

CATHERINE ROUCAIROL

Un nouvel algorithme pour le problème d'affectation quadratique

RAIRO. Recherche opérationnelle, tome 13, n° 3 (1979), p. 275-301

http://www.numdam.org/item?id=RO_1979__13_3_275_0

© AFCET, 1979, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

UN NOUVEL ALGORITHME POUR LE PROBLÈME D'AFFECTATION QUADRATIQUE (*)

par Catherine ROUCAIROL ⁽¹⁾

Résumé. — *Les problèmes d'affectation à fonction économique quadratique se présentent dans de nombreux domaines d'activité pratique : économie, électronique, aide à la conception en architecture, ergonomie, informatique... Leur complexité est telle (problèmes NP complet), que les méthodes existantes n'abordaient jusqu'à présent que des problèmes de taille 10×10 en des temps peu raisonnables. Nous proposons une méthode originale dite « méthode par réduction », qui permet d'obtenir l'optimum des problèmes de taille inférieure ou égale à 10 plus rapidement que par les méthodes utilisées auparavant, et des solutions satisfaisantes pour des problèmes de taille supérieure : l'écart entre la solution trouvée et la solution optimale pouvant être borné, il est alors possible de juger de la qualité de cette solution.*

Abstract. — *Assignment problems with quadratic objective function (QAP) can be found in economics (plant location problem), electronics (backboard wiring problem), computer aided design in architecture (layout of hospitals), ergonomics (development of new typewriter keyboards). We propose a new method "the reduction method" based upon a reduction procedure followed by a branch and bound; this method produces optimal solutions to QAP's ($n \leq 10$) faster than any other known method and for $n > 10$, good solution together with an upper bound on its deviation from optimality.*

INTRODUCTION

Des besoins concrets (aide à la conception en architecture) nous ont poussé à rechercher une procédure *pratique* permettant de résoudre les problèmes d'affectation quadratique de taille moyenne (entre 10 et 30), pour lesquels les méthodes proposées précédemment étaient défaillantes.

La méthode par réduction des matrices et décomposition en parties linéaire et quadratique de la fonction économique, que nous décrivons ci-après, a été programmée et essayée sur de nombreux exemples et semble efficace.

Elle permet d'aborder en un temps raisonnable les problèmes de taille voulue et de connaître une borne de l'erreur commise entre la solution retenue et l'optimum théorique.

(*) Reçu juillet 1978.

(1) Institut de Programmation, Université Paris-VI.

I. DÉFINITION DU PROBLÈME D'AFFECTATION QUADRATIQUE

Le problème d'affectation se présente fréquemment en recherche opérationnelle. Il consiste à réaliser une bijection des éléments i d'un ensemble I sur ceux j d'un ensemble J , de même cardinalité, de telle manière qu'une certaine fonction de coût, dépendant du choix des couples (i, j) soit minimale.

Lorsque cette fonction de coût est linéaire, c'est un problème classique et sa solution est donnée par un algorithme polynomial [en $O(n^4)$, si $|I| = |J| = n$] : la méthode hongroise.

Pourtant, il existe des problèmes appartenant à de nombreux domaines, aussi variés que l'électronique, l'économie, l'architecture, l'informatique, etc., pour lesquels la fonction de coût est quadratique. Le problème d'affectation s'appelle alors « affectation quadratique »; c'est un problème NP complet et donc beaucoup plus difficile à résoudre que l'affectation linéaire.

Programme quadratique équivalent

Résoudre un problème d'affectation quadratique, revient à résoudre le programme quadratique suivant :

Problème d'affectation quadratique (noté A.Q.) :

$$(I) \left\{ \begin{array}{ll} x_{ij} = 0 \text{ ou } 1, & i \in I, \quad j \in J, \quad |I| = |J|, \\ \sum_{j \in J} x_{ij} = 1, & \forall i \in I, \\ \sum_{i \in I} x_{ij} = 1, & \forall j \in J, \\ [\text{MIN}] \sum_{i \in I} \sum_{j \in J} \sum_{k \in I} \sum_{l \in J} c_{ijkl} x_{ij} x_{kl}. \end{array} \right. \quad \begin{array}{l} (1) \\ (2) \\ (3) \\ (4) \end{array}$$

Les trois premières contraintes définissent une affectation :

si x_{ij} vaut 1, l'élément i de I est affecté à l'élément j de J (1^{re} contrainte);
à un élément de I est affecté un et un seul élément de J (2^e et 3^e contraintes).

L'affectation optimale est celle qui minimise une fonction de coût à valeurs positives, d'où la fonction économique (4) où :

c_{ijkl} représente le coût lié à l'affectation respective de i en j et de k en l .

Une notation plus condensée permet d'alléger l'écriture du problème et de faciliter la présentation des méthodes de résolution.

En effet, une affectation de n éléments peut être définie comme une permutation ρ de l'ensemble des n premiers nombres entiers. Le problème A.Q. s'écrit alors :

$$\min_{\rho \in P} \sum_{i=1}^n \sum_{j=1}^n c_{i\rho(i)j\rho(j)},$$

P ensemble des permutations de $\{1, 2, \dots, n\}$.

Exemples de problèmes d'affectation quadratique

Les problèmes réels qui ont la structure mathématique d'un problème A.Q., se rencontrent dans de nombreux contextes. Les trois domaines où ce problème se pose le plus fréquemment sont ceux de l'économie, l'électronique, l'architecture.

Il est remarquable que les éléments de la matrice des coûts y soient toujours en fait le produit des éléments de deux matrices d'ordre n , à éléments non négatifs :

$$c_{ijkl} = f_{ik} \cdot d_{jl}, \quad \text{où } F = (f_{ik}), \quad D = (d_{jl}).$$

Dans ces conditions, le problème d'affectation quadratique s'écrit simplement :

$$\left\{ \begin{array}{l} x_{ij} = 0 \text{ ou } 1, \quad i \in I, j \in J, \quad |I| = |J| = n, \\ \sum_{j \in J} x_{ij} = 1, \quad \forall i \in I, \\ \sum_{i \in I} x_{ij} = 1, \quad \forall j \in J, \\ \text{[MIN]} \sum_{i \in I} \sum_{j \in J} \sum_{k \in I} \sum_{l \in J} f_{ik} \cdot d_{jl} x_{ij} x_{kl}. \end{array} \right.$$

En économie, le problème fut posé pour la première fois par Koopmans et Beckmann [11], en 1957, sous le nom anglais d'« indivisible resources allocation problem ». Il s'agit d'implanter diverses unités de production à des emplacements différents déjà déterminés. Des quantités fixées de produits (marchandises, matières premières, ...) doivent circuler entre ces usines, et leur coût de transport dépend de la distance parcourue.

L'implantation des usines cherchée est évidemment celle qui assurera un coût de transport minimal :

I est l'ensemble des n usines;

J , celui des n emplacements possibles;

$x_{ij} = 1$ si l'usine i est implantée en j , $= 0$ sinon;

f_{ik} , la quantité de produits à transporter de l'usine i à l'usine k ;

d_{jl} , le coût de transport d'une unité de produits de l'emplacement j à l'emplacement l .

Dans l'industrie des automatismes et ordinateurs, le problème se pose lors de la conception de la plaquette qui doit supporter les circuits intégrés. Pour obtenir un bon tracé de connexions, il faut chercher à implanter les modules de circuits intégrés sur cette plaquette de manière à minimiser la longueur totale des liaisons :

I est alors l'ensemble des modules;

J , celui des emplacements possibles pour ces modules sur la plaquette;

$x_{ij} = 1$, si le module i est placé en j , $= 0$ sinon;

f_{ik} , le nombre de liaisons du module i au module k ;

d_{jl} , la longueur d'une liaison, c'est-à-dire la distance euclidienne ou rectangulaire entre l'emplacement j et l'emplacement l .

Lors de la conception des projets architecturaux tels que planning de bureaux, réorganisation de services dans l'hôpital..., l'architecte peut répertorier les emplacements où peuvent se dérouler différentes activités sur un site, puis se fonder pour implanter ces activités sur un critère de contiguïté. Plus certaines activités apparaissent comme complémentaires, c'est-à-dire, plus les flux de personnel, information ou matériel, circulant entre elles, sont importants, plus ces activités devront être voisines sur le futur plan. L'architecte cherche alors le « schéma de plan » qui minimise les distances de circulations entre activités :

I est alors l'ensemble des activités devant se dérouler dans le projet;

J , l'ensemble des emplacements possibles;

$x_{ij} = 1$, si l'activité i est implantée en j , $= 0$ sinon;

f_{ik}^h , un nombre traduisant l'importance de la liaison entre deux activités i et k mesurée par la quantité de flux h s'écoulant entre i et k ;

d_{jl}^h , la distance sur le site mesurée à travers les axes de circulation existant pour le flux h , entre les emplacements j et l .

