fictives}) = 0 : (ijkl) \neq (m+1, n+1, p+1, q+1)\right\}\\
&\cup \{x_{m+1,n+1,p+1,q+1} = \varepsilon\}
\end{aligned}$$

(Une variable est dites fictive si l'un au moins, de ses indices est fictif, i. e., $i = m+1, j = n+1, k = p+1$ ou $l = q+1$).

Prenons

$$\begin{aligned}
F &= \left\{P_{ijkl} \in \Re^{M+4} : 0 < x^{(r)}_{ijkl} < d_{ijkl},\ (i,j,k,l) \in I \times J \times K \times L\right\}\\
G &= \left\{P_{ijkl} \in \Re^{M+4} : 0 < x^{(r)}_{ijkl} < d_{ijkl},\ (i,j,k,l) \in \overline{I} \times \overline{J} \times \overline{K} \times \overline{L}\right\}.
\end{aligned}$$

On remarque que

$$G = F \cup \{P_{m+1,n+1,p+1,q+1}\}.$$

Comme $\overline{x}^{(r)}$ est une solution réalisable de base du problème $(PT4C(\tilde{M}))$, alors G est libre est par conséquent F est libre. Si on élimine les quatre composantes nulles de chaque vecteur $P_{ijkl} \in F$ correspondant aux lignes $(m+1), (m+n+2), (m+n+p+3)$ et $(m+n+p+q+4)$, on obtient que l'ensemble

$$\hat{F} = \left\{P_{ijkl} \in \Re^{M} : 0 < x^{(r)}_{ijkl} < d_{ijkl},\ (i,j,k,l) \in I \times J \times K \times L\right\}$$

est libre, par conséquent la solution $x^{(r)}$ est de base.

2. Supposons que $\overline{x}^{(*)}$ est une solution optimale du problème $(PT4C(\tilde{M}))$ telle que

$$x^{(*)}_{m+1,n+1,p+1,q+1} = \acute{\varepsilon} < \varepsilon.$$

Dans ce cas, le second membre dans les équations (1), (2), (3) et (4) ci-dessus, devient $(\varepsilon - \acute{\varepsilon}) > 0$, ce qui montre qu'il existe au moins une variable non nulle (strictement positive) parmi les variables fictives $x^{(*)}_{m+1,jkl}$, $x^{(*)}_{i,n+1,kl}$, $x^{(*)}_{ij,p+1,l}$ et $x^{(*)}_{ijk,q+1}$ (autres que $x^{(*)}_{m+1,n+1,p+1,q+1}$).

La valeur optimale de l'objectif correspondant à la solution optimale $\overline{x}^{(*)}$ est

$$\begin{aligned}\overline{Z}^{(*)} &= \sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{k=1}^{p+1}\sum_{l=1}^{q+1} c_{ijkl}x^{(*)}_{ijkl} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} c_{ijkl}x^{(*)}_{ijkl} + \tilde{M}\Omega,\end{aligned}$$

où

$$\begin{aligned}\Omega = &\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} x^{(*)}_{m+1,jkl} + \sum_{i=1}^{m+1}\sum_{k=1}^{p}\sum_{l=1}^{q} x^{(*)}_{i,n+1,kl} + \sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{l=1}^{q} x^{(*)}_{ij,p+1,l} \\ &+ \sum_{i=1}^{m+1}\sum_{j=1}^{n+1}\sum_{k=1}^{p+1} x^{(*)}_{ijk,q+1}\end{aligned}$$

Il est clair que $\Omega > 0$.

Supposons maintenant, que le problème $(PT4C)$ admet une solution réalisable

$$x^0 = \left\{x^0_{ijkl} \,:\, (i,j,k,l) \in I \times J \times K \times L\right\},$$

et prenons

$$\begin{aligned} x_f = &\left\{x^0_{m+1,jkl} = 0 : (j,k,l) \in \overline{J} \times \overline{K} \times \overline{L}\right\} \\ &\cup \left\{x^0_{i,n+1,kl} = 0 \ : \ (i,k,l) \in \overline{I} \times \overline{K} \times \overline{L}\right\} \\ &\cup \left\{x^0_{ij,p+1,l} = 0 \ : \ (i,j,l) \in \overline{I} \times \overline{J} \times \overline{L}\right\} \\ &\cup \left\{x^0_{ijk,q+1} = 0 \ : \ (i,j,k) \in \overline{I} \times \overline{J} \times \overline{K}\right\} \\ &\cup \left\{x^0_{m+1,n+1,p+1,q+1} = \varepsilon\right\}. \end{aligned}$$

Alors l'ensemble $\overline{x}^0 = (x^0 \cup x_f)$ est une solution réalisable de base du problème $(PT4C(\tilde{M}))$.

La valeur de l'objectif correspondant à la solution $\overline{x}^0$ est

$$\begin{aligned} \overline{Z}^{(0)} &= \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \sum_{l=1}^{q} c_{ijkl} x^{(0)}_{ijkl} + \varepsilon c_{m+1,n+1,p+1,q+1} \\ &= \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \sum_{l=1}^{q} c_{ijkl} x^{(0)}_{ijkl}, \text{ car } c_{m+1,n+1,p+1,q+1} = 0. \end{aligned}$$

Comme $\tilde{M}$ est un nombre suffisamment grand, on peut le choisir de telle façon que $\overline{Z}^{(*)} > \overline{Z}^{(0)}$, ce qui est contraire à l'hypothèse . □

Lemme 4 *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*

$$Z^{(r+1(} = Z^{(r)} - (-1)^{\eta} \theta^{(r)} \triangle^{(r+1)}_{i_0 j_0 k_0 l_0}$$

avec

$$\eta = \begin{cases} 1 & si \quad x^{(r)}_{i_0 j_0 k_0 l_0} = 0, \\ 0 & si \quad x^{(r)}_{i_0 j_0 k_0 l_0} = d_{i_0 j_0 k_0 l_0}. \end{cases}$$

D'où $\quad Z^{(r+1)} < Z^{(r)}$.

Preuve

On considère deux cas :

Premier cas : $x^{(r)}_{i_0j_0k_0l_0} = 0$.

