REVUE FRANÇAISE D'AUTOMATIQUE, INFORMATIQUE, RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

M. GONDRAN
J. L. LAURIÈRE

Un algorithme pour le problème de partitionnement

Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, tome 8, n° V1 (1974), p. 27-40

<http://www.numdam.org/item?id=RO_1974__8_1_27_0>

© AFCET, 1974, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique, informatique, recherche opérationnelle. 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/

R.A.I.R.O.
(8e année, V-1, 1974, p. 27-40)

UN ALGORITHME POUR LE PROBLEME DE PARTITIONNEMENT

par M. GONDRAN (1) et J. L. LAURIÈRE (2)

Résumé. — *On propose pour le problème de partitionnement un algorithme d'énumération implicite basé sur la réduction de la fonction économique, l'utilisation d'implications et de pénalités. Un grand choix d'heuristiques permet alors de réduire fortement l'arborescence.*

1. INTRODUCTION

Le problème de partitionnement PP

(1.1) minimiser $z = cx$

(1.2) avec $Ax = e$

(1.3) $x_j = 0$ ou 1

ou (*i*) $A = (a_{ij})$ est une $m \times n$ matrice avec $a_{ij} = 0$ ou 1,

(*ii*) e est un $m \times 1$ vecteur de 1,

(*iii*) c est un $1 \times n$ vecteur à coefficients positifs,

étant un problème très contraint ($Ax = e$), se résoud bien par énumération implicite (cf. Pierce [6], Garfinkel et Nemhauser [2]).

Pourtant ces auteurs n'ayant pas de bonne évaluation par défaut, développent de grandes arborescences. Pour réduire cette arborescence Heurgon [4], Pierce et Lasky [7] utilisent la solution continue du problème PP.

Nous présentons ici pour le problème PP une nouvelle méthode d'énumération implicite du type PSES (cf. Roy [8]) dont les principales caractéristiques sont les suivantes :

(*i*) La fonction économique est *réduite* à partir d'une combinaison linéaire des contraintes (cf. paragraphe 2). Cela permet de définir une évaluation par

(1) Service IMA.
Électricité de France. Direction des études et recherches.
(2) Université de Paris-VI, Institut de Programmation.

défaut et des coûts réduits pour chaque variable *sans avoir à résoudre un PL.* Cette évaluation est meilleure que celle donnée par Vo-khac dans [9] et presque aussi bonne que celle donnée par la programmation linéaire.

(*ii*) Les *implications* dues aux contraintes (1.2) sont utilisées systématiquement (cf. paragraphe 3). Pour cela, nous devons stocker la matrice A en lignes *et* en colonnes.

(*iii*) L'utilisation des coûts réduits permet la définition et l'utilisation de *pénalités* d'une façon analogue que dans Little, Murty, Sweeney et Karel [5] pour le problème du voyageur de commerce (cf. paragraphe 4).

(*iv*) Les *heuristiques* permettant de réduire la fonction économique et de choisir la variable à séparer se font à l'aide d'indicateurs calculés d'une manière dynamique.

(*v*) La méthode précédente s'adapte avec succès à la résolution d'un grand nombre de problèmes. Au paragraphe 5 nous l'appliquerons aux problèmes d'affectation que la matrice des coûts soit rectangulaire ou non. Au paragraphe 6 nous l'adaptons aux problèmes de partitionnement « bottleneck ». Dans [3], nous donnons un algorithme d'énumération implicite du même type pour résoudre le problème de recouvrement.

2. REDUCTION DE LA FONCTION ECONOMIQUE

Considérons la i-ième contrainte :

$$\sum_{j \in J} a_{ij} x_j = 1$$

et soit $J_i = \{j | a_{ij} = 1\}$.

Alors si $c_{j_0} = \min_{j \in J_i} c_j$, la fonction économique z peut s'écrire :

$$z = c_{j0} + \sum_{j \in J} c'_j x_j$$

avec $c'_j = c_j - a_{ij} c_{j_0} \geqslant 0$.