II. MÉTHODES DE RÉOLUTION EXISTANTES

Pour résoudre ce problème qui rappelons-le, est un problème NP complet admettant le problème du voyageur de commerce comme cas particulier (il suffit de prendre pour matrice du circuit hamiltonien $(1, 2, \dots, n, 1)$:

$$F = \begin{bmatrix} 0 & 1 & & & 0 \\ & 0 & 0 & 1 & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 1 & & & & 0 \end{bmatrix}$$

et pour D la matrice des distances entre villes), il existe deux sortes de méthodes : exactes ou approchées.

Méthodes visant un optimum

Parmi les méthodes dont l'objectif est d'atteindre un optimum, certaines [13, 2] proposent de :

trouver un programme linéaire équivalent au programme quadratique.

Une formulation, donnée par Lawler [13], a dû être abandonnée car le nombre des variables, ainsi que celui des contraintes du programme linéaire équivalent excèdent alors n^4 , ce qui place les problèmes de taille initiale 10 largement hors des limites où le traitement des programmes en entiers peut encore se pratiquer effectivement !

Les autres méthodes utilisent des *procédures de recherche arborescente* : Lawler [13], Gilmore [8], Pierce et Crowston [16], Burkard [3], Gavett et Plyter [7]. L'affectation optimale cherchée y est construite pas à pas : le critère « un élément i (ou un couple d'éléments) est affecté à un élément j » permet de séparer l'ensemble des affectations de n éléments. La construction de l'arborescence des solutions est guidée par le calcul d'une borne inférieure du coût des affectations appartenant à l'ensemble séparé. Mais, en dépit des améliorations qui leur ont été apportées (Roucairol, [17]), ces algorithmes ne permettent de traiter que des problèmes de taille inférieure à 10, et encore pas toujours. De plus, les temps obtenus ne sont souvent guère raisonnables.

Pour une affectation de $n=8$ éléments, 1 mn 30 s sur Univac 494, $n=9$ éléments, 2 minutes sur Univac 494.

Il était donc fatal que des méthodes d'approximation fussent tentées.

Méthodes d'approximation

Qu'elles soient dérivées des méthodes par construction d'arborescence précédentes (une seule branche est choisie à l'aide d'heuristiques) : méthodes de Gilmore [8], Hilliers et Connors [10], qu'elles soient fondées sur des heuristiques : Müller Merbach [14], une représentation graphique de la matrice C des coûts : Gaschutz et Ahrens [6], ou utilisent des notions de probabilité : Graves et Whinston [9], ou encore procèdent par amélioration d'une affectation tirée au hasard (transposition des affectations de certains éléments) : Buffa [1], Nugent *et al.* [15], Roucairol [17], aucune ne donne une idée de la qualité de la solution retenue finalement.

Les seuls éléments de jugement fournis par leurs auteurs sont la comparaison avec les résultats obtenus sur de mêmes exemples « célèbres » empruntés à la littérature (exemples de Steinberg, Armour et Buffa, Nugent). Les méthodes les plus performantes semblent alors être les dernières citées, c'est-à-dire celles qui consistent à tirer un échantillon d'affectations au hasard, puis à les rendre λ -optimales, c'est-à-dire de coût inférieur à celui de toute affectation obtenue en effectuant une permutation quelconque de tout sous-ensemble de λ éléments.

Pour $\lambda=3$ et un échantillon de taille 10, nous avons obtenu de bonnes solutions ([17] et tableau III) sur les exemples de la littérature ($n \geq 20$) cités précédemment.

III. UNE MÉTHODE VISANT L'OPTIMUM : « LA MÉTHODE PAR RÉDUCTION »

La méthode par réduction consiste à mettre la fonction de coût sous la forme de la somme d'une fonction linéaire et d'une fonction quadratique, dite « réduite ». Cette décomposition permet de trouver aisément à la fois une borne inférieure et une borne supérieure de la solution optimale. Nous utilisons ensuite une procédure P.S.E.P. fondée sur la notion de regret; l'obtention des deux bornes précédentes permet de guider efficacement la construction de l'arborescence.

Le problème d'affectation quadratique (A.Q.) est posé sous la forme :

Chercher la permutation ρ de $\mathcal{J} = \{1, 2, \dots, n\}$ qui minimise

$$(A.Q.) \quad \begin{cases} Z(\rho) = \sum_i \sum_{j \neq i} c_{ij} d_{\rho(i)\rho(j)}, \\ \text{avec } C = (c_{ij}), D = (d_{kl}), \text{ matrices à éléments non négatifs, d'ordre } n, \text{ à} \\ \text{diagonale nulle }^{(2)}. \end{cases}$$

1. Première phase : réduction

1.1. DÉFINITION

Réduire une matrice à éléments non négatifs, d'ordre n , $M = (m_{ij})$, c'est trouver $2n$ nombres non négatifs α_i, β_j tels que la matrice M' d'éléments

$$m'_{ij} = m_{ij} - \alpha_i - \beta_j,$$

soit une matrice à éléments non négatifs, comportant au moins un zéro par ligne et colonne.

1.2. PROBLÈME D'AFFECTATION QUADRATIQUE RÉDUIT

Soient $C' = (c'_{ij})$ la matrice obtenue par réduction de C :

$$c'_{ij} = c_{ij} - \alpha_i - \beta_j, \quad \alpha_i, \beta_j, c'_{ij} \geq 0.$$

$D' = (d'_{kl})$ celle obtenue par réduction de D :

$$d'_{kl} = d_{kl} - \alpha'_k - \beta'_l, \quad \alpha'_k, \beta'_l, d'_{kl} \geq 0.$$

Nous appelons *problème d'affectation quadratique réduit* le problème d'affectation de fonction économique

$$Z'(\rho) = \sum_i \sum_{j \neq i} c'_{ij} d'_{\rho(i)\rho(j)}$$

⁽²⁾ Cette hypothèse simplifie l'écriture du problème. Le cas où les diagonales de C et D ne sont pas nulles revient à ajouter le terme linéaire $\sum_i c_{ii} d_{\rho(i)\rho(i)}$ à $Z(\rho)$.

Une relation lie la fonction économique du problème d'affectation quadratique et celle du problème réduit. En développant la formule ci-dessous :

$$Z'(\rho) = \sum_i \sum_{j \neq i} (c_{ij} - \alpha_i - \beta_j) (d_{\rho(i)\rho(j)} - \alpha'_{\rho(i)} - \beta'_{\rho(j)}).$$

Il vient

$$\begin{aligned} Z'(\rho) = & \sum_i \sum_{j \neq i} c_{ij} d_{\rho(i)\rho(j)} - \sum_i \alpha_i \sum_{j \neq i} d_{\rho(i)\rho(j)} - \sum_{j \neq i} \beta_j \sum_i d_{\rho(i)\rho(j)} \\ & - \sum_i \sum_{j \neq i} c_{ij} \alpha'_{\rho(i)} + (n-1) \sum_i \alpha_i \alpha'_{\rho(i)} \\ & + (\sum_i \alpha_i) (\sum_i \beta_i) - \sum_i \beta_i \alpha'_{\rho(i)} - \sum_{j \neq i} \beta'_{\rho(j)} \sum_i c_{ij} \\ & + (\sum_i \alpha_i) (\sum_i \beta_i) - \sum_i \alpha_i \beta'_{\rho(i)} + (n-1) \sum_j \beta_j \beta'_{\rho(j)}. \end{aligned}$$

Ce qui s'écrit encore :

$$\begin{aligned} Z'(\rho) = & Z(\rho) - \sum_i [\alpha_i \sum_{j \neq i} d_{\rho(i)\rho(j)} + \beta_i \sum_{j \neq i} d_{\rho(j)\rho(i)} \\ & + \alpha'_{\rho(i)} \sum_{j \neq i} c_{ij} + \beta'_{\rho(i)} \sum_{j \neq i} c_{ji} \\ & - (n-1)(\alpha_i \alpha'_{\rho(i)} + \beta_i \beta'_{\rho(i)} + \alpha_i \beta'_{\rho(i)} + \beta_i \alpha'_{\rho(i)})] \\ & + (\sum_i \alpha_i) (\sum_i \beta_i) + (\sum_i \alpha_i) (\sum_i \beta_i). \end{aligned}$$

Appelons le terme constant positif γ :

$$\gamma = (\sum_i \alpha_i) (\sum_i \beta_i) + (\sum_i \alpha_i) (\sum_i \beta_i)$$

et l'intérieur du crochet

$$k_{ip} = \alpha_i \sum_{l \neq p} d_{pl} + \beta_i \sum_{l \neq p} d_{lp} + \alpha'_p \sum_{r \neq i} c_{ir} + \beta'_p \sum_{r \neq i} c_{ri} - (n-1)(\alpha_i \alpha'_p + \beta_i \beta'_p) + \alpha_i \beta'_p + \alpha'_p \beta_i.$$

k_{ip} est l'élément d'une matrice K d'ordre n .