On a

$$Z^{(r+1)} = \sum_{(i,j,k,l)\notin\sigma^{(r)}} c_{ijkl}x^{(r)}_{ijkl} + \sum_{(i,j,k,l)\in\sigma^{(r)}} c_{ijkl}x^{(r+1)}_{ijkl}.$$

Prendre

$$\sum_{(i,j,k,l)\notin\sigma^{(r)}} c_{ijkl}x^{(r)}_{ijkl} = K.$$

Alors

$$Z^{(r+1)} = K + \sum_{(i,j,k,l)\in\sigma^{(r)}} c_{ijkl}(x^{(r)}_{ijkl} + \alpha_{ijkl}\theta^{(r)}).$$

Donc, si $\alpha_{i_0j_0k_0l_0} = 1$ et $\hat{\sigma}^{(r)} = \sigma^{(r)} - \{(i_0, j_0, k_0, l_0)\}$ alors

$$Z^{(r+1)} = Z^{(r)} + \theta^{(r)}(c_{i_0j_0k_0l_0} + \sum_{(i,j,k,l)\in\hat{\sigma}^{(r)}} \alpha_{ijkl}(u^{(r)}_i + v^{(r)}_j + w^{(r)}_k + t^{(r)}_l) \quad (3.15)$$

Posons

$$\begin{aligned}
i(\hat{\sigma}^{(r)}) &= \left\{(j,k,l) \text{ tel que } (i,j,k,l) \in \hat{\sigma}^{(r)}\right\},\\
j(\hat{\sigma}^{(r)}) &= \left\{(i,k,l) \text{ tel que } (i,j,k,l) \in \hat{\sigma}^{(r)}\right\},\\
k(\hat{\sigma}^{(r)}) &= \left\{(i,j,l) \text{ tel que } (i,j,k,l) \in \hat{\sigma}^{(r)}\right\},\\
l(\hat{\sigma}^{(r)}) &= \left\{(i,j,k) \text{ tel que } (i,j,k,l) \in \hat{\sigma}^{(r)}\right\}.
\end{aligned}$$

On vérifie facilement, que l'on a pour tout $i \neq i_0$,

$$\sum_{(j,k,l)\in i(\hat{\sigma}^{(r)})} \alpha_{ijkl} = 0. \quad (3.16)$$

et pour $i = i_0$,

$$u^{(r)}_{i_0}(\sum_{(i_0,j,k,l)\in\hat{\sigma}^{(r)}} \alpha_{i_0jkl} + 1) = 0,$$

par conséquent

$$\sum_{(i_0,j,k,l)\in\hat{\sigma}^{(r)}} \alpha_{i_0jkl} u_{i_0}^{(r)} = -u_{i_0}^{(r)}. \tag{3.17}$$

On obtient des résultats analogues aux (3.15) et (3.16) pour j, k, l. Donc

$$\sum_{(i,j,k,l)\in\hat{\sigma}^{(r)}} \alpha_{ijkl}(u_i^{(r)} + v_j^{(r)} + w_k^{(r)} + t_l^{(r)}) = -(u_0^{(r)} + v_0^{(r)} + w_0^{(r)} + t_0^{(r)}).$$

En substituant cette valeur dans (3.14), on obtient

$$Z^{(r+1)} = Z^{(r)} + \theta^{(r)} \triangle_{i_0j_0k_0l_0}^{(r+1)},$$

ce qui montre que $Z^{(r+1)} < Z^{(r)}$, car $\theta^{(r)} > 0$ et $\triangle_{i_0j_0k_0l_0}^{(r+1)} < 0$.

Deuxième cas : $x_{i_0j_0k_0l_0}^{(r)} = d_{i_0j_0k_0l_0}$.

On a

$$\begin{aligned} Z^{(r+1)} &= K + \sum_{(i,j,k,l)\in\sigma^{(r)}} c_{ijkl}(x_{ijkl}^{(r)} - \alpha_{ijkl}\theta^{(r)}) \\ &= Z^{(r)} - \theta^{(r)} \sum_{(i,j,k,l)\in\sigma^{(r)}} c_{ijkl}\alpha_{ijkl}. \end{aligned}$$

Par une démonstration analogue à celle utilisée dans le premier cas, on obtient

$$Z^{(r+1)} = Z^{(r)} - \theta^{(r)} \triangle_{i_0j_0k_0l_0}^{(r+1)},$$

ce qui montre que $Z^{(r+1)} < Z^{(r)}$, car $\theta^{(r)} > 0$ et $\triangle_{i_0j_0k_0l_0}^{(r+1)} > 0$. □

Théorème 10 *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} guarantie 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. □

3.4.4 Exemple

Considérons un problème de transport de la forme $(PT4C)$ avec :
$m = n = p = q = 2, \alpha_1 = 15, \alpha_2 = 7, \beta_1 = 12, \beta_2 = 10, \gamma_1 = 6, \gamma_2 = 16,$
$\delta_1 = 8, \delta_2 = 14.$
Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau suivant :

(i,j,k,l)	1111	1112	1121	1122	1211	1212	1221	1222	2111	2112
d_{ijkl}	5	7	3	4	1	2	2	4	1	2
c_{ijkl}	5	1	2	7	3	6	4	3	5	8

(i,j,k,l)	2121	2122	2211	2212	2221	2222
d_{ijkl}	2	4	1	3	5	2
c_{ijkl}	9	4	4	6	2	5

Nous avons utilisé la phase 1 pour déterminer une solution initiale de base pour le problème . En passant par la construction du problème $(PT4C(\tilde{M}))$, une solution initiale de base x^0 pour $(PT4C)$ est obtenue au bout de quatre itérations. La phase 2 nous a donné une solution optimale x^* en une seule itération.

Les composantes non nulles de x^* sont :

$x_{1121} = 3, \quad x_{1221} = 2, \quad x_{1112} = 6, \quad x_{1222} = 4, x_{2221} = 3, \quad x_{2122} = 3$
et $\quad x_{2222} = 1.$

La valeur optimale de l'objectif est $z^* = 55$.

3.5 Etude numérique

Notre premier objectif est la mise en œuvre de l'algorithme Al_{PT4C}. En effet, nous allons construire des exemples de tailles différentes permettant de mesurer la stabilité et la robustesse de cet algorithme. L'estimation des performances numériques de l'algorithme sera traitée dans le chapitre 4 dans un cadre comparatif adéquat.

3.5.1 Tests numériques

Nos tests sont réalisés sur un Pentium IV sous Windows et les programmes sont écrits en Borland Delphi 4.

Dans les exemples suivants, nous présentons les solutions optimales par leurs composantes non nulles seulement.