On fait ainsi apparaître au moins un zéro, $c'_{j_0} = 0$, dans la fonction économique.

En faisant ainsi avec chacune des m contraintes, on fera apparaître un certain nombre de zéros dans la fonction économique. On obtient alors une fonction économique de la forme :

$$z = z_0 + \sum_{j \in J} c'_j x'_j \tag{2.1}$$

avec $c'_j \geqslant 0$.

Une évaluation par défaut de PP sera donc :

$$\underline{z}(PP) = z_0.$$

Pratiquement, nous avons un choix sur l'ordre avec lequel les lignes sont utilisées. Pour cela, nous appliquerons à chaque étape les critères heuristiques suivants dans l'ordre aux lignes candidates :

Critère 1 $\quad \min_{i \in I} n_i \quad$ ou $\quad n_i = |J_i|$,

Critère 2 $\quad \min_{i \in I} \gamma_i = \text{card}\{j \in J_i | c_j = c_{j_0}\}$,

Critère 3 $\quad \max_{i \in I} \min_{j \in J_i} c_j$

Critère 4 $\quad \min_{j_0 \in J} m_{j_0} \quad$ ou $\quad m_{j_0} = \sum_{k \in I} a_{kj_0}$.

La réduction précédente est une sous-optimisation de ce que donne la programmation linéaire. En effet, si on cherche à réduire les coûts initiaux par des combinaisons linéaires des contraintes :

$$c'_j = c_j - \sum_{i \in I} u_i a_{ij} \geqslant 0 \tag{2.2}$$

de manière à obtenir la meilleure évaluation possible z_0, on est amené à résoudre le programme linéaire suivant :

$$\max \sum_{i \in I} u_i \tag{2.3}$$

$$\text{avec} \quad \sum_{i \in I} u_i a_{ij} \leqslant c_j \tag{2.4}$$

qui est le programme dual du problème *PP* en continu. L'évaluation z_0 maximale est donc la solution du problème *PP* en continu (cf. Pierce et Lasky [7]).

L'intérêt essentiel de la réduction précédente est de ne pas avoir à résoudre un *PL*.

EXEMPLE 1 (Garfinkel et Nemhauser [1], p. 315)

$$\min z = 3x_1 + 7x_2 + 5x_3 + 8x_4 + 10x_5 + 4x_6 + 6x_7 + 9x_8$$

sous les contraintes (1, 2), (3, 4, 5), (5, 6, 7), (7, 8), (2, 4, 6) où (2, 4, 6) correspond par exemple à la contrainte $x_2 + x_4 + x_6 = 1$.

On choisit donc successivement les contraintes (7, 8), (1, 2), (3, 4, 5) et (2, 4, 6). On trouve :

$$z = 17 + x_2 + 5x_5 + x_6 + 3x_8.$$

EXEMPLE 2 (Vo-khac [9], p. 10)

$$\min z = 30x_1 + 49x_2 + 51x_3 + 52x_4 + 52x_5 + 55x_6 + 67x_7 + 67x_8 + 70x_9 + 81x_{10} + 81x_{11} + 83x_{12}$$

sous les contraintes (1, 2, 3, 4, 8, 9), (1, 2, 3, 6, 7, 10), (1, 4, 5, 6, 8, 11), (2, 5, 7, 8, 9, 12), (3, 4, 9, 10, 11, 12), (5, 6, 7, 10, 11, 12).

On choisit successivement les contraintes n^os 6, 1 et 5. On trouve :

$$z = 103 + 19x_2 + x_4 + 3x_6 + 15x_7 + 37x_8 + 19x_9 + 29x_{10} + 8x_{11} + 10x_{12}.$$

EXEMPLE 3 (Vo-khac [9], p. 11)

$$\min z = x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + \ldots + 18x_{18} + 19x_{19} + 20x_{20}$$

sous les contraintes (1, 2, 5, 7, 13), (2, 7, 14, 18), (3, 9, 10, 11), (4, 6, 8, 12), (4, 9, 11, 13, 19), (5, 10, 15), (5, 16, 20), (8, 14, 15, 17).