Posons

$$K(\rho) = \sum_i k_{i\rho(i)}.$$

D'où

$$Z(\rho) = Z'(\rho) + K(\rho) - \gamma.$$

PROPOSITION : La matrice $K=(K_{ip})$ est une matrice à éléments non négatifs

$$\sum_{l \neq p} d_{pl} \geq \sum_{l \neq p} (\alpha'_p + \beta'_l) = (n-1) \alpha'_p + (\sum_l \beta'_l) - \beta'_p.$$

De même

$$\sum_{l \neq p} d_{lp} \geq (\sum_l d'_l) - \alpha'_p + (n-1) \beta'_p.$$

Une borne inférieure de k_{ip} peut ainsi être calculée

$$\begin{aligned} k_{ip} &\geq \alpha_i [(n-1) \alpha'_p + (\sum_l \beta'_l) - \beta'_p] + \beta_i [(\sum_l \alpha'_l) - \alpha'_p + (n-1) \beta'_p] \\ &\quad + \alpha'_p \sum_{r \neq i} c_{ir} + \beta'_p \sum_{r \neq i} c_{ri} - (n-1)(\alpha_i \alpha'_p + \beta_i \beta'_p) + \alpha_i \beta'_p + \beta_i \alpha'_p. \\ k_{ip} &\geq \alpha_i (\sum_l \beta'_l) + \beta_i (\sum_l \alpha'_l) + \alpha'_p \sum_{r \neq i} c_{ir} + \beta'_p \sum_{r \neq i} c_{ri}, \end{aligned}$$

la borne inférieure de k_{ip} étant une somme de termes positifs

$$\forall i, \quad p \in \mathcal{I} \times \mathcal{J}, \quad k_{ip} \geq 0.$$

THÉORÈME :

$$\forall p, \quad Z(p) = Z'(p) + K(p) - \gamma,$$

où $Z'(p)$ est la fonction économique du problème d'affectation quadratique réduit, $K(p)$ la fonction économique d'un problème d'affectation linéaire de matrice K et γ une constante positive.

Ce théorème découle directement de ce qui précède. Nous avons donc pu décomposer la fonction coût d'un problème A.Q. en la somme d'une fonction quadratique et d'une fonction linéaire. Cette décomposition va permettre de trouver aisément à la fois une borne inférieure et une « bonne » borne supérieure de la solution optimale du problème A.Q..

1.3. BORNE INFÉRIEURE DE $Z(p)$

$$\forall p \in P, \quad Z(p) \geq \min_p K(p) - \gamma.$$

$$Z(p) \text{ s'écrivant : } Z(p) = Z'(p) + K(p) - \gamma.$$

$$K(p) - \gamma = Z(p) - Z'(p) = \sum_i \sum_{j \neq i} c_{ij} d_{p(i)p(j)} - c'_{ij} d'_{p(i)p(j)} \geq 0$$

(car $\forall i, j, c_{ij} \geq c'_{ij}, d_{p(i)p(j)} \geq d'_{p(i)p(j)}$)

Si p_k est la permutation qui minimise $K(p) = \sum_i k_{ip(i)}$:

$$K(p_k) = \min_p K(p),$$

alors

$$\forall \rho, \quad Z(\rho) \geq K(\rho) - \gamma \geq \min_{\rho} K(\rho) - \gamma.$$

La recherche de ρ_k amène à résoudre un problème d'affectation linéaire de matrice des coûts K . Nous utiliserons la classique méthode hongroise [12].

1.4. BORNE SUPÉRIEURE

Soit ρ^* , la permutation cherchée

$$Z(\rho^*) = \min_{\rho} Z(\rho).$$

On a

$$Z(\rho^*) \leq Z(\rho_k).$$

Le calcul de $Z(\rho_k)$ est simple lorsqu'on connaît $K(\rho_k) - \gamma$ puisque

$$Z(\rho_k) = K(\rho_k) - \gamma + Z'(\rho_k),$$

Finalement :

$$\min_{\rho} K(\rho) - \gamma \leq Z(\rho^*) \leq Z(\rho_k),$$

$Z'(\rho_k)$ représente l'écart entre la borne inférieure et la borne supérieure de $Z(\rho^*)$.

Nous pouvons donc en déduire que :

PROPOSITION :

$$Z'(\rho_k) = 0 \Rightarrow \rho^* = \rho_k.$$

1.5. EXEMPLE

Nous allons présenter deux types différents de réduction sur l'exemple traditionnellement utilisé dans la littérature (Gavett-Plyter, [7]).

Il s'agit de trouver la permutation de $\mathcal{J} = \{1, 2, 3, 4\}$ qui minimise

$$Z(\rho) = \sum_i \sum_{j \neq i} c_{ij} d_{\rho(i)\rho(j)}.$$

où

		1	2	3	4			1	2	3	4	
	1		28	25	13		1		6	7	2	
$C = 2$	28			15	4		$D = 2$		6		5	6
3	25	15			23		3		7	5		1
4	13	4	23				4		2	6	1	

Réduction 1 (Réduction 1)

La façon la plus simple de procéder pour réduire une matrice est de soustraire à chacune de ses lignes son minimum, puis de procéder de même pour chaque colonne

$$\begin{array}{c}
 C = \begin{array}{|c|c|c|c|} \hline & 28 & 25 & 13 \\ \hline 28 & & 15 & 4 \\ \hline 25 & 15 & & 23 \\ \hline 13 & 4 & 23 & \\ \hline \end{array} \quad \begin{array}{l} \alpha_1 = 13 \\ \alpha_2 = 4 \\ \alpha_3 = 15 \\ \alpha_4 = 4 \end{array} \rightarrow \begin{array}{|c|c|c|c|} \hline & 15 & 12 & 0 \\ \hline 24 & & 11 & 0 \\ \hline 10 & 0 & & 8 \\ \hline 9 & 0 & 19 & \\ \hline \end{array} \rightarrow C' = \begin{array}{|c|c|c|c|} \hline & 15 & 1 & 0 \\ \hline 15 & & 0 & 0 \\ \hline 1 & 0 & & 8 \\ \hline 0 & 0 & 8 & \\ \hline \end{array}
 \end{array}$$

$\beta_1 = 9 \quad \beta_3 = 11$
 α'

$$D' = \begin{array}{|c|c|c|c|} \hline & 0 & 5 & 0 \\ \hline 0 & & 0 & 1 \\ \hline 5 & 0 & & 0 \\ \hline 0 & 1 & 0 & \\ \hline \end{array} \quad \begin{array}{l} 2 \\ 5 \\ 1 \\ 1 \end{array}$$

$\beta' \quad 1 \quad 4 \quad 0 \quad 0$

Calculons K dans le cas où C et D sont symétriques :

$$k_{ip} = (\alpha_i + \beta_i) \left(\sum_l d_{pl} \right) + \left(\sum_r c_{ir} \right) (\alpha'_p + \beta'_p) - 3(\alpha_i \alpha'_p + \beta_i \beta'_p) + \beta_i \alpha'_p + \alpha_p \beta'_i.$$

$$\begin{array}{c}
 \begin{array}{|c|c|c|c|c|} \hline & 1 & 2 & 3 & 4 \\ \hline K = 1 & 454 & 762 & 322 & 234 \\ \hline 2 & 181 & 447 & 87 & 71 \\ \hline 3 & 493 & 767 & 367 & 263 \\ \hline 4 & 160 & 384 & 80 & 64 \\ \hline \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{|c|c|c|c|c|} \hline & 4 & 3 & 1 & 2 \\ \hline K = 3 & 263 & 367 & 493 & 767 \\ \hline 1 & 234 & 322 & 454 & 762 \\ \hline 2 & 71 & 87 & 181 & 447 \\ \hline 4 & 64 & 80 & 160 & 384 \\ \hline \end{array}
 \end{array}$$

$$\gamma = 36.5 + 9.20 = 360.$$

Pour chercher la permutation ρ_k qui minimise $\sum_i k_{ip(i)}$, nous pouvons sur cet exemple ne pas employer l'algorithme hongrois. En effet, nous pouvons remarquer (matrice de droite) qu'il existe une permutation des lignes et des colonnes de K telle que :

tous les éléments d'une ligne de K soient rangés par ordre croissant, tous les éléments d'une colonne de K par ordre décroissant.