Exemple 1 *Considérons un problème de transport de la forme* $(PT4C)$ *avec :*
$m = n = p = q = 2$, $\alpha_1 = 15$, $\alpha_2 = 7$, $\beta_1 = 12$, $\beta_2 = 10$, $\gamma_1 = 6$, $\gamma_2 = 16$, $\delta_1 = 8, \delta_2 = 14$.
Les quantités c_{ijkl} *et* d_{ijkl} *sont données par le tableau suivant :*

(i,j,k,l)	*1111*	*1112*	*1121*	*1122*	*1211*	*1212*	*1221*	*1222*	*2111*	*2112*
d_{ijkl}	*5*	*7*	*3*	*4*	*1*	*2*	*2*	*4*	*1*	*2*
c_{ijkl}	*5*	*1*	*2*	*7*	*3*	*6*	*4*	*3*	*5*	*8*

(i,j,k,l)	*2121*	*2122*	*2211*	*2212*	*2221*	*2222*
d_{ijkl}	*2*	*4*	*1*	*3*	*5*	*2*
c_{ijkl}	*9*	*4*	*4*	*6*	*2*	*5*

Les composantes non nulles de la solution optimale : $x^* = (x_{ijkl})$ sont :

$$\begin{array}{ll} x_{1112} = 6, & x_{1121} = 3 = d_{1121}, \\ x_{2221} = 3, & x_{1222} = 4 = d_{1222}, \\ x_{2122} = 3, & x_{1221} = 2 = d_{1221}, \\ & x_{2222} = 1. \end{array}$$

Le nombre d' itérations est = 07. Le temps d'exécution est $= 0,22$ sec.

La valeur optimale est $z^* = 55$.

Exemple 2 *Considérons un problème de transport de la forme* $(PT4C)$ *avec :*
$m = n = p = q = 2, \alpha_1 = 15, \alpha_2 = 15, \beta_1 = 10, \beta_2 = 20, \gamma_1 = 12, \gamma_2 = 18, \delta_1 = 9,\ \delta_2 = 21$.

Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau suivant :

(i,j,k,l)	1111	1112	1121	1122	1211	1212	1221	1222	2111	2112
d_{ijkl}	15	25	45	12	10	10	10	12	11	11
c_{ijkl}	5	4	5	4	5	5	12	12	1	2

(i,j,k,l)	2121	2122	2211	2212	2221	2222
d_{ijkl}	11	22	5	6	8	9
c_{ijkl}	10	10	10	1	1	10

Les composantes non nulles de la solution optimale : $x^* = (x_{ijkl})$ sont :

$$\begin{array}{ll} x_{1122} = 9,5, & x_{1211} = 0,5, \\ x_{1212} = 5, & x_{2111} = 0,5, \\ x_{2221} = 8 = d_{2221}, & x_{2212} = 6 = d_{2212}, \\ \multicolumn{2}{c}{x_{2222} = 0,5.} \end{array}$$

Le nombre d' itérations est $= 03$. Le temps d'exécution est $= 0,16$ sec.

La valeur optimale est $z^* = 85$.

Exemple 3 *Considérons un problème de transport de la forme $(PT4C)$ avec :*
$m = 3$,$n = p = q = 2$,$\alpha_1 = 6, \alpha_2 = 8, \alpha_3 = 12$,$\beta_1 = 14, \beta_2 = 12, \gamma_1 = 9$, $\gamma_2 = 17, \delta_1 = 8, \delta_2 = 18$.
Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau suivant :

(i,j,k,l)	1111	1112	1121	1122	1211	1212	1221	1222	2111	2112
d_{ijkl}	12	15	14	16	22	13	21	20	25	29
c_{ijkl}	5	2	3	7	8	5	4	2	6	9
(i,j,k,l)	2121	2122	2211	2212	2221	2222	3111	3112	3121	3122
d_{ijkl}	26	20	9	8	14	12	14	15	18	24
c_{ijkl}	8	5	4	7	4	1	5	2	3	6

(i,j,k,l)	3211	3212	3221	3222
d_{ijkl}	24	23	21	23
c_{ijkl}	8	9	8	7

Les composantes non nulles de la solution optimale : $x^* = (x_{ijkl})$ sont :

$$\begin{array}{ll} x_{1121} = 2, & x_{1221} = 3, \\ x_{1222} = 1, & x_{2222} = 8, \\ x_{3112} = 9, & x_{3121} = 3. \end{array}$$

Le nombre d' itérations est $= 01$. Le temps d'exécution est $= 0,11$ sec.

La valeur optimale est $z^* = 55$.

Exemple 4 *Considérons un problème de transport de la forme* $(PT4C)$ *avec :*
$m = p = 3$,$n = q = 2$,$\alpha_1 = 15, \alpha_2 = 15, \alpha_3 = 10$,$\beta_1 = 15, \beta_2 = 25, \gamma_1 = 12$, $\gamma_2 = 14, \gamma_3 = 14, \delta_1 = 18, \delta_2 = 22$.
Les quantités c_{ijkl} *et* d_{ijkl} *sont données par le tableau suivant :*

(i,j,k,l)	*1111*	*1112*	*1121*	*1122*	*1131*	*1132*	*1211*	*1212*	*1221*	*1222*
d_{ijkl}	*9*	*8*	*12*	*10*	*11*	*12*	*13*	*14*	*15*	*10*
c_{ijkl}	*8*	*9*	*4*	*5*	*7*	*8*	*9*	*10*	*15*	*12*
(i,j,k,l)	*1231*	*1232*	*2111*	*2112*	*2121*	*2122*	*2131*	*2132*	*2211*	*2212*
d_{ijkl}	*11*	*12*	*8*	*9*	*52*	*24*	*25*	*12*	*10*	*10*
c_{ijkl}	*1*	*1*	*14*	*5*	*9*	*9*	*6*	*7*	*5*	*10*
(i,j,k,l)	*2221*	*2222*	*2231*	*2232*	*3111*	*3112*	*3121*	*3122*	*3131*	*3132*
d_{ijkl}	*11*	*12*	*22*	*21*	*21*	*10*	*10*	*10*	*21*	*21*
c_{ijkl}	*21*	*41*	*54*	*20*	*20*	*15*	*8*	*9*	*12*	*11*
(i,j,k,l)	*3211*	*3212*	*3221*	*3222*	*3231*	*3232*				
d_{ijkl}	*9*	*8*	*7*	*10*	*9*	*4*				
c_{ijkl}	*10*	*5*	*8*	*9*	*6*	*7*				

Les composantes non nulles de la solution optimale : $x^* = (x_{ijkl})$ sont :

$$\begin{array}{lll} x_{1121} = 1, & x_{1231} = 2, & x_{2112} = 7, \\ x_{2122} = 3, & x_{2211} = 5, & x_{3121} = 4, \\ x_{3221} = 6, & x_{1232} = 12 = d_{1232}. & \end{array}$$

Le nombre d' itérations est $= 03$. Le temps d'exécution est$= 0,17$ sec.

