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

M. GONDRAN
J. L. LAURIÈRE

Un algorithme pour les problèmes de recouvrement

Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, tome 9, nº V2 (1975), p. 33-51

<http://www.numdam.org/item?id=RO_1975__9_2_33_0>

R.A.I.R.O.
(9e année, juin 1975, V-2, p. 33 à 51)

UN ALGORITHME POUR LES PROBLEMES DE RECOUVREMENT (*)

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

Résumé. — *Les auteurs définissent, pour les problèmes de recouvrement, un algorithme efficace du type PSEP.*

1. INTRODUCTION

Considérons le problème de recouvrement (PR) :

$$\text{minimiser } z = cx \tag{1.1}$$

avec

$$Ax \geqslant e \tag{1.2}$$

$$x_j = 0 \text{ ou } 1$$

où :

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

e est un $m \times 1$ vecteur de 1 ;

c est un $1 \times n$ vecteur à coefficients positifs.

Le problème de partitionnement (PP) est obtenu en remplaçant (1.2) par $Ax = e$.

Ces deux problèmes sont d'une grande importance dans de nombreux problèmes combinatoires et industriels.

Citons, par exemple, le problème de Quine en logique et théorie des relais, les problèmes de coloriage dans les graphes et les hypergraphes, les problèmes de rotation des équipages dans les compagnies aériennes [1], d'habillage des horaires dans les transports urbains [2], les problèmes de découpage électoral [3], de typologie, etc...

(*) Reçu juillet 1974.

(1) Electricité de France, Direction des Etudes et Recherches, Service Informatique et Mathématiques Appliquées, Département Traitement de l'Information et Etudes Mathématiques.

(2) CNRS groupe de recherche Structure de l'information T45 Place Jussieu Paris.

Toutefois, un problème réel ne se formalisera pas exactement comme un (PR). La fonction économique ne sera pas toujours aussi simple que (1.1) et on devra ajouter un certain nombre (en général faible) de contraintes supplémentaires (cf. Roy [4] et [5]).

Une très grande variété d'algorithmes ont été donnés pour le problème de recouvrement simple (cf. Glenn Martin [6], Salkin et Koncal [7], House, Nelson et Rado [13], Bellmore et Ratliff [14], Gondran [8] et Delorme [9] pour l'utilisation de coupes ; Lemke, Salkin et Koncal [10], Pierce et Lasky [11] pour une énumération implicite utilisant la résolution d'un programme linéaire ; Thiriez [12] pour l'utilisation de la théorie des groupes).

Par contre, peu d'algorithmes ont été donnés pour le problème de recouvrement généralisé (cf. Roy [4]).

Nous présentons ici pour le (PR) simple et pour un grand nombre de (PR) généralisé un nouvel algorithme du type P.S.E.P. (cf. Roy [15]) voisin de celui que nous avions défini pour le (PP) [16].

Ses 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 défaut de la fonction économique et des coûts réduits pour chaque variable (naturelle et d'écart) *sans avoir à résoudre un programme linéaire continu.*

Pour améliorer (en probabilité) cette réduction, nous serons amenés à définir plusieurs fonctions économiques réduites.

L'évaluation ainsi obtenue est presque aussi bonne que celle donnée par la programmation linéaire.

(*ii*) Une bonne solution réalisable est recherchée dès le début de l'algorithme (cf. paragraphe 3). Pour cela, nous définissons plusieurs heuristiques pour déterminer *plusieurs solutions réalisables* ; on obtient ainsi une bonne solution réalisable (en probabilité). Cela est *très important* dans toutes les méthodes P.S.E.P. En effet, il se peut qu'une heuristique soit mauvaise pour un problème donné. Donc, si on applique plusieurs heuristiques, cela permet d'obtenir des résultats beaucoup plus homogènes.

(*iii*) L'utilisation des coûts réduits permet la définition et l'utilisation de *pénalités* (cf. paragraphe 4).

(*iv*) Les *implications* dues aux contraintes (1.2) sont utilisées systématiquement (cf. paragraphe 5). Pour cela, nous devons stocker la matrice A en lignes *et* en colonnes.

(*v*) La méthode précédente s'adapte avec succès à la résolution d'un grand nombre de problèmes.

Au paragraphe 6, on montrera qu'elle s'applique à certaines classes de problèmes de recouvrement généralisé.

2. RÉDUCTION DE LA FONCTION ÉCONOMIQUE

La i-ième contrainte du problème de recouvrement (PR) s'énonce :

$$\sum_{i \in J} a_{ij} x_j = 1 + s_i$$

avec :

$$s_i \in \mathbf{N} .$$

Pour un i donné, soit $J_i = \{ j \mid a_{ij} = 1 \}$.