C'est donc la permutation optimale :

$$\rho_k = \begin{pmatrix} 3 & 1 & 2 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix},$$

$$K(\rho_k) = 1\,150, \quad Z'(\rho_k) = 166, \quad Z(\rho_k) = 790 + 166 = 956,$$

$$Z(\rho) \geq 1\,150 - 360 = 790, \quad \underline{790 \leq Z(\rho) \leq 956}.$$

Réduction 2 (Réd 2)

De la réduction des matrices C et D dépend l'importance de la grandeur du terme linéaire par rapport à celle du terme quadratique. C'est pourquoi, nous avons défini une autre réduction qui permet de diminuer l'importance du terme quadratique.

Elle est fondée sur le principe suivant :

« diminuer au maximum d'abord les coefficients les plus grands de la matrice » :

$$C = \begin{bmatrix} & 28 & 25 & 13 \\ 28 & & 15 & 4 \\ 25 & 15 & & 23 \\ 13 & 4 & 23 & \end{bmatrix} \quad \alpha_1 = 13$$

$$\beta_1 = 13$$

Dans le cas de la matrice C , il vient donc

$$\max_{i,j} c_{ij} = c_{12} = 28;$$

$$\min_{j \neq 1} c_{1j} = 13 > \min_{i \neq 2} c_{i2} = 4, \quad \text{d'où} \quad \alpha_1 = 13; \quad \beta_4 = 0.$$

C étant symétrique, $\beta_1 = 13$, $\alpha_4 = 0$.

$$\begin{bmatrix} & 15 & 12 & 0 \\ 15 & & 15 & 4 \\ 12 & 15 & & 23 \\ 0 & 4 & 23 & \end{bmatrix} \alpha_3 = 12 \quad \rightarrow \quad \begin{bmatrix} & 15 & 0 & 0 \\ 15 & & 3 & 4 \\ 0 & 3 & & 11 \\ 0 & 4 & 11 & \end{bmatrix} \alpha_2 = 3 \quad \rightarrow \quad \begin{bmatrix} & 12 & 0 & 0 \\ 12 & & 0 & 1 \\ 0 & 0 & & 11 \\ 0 & 1 & 11 & \end{bmatrix}$$

$$\beta_3 = 12 \quad \beta_2 = 3$$

De la même façon

$$D' = \begin{bmatrix} & 0 & 4 & 0 \\ 0 & & 1 & 2 \\ 4 & 0 & & 0 \\ 0 & 2 & 0 & \end{bmatrix} \quad \alpha' = 2 \quad \gamma = 392$$

$$K = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 550 & 762 & 418 & 234 \\ 254 & 430 & 160 & 54 \\ 516 & 720 & 390 & 216 \\ 160 & 320 & 80 & 0 \end{bmatrix} \end{matrix}$$

$$\beta' \begin{matrix} 2 & 4 & 1 & 0 \end{matrix}$$

$$\rho_k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix},$$

$$\min_{\rho} K(\rho) = K(\rho_k) = 1\,198,$$

$$Z(\rho) \geq 1\,198 - 392 = 806.$$

$$Z'(\rho_k) = 0 \Rightarrow \rho_k = \rho^*, \quad Z(\rho^*) = 806.$$

La réduction (Réd 2) des matrices C et D a permis de trouver directement l'affectation optimale.

2. Deuxième phase : procédure arborescente

L'affectation optimale est construite pas à pas grâce à une procédure par séparation et évaluation progressive (P.S.E.P.) selon la terminologie de B. Roy.

La racine de l'arborescence sommet S_0 est l'ensemble P des permutations ρ de $I = \{1, 2, \dots, n\}$.

2.1. SÉPARATION

Le principe de séparation est :

« un élément i est affecté ou non à un élément j ». C'est une application Γ définie sur les parties de P , $\mathcal{P}(P)$ telle que :

pour tout sommet S_k de l'arborescence A :

$$\Gamma(S_k) = \{S_{k+1}, S_{k+2}\},$$

avec

$$S_{k+1} = \{\rho \in S_k \mid \rho(i) = j\}, \quad S_{k+2} = \{\rho \in S_k \mid \rho(i) \neq j\},$$

ρ étant une permutation de I .

Nous dirons que le sommet S_{k+1} correspond à un choix positif, le sommet S_{k+2} à un choix négatif.

2.2. ÉVALUATION

Définitions préalables

Une permutation partielle de I sur X est une application bijective de $X \subset I$ dans une partie de I .

A une telle permutation, correspond donc une affectation partielle : seuls $|X|$ éléments sont affectés.

Nous dirons qu'une permutation partielle β de I sur Y est une *extension* d'une autre permutation partielle α de I sur $X = I \setminus Y$ ($X \cap Y = \emptyset$) si l'application $\rho = \alpha\beta$ et définie par : $\forall i \in X, \rho(i) = \alpha(i)$ et $\forall i \in Y, \rho(i) = \beta(i)$, est une permutation de I .

Soit S_I un sommet pendant de l'arborescence construit après un certain nombre de séparations, donc de choix positifs et négatifs; supposons que k éléments aient déjà été affectés. Toute permutation ρ appartenant à S_I peut donc s'écrire : $\rho = \alpha\beta$ avec α , permutation partielle de I définie sur I_1 , $I_1 \subset I$, $|I_1| = k$; β , permutation extension de α sur I .

Notons $\bar{I}_1 = I \setminus I_1$.

La valeur de la fonction économique du problème A.Q., associée à la permutation ρ , est donc

$$\begin{aligned} \forall \rho = \alpha\beta \in S_I, \quad Z(\rho) &= Z'(\rho) + K(\rho) - \gamma, \\ Z'(\rho) &= \sum_{i \in I_1} \sum_{j \in I_1} c'_{ij} d'_{\alpha(i) \alpha(j)} + \sum_{i \in I_1} \sum_{j \in \bar{I}_1} c'_{ij} d'_{\alpha(i) \beta(j)} \\ &\quad + \sum_{j \in \bar{I}_1} \sum_{i \in I_1} c'_{ji} d'_{\beta(j) \alpha(i)} + \sum_{i \in \bar{I}_1} \sum_{j \in \bar{I}_1} c'_{ij} d'_{\beta(i) \beta(j)}. \end{aligned}$$

ce que nous noterons

$$Z'(\rho) = Z'(\alpha) + Z'(\alpha\beta) + Z'(\beta\alpha) + Z'(\beta),$$

$$K(\rho) = \sum_{s \in I_1} k_{s\alpha}(s) + \sum_{s \in \bar{I}_1} k_{s\beta}(s) = K(\alpha) + K(\beta).$$

De telle sorte que

$$\forall \rho \in S_I, \quad Z(\rho) = \begin{cases} Z'(\alpha) + K(\alpha) - \gamma, & (1) \\ + K(\beta) + Z'(\alpha\beta) + Z'(\beta\alpha), & (2) \\ + Z'(\beta). & (3) \end{cases}$$

□ (1) représente le coût, que nous pouvons noter $Z(\alpha)$ de l'affectation partielle α . Il est connu, puisqu'au sommet S_I , α est connue.

□ Nous allons montrer que (2) est une fonction linéaire, dont on peut calculer une borne inférieure.

Posons

$$s \in \bar{I}_1, \quad b_{s\beta}(s) = k_{s\beta}(s) + \sum_{r \in I_1} (c'_{rs} d'_{\alpha(r) \beta(s)} + c'_{sr} d'_{\beta(s) \alpha(r)}).$$

(2) peut s'écrire alors :

$$\sum_{s \in \bar{I}_1} b_{s\beta}(s) = B(\beta),$$

$$\forall \beta, \quad b = \min_{\beta} B(\beta) \leq B(\beta).$$

Chercher une borne inférieure b de (2) revient donc à chercher la permutation qui minimise une fonction linéaire, donc à résoudre un problème d'affectation

linéaire de matrice des coûts $B = (b_{i\beta(i)})$. La méthode utilisée est l'algorithme hongrois [12] (une description détaillée, algorithme et démonstration, pourra être trouvée dans *Chemins et flots, ordonnancements*, par R. Faure, C. Roucairol et P. Tolla, [5]).

Néanmoins, elle peut être interprétée comme une méthode consistant à :

(i) réduire la matrice des coûts d'affectation (ici B) en employant une réduction analogue à Réd 1.