La valeur optimale est $z^* = 185$.

Exemple 5 *Considérons un problème de transport de la forme* $(PT4C)$ *avec :*
$m = n = p = 3$,$q = 2$,$\alpha_1 = 9, \alpha_2 = 11, \alpha_3 = 10$,$\beta_1 = 8, \beta_2 = 8, \beta_3 = 14$, $\gamma_1 = 7, \gamma_2 = 8, \gamma_3 = 15, \delta_1 = 6, \delta_2 = 24$.
Les quantités c_{ijkl} *et* d_{ijkl} *sont données par le tableau suivant :*

(i,j,k,l)	1111	1112	1121	1122	1131	1132	1211	1212	1221	1222
d_{ijkl}	14	16	18	22	30	31	32	33	19	17
c_{ijkl}	8	5	5	6	4	7	8	9	12	10
(i,j,k,l)	1231	1232	1311	1312	1321	1322	1331	1332	2111	2112
d_{ijkl}	15	10	11	19	25	26	27	30	33	32
c_{ijkl}	11	23	10	9	8	10	9	8	7	16
(i,j,k,l)	2121	2122	2131	2132	2211	2212	2221	2222	2231	2232
d_{ijkl}	31	29	28	27	11	25	26	24	20	21
c_{ijkl}	15	20	19	21	11	12	13	15	14	10
(i,j,k,l)	2311	2312	2321	2322	2331	3232	3111	3112	3121	3122
d_{ijkl}	22	23	10	9	22	27	18	16	15	33
c_{ijkl}	6	18	19	17	14	16	12	10	6	5
(i,j,k,l)	3131	3132	3211	3212	3221	3222	3231	3232	3311	3312
d_{ijkl}	34	35	36	37	29	28	27	26	25	32
c_{ijkl}	15	14	12	11	19	18	17	11	10	10
(i,j,k,l)	3321	3322	3331	3332						
d_{ijkl}	31	31	30	27						
c_{ijkl}	11	12	20	14						

Les composantes non nulles de la solution optimale : $x^* = (x_{ijkl})$ sont :

$$\begin{array}{lll} x_{1221} = 1, & x_{1332} = 8, & x_{2232} = 5, \\ x_{2311} = 1, & x_{3122} = 8, & x_{3232} = 2. \end{array}$$

Le nombre d' itérations est $= 07$. Le temps d'exécution est $= 0,11$ sec.
La valeur optimale est$z^* = 221$.

Récapitulation :

Exemple	Taille du Prob. MxN	Nombre d'itérations	Temps d'exécution
1	8x16	7	0,22 sec
2	8x16	3	0,16 sec
3	9x24	1	0,11 sec
4	10x36	3	0,17 sec
5	11x54	7	0,11 sec

3.6 Commentaires

A travers les tests numériques qu'on a effectué, on se rend compte de la stabilité et la robustesse de notre algorithme. Cependant, pour les problèmes dégénérés (les plus fréquents en pratique), l'algorithme Al_{PT4C} diverge. A ce propos, nous avons introduit une technique pour surmonter ce phénomène, laquelle sera présentée en détail dans le chapitre 4 .

Chapitre 4

Etude comparative d'algorithmes pour les problèmes de transport à quatre indices avec capacités

4.1 Introduction

Comme nous l'avons signalé précédement, un problème de transport à quatre indices avec capacités peut s'écrire sous la forme d'un problème de programmation linéaire et peut être résolu à l'aide d'algorithmes classiques comme celui du simplexe, celui de Karmarkar, ...

Cependant, une telle formulation conduit à introduire de nombreuses variables d'écart entraînant une augmentation de la taille du problème, alors du coût de résolution, et donc un mauvais comportement en raison de la propagation des erreurs d'arrondis.

Nous proposons dans ce chapitre une modifiée de l'algorithme Al_{PT4C} présenté au chapitre 3. Nous testerons ensuite cet algorithme en utilisant comme « bench mark » l'algorithme du simplexe pour la résolution du problème.

4.2 Algorithme du simplexe

L'algorithme du simplexe nécessite la reformulation du problème (3.7) comme suit :

$$\min_{\hat{x}}[\hat{c}^t\hat{x} : \hat{x} \geq 0,\ \hat{A}\hat{x} = \hat{b}], \tag{4.1}$$

où $x = (x_{ijkl}) \in \Re^N$, $y \in \Re^N$, $b \in \Re^M$, $d \in \Re^N$, $\hat{x} = (x, y)^t$, $\hat{c} = (c, 0)^t$, 0 est un N-vecteur nul, $\hat{b} = (b, d)^t$, $\hat{A}$ est une $(M+N) \times 2N$ matrice correspondant au contraintes $Ax = b$ et $x_{ijkl} + y_\tau = d_{ijkl}$, avec $i = 1, ..., m$, $j = 1, ..., n$, $k = 1, ..., p$, $l = 1, ..., q$ et $\tau = 1, ..., N$.

On remarque que

$$\hat{A} = \left[\begin{array}{c|c} A & 0 \\ \hline I_N & I_N \end{array}\right],$$

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

Notons que $rang(\hat{A}) = M + N - 3$.

4.3 Algorithme Al_{PT4C} modifié

Cet algorithme noté Alm_{PT4C} est une modifiée de l'algorithme Al_{PT4C} présenté au paragraphe (3.4.3). Les modifications proposées sur ce dernier, permettant d'éviter le cyclage et d'optimiser le volume calculatoire.

En effet, on propose dans le point **I)** ci-dessous, une procédure pour déterminer l'ensemble $I^{(r)}$ des cases intéressantes (i, j, k, l) (i. e., compléter la matrice A_x) dans le cas où la solution de base $x^{(r)}$ est dégénérée.

Dans le point **II)**, on propose une procédure modifiant les deux points c) et d) de la phase 2 de l'algorithme Al_{PT4C}.

I) *Procédure pour déterminer l'ensemble* $I^{(r)}$ *des cases intéressantes* :

1. 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, on a :

$$N_h = N - N_b,\ N = mnpq.$$

– 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 $s = rang(A) - N_b$.