On choisit successivement les contraintes n^os 7, 8, 5, 3 et 2. On trouve :

$$z = 22 + x_1 + 6x_6 + 7x_7 + 2x_9 + 5x_{10} + 4x_{11} + 12x_{12} + 9x_{13} + 4x_{14} + 7x_{15} + 11x_{16} + 9x_{17} + 16x_{18} + 15x_{19} + 15x_{20}.$$

EXEMPLE 4 (Garfinkel et Nemhauser [2], p. 851)

$$\min z = 18x_1 + 22x_2 + 14x_3 + 36x_4 + 17x_5 + 14x_6 + 8x_7 + 24x_8 + 14x_9 + 7x_{10}$$

sous les contraintes (3, 5), (4, 5, 6), (2, 4, 8, 9), (3, 4, 8, 10), (1, 2, 7, 9), (1, 2, 4, 6, 8).

On choisit successivement les contraintes n^os 1, 2, 3 et 6. On trouve :

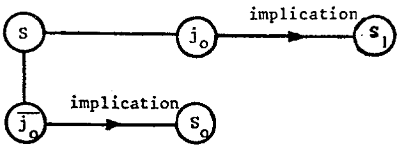
$$z = 39 + 10x_1 + 11x_4 + 3x_6 + 8x_7 + 2x_8 + 7x_{10}.$$

Si on avait remplacé le critère 3 par $\min\limits_{i \in I} \min\limits_{j \in Ji} c_j$, l'ordre du choix des contraintes aurait été n° 1, 2, 5, 6 et 3. On aurait trouvé :

$$z = 39 + 19x_4 + x_6 + 10x_8 + 2x_9 + 7x_{10}.$$

3. IMPLICATIONS

Plaçons-nous à un sommet S de l'arborescence. Soit $J(S)$ l'ensemble des variables libres. Ce sommet est séparé (cf. paragraphe 4) sur une variable x_{j0} en deux pseudo-sommets $j_0(x_{j_0} = 1)$ et $\bar{j}_0(x_{j_0} = 0)$:



Les implications se feront sur les pseudo-sommets j_0 et $\bar{j}_0$.

Soient $I_j = \{i | a_{ij} = 1\}$ et $I(S)$ l'ensemble des contraintes non saturées. Alors on définira pour toute contrainte i non saturée ($i \in I(S)$) :

$$J_i(S) = J_i \cap J(S).$$

Implications sur j_0

α) $x_{j_0} = 1$ sature les équations d'indices I_{j_0} et implique donc $x_j = 0$ pour tous les j, éléments de $\bigcup_{i \in I_{j_0}} J_i(S) \setminus \{i_0\}$.

On pose alors $J(S_1) = J(S) \setminus \bigcup_{i \in I_{j_0}} J_i(S)$ et $I(S_1) = I(S) \setminus I_{j_0}$.

β) Si $I(S_1) = \emptyset$, le sommet S_1 a une *seule* solution réalisable (que nous avons obtenue). Ce sommet est donc terminal. FIN des implications.

Sinon, considérons pour $i \in I(S_1)$ les nombres $n_i(S_1) = |J_i(S_1)|$.

Si $n_i(S_1) \geqslant 2$ pour tout $i \in I(S_1)$, FIN des implications.

S'il existe un $n_i(S_1) = 0$, le sommet S_1 n'admet pas de solution réalisable. Ce sommet est donc terminal. FIN des implications.

Si $n_i(S_1) \geqslant 1$ pour tout $i \in I(S_1)$ et $n_{i_0}(S_1) = 1$, alors $J_{i_0}(S_1) = \{k_0\}$ implique $x_{k_0} = 1$. Alors GOTO α avec $j_0 = k_0$.