Alors, si

$$c_i^0 = \min_{j \in J_i} c_i \text{ et } c'_j = c_j - a_{ij} c_i^0 \; (j \in J) ,$$

la fonction économique peut s'écrire :

$$z = \sum_{j \in J} c_j x_j = c_i^0 \left(\sum_{j \in J} a_{ij} x_j \right) + \sum_{j \in J} c'_j x_j$$

ou bien :

$$z = c_i^0 + c_i^0 s_i + \sum_{j \in J} c'_j x_j .$$

En procédant à cette « réduction » des coûts, successivement, pour chacune des contraintes, on obtient une fonction économique de la forme :

$$\boxed{z = z_0 + \sum_{i \in I} c_i^0 s_i + \sum_{j \in J} c'_j x_j} \qquad (2.1)$$

où z_0 est la somme des quantités retranchées et où :

$$\forall i \in I \quad c_i^0 \geqslant 0, \quad \forall j \in J \quad c'_j \geqslant 0 .$$

Une évaluation par défaut de (PR) sera donc :

$$\boxed{\underline{z}(PR) = z_0} .$$

Pratiquement, on a le choix de l'ordre dans lequel on considère les contraintes. Comme dans le problème de partitionnement [16] cet ordre sera déduit des quatre critères heuristiques suivants :

- *Critère 1* :

$$\min_{i \in I} n_i \quad \text{avec} \quad n_i = | J_i |$$

puis en cas d'égalité sur un sous-ensemble I' de contraintes $(I' \subset I)$:

- *Critère 2* :

$$\min_{i \in I'} \min_{j \in J_i} c_j$$

en cas d'égalité sur $I'' \subset I'$:

- *Critère 3* :

$$\min_{i \in I''} |\{ j \in J_i \mid c_j = c_i^0 \}|$$

en cas d'égalité sur $I''' \subset I''$:

- *Critère 4* :

$$\max_{\substack{j_0 \in J_i \\ i \in I'''}} \sum_{k \in I} a_{kj_0} .$$

Comme dans le cas du problème de partitionnement, la réduction précédente fournit une borne inférieure de z moins bonne que la programmation linéaire en continu. En effet, la meilleure évaluation par défaut, obtenue par combinaison linéaire des contraintes et réduction de la fonction objectif est la valeur de y à l'optimum du programme :

$$\begin{cases} \max y = \sum_{i \in I} u_i & \\ \sum_{i \in I} u_i \, a_{ij} \leqslant c_j & (\text{car } c_j' \geqslant 0) \\ u_i \geqslant 0 & (\text{car } c_i^0 \geqslant 0) \end{cases}$$

qui est le dual du problème (PR) en continu.

EXEMPLE 1

Garfinkel et Nemhauser [17], p. 310.

						Matrice A
i \ j	1	2	3	4	5	n_i
1	1	0	1	1	0	3
2	0	1	0	1	0	2
3	1	0	0	0	1	2
4	0	0	1	0	1	2
c_j	7	3	7	12	6	
						c_i^0
Réductions L_2		0		9		3
Réductions L_3	1				0	6
Réductions L_1	0		6	8		1
c_j'	0	0	6	8	0	$z_0 = 10$

Figure 1.

— Réductions successives :

$n_i = 2$ (critère 1) : L_2, L_3, L_4 on prend d'abord L_2 correspondant à

$$\min_i \min_j c_j = 3 ;$$

$n_i = 3$ mène à considérer L_1. D'où :

$$Z = 10 + 3\, s_2 + 6\, s_3 + s_1 + 6\, x_3 + 8\, x_4 .$$

EXEMPLE 2

Lemke, Salkin, Spielberg [10], p. 1012.