Ainsi, si $B^{(1)} = (b'_{i\beta(i)})$ est la matrice obtenue par réduction de B , $b^{(1)}$ une constante égale à la somme des éléments soustraits de la matrice B lors de la réduction, $B(\beta)$ peut s'écrire :

$$B(\beta) = \sum_{i=1}^n b'_{i\beta(i)} + b^{(1)},$$

$$B(\beta) = B^{(1)}(\beta) + b^{(1)};$$

(ii) effectuer successivement sur la matrice $B^{(i)}$ des transformations admissibles c'est-à-dire telles que

$$B^{(i)}(\beta) = B^{(i+1)}(\beta) + b^{(i+1)},$$

où $B^{(i+1)}$ est une matrice réduite, $b^{(i+1)}$ une constante,

jusqu'à ce qu'il existe sur la matrice réduite obtenue notée B^* une affectation de coût nul :

$\exists k$ tel que $B^{(k)}$ notée B^* est telle que

$$\min_{\beta} B^*(\beta) = 0 = B^*(\beta^*),$$

$B(\beta)$ s'écrit alors, en notant $b = \sum_{i=1}^k b^{(i)}$:

$$B(\beta) = B^*(\beta) + b,$$

ce qui entraîne

$$\min_{\beta} B(\beta) = \min_{\beta} B^*(\beta) + b.$$

□ (3) est le coût de l'affectation β pour le problème d'affectation réduit. A son propos, deux points de vue s'affrontent.

Une borne inférieure peut être cherchée en utilisant la notion de « produit ordonné de deux vecteurs ».

Le *produit ordonné* de deux vecteurs (a_i) et (b_i) , $i \in I$, noté $PO_I(a, b)$, qui est par définition le produit scalaire d'un vecteur dont les composantes sont celles de (a_i)

rangées par ordre croissant, par un vecteur de composantes celles de (b_i) rangées par ordre décroissant, est tel que

$$\min_p \sum_i a_i b_{p(i)} = PO_I(a, b),$$

où p est une permutation de $I = \{1, 2, \dots, n\}$.

On a alors

$$\forall \beta, \quad Z'(\beta) = \sum_{i \in \bar{I}_1} \sum_{j \in \bar{I}_1} c'_{ij} d_{\beta(i)\beta(j)} \leq \sum_{i \in \bar{I}_1} PO_{\bar{I}_1}(c'_i, d'_{\beta(i)}).$$

Cette borne inférieure serait alors à inclure dans $B(\beta)$:

$$B(\beta) = \sum_{i \in \bar{I}_1} [b_{i\beta(i)} + PO_{\bar{I}_1}(c'_i, d'_{\beta(i)})].$$

Mais, pour éviter le calcul des produits ordonnés, on peut également supposer que si les matrices réduites C' et D' contiennent beaucoup d'éléments nuls, la valeur de $Z'(\beta)$ est négligeable devant celle de $B(\beta)$.

Borne inférieure de $Z(\rho)$

Puisque

$$Z(\rho) = Z(\alpha) + b + B^*(\beta) + Z'(\beta), \quad (1)$$

$$\forall \rho \in S_I, \quad Z(\rho) \geq Z(\alpha) + b.$$

2.3. CHOIX DE L'ÉLÉMENT A AFFECTER LORS DE LA SÉPARATION

L'affectation d'un nouvel élément permettant de séparer S_I est choisie dans B^* . Le raisonnement suivant est fait :

- si une affectation $p(s)=t$ telle que $b_{st}^*=0$ est sélectionnée, la borne inférieure de $Z(\rho)$ est inchangée;
- si cette affectation est interdite ($b_{st}^* = +\infty$), le coût des permutations correspondantes sera augmenté au moins du regret r_{st} (notion utilisée dans divers algorithmes [4]) :

$$r_{st} = \min_{\substack{j \in \bar{I}_1 \\ j \neq s}} b_{sj}^* + \min_{\substack{i \in \bar{I}_1 \\ i \neq t}} b_{it}^*.$$

L'affectation choisie sera celle qui entraînerait la plus forte augmentation de la borne inférieure, si nous ne la choisissons pas.

Pour ce faire, il suffit de calculer le regret de chaque zéro correspondant à la permutation optimale β^* (les autres sont de regret nul) et de retenir le plus fort.

Si $r_{i_*j_*} = \max_i r_{i\beta^*(i)}$, deux sommets successeurs de S_I sont créés dans l'arborescence

$$\Gamma(S_I) = \{S_{I+1}, S_{I+2}\},$$

$$S_{I+1} = \{\rho \in S_I \mid \rho(i^*) = j^*\}, \quad S_{I+2} = \{\rho \in S_I \mid \rho(i^*) \neq j^*\}.$$

2.4. FONCTION D'ÉVALUATION PAR DÉFAUT

La fonction d'évaluation associée à un sommet S de l'arborescence est

$$v : \mathcal{P}(P) \rightarrow \mathbf{R},$$

$$\forall \rho \in S, \quad v(S) \leq Z(\rho).$$

D'après (1), et la définition de S_{l+1} et S_{l+2} , nous prendrons

$$v(S_{l+1}) = Z(\alpha) + b,$$

$$v(S_{l+2}) = Z(\alpha) + b + r_{i^*j^*}.$$

2.5 FONCTION D'ÉVALUATION PAR EXCÈS

Lors de la séparation de chaque sommet, tel que S_l , nous pouvons calculer aisément une borne supérieure de $Z(\rho)$:

dès que β^* est connue, la valeur de la permutation

$$\rho^0 = (\alpha\beta^*) \text{ est : } Z(\rho^0) = Z(\alpha) + b + Z'(\beta^*),$$

$$\forall \rho \in P, \quad Z(\rho) \leq Z(\rho^0).$$

2.6. IMPLICATIONS

Les fonctions d'évaluation calculées aux deux sommets créés par séparation nécessitant un nombre d'opérations polynomial (algorithme hongrois), nous essaierons de recueillir le plus d'informations sur les permutations appartenant à un sommet avant de passer à la séparation suivante.

A partir de S_l : Implication triviale (I_1) : si $Z'(\beta^*) = 0$ alors $\rho^* = \rho^0$. La permutation optimale ρ^* étant trouvée, le problème A.Q. est alors résolu.

Dans le cas contraire, appelons $\bar{\rho}$ la meilleure permutation trouvée jusqu'alors, c'est-à-dire celle qui correspond à la valeur la plus faible prise par $Z(\rho)$.

Posons : $\Delta = Z(\bar{\rho}) - (Z(\alpha) + b)$.

Δ représente l'écart entre la borne inférieure des permutations appartenant au sommet S_l et la meilleure permutation $\bar{\rho}$;

(I_2) si $\Delta < 0$, l'exploration de S_l est abandonnée.

Nous posons dans ce cas : $v(S_l) = +\infty$;

(I_3) si $\Delta = 0$, $\rho^* = \bar{\rho}$.

Si $\Delta > 0$, deux sommets successeurs de S_l sont créés par séparation.

A partir du sommet correspondant au choix positif : la ligne i^* et la colonne j^* sont supprimées de la matrice B^* . Dans la matrice ainsi obtenue :

(I_4) si $b_{ij}^* \geq \Delta$ alors l'affectation de i à j est interdite (en effet, $\forall \rho \in S_{l+1}$ tel que $Z(\rho) < Z(\bar{\rho})$, $\rho(i) \neq j$).

Nous posons alors : $b_{ij}^* = +\infty$.

Si le nombre d'éléments infinis dans B^* est important, c'est-à-dire s'il dépasse $n-k-3$. ($n-k-1$ est la dimension de B^*), il se peut qu'il ne reste sur une ligne ou colonne de B^* :

(I₅) aucun élément fini : l'exploration de S_{l+1} est alors abandonnée.

(I₆) un seul élément fini : le choix de l'affectation correspondante est forcée;

(I₇) deux éléments finis : il n'existe alors que deux possibilités pour affecter l'élément qui sont alors énumérées.

A partir du sommet correspondant au choix négatif : $\rho(i^*) \neq j^*$.

Nous posons dans B^* :

$$b_{i^*j^*}^* = +\infty; \quad \forall j \in \bar{I}_1, \quad j \neq j^*, \quad b_{i^*j}^* = b_{i^*j}^* - \min_{j \neq i^*} b_{i^*j}^*,$$

$$\forall i \in \bar{I}_1, \quad i \neq i^*, \quad b_{ij}^* = b_{ij}^* - \min_{i \neq j^*} b_{ij}^*.$$

Comme $\forall \rho \in S_{l+2}$, $Z(\rho) \geq Z(\alpha) + b + r_{i^*j^*}$:

$$\Delta : = \Delta - r_{i^*j^*}.$$

Des implications identiques à I₄, I₅, I₆, I₇ peuvent être faites alors sur la matrice ci-dessus obtenue.