Implications sur $\bar{j}_0$

$x_{j_0} = 0$ implique $J(S_0) = J(S) \setminus \{j_0\}$ et $I(S_0) = I(S)$.
GOTO β avec $S_1 = S_0$.

Les implications précédentes étant faites, il ne reste plus qu'à réduire la fonction économique sur le problème ainsi réduit (cf. paragraphe 2).

Ces implications sont très rapides car on a stocké la matrice A en lignes *et* en colonnes.

4. PRINCIPE DE SEPARATION

En chaque sommet S, on a :

$$z = \underline{z}(S) + \sum_{j \in J(S)} c'_j(S) x_j. \quad (4.1)$$

La séparation du sommet S se fera sur une variable dont le coût réduit $c'_j(S)$ est nul.

Le choix se fera alors en deux étapes : choix d'une contrainte non encore saturée, puis choix d'une variable de coût réduit nul dans cette contrainte.

Pour chaque contrainte non saturée $i \in I(S)$, définissons l'ensemble :

$$J_i^0(S) = \{ j \mid j \in J_i(S), c_j'(S) = 0 \},$$

c'est-à-dire l'ensemble des variables libres de coûts réduits nuls correspondant à la contrainte i. On a $n_i^0(S) = |J_i^0(S)| \geqslant 1$.

Nous choisirons la contrainte la plus contraignante vis-à-vis de l'augmentation de la fonction économique, c'est-à-dire celle qui minimise $n_i^0(S)$.

Pour départager les ex-æquo, considérons pour chaque contrainte i tel que $n_i^0(S)$ soit minimum, la pénalité p_i définie de la manière suivante :

$$p_i = \min_{j \in J_i(S) \setminus J_i^0(S)} c_j'(S). \tag{4.2}$$

On choisit alors la contrainte qui a la pénalité maximale. Comme troisième critère, on pourra prendre la contrainte qui minimise $n_i(S)$.

La contrainte i étant choisie, le choix de la variable à séparer se fait d'une des façons suivantes :

— Si $n_i^0(S) = 1$, $J_i^0(S) = \{ j_0 \}$. Le sommet S donne naissance au pseudo-sommet $j_0(x_{j0} = 1)$ avec pour évaluation par défaut $\underline{z}(S)$ et au pseudo-sommet $\bar{j}_0(x_{j0} = 0)$ avec pour évaluation par défaut $\underline{z}(S) + p_i$.

Alors si nous connaissons une solution réalisable x^* telle que $\underline{z}(S) + p_i > z(x^*)$, le pseudo-sommet $\bar{j}_0$ est terminal et l'implication donnant S_0 est inutile.

— Si $n_i^0(S) \geqslant 2$, plusieurs variables peuvent être candidates. Une heuristique simple est de choisir la variable qui sature le maximum de contrainte, c'est-à-dire de choisir l'indice de $J_i^0(S)$ qui maximise $m_j = |I_j|$.

Une heuristique qui paraît meilleure car elle préserve les choix futurs est de choisir le minimum de $\dfrac{C_j(S)}{m_j}$ ou $C_j(S) = \left| \bigcup_{i \in I_j} J_i^0(S) \right|$.

En effet, on tient ainsi compte du nombre de variables de coûts réduits nuls qu'on annule en fixant x_{j_0} à 1.

Le sommet S donne alors naissance à deux pseudo-sommets j_0 et $\bar{j}_0$ d'évaluation par défaut $\underline{z}(S)$.

REMARQUE 1

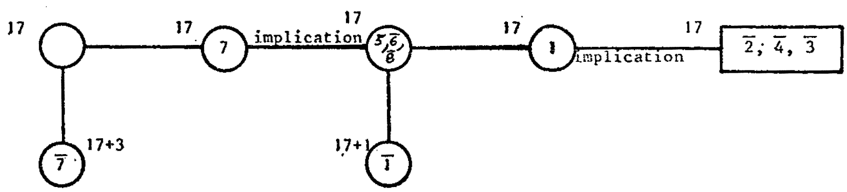
On se trouve dans le cas $n_i^0(S) \geqslant 2$ lorsque les coûts c_j ont une petite dispersion; par exemple lorsque $c_j = 1 \; \forall j \in J$. Le problème PP est dans ce cas plus difficile; l'arborescence sera donc plus importante.