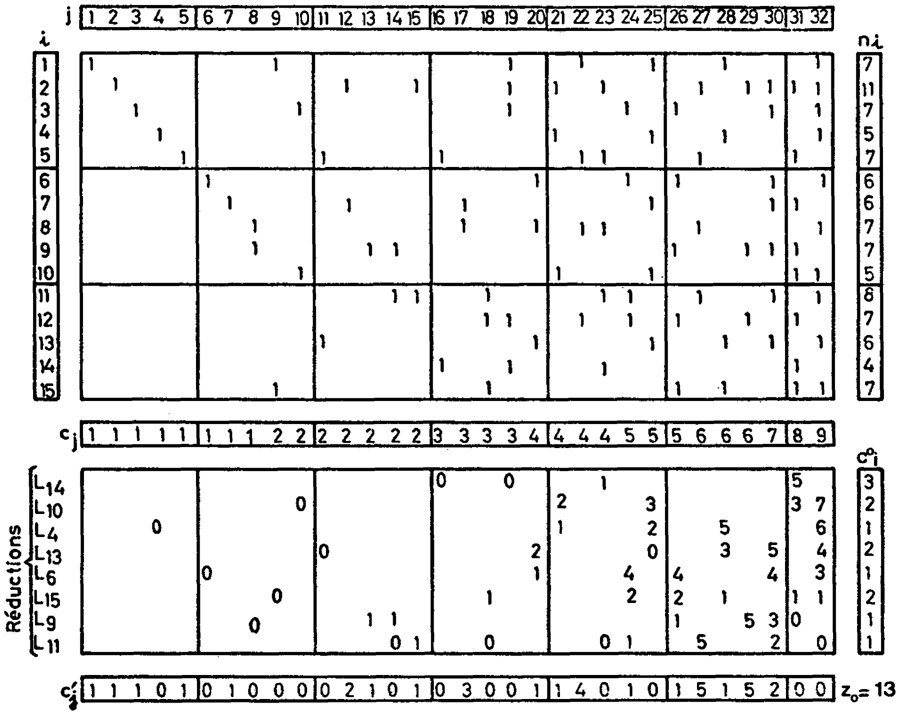

Figure 2.

Les réductions successives sont faites en considérant les contraintes dans l'ordre des n_i croissants :

$n_i = 4$ L_{14}

$n_i = 5$ L_4, L_{10} ; choix L_{10} (critère 2), puis L_4

$n_i = 6$ L_6, L_7, L_{13}, puis L_6

$n_i = 7$ L_1, L_3, L_5, L_8, L_9, L_{15} ; choix L_{15}, L_9

$n_i = 8$ L_{11}

$n_i = 11$ L_2

D'où :

$$\begin{aligned} z = 13 &+ 3\,s_{14} + 2\,s_{10} + s_4 + 2\,s_{13} + s_6 + 2\,s_{15} + s_9 + s_{11} \\ &+ x_1 + x_2 + x_3 + x_5 + x_7 + 2\,x_{12} + x_{13} + x_{15} \\ &+ 3\,x_{17} + x_{20} + x_{21} + 4\,x_{22} + x_{24} + x_{26} \\ &+ 5\,x_{27} + x_{28} + 5\,x_{29} + 2\,x_{30}\,. \end{aligned}$$

3. SOLUTIONS INITIALES

3.1. Heuristique H_1

Contrairement au cas du problème de partitionnement (PP), on est sûr, à ce stade, qu'il existe une solution constituée par un sous-ensemble des variables « naturelles » x de coûts réduits nuls, i.e. que le problème (PR') :

$$(PR') \begin{cases} \sum_{j \in J^0} a_{ij} x_j = 1 + s_i & (i \in I) \\ \text{où } J^0 = \{ j \mid c'_j = 0 \} \end{cases}$$

admet une solution.

L'heuristique H_1 suivante permet de construire une telle solution :

H_1

Considérant les contraintes non satisfaites suivant l'ordre donné par :

$$\min_i | \{ j \mid j \in J^0, a_{ij} = 1 \} | ,$$

pour chaque contrainte i_0, on fixe à 1 la variable x_{j_0}, $j_0 \in J_{i_0}$, dont le coût initial est le plus faible :

$$j_0 \in J^0 \cap J_{i_0} \qquad \text{et} \qquad c_{j_0} = \min_{J^0 \cap J_{i_0}} c_j \Rightarrow x_{j_0} := 1 .$$

Soit $\bar{z}$ la valeur de la fonction économique associée à cette solution X_0, il est clair que l'optimum Z^* de (PR) satisfait aux inéquations :

$$z_0 \leqslant Z^* \leqslant \bar{z} . \tag{3.1}$$

De plus, s'il existe une solution avec

$$\forall_i, s_i \times c_i^0 = 0, \qquad \forall j, x_j \times c'_j = 0$$

cette solution est la solution optimale de (PR).

Par ailleurs des solutions initiales réalisables peuvent être construites d'entrée de jeu sur le problème (PR), *avant* les réductions du paragraphe 2.