2.7. EXEMPLE

Chercher la permutation ρ de $I = \{1, 2, 3, 4\}$ qui minimise

$$Z(\rho) = \sum_i \sum_{j \neq i} c_{ij} d_{\rho(i)\rho(j)} :$$

$$C = \begin{array}{|c|c|c|c|} \hline & 8 & 8 & 12 \\ \hline 8 & & 5 & 2 \\ \hline 8 & 5 & & 11 \\ \hline 12 & 2 & 11 & \\ \hline \end{array} \quad D = \begin{array}{|c|c|c|} \hline & 1 & 4 & 3 \\ \hline 1 & & 2 & 1 \\ \hline 4 & 2 & & 3 \\ \hline 3 & 1 & 3 & \\ \hline \end{array}$$

Une réduction de type 2 a permis d'obtenir les matrices réduites C' , D' et de calculer K :

$$C' = \begin{array}{|c|c|c|c|} \hline & 0 & 0 & 2 \\ \hline 0 & & 5 & 0 \\ \hline 0 & 5 & & 9 \\ \hline 2 & 0 & 9 & \\ \hline \end{array} \quad \alpha \quad D' = \begin{array}{|c|c|c|c|} \hline & 0 & 1 & 1 \\ \hline 0 & & 0 & 0 \\ \hline 1 & 0 & & 0 \\ \hline 1 & 0 & 0 & \\ \hline \end{array} \quad \alpha' \quad K = 1 \quad \begin{array}{|c|c|c|c|} \hline & 176 & 64 & 240 & 160 \\ \hline 2 & 30 & 0 & 60 & 30 \\ \hline 3 & 48 & 0 & 96 & 48 \\ \hline 4 & 80 & 16 & 132 & 76 \\ \hline \end{array}$$

$$\beta' \quad 8 \quad 0 \quad 0 \quad 2 \quad \beta' \quad 1 \quad 0 \quad 2 \quad 1$$

$$Z(\rho) = Z'(\rho) + K(\rho) - \gamma, \quad \gamma = 10.4.2 = 80.$$

Recherchons $\min_p K(p) = K(p_k)$.

La matrice réduite K^* obtenue par l'algorithme hongrois est

	1	2	3	4
1	52	$\boxed{0}$	86	36
2	0	30	$\boxed{0}$	0
3	$\boxed{0}$	12	18	0
4	4	0	26	$\boxed{0}$

$$p_k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}, \quad K(p_k) = 248,$$

$$\forall p \in I, \quad K(p) = K^*(p) + K(p_k).$$

$$Z'(p_k) = 28,$$

A la racine de l'arborescence, que nous allons créer, correspond donc l'ensemble des permutations de $I = \{1, 2, 3, 4\}$ avec une borne inférieure de $248 - 80 = 168$, et une borne supérieure de $168 + 28 = 196$.

Puisque nous cherchons une permutation meilleure que p_k , toute affectation $\begin{pmatrix} i \\ j \end{pmatrix}$ dont le coût entraînerait un dépassement de cette borne supérieure 196 doit être interdite.

Comme $Z(p) = Z'(p) + K^*(p) + 168$, tout élément (i, j) de K^* supérieur à $196 - 168 = 28$ est donc interdit.

C'est ainsi que nous poserons

$$k_{11}^* = k_{13}^* = k_{14}^* = k_{22}^* = +\infty,$$

	1	2	3	4
1	∞	0	∞	∞
2	0	∞	0	0
3	0	12	18	0
4	4	0	26	0

Il reste alors un seul élément auquel affecter 1, l'élément 2.

Barrons la ligne 1 et la colonne 2 de K^* . Cherchons alors la meilleure permutation par la méthode hongroise sur la matrice $B : b_{ij} = k_{ij}^* + 2c'_{1i} \cdot d'_{j2}$

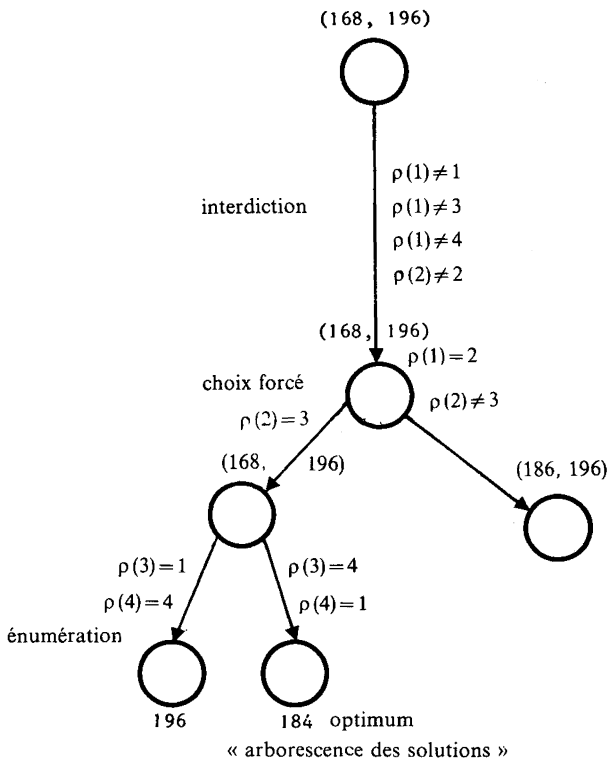
On trouve $\begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 4 \end{pmatrix}$, c'est-à-dire β^* :

$$\begin{array}{c}
 \begin{array}{cc} & \begin{array}{ccc} 1 & 3 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{array}{|ccc|} \hline 0 & 0 & 0 \\ 0 & 18 & 0 \\ 4 & 26 & 0 \\ \hline \end{array}
 \end{array}
 +
 \begin{array}{|ccc|} \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline \end{array}
 =
 \begin{array}{|ccc|} \hline 0 & \overset{18}{\boxed{0}} & 0 \\ \boxed{0} & 18 & 0 \\ 4 & 26 & \boxed{4} \\ \hline \end{array}
 \end{array}$$

Dans B , les affectations de coût nul n'ont pas toutes le même poids. Celle qu'on regretterait le plus d'abandonner est $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$, car, si on ne la retenait pas, il faudrait payer au moins :

$$r_{23} = \underbrace{\min_{j=1,3,4} b_{2j}}_{\text{(affectation de 2 à un autre élément)}} + \underbrace{\min_{i=2,3,4} b_{i3}}_{\text{(affectation d'un autre élément à 3)}} = 18.$$

Au contraire, à $\begin{pmatrix} 4 \\ 4 \end{pmatrix}$ ne correspond qu'un regret de 4, et aux autres des regrets nuls.



Séparons donc le sommet pendant de l'arborescence sur l'affectation $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$;

$Z'\begin{pmatrix} 1 \\ 2 \end{pmatrix}=0, K\begin{pmatrix} 1 \\ 2 \end{pmatrix}=0$; la borne de $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$ est donc 168, alors que celle de non $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$ devient $168 + 18 = 186$.

Repartant du sommet dont la borne inférieure est la plus faible, nous constatons qu'il n'existe plus qu'une alternative pour 3 : $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ ou $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$.

$$\begin{pmatrix} 3 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \quad Z \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix} = 196.$$

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \quad Z \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} = 184$$

ce qui constitue l'optimum.

3. Résultats

La complexité du problème A.Q. est telle qu'il nous faut séparer les exemples de petite taille (nombre d'éléments à affecter $n < 10$) de ceux de taille plus importante, $n \geq 10$).

3.1. EXEMPLES DE TAILLE $n < 10$

Nous avons évalué les performances de notre algorithme sur une quarantaine d'exemples tirés de la littérature (tableau I). Les coefficients des matrices C et D sont tirés au hasard. Le pourcentage d'éléments nuls ainsi que l'intervalle de variation des coefficients diffèrent suivant les exemples.

La méthode par réduction donne l'optimum de ces problèmes en un temps nettement plus court que par les autres méthodes (en moyenne 2 secondes sur IBM 370/168).

Légende des tableaux

N , taille du problème : nombre d'éléments à affecter.

$Z(p^*)$, coût de la solution optimale.

$I_1(I_2)$, intervalle de variation des coefficients des matrices C et D , respectivement $[0,10]$ ($[0,1000]$).

Temps, temps en seconde.

NS , nombre de sommets de l'arborescence construite.

*, cas où la phase de réduction n'a pas lieu, C et D étant déjà données sous forme réduite.