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

i) **Si** l'ensemble $(E_s \cup E_b)$ est libre, alors une base $I^{(r)}$ est déterminée. **Fin**.

ii) Remplacer le premier (le deuxième, le troisième,..., ou le s^{ieme}) élément dans E_s par le $(s+1)^{ieme}$ élément dans E_h, ou prendre un autre sous ensemble E_s quelconque de E_h, et répéter **i)** jusqu'à ce qu'une base $I^{(r)}$ soit déterminée. **Fin**.

II) *Procédure pour savoir si une solution réalisable de base $x^{(r)}$ est optimale ou l'améliorer le cas échéant :*

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

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

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

$\Gamma_0^{(r)} = \left\{ \triangle_{ijkl}^{(r)} \quad \text{tel que} \quad x_{ijkl}^{(r)} = 0 \right\},$
et
$\Gamma_d^{(r)} = \left\{ \triangle_{ijkl}^{(r)} \quad \text{tel que} \quad x_{ijkl}^{(r)} = d_{ijkl} \right\},$

où les éléments $\triangle_{ijkl}^{(r)}$ sont représentés de la même façon que les variables x_{ijkl} dans le tableau de transport.

- En allant de gauche à droite, choisir $\triangle_{i_0 j_0 k_0 l_0}^{(r)}$ comme le premier élément $\triangle_{ijkl}^{(r)} < 0$ trouvé dans le tableau $\Gamma_0^{(r)}$.
  Noter dans ce cas, que $\triangle_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_0^{(r)}$.

- Si tous les éléments de $\Gamma_0^{(r)}$ sont non négatifs, alors choisir pareillement, $\triangle_{i_0 j_0 k_0 l_0}^{(r)}$ comme le premier élément $\triangle_{ijkl}^{(r)} > 0$ trouvé dans le tableau $\Gamma_d^{(r)}$. Noter dans ce cas, que $\triangle_{i_0 j_0 k_0 l_0}^{(r)} \in \Gamma_d^{(r)}$.

- Si tous les éléments de $\Gamma_d^{(r)}$ sont non positifs, alors **la solution de base** $x^{(r)}$ **est optimale. Fin.**

4.4 Résultats numériques

Les deux algorithmes précédents ont été programmés sur Pentium IV sous Windows en Borland Delphi 4. Nous avons traité des exemples de tailles différentes.

Dans le programme linéaire (3.7), on présente le vecteur $x \in \Re^N$ comme

$$x = (x_{ijkl}) = (x_{1111}, x_{1112}, .., x_{111q}, x_{1121}, .., x_{11pq}, x_{1211}, .., x_{1npq}, x_{2111}, .., x_{mnpq}).$$

Cependant, on le présente dans (4.1) comme

$$x = (x_t) = (x_1, x_2, ..., x_q, x_{q+1}, ..., x_{pq}, x_{pq+1}, ..., x_{npq}, x_{npq+1}, ..., x_N).$$

Dans tous les exemples ci-dessous, la solution optimale x^* sera représentée seulement par ses composantes non nulles.

Exemple 6 *Considérons un problème de transport sous forme* $(PT4C)$ *avec :*
$m = n = p = 2$ *et* $q = 1$, $\alpha_1 = 5$, $\alpha_2 = 10$, $\beta_1 = 9$, $\beta_2 = 6$, $\gamma_1 = 12$, $\gamma_2 = 3$, *et* $\delta_1 = 15$.
Les quantités c_{ijkl} *et* d_{ijkl} *sont données par le tableau :*

$ijkl$	*1111*	*1121*	*1211*	*1221*	*2111*	*2121*	*2211*	*2221*
d_{ijkl}	*15*	*10*	*8*	*9*	*7*	*11*	*11*	*9*
c_{ijkl}	*5*	*4*	*6*	*7*	*2*	*1*	*5*	*4*

Résolution :

Simplexe :

Le programme linéaire (4.1) s'écrit comme suit :

$\min Z = 5x_1 + 4x_2 + 6x_3 + 7x_4 + 2x_5 + x_6 + 5x_7 + 4x_8$

Sous les contraintes

$x_1 + x_2 + x_3 + x_4 = 5$

$x_5 + x_6 + x_7 + x_8 = 10$

$x_1 + x_2 + x_5 + x_6 = 9$

$x_3 + x_4 + x_7 + x_8 = 6$

$x_1 + x_3 + x_5 + x_7 = 12$

$x_2 + x_4 + x_6 + x_8 = 3$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 15$

$x_1 \leq 15, \quad x_2 \leq 10, \quad x_3 \leq 8, \quad x_4 \leq 9, \quad x_5 \leq 7, \quad x_6 \leq 11, \quad x_7 \leq 11,$

$x_8 \leq 9$ et $x_t \geq 0$ pour $t = 1, \ldots, 8$.

Les composantes $x_t \neq 0$ de x^* sont présentées dans le tableau suivant :

t	3	5	6	7
x_t	5	6	3	1

Nombre d'itérations = 9. Temps d'exécution = 1 sec.

La valeur optimale est $z^* = 50.00$

Alm$_{\mathbf{PT4C}}$:

Le programme linéaire (3.7) s'écrit comme :

$\min Z = 5x_{1111} + 4x_{1121} + 6x_{1211} + 7x_{1221} + 2x_{2111} + x_{2121} + 5x_{2211} + 4x_{2221}$

Sous les contraintes

$x_{1111} + x_{1121} + x_{1211} + x_{1221} = 5$

$x_{2111} + x_{2121} + x_{2211} + x_{2221} = 10$

$x_{1111} + x_{1121} + x_{2111} + x_{2121} = 9$

$x_{1211} + x_{1221} + x_{2211} + x_{2221} = 6$

$x_{1111} + x_{1211} + x_{2111} + x_{2211} = 12$

$x_{1121} + x_{1221} + x_{2121} + x_{2221} = 3$

$x_{1111} + x_{1121} + x_{1211} + x_{1221} + x_{2111} + x_{2121} + x_{2211} + x_{2221} = 15$

$0 \leq x_{1111} \leq 15, \quad 0 \leq x_{1121} \leq 10, \quad 0 \leq x_{1211} \leq 8, \quad 0 \leq x_{1221} \leq 9,$

$0 \leq x_{2111} \leq 7, \quad 0 \leq x_{2121} \leq 11, \quad 0 \leq x_{2211} \leq 11, \quad 0 \leq x_{2221} \leq 9.$

Les composantes $x_{ijkl} \neq 0$ de x^* sont présentées dans le tableau suivant :

$ijkl$	1211	2111	2121	2211
x_{ijkl}	5	6	3	1

Nombre d'itérations = 2. Temps d'exécution = 0.010 sec.