Deux heuristiques sont maintenant présentées dans ce but. Chacune d'elles, comme H_1 elle-même, peut être utilisée :

— soit en considérant à chaque pas, toutes les contraintes non saturées ; l'ensemble de ces contraintes est noté I^0 ;

— soit en ne retenant à chaque pas, parmi les contraintes de I^0 que celles pour lesquelles $n_i = \sum_j a_{ij}$ est minimal.

Dans les deux cas, on ne s'intéresse bien sûr qu'aux seules variables qui permettent de saturer au moins une contrainte de I_0.

3.2. Heuristique H_2

Cette heuristique est basée sur la définition du coût moyen d'une contrainte, soit :

$$\frac{\vec{a}_i . \vec{c}}{|\vec{a}_i|} = \frac{\sum_j a_{ij} c_j}{\sum_j a_{ij}} .$$

H_2 Faire $x_{j_0} := 1$ pour l'indice j_0 qui rend maximale la quantité :

$$\frac{1}{c_j} \times \sum_{i \in I^0 \cap I_j} \frac{\vec{a}_i . \vec{c}}{|\vec{a}_i|} \qquad \text{avec} \qquad I_j = \{ i \mid a_{ij} = 1 \}$$

et cela tant qu'il reste une contrainte non satisfaite.

3.3. Heuristique H_3

Une autre idée est de prendre 1 comme coût moyen :

Il y a $m = |I|$ contraintes à satisfaire, donc la somme de tous les 1 dans les colonnes d'une solution réalisable quelconque doit valoir au moins m :

$$\sum_j \left(\sum_i a_{ij} \right) x_j \geqslant m \tag{3.2}$$

et l'on doit minimiser :

$$\sum_j c_j x_j . \tag{3.3}$$

On est donc conduit ainsi à un problème de sac à dos [11] dont on donne une solution approchée (H_3 est en fait une heuristique pour le problème (3.2) (3.3)).

H_3 Faire $x_j^0 := 1$ pour l'indice j_0 qui rend la quantité $\frac{c_j}{|I^0 \cap I_j|}$ minimale, avec c_j maximal en cas d'égalité et cela tant qu'il reste une contrainte non satisfaite.

H_3 permet de plus de calculer une nouvelle borne inférieure z'_0 de z :

$$z'_0 = \sum_{j+1}^{k} c_j + \left\lceil \left(m - \sum_{j=1}^{k} |I_j| \right) \times \frac{c_{k+1}}{|I_{k+1}|} \right\rceil^{(1)}$$

où k est l'indice tel que :

$$\sum_{j=1}^{k} |I_j| \leqslant m \qquad \sum_{j=1}^{k+1} |I_j| > \mathrm{m} .$$

(1) Partie entière supérieure.

Six heuristiques sont ainsi définies qui pourront être toutes mises en œuvre au début d'un problème de dimensions importantes.

En outre, diverses compositions de ces heuristiques entre elles sont possibles.

Les efficacités respectives sur un problème donné dépendront alors de la redondance de celui-ci.

L'évaluation par excès $\bar{z}$ de l'optimum de (PR) sera bien entendu la plus faible des valeurs prises par z sur l'ensemble des solutions réalisables obtenues par ces heuristiques et

$$\underline{z} \leqslant z \leqslant \bar{z} \tag{3.4}$$

avec de même $\underline{z} = \max(z_0, z'_0)$.

3.4. Premières implications

En utilisant (2.1), il est clair que les variables dont le coût réduit est supérieur à la différence $\bar{z} - z_0$ ne sauraient appartenir à la solution optimale et peuvent être définitivement éliminées :

$$c'_j > \bar{z} - z_0 \Rightarrow x_j = 0 \text{ à l'optimum}$$
$$c'_i > \bar{z} - z_0 \Rightarrow s_i = 0 \text{ à l'optimum.}$$

4. PRINCIPE DE SÉPARATION

Nous obtiendrons la solution optimale de (PR) par une procédure arborescente fixant progressivement les valeurs des s_i et x_j.

Après un certain nombre de choix associés à un sommet S de l'arborescence de résolution, définissons les ensembles :

$$K^0(S) = \{ j/x_j = 0 \}$$
$$K^1(S) = \{ j/x_j = 1 \}, \qquad K(S) = K^0 \cup K^1$$

est l'ensemble des variables naturelles déjà fixées ;

$$J(S) = J - K(S)$$

est l'ensemble des variables libres.

Pour chaque contrainte i, posons :

$$b_i = \sum_{j \in K(S)} a_{ij} x_j \qquad \text{et} \qquad s'_i = s_i - b_i .$$

L'équation (2.1) s'écrit alors :

$$z = \left[z_0 + \sum_{