REMARQUE 2

Le principe de séparation est basé sur la fonction économique réduite (4.1). Or on peut définir plusieurs réductions lorsque l'on change les critères ou même seulement l'ordre des critères du paragraphe 2, (cf. par exemple l'exemple 4 du paragraphe 2). L'utilisation de plusieurs réductions permet alors d'obtenir de meilleures pénalités et des évaluations par défaut plus précises.

Reprenons les exemples du paragraphe 2. On prendra les réductions obtenues au paragraphe 2 et on ne fera pas les implications sur $\bar{j}_0$.

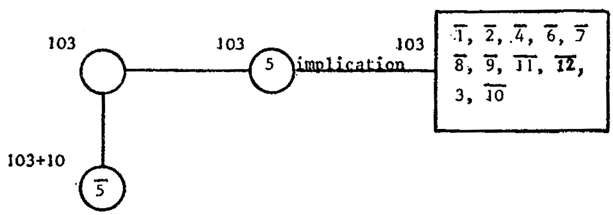
EXEMPLE 1



La solution optimale est donc :

$$x_1 = x_4 = x_7 = 1 \quad , \quad z = 17.$$

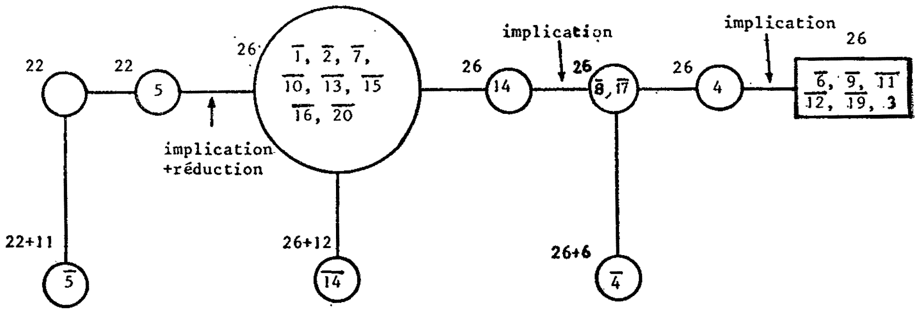
EXEMPLE 2



La solution optimale est donc :

$$x_3 = x_5 = 1 \quad , \quad z = 103.$$

EXEMPLE 3



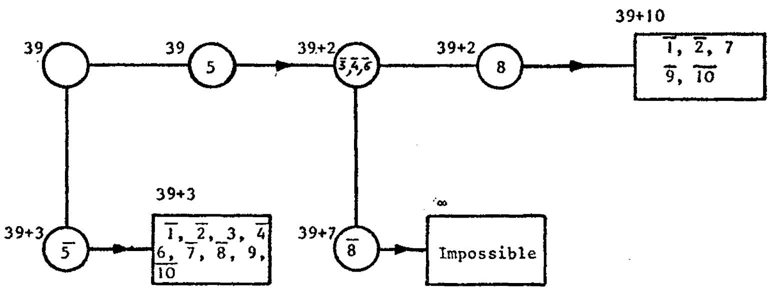
La solution optimale est donc :

$$x_3 = x_4 = x_5 = x_{14} = 1 \quad , \quad z = 26.$$

EXEMPLE 4

— Avec la fonction économique réduite :