■, pas de résultat en un temps limité, donné par l'auteur.

Cycle mémoire : Univac 494 (750 ns), IBM 370/168 (80).

TABLEAU I

Exemples	N	Éléments nuls (%)	Z (p*)	Univac 494 Temps				IBM 370/168			
				Pierce et Crowston	Gilmore	Lawler	Burkard	Réduction 1		Réduction 2	
								Temps	NS	Temps	NS
Gavett Plyter	4		806					0,01	22	< 0,01	0
Personnel	5		368					0,04	40	0,03	44
Nugent	5		50	0,51	0,69	0,26	0,34	0,01	6	< 0,01	29
	6		86	3,53	5,17	2,17	1,58	0,08	62	0,08	62
	8		214	45,78	■	35,03	43,00	1,30	876	1,44	915
Burkard	6	18	70 737	2,18	7,22	1,76	1,14	0,08	45	0,06	45
	21	37	21 922	5,59	11,52	3,83	1,11	0,18	192	0,27	283
	22	63	404	1,52	6,63	2,38	0,40	0,02	21	0,02	21
	23	86	0	0,62	0,11	0,67	0,09	< 0,01	*	*	*
	24	14	112 606	14,63	■	15,20	7,44	1,49	741	1,39	751
	25	39	21 851	12,48	32,30	11,08	3,65	0,58	302	0,55	302
	26	50	1 925	12,19	16,94	10,79	2,01	0,48	*	*	*
	27	86	0	1,15	3,49	1,63	0,12	0,03	*	*	*
	28	13	112 102	99,04	■	64,72	52,85	13,73	3 569	7,30	2 502
	29	32	31 636	38,63	■	41,54	20,44	2,49	950	3,12	1 290
	30	59	1 760	60,19	136,00	40,32	11,11	2,51	*	*	*
	31	80	0	2,46	11,19	3,85	0,27	0,03	*	*	*
	32	37	25 388	■	■	■	63,61	48,36	8 310	55,84	9 253
	33	62	1 176	122,75	■	■	9,50	1,83	*	*	*
	34	76	0	6,54	11,25	■	0,33	0,02	47	0,04	10
	35	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	587	2,05	4,03	1,80	1,03	0,08	61	0,05	47
Little 41	36	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	5 409 470	1,46	3,86	0,91	1,41	0,04	642	1,27	818
	37	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	899	13,08	137,98	13,96	6,91	1,10	428	0,70	348
	38	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	8 509 308	59,81	117,64	10,72	8,94	0,80	2 605	4,09	1 949
	39	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	891	75,48	■	64,08	44,88	8,25	10,29	7,86	2 616
	40	$\left. \begin{matrix} I_1 \\ I_2 \end{matrix} \right\}$	11 902 372	■	■	■	46,34	10,29	2 772	7,86	2 616
Results	6		63	6,83	9,52	3,63	1,44	0,14	233	0,08	122

TABLEAU II

Exemples personnels $N = 10, 15, 20, 25, 30$
(coefficients tirés au hasard dans $[1, 100]$)

	Coût de la solution 3-optmale	Meilleure borne inférieure	Au bout d'un temps IBM 370/168	Au plus à $\alpha\%$ de l'optimum	Nombre de sommets de l'arborescence	Coût de la meilleure solution trouvée
$N = 10$						
Réd 1	2 349 610 2 230 066	1 334 809 1 623 586	0,56 s 3,99 s 18,47 s	76 37 0	6 336	2 230 066 optimale
Réd 2	3 349 610 2 230 066	1 324 561 1 550 244	0,57 s 2,63 s 1 mn 6 s	77 43 0	10 428	2 230 066
$N = 10$						
Réd 1	115 612 112 084	102 711 105 932	0,38 s 0,56 s 1 mn 47 s	12 5 0	14 242 14 242	112 084 optimale
Réd 2	112 990 112 932 112 084	100 850 107 133 109 965	0,58 s 7,69 s 28,30 s 1 mn 10 s	12 5 1 0	10 574 15 341	112 084
$N = 15$						
Réd 1	497 804 491 674	457 162 465 034 468 204 469 220	1,83 s 42 s 3 mn 5 mn	8 5 5 4,5	430 2 034 3 140	491 674
Réd 2	493 662	440 890 459 292 460 847	2,98 s 3 mn 5 mn	11 7 6,5	1 894 3 143	493 666
$N = 20$						
Réd 1	879 888 738 718	637 434 656 010 670 457	14,42 s 46,62 s 3 mn	27 11 9	339 1 878	738 718
Réd 2	854 176 764 516	630 451 643 196 650 127 653 399	14,37 s 60,00 s 3 mn 5 mn	26 16 15 15	362 1 124 2 029	754 516
$N = 25$						
Réd 1	1 190 116	945 580 953 598	36,56 s 3 mn	20 20	166	1 190 116
Réd 2	1 189 352	940 360 950 188 952 432	36,82 s 3 mn 5 mn	21 20 19,8	164 328	1 189 352
$N = 30$						
Réd 1	1 835 010	1 434 082 1 440 309	55,32 s 3 mn	22 21	66	1 835 010
Réd 2	1 814 406	1 431 229 1 437 604 1 441 265	54 s 3 mn 5 mn	21 20 20	62 134	1 814 406

TABLEAU III

Coût de la solution 3-optimale	Meilleure borne inférieure	Au bout d'un temps IBM 370/168	Au plus à α % de l'optimum	Nombre de sommets de l'arborescence construite	Meilleures solutions connues
Exemple Nugent, $N = 20$					
2 608	2 027 2 085	8 s 5 mn	20	1 494	<i>Hilliers et Connors</i> (1966) 2 638 – 9 mn sur GE 265 <i>Burkard</i> (1976) 2 574 – 1 mn sur CDC Cyber 72/76 <i>Personnel</i> (10 solutions 3-optimales) 2 596 – 2 mn sur IBM 370/168.
Exemple Armour-Buffer, $N = 20$ (D = matrice de « distance euclidienne »)					
48 837 820	26 221 320 30 252 040	11 s 5 mn	38	1 769	<i>Burkard</i> (1976) 47 209 360 – 3 mn sur CDC Cyber 72/76 <i>Personnel</i> (10 solutions 3-optimales) 46 768 360 – 1 mn sur IBM 370/168.
Exemple Steinberg, $N = 36$ (D = matrice de « distance euclidienne au carré »)					
8 597	3 247 3 247 3 480	84 s 5 mn 10 mn	59	474 1 028	<i>Graves et Whinston</i> (1970) 10 253 = 1 mn 30 s <i>Burkard</i> (1976) 7 926 – 21 mn sur CDC 72/76

3.2. EXEMPLES DE TAILLE $n \geq 10$

Pour les exemples de taille $n \geq 10$, le temps passé à la recherche d'un optimum croît de façon exponentielle avec n . Aussi, pour limiter la taille de l'arborescence construite, nous avons utilisé la notion de *solution 3-optimale* afin d'améliorer la borne supérieure calculée en chaque sommet de l'arborescence.

Une affectation est dite 3-optimale s'il est impossible d'obtenir une affectation de coût inférieur en réalisant une permutation quelconque de tout sous-ensemble de 3 éléments.

Ainsi, lors de la construction de l'arborescence, chaque fois qu'une solution admissible meilleure que la meilleure solution trouvée jusqu'alors apparaît, elle est rendue 3-optimale.

L'erreur relative commise en prenant cette solution comme solution du problème, est calculée comme suit :

si $Z(\rho^*)$ est le coût de la solution optimale; $Z(\rho^0)$ celui de la solution 3-optimale obtenue; $v(S)$ la plus petite borne inférieure des sommets pendants de l'arborescence

$$\frac{Z(\rho^0) - Z(\rho^*)}{Z(\rho^0)} \leq \frac{Z(\rho^0) - v(S)}{Z(\rho^0)} = \alpha.$$

Plus la construction de l'arborescence avance, plus $v(S)$ augmente et plus l'écart relatif décroît. On peut alors décider d'arrêter le déroulement de l'algorithme, lorsqu'une solution est au plus à $\alpha\%$ de l'optimum.

C'est ainsi que nous avons résolu des problèmes de taille comprise entre 10 et 15. L'optimum y est atteint en moyenne au bout de 1 minute sur IBM 370/168.

Pour des exemples de taille comprise entre 15 et 30, une ou deux solutions 3-optimales peuvent être obtenues en un temps de 5 minutes sur IBM 370/168. (tableaux II et III).