La valeur optimale est $z^* = 50.00$

Exemple 7 *Considérons un problème de transport sous forme* $(PT4C)$ *avec :*
$m = n = p = q = 2,\ \alpha_1 = 14,\ \alpha_2 = 12,\ \beta_1 = 10,\ \beta_2 = 16,\ \gamma_1 = 9,\ \gamma_2 = 17,$
$\delta_1 = 8,\ et\ \delta_2 = 18.$
Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau :

$ijkl$	*1111*	*1112*	*1121*	*1122*	*1211*	*1212*	*1221*	*1222*	*2111*	*2112*
d_{ijkl}	*12.5*	*15.2*	*14.1*	*16*	*22*	*13.4*	*21*	*20*	*10*	*25.2*
c_{ijkl}	*5*	*2.1*	*3*	*7*	*8.5*	*5*	*4.1*	*2.1*	*3.2*	*6.5*

$ijkl$	*2121*	*2122*	*2211*	*2212*	*2221*	*2222*
d_{ijkl}	*29.3*	*26*	*20*	*9*	*8*	*14*
c_{ijkl}	*9*	*8.2*	*5*	*4*	*7*	*1.1*

Résolution :

Simplexe :

Les composantes $x_t \neq 0$ de x^* sont présentées dans le tableau suivant :

t	2	3	7	9	16
x_t	7.5	1	5.5	1.5	10.5

Nombre d'itérations = 12. Temps d'exécution = 3 sec.

La valeur optimale est $z^* = 57.65$.

Alm$_{\mathbf{PT4C}}$:

Les composantes $x_{ijkl} \neq 0$ de x^* sont présentées dans le tableau suivant :

$ijkl$	1112	1121	1221	2111	2222
x_{ijkl}	7.5	1	5.5	1.5	10.5

Nombre d'itérations = 2. Temps d'exécution = 0.010 sec.

La valeur optimale est $z^* = 57.65$.

Exemple 8 *Considérons un problème de transport sous forme* $(PT4C)$ *avec :*
$m = 3,\ n = p = q = 2,\ \alpha_1 = 6,\ \alpha_2 = 8,\ \alpha_3 = 12,\ \beta_1 = 14,\ \beta_2 = 12,\ \gamma_1 = 9,\ \gamma_2 = 17,\ \delta_1 = 8,$ *et* $\delta_2 = 18$.
Les quantités c_{ijkl} *et* d_{ijkl} *sont données par le tableau :*

$ijkl$	*1111*	*1112*	*1121*	*1122*	*1211*	*1212*	*1221*	*1222*	*2111*	*2112*
d_{ijkl}	*12*	*15*	*14*	*16*	*22*	*13*	*21*	*20*	*25*	*29*
c_{ijkl}	*5*	*2.1*	*3*	*7*	*8*	*5*	*4*	*2*	*6*	*9*
$ijkl$	*2121*	*2122*	*2211*	*2212*	*2221*	*2222*	*3111*	*3112*	*3121*	*3122*
d_{ijkl}	*26*	*20*	*9*	*8*	*14*	*12*	*14*	*15*	*18*	*24*
c_{ijkl}	*8*	*5*	*4*	*7*	*4*	*1*	*5*	*2*	*3*	*6*

$ijkl$	*3211*	*3212*	*3221*	*3222*
d_{ijkl}	*24*	*23*	*21*	*23*
c_{ijkl}	*8*	*9*	*8*	*7*

Résolution :

Simplexe :

Les composantes $x_t \neq 0$ de x^* sont présentées dans le tableau suivant :

t	3	8	13	16	18	19
x_t	2	4	1.5	6.5	7.5	4.5

Nombre d'itérations = 12. Temps d'exécution = 9 sec.

La valeur optimale est $z^* = 55.00$.

Alm$_{\mathbf{PT4C}}$:

Les composantes $x_{ijkl} \neq 0$ de x^* sont présentées dans le tableau suivant :

$ijkl$	1121	1221	1222	2222	3112	3121
x_{ijkl}	2	3	1	8	9	3

Nombre d'itérations = 2. Temps d'exécution = 0.110 sec.

La valeur optimale est $z^* = 55.00$.

Exemple 9 *Considérons un problème de transport sous forme* $(PT4C)$ *avec :*
$m = 3,\ n = 2,\ p = 3,\ q = 2,\ \alpha_1 = 15,\ \alpha_2 = 15,\ \alpha_3 = 10,\ \beta_1 = 15,\ \beta_2 = 25,$
$\gamma_1 = 12,\ \gamma_2 = 14,\ \gamma_3 = 14,\ \delta_1 = 18,\ et\ \delta_2 = 22.$
Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau :

$ijkl$	*1111*	*1112*	*1121*	*1122*	*1131*	*1132*	*1211*	*1212*	*1221*	*1222*
d_{ijkl}	*9*	*8*	*12*	*10*	*11*	*12*	*13*	*14*	*15*	*10*
c_{ijkl}	*8*	*9*	*4*	*5*	*7*	*8*	*9*	*10*	*15*	*12*
$ijkl$	*1231*	*1232*	*2111*	*2112*	*2121*	*2122*	*2131*	*2132*	*2211*	*2212*
d_{ijkl}	*11*	*12*	*8*	*9*	*52*	*24*	*25*	*12*	*10*	*10*
c_{ijkl}	*1*	*1*	*14*	*5*	*9*	*9*	*6*	*7*	*5*	*10*
$ijkl$	*2221*	*2222*	*2231*	*2232*	*3111*	*3112*	*3121*	*3122*	*3131*	*3132*
d_{ijkl}	*11*	*12*	*22*	*21*	*21*	*10*	*10*	*10*	*21*	*21*
c_{ijkl}	*21*	*41*	*54*	*20*	*20*	*15*	*8*	*9*	*12*	*11*

$ijkl$	*3211*	*3212*	*3221*	*3222*	*3231*	*3232*
d_{ijkl}	*9*	*8*	*7*	*10*	*9*	*4*
c_{ijkl}	*10*	*5*	*8*	*9*	*6*	*7*

Résolution :

Simplexe :

Les composantes $x_t \neq 0$ de x^* sont présentées dans le tableau suivant :

t	3	11	12	14	16	19	27	33
x_t	1	2	12	7	3	5	4	6

Nombre d'itérations = 23. Temps d'exécution = 6 sec.