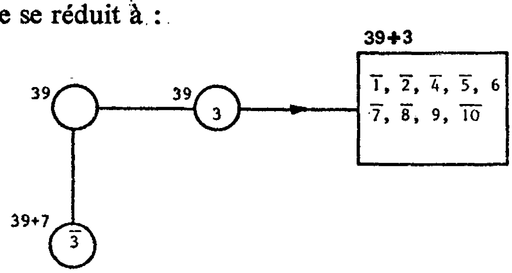
$$z = 39 + 10x_1 + 11x_4 + 3x_6 + 8x_7 + 2x_8 + 7x_{10}.$$



— Avec la fonction économique réduite :

$$z = 39 + 19x_4 + x_6 + 10x_8 + 2x_9 + 7x_{10}$$

l'arborescence se réduit à :



La solution optimale est donc :

$$x_3 = x_6 = x_9 = 1 \quad , \quad z = 42.$$

Il semble qu'il faille toujours remplacer le critère 3 du paragraphe 2 par $\min_{i \in I} \min_{j \in J_i} c_j$.

5. PROBLEMES D'AFFECTATION

Le problème d'affectation, lorsque le tableau des coûts est carré, est un problème de partitionnement particulier (la matrice A est la matrice d'incidence aux sommets d'un graphe bipartie).

On peut donc appliquer l'algorithme précédent sur la matrice des coûts presque sans modification. On prendra cependant une pénalité différente, analogue à celle de Little, Murty, Sweeney et Karel [5] pour le problème du voyageur de commerce.

EXEMPLE 5

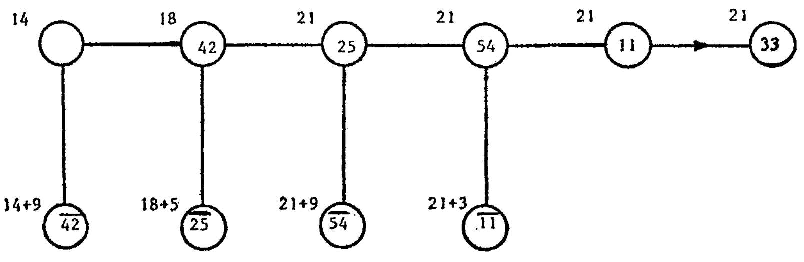
Trouver l'affectation de coût minimal pour la matrice des coûts suivants :

K_1	K_2	K_3	K_4	K_5	
7	3	5	7	10	L_1
6	∞	∞	8	7	L_2
6	5	1	5	∞	L_3
11	4	∞	11	15	L_4
∞	4	5	2	10	L_5

Pour réduire la matrice des coûts, on choisit successivement les contraintes $L_2, K_3, K_2, K_5, L_4, L_5$ et K_4. On trouve $z_0 = 14$ et la matrice des coûts réduits :

7	0	4	6	9
0	∞	∞	1	0
6	2	0	4	∞
10	0	∞	9	13
∞	0	3	0	8

L'arborescence de la résolution est alors :



d'où la solution optimale :

$$x_{11} = x_{33} = x_{54} = x_{25} = x_{42} \quad , \quad z = 21.$$

(On n'a pas représenté ici les implications qui sont implicites dans le cas des problèmes d'affectation.)

Malgré une très mauvaise évaluation de départ ($z_0 = 14$), on trouve en une descente la solution optimale.

Pour améliorer cette évaluation il semble que dans le cas des problèmes d'affectation, il faille prendre les critères suivants :

critère 1 $\min_{i \in I} \min_{j \in J_i} c_j$

critère 2 $\min_{i \in I} n_i$

critère 3 $\min_{i \in I} \gamma_i$

Dans le problème précédent, on trouve $z_0 = 21$ en choisissant successivement les contraintes K_3, L_5, K_2, L_1, L_4, K_1 et K_5.

Le problème d'affectation à tableau rectangulaire $I \times J\,(|I| < |J|)$ peut s'écrire (lorsque le tableau est plein) :

(5.1) minimiser $z = \sum_{i \in I, j \in J} c_{ij} x_{ij}$

(5.2) avec $\sum_{j \in J} x_{ij} = 1$

(5.3) $\sum_{i \in I} x_{ij} \leqslant 1$

(5.4) $x_{ij} = 0$ ou 1.