REMARQUES :

- Il faut remarquer que pour tester notre méthode, dont l'une des principales caractéristiques est la phase de réduction, nous avons choisi des exemples où cette phase pourrait avoir lieu (exemples personnels du tableau II). Lorsque les matrices C ou D sont déjà données sous forme réduites, c'est-à-dire comportent déjà au moins un zéro par ligne et colonne (exemples du tableau III), la deuxième phase de la méthode permet néanmoins d'obtenir rapidement des solutions 3-optimales et de donner une borne.

- Sur les exemples traités, Réd 2 est généralement plus efficace que Réd 1 : la borne inférieure obtenue par Réd 2 est souvent plus élevée que celle donnée par Réd 1. Mais, il paraît difficile d'après nos résultats de caractériser le type de données sur lesquelles une réduction est meilleure qu'une autre.

- A propos des exemples du tableau III, il faut souligner que :

- tirer dix solutions au hasard et les rendre 3-optimales a conduit à de bonnes solutions;

- les résultats de Burkard sont dus à l'utilisation d'une méthode en 2 phases. Dans une première phase, il applique sa méthode dite « de perturbation » (procédure arborescente) pour obtenir une solution sous-optimale (résultats non communiqués). Dans une seconde phase, il l'améliore grâce à un nombre arbitraire d'échanges aléatoires de deux, puis trois éléments...

Outre le fait que ce type de procédure puisse bien entendu s'appliquer à n'importe lequel des algorithmes de résolution proposés, et en particulier à notre

méthode, il est remarquable que la solution 3-optimale que nous obtenons rapidement soit proche de celle trouvée par Burkard au terme de la seconde phase.

— Une autre notion que celle de produit ordonné peut être utilisée pour trouver une borne inférieure de $Z'(\beta)$; Graves et Whinston [9] calcule la borne inférieure d'une classe de permutation C_α s'écrivant $\rho = \alpha\beta$ avec β extension de α sur I en construisant une fonction à valeurs réelles $G(\rho)$ telle que

$$\text{Prob} \{ Z(\rho) \leq G(\rho) \mid \rho \in C_\alpha \} \leq \gamma, \quad 0 \leq \gamma \leq 1$$

(la forme explicite de $G(\rho)$ est déduite de l'inégalité de Bienaymé Tchebicheff).

Le critère de séparation de l'arborescence construite est alors : choisir le couple d'éléments (i_h, j_h) qui minimise la valeur moyenne du coût des permutations de la classe C_{α_h} [où α_h est la permutation α à laquelle on adjoint $\begin{pmatrix} i_h \\ j_h \end{pmatrix}$].

Bien que cette notion ait surtout été testée par ses auteurs sur des exemples de problèmes de voyageur de commerce [un seul exemple d'affectation quadratique (cf. tableau III)], il serait intéressant lors de travaux ultérieurs de l'introduire dans la seconde phase de notre méthode, comme le suggère un des correcteurs, pour voir s'il permet un meilleur parcours de l'arborescence surtout lorsque la taille du problème est grande.

Conclusion

Bien que le problème d'affectation quadratique constitue la formalisation naturelle d'un grand nombre de problèmes concrets, appartenant à des domaines variés, sa complexité est telle que les méthodes jusqu'alors proposées pour le résoudre n'étaient pas d'une efficacité suffisante pour être pratiquement utilisables.

La méthode que nous avons imaginée et mise au point dite « méthode par réduction » nous paraît, au contraire, susceptible d'être employée dans la pratique et ceci pour deux raisons :

1) Nous atteignons l'optimum des problèmes de taille inférieure ou égale à 10 en un temps plus court que par les autres méthodes. D'ailleurs, la plupart des méthodes connues ne se proposent même pas l'obtention de l'optimum, en un temps raisonnable, bien que leurs auteurs ne prétendent pas qu'il soit possible de les appliquer aux problèmes de taille supérieure à 10;

2) Pour les problèmes de taille supérieure à 10, nous obtenons une « bonne solution » et calculons un majorant de l'écart entre le coût de cette solution et

celui de la solution optimale. Et, bien entendu, plus le déroulement de l'algorithme est prolongé, plus l'écart relatif décroît.

D'autre part, la notion de solution intermédiaire, dite 3-optimale, que nous avons introduite, permet d'accentuer l'aspect opérationnel de notre méthode : chaque fois qu'une solution admissible meilleure que la meilleure solution connue jusqu'alors est trouvée, elle est améliorée de telle façon qu'il soit impossible d'obtenir une affectation de coût inférieur en réalisant une permutation quelconque des affectations de tout sous-ensemble de trois éléments.

Ainsi, sur les exemples de taille 10 que nous avons traités, nous avons constaté que la seconde solution 3-optimale, obtenue en quelques secondes, était toujours la solution optimale. En pratique, il sera donc possible, dans ce cas, de s'arrêter dès l'obtention de cette seconde solution.

Enfin, pour des problèmes de taille comprise entre 10 et 30, nous avons remarqué qu'un temps de 5 mn sur IBM 370/168 était suffisant pour trouver une ou deux « bonnes solutions » 3-optimales.

BIBLIOGRAPHIE *

1. G. C. ARMOUR et E. S. BUFFA, *A Heuristic Algorithm and Simulative Approach to Relative Location of Facilities*, Management Science, vol. 9, janvier 1963, p. 294-309.
2. F. BROECKX et L. KAUFMAN, *An Algorithm for the Q.A. Problem Using Bender's Decomposition*, Papers faits, Euro II, Stockholm, novembre 1976.
3. R. E. BURKARD, *Die Störungsmethode zur Lösung quadratischer Zuordnungsprobleme*, Op. Res. Verf. 16, 1973, p. 34-108; *Quadratic assignment problems: suboptimal methods*, Summer School in Combinatorial Optimization, SOGESTA, Urbino, Italy 1978; R. E. BURKARD et J. OFFERMAN, *Entwurf von Schreibmaschintastaturen mittels quadratischer Zuordnungsprobleme*, Op. Res. 21, 1977, p. 121-132.
4. R. FAURE, *Une heuristique efficace : la notion de regret en Recherche Opérationnelle*, Polskie Toutargustwo Cybernetyczne, Referaty na III Symposium p.n. Methody Henryzy, 25, Wresnia 1976, p. 55-93.
5. R. FAURE, C. ROUCAIROL et P. TOLLA, *Chemins, flots et ordonnancements*, Recherche Opérationnelle Appliquée, t. 1, Collection Programmation, Gauthier-Villars, Paris 1976.
6. G. K. GASCHUTZ et J. H. AHRENS, *Suboptimal Algorithms for the Quadratic Assignment Problem*, Naval Research Logistics Quarterly, vol. 15, mars 1968, p. 49-62.
7. J. W. GAVETT et N. V. PLYTER, *The Optimal Assignment of Facilities to Locations by Branch and Bound*, Operations Research, vol. 14, mars-avril 1966, p. 210-232.
8. P. C. GILMORE, *Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem*, S.I.A.M. J., vol. 10, juin 1962, p. 305-313.

(*) Juillet 78

9. G. W. GRAVES et A. B. WHINSTON, *An Algorithm for the Quadratic Assignment Problem*, Integer and non linear programming, p. 473-496, North Holland Publishing Company, J. ABADIE, éd., 1970.
10. F. S. HILLIER et M. M. CONNORS, *Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities*, Management Science, vol. 13, septembre 1966, p. 42-57.
11. T. C. KOOPMANS et M. BECKMANN, *Assignment Problems and the Location of Economic Activities*, Econometrica, vol. 25, janvier 1957, p. 53-76.
12. H. W. KUHN, *The Hungarian Method for Assignment Problems*, Nav. Res. Logist. Quart., vol. 2, 1955.
13. E. L. LAWLER, *The Quadratic Assignment Problem*, Management Science, vol. 9, juillet 1963, p. 586-599.
14. H. MULLER-MERBACH, *Operations Research*, p. 313-320, Verlag Wahlen, München, 1973.
15. C. E. NUGENT, T. E. VOLLMAN et J. RUMML, *An Experimental Comparison of Techniques for the Assignment of Facilities to Locations*, Operations Research, vol. 16, janvier-février 1968, p. 150-173.
16. J. F. PIERCE et W. B. CROWSTON, *Tree Search Algorithms for Quadratic Assignment Problems*, Naval Research Logistics Quarterly, vol. 18, 1971, p. 1-36.
17. C. ROUCAIROL, *Affectation quadratique*, Thèse de 3^e cycle, Université Paris-VI, octobre 1976.

UNIVERSITÉ PAUL SABATIER
LABORATOIRE DE STATISTIQUE ET PROBABILITÉS
31077 TOULOUSE CEDEX
38, ROUTE DE NARBONNE