La valeur optimale est $z^* = 185.00$.

Alm$_{\mathbf{PT4C}}$:

Les composantes $x_{ijkl} \neq 0$ de x^* sont présentées dans le tableau suivant :

$ijkl$	1121	1231	1232	2112	2122	2211	3121	3221
x_{ijkl}	1	2	12	7	3	5	4	6

Nombre d'itérations = 3. Temps d'exécution = 0.030 sec.

La valeur optimale est $z^* = 185.00$.

Exemple 10 *Considérons un problème de transport sous forme* $(PT4C)$ *avec :*
$m = 4,\ n = 3,\ p = 2,\ q = 2,\ \alpha_1 = 13.5,\ \alpha_2 = 7.5,\ \alpha_3 = 10,\ \alpha_4 = 9,$ $\beta_1 = 24.75,\ \beta_2 = 6.25,\ \beta_3 = 9,\ \gamma_1 = 21.5,\ \gamma_2 = 18.5,\ \delta_1 = 5,\ et\ \delta_2 = 35.$
Les quantités c_{ijkl} et d_{ijkl} sont données par le tableau :

$ijkl$	*1111*	*1112*	*1121*	*1122*	*1211*	*1212*	*1221*	*1222*	*1311*	*1312*
d_{ijkl}	*15*	*30*	*75.5*	*35*	*14*	*13.25*	*12*	*10*	*21*	*23.1*
c_{ijkl}	*17*	*8*	*8.5*	*4*	*3.1*	*2*	*1*	*1*	*10*	*9.5*
$ijkl$	*1321*	*1322*	*2111*	*2112*	*2121*	*2122*	*2211*	*2212*	*2221*	*2222*
d_{ijkl}	*57*	*60.5*	*12.12*	*11*	*15.5*	*50*	*55*	*28*	*30*	*35*
c_{ijkl}	*8*	*7*	*6*	*3*	*4*	*6.5*	*10*	*11.12*	*9*	*3*
$ijkl$	*2311*	*2312*	*2321*	*2322*	*3111*	*3112*	*3121*	*3122*	*3211*	*3212*
d_{ijkl}	*39*	*40*	*60*	*69.8*	*73*	*74*	*74.7*	*19*	*18*	*17*
c_{ijkl}	*7*	*10*	*3.5*	*9*	*11*	*12*	*13*	*14.1*	*50*	*70*

$ijkl$	*3221*	*3222*	*3311*	*3312*	*3321*	*3322*	*4111*	*4112*	*4121*	*4122*
d_{ijkl}	*14*	*8.5*	*6*	*7*	*8*	*10*	*9.3*	*11*	*11*	*13*
c_{ijkl}	*21*	*25*	*34*	*40.4*	*9*	*6*	*7*	*3*	*5*	*8*

$ijkl$	*4211*	*4212*	*4221*	*4222*	*4311*	*4312*	*4321*	*4322*
d_{ijkl}	*35*	*40*	*45.8*	*46*	*50*	*51*	*52*	*60.4*
c_{ijkl}	*9.6*	*10*	*11*	*30*	*31.2*	*32*	*34*	*34*

Résolution :

Simplexe :

Les composantes $x_t \neq 0$ de x^* sont présentées dans le tableau suivant :

t	4	5	6	7	14	25	36	38
x_t	7.25	1.75	2.25	2.25	7.5	1	9	9

Nombre d'itérations = 26. Temps d'exécution = 10 sec.

La valeur optimale est $z^* = 155.675$.

Alm$_{\mathbf{PT4C}}$:

Les composantes $x_{ijkl} \neq 0$ de x^* sont présentées dans le tableau suivant :

$ijkl$	1122	1211	1212	1221	2112	3111	3322	4112
x_{ijkl}	7.25	1.75	2.25	2.25	7.5	1	9	9

Nombre d'itérations = 12. Temps d'exécution = 0.060 sec.

La valeur optimale est $z^* = 155.675$.

Le tableau suivant regroupe les résultats obtenus.

Tableau de comparaison

Exemple Numéro	Taille du problème		Nombre d'itérations		Valeur optimale (Z^*)		Temps d'exécution en secondes	
	Simplexe	Alm_{PT4C}	Simplexe	Alm_{PT4C}	Simplexe	Alm_{PT4C}	Simplexe	Alm_{PT4C}
6	15×16	7×8	9	2	50.00	50.00	1	0.010
7	24×32	$8{\times}16$	12	2	57.65	57.65	3	0.010
8	33×48	9×24	12	2	55.00	55.00	8	0.060
9	46×72	10×36	23	3	185.00	185.00	6	0.030
10	59×96	11×48	26	12	155.67	155.67	10	0.060
11	84×144	12×72	79	57	3504.72	3504.72	15	0.050
12	121×216	13×108	43	24	1802.34	1802.34	15	0.100
13	337×640	17×320	62	33	391.00	391.00	10	0.060

4.5 Commentaires

On note la supériorité de l'algorithme Alm_{PT4C} vis à vis de l'algorithme du simplexe tant en nombre d'itérations qu'en temps d'exécution. L'algorithme Alm_{PT4C} permet de traiter les problèmes de dégénérescence, ce que l'algorithme Al_{PT4C} ne permettait pas de faire .

Conclusion

Dans cette étude, nous avons pu apporter des réponses intéressantes, concernant le problème de transport à quatre indices avec capacités de somme axiale, non traité jusqu'à présent.

Sur le plan théorique, nous avons établi des conditions et de réalisabilté et d'optimalité. Ces résultats nous ont permis de construire un algorithme pour la résolution du problème.

L'étude numérique de l'algorithme est effectuée en deux étapes. La première est consacrée à l'implantation de la version originelle de l'algorithme, elle nous a permis de constater un comportement numérique encourageant. La deuxième étape est réalisée avec des aménagements réduisant considérablement le volume calculatoire et traitant convenablement les problèmes de dégénérescence.

Les résultats que nous avons obtenus sont indépendants du nombre d'indices, par conséquent, ils peuvent s'étendre pour un problème ayant un nombre d'indices supérieur à quatre.

Bibliographie

[1] S. Achmanov, Programmation linéaire, trad. Irina Petrova, Mir, Moscou, (1984).

[2] E. Balas and M.J. Saltzman, An algorithm for the three-index assignment problem, Oper. Res. 39, (1991), 150–161.

[3] M. S. Bazaraa, J. J. Javis and H. D. Sherali, Linear programming and network flows, John Wiley & Sons, 1990.

[4] S. A. Bulut, Construction and Algebraic Characterizations of the Planar and Axial Transportation Problems, J. Math. Anal. Appl., 220, (1998), 535–552.

[5] D. Cardoso and J. Climaco, The generalized simplex method, Oper. Res. Lett., 12, 5, (1992), 337-348.

[6] H. Chen, P. Pardalos and M. Saunders, The simplex algorithm with a new primal and dual pivot rule, Oper. Res. Lett., 16, (1994), 121-127.

[7] D. Cheung and F. Cucker, Solving linear programs with finite precision : II. Algorithms, J. Complexity, 22 , 3, (2006), 305 – 335.

[8] A. Corban, A multidimensional transportation problem, Rev. Roumaine Math. Pures Appl., 9, (1964), 721–735.

[9] A. Corban, On a three-dimensional transportation problem, Rev. Roumaine Math. Pures Appl., 11, (1966), 57–75.

[10] J.-C. Culioli, Introduction à loptimisation, Ellipses, 1994.

[11] G. B. Dantzig, Maximization of a linear function of variables subject to linear inequalities, in Koopmans, T.C. (ed.), Activity analysis of production and allocation, John Wiley & Sons, New York, (1951), 339347.

[12] K. Dosios and K. Paparrizos, Resolution of the problem of degeneracy in a primal and dual simplex algorithm, Oper. Res. Lett., 20, (1997), 45-50.

[13] S. C. Eisenstat, M.H. Schultz and A.H. Sherman, Algorithms and data structures for sparse symmetric Gaussian elimination, SIAM J. Sci. Sta. Comput., 2, (1981), 225-237.

[14] J.J.H. Forrest and D. Goldfarb, Steepest-edge simplex algorithms for linear programming, Math. Program., 57, (1992), 341-374.

[15] A.M. Frieze and J. Yadegar, An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice, J. Oper. Res. Sot., 32, (1981), 989–995.

[16] S. I. Gass and S. Vinjamuri, Cycling in linear programming problems, Comput. Oper. Res., 31, (2004), 303–311.

[17] K. B. Haley, The multi-index transportation problem, Oper. Res., 11, (1963), 368–379.

[18] J. A. J. Hall and K. I. M. McKinnon, The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling, Math. Program. Ser. B, 100, 1, (2004), 133–150.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal, Convex Analysis and Minimization Algorithms, T 1 et T 2, Springer-Verlag, Berlin, 1993.

[20] K. Holmberg, Exact solution methods for uncapacitated location problems with convex transportation costs, European J. Oper. Res., 114, (1999), 127–140.

[21] G. Huang and A. Lim, A hybrid genetic algorithm for the Three-Index Assignment Problem, European J. Oper. Res., 172, (2006), 249-257.

[22] N. Karmarkar, An Interior-Point Approach to NP-complete Problems, Proceedings of the 1st Integer Programming and Combinatorial Optimization Conference, (1990), 351–366.

[23] V. Klee and G.J. Minty, How Good is the Simplex Algorithm ? In O. Shisha, editor, Inequalities, III, Academic Press, New York, NY, (1972), 159–175.

[24] M. K. Kravtsov and A. P. Krachkovskii, Asymptotics of Multi-index Axial Transportation Polyhedra, Diskret. Mat., 10, 4, (1998), 61–81.

[25] M. K. Kravtsov, A Counter example to the Hypothesis of Maximum Number of Integer Verrtices of the Multi-index Axial Transportation Polyhedron, Diskret. Mat., 12, 1, (2000), 107–112.

[26] M. K. Kravtsov and E. V. Lukshin, On the Noninteger Polyhedron Vertices of the Three-index Axial Transportation Problem, Autom. Remote Control, 65, 3, (2004), 422–430.

[27] S. -T. Liu, The total cost bounds of the transportation problem with varying demand and supply, Omega, 31, (2003), 247 - 251.

[28] S. -T. Liu, and Chiang Kao, Solving fuzzy transportation problems based on extension principle, European J. Oper. Res., 153, (2004), 661-674.

[29] M. Minoux, Programmation Mathématique : Théorie et algorithmes, T 1 et T 2, Dunod, 1983.

[30] K. Paparrizos, N. Samaras and G. Stephanides, A new efficient primal dual simplex algorithm, Comput. Oper. Res., 30, (2003), 1383–1399.

[31] M. Queyranne and F.C.R. Spieksma, Approximation algorithms for multi-index transportation problems with decomposable costs, Discrete Appl. Math., 76, (1997), 239–253.

[32] C.S. ReVelle and H.A. Eiselt, Location analysis : A synthesis and survey, European J. Oper. Res., 165, (2005), 1-19.

[33] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, 1970.

[34] M. Sakharovitch, Graphes et programmation linéaire, Hermann, Paris, (1984).

[35] M. Sakharovitch, Programmation discrète, Hermann, Paris, (1984).

[36] R.R.K. Sharma and S. Prasad, Obtaining a good primal solution to the uncapacitated transportation problem, European J. Oper. Res. , 144, (2003), 560–564.

[37] R.R.K. Sharma and K.D. Sharma, A new dual based procedure for the transportation problem, European J. Oper. Res., 122, (2000), 611–624.

[38] M. Simonard, Programmation linéaire, Dunod, Paris, T 1 et T 2, 1972.

[39] I. M. Stancu-Minasian, A three-dimensional transportation problem with a special structured objective function, Bull. Math. Soc. Sci. Math. Roumanie (N.S.) , 18(66), 3/4, (1974), 385–397.

[40] G. H. Tzeng, D. Teodorović and M. J. Hwang, Fuzzy bicriteria multi-index transportation problems for coal allocation planning of Taipower, European J. Oper. Res., 95, (1996), 62–72.

[41] J.R. Vera, On the complexity of linear programming under finite precision arithmetic, Math. Program., 80, (1998), 91-123.

[42] R. Zitouni and A. Keraghel, Resolution of a capacitated transportation problem with four subscripts, Kybernetes, 32, 9/10, (2003), 1450–1463.

[43] R. Zitouni and A. Keraghel, A note on the algorithm of resolution of a capacitated transportation problem with four subscripts, à apparaître dans le journal (FJMS).