La réduction du paragraphe 2 peut donc se faire en utilisant seulement les équations (5.2), c'est-à-dire les lignes de la matrice des coûts. Il y a dans ce cas aucune ambiguïté.

Les implications α du paragraphe 3, se feront de la même manière que pour le problème *PP*.

Enfin, les pénalités se calculeront seulement sur les lignes de la matrice des coûts.

EXEMPLE 6

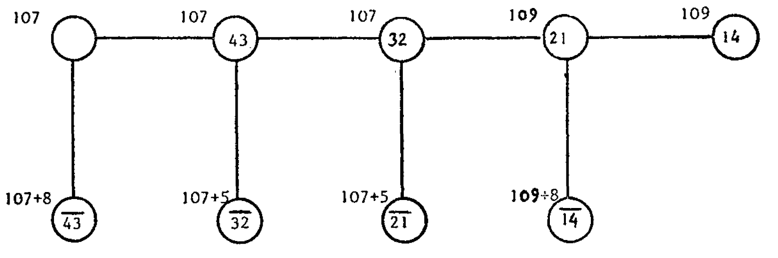
Trouver l'affectation de coût minimal pour la matrice des coûts suivants :

9	22	58	11	19	27
43	78	72	50	63	48
41	28	91	37	45	33
74	42	27	49	39	35

La réduction donne $z_0 = 107$ et la matrice des coûts réduits :

0	13	39	2	10	18
0	35	29	7	20	5
13	0	63	9	17	5
47	15	0	22	12	8

L'arborescence de la résolution est alors :



La solution optimale est donc :

$$x_{14} = x_{21} = x_{32} = x_{43} = 1 \quad ; \quad z = 109.$$

Il semblerait que l'on obtienne très souvent la solution optimale d'un problème d'affectation en une descente. La méthode précédente permet donc, comme toutes les PSES, d'obtenir une très bonne solution en une descente (cf. Vogel pour le problème de transport).

Si la solution n'est pas optimale, on a alors le choix entre l'exploration des sommets pendants non terminaux (s'ils sont peu nombreux) ou l'utilisation de l'algorithme de Ford Fulkerson en prenant cette solution comme solution de départ.

6. LE PROBLEME DE PARTITIONNEMENT « bottleneck »

Le problème de partionnement « bottleneck » ou on remplace la fonction économique $z = \sum_{j \in J} c_j x_j$ par

(6.1) $$z = \max_{j \in J} c_j x_j$$

se rencontre dans de nombreuses formalisations de problèmes (Edmonds et Fulkerson [14], Garfinkel et Nemhauser [11], Roy [8] chap. 7).

L'algorithme des paragraphes 2, 3 et 4 s'applique alors sans grands changements.

La réduction du paragraphe 2 se simplifie; en effet la fonction économique s'écrit :

(6.2) $$z = z_0 + \max_{j \in J} c'_j x_j$$

avec

(6.3) $$z_0 = \max_{i \in I} \min_{j \in J_i} c_j$$

(6.4) $$c'_j = \max(0, c_j - z_0)$$

L'évaluation par défaut est alors z_0.

Les implications et les pénalités se font alors de la même façon que pour le problème *PP*.

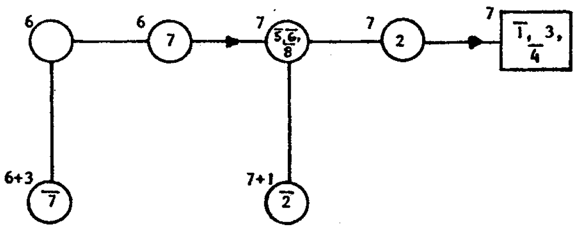
EXEMPLE 7

$$\min z = \max(3x_1, 7x_2, 5x_3, 8x_4, 10x_5, 4x_6, 6x_7, 9x_8)$$

sous les contraintes (1, 2), (3, 4, 5), (5, 6, 7), (7, 8), (2, 4, 6).

La réduction donne $z = 6 + \max(x_2, 2x_4, 4x_5, 3x_8)$.

L'arborescence de la résolution est alors :



La solution optimale est donc $x_2 = x_3 = x_7 = 1$, $z = 7$.

Comme nous l'avons fait au paragraphe 5 pour les problèmes d'affectations, nous pouvons appliquer l'algorithme précédent aux problèmes d'affectations « bottleneck ».

On ne fera aucun changement pour une matrice des coûts carrée; dans le cas d'une matrice des coûts rectangulaire on fera les mêmes changements qu'au paragraphe 5.

EXEMPLE 8 (Garfinkel [10] p. 1749).

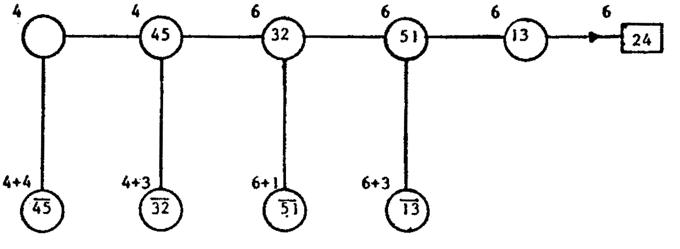
Trouver l'affectation de coût min max pour la matrice des coûts suivants :

3	6	5	9	8
6	7	4	2	8
7	3	10	14	11
9	6	5	6	2
6	3	8	7	9

La réduction donne $z_0 = 4$ et la matrice des coûts réduits.

0	2	1	5	4
2	3	0	0	4
3	0	6	10	7
5	2	1	2	0
2	0	4	3	5

L'arborescence de la résolution est alors :



La solution optimale est donc $x_{45} = x_{32} = x_{51} = x_{13} = x_{24} = 1, z = 6$.

BIBLIOGRAPHIE

[1] GARFINKEL R. S. and NEMHAUSER G. L., « Integer Programming », chap. 8, John Wiley and Sons, 1972.

[2] GARFINKEL R. S. and NEMHAUSER G. L., « The Set Partitionning Problem : Set Covering with Equality Constraints », *Opns. Res.*, 17 1969, p. 848-856.

[3] GONDRAN M., « Un algorithme pour le problème de recouvrement », note EDF, à paraître.

[4] HEURGON E., « Un problème de recouvrement : l'habillage des horaires d'une lignes d'autobus », *R.I.R.O.*, 6e année, V-1 1972, p. 13-29.

[5] LITTLE J., MURTY K., SWEENEY D. and KAREL C., « An Algorithm for the Traveling Salesman Problem », *Opns. Res.*, 11 1963, p. 979-989.

[6] PIERCE J. F., « Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems », *Man. Sci.*, 15 1968, p. 191-209.

[7] PIERCE J. F. and LASKY J. S., « Improved Combinatorial Programming Algorithms for a class of All-Zero-One Integer Programming Problems », *Man. Sci.*, 19 1973, n° 5, p. 528-543.

[8] ROY B., « Algèbre Moderne et Théorie des graphes », tome 2, chap. 10, Dunod, 1970.

[9] VO-KHAC K., « Utilisation des coûts pondérés et des variables bivalentes dans le problèmes des tournées : sectorisation sous contraintes nombreuses », *R.A.I.R.O.*, 6e année, V-2 1972, p. 3-20.

[10] GARFINKEL R. S., « An improved algorithm for the bottleneck assignment problem », *Opns. Res.*, 19 1971, p. 1747-1751.

[11] GARFINKEL R. S. and NEMHAUSER G. L., « Optimal political Districting by Implicit Enumeration Technique », *Man. Sci.*, 16 B 1970, p. 495-508.

[12] EDMONDS J. and FULKERSON D. R., « Bottleneck Extrema », *J. Comb. Theory*, 8 1970, p. 299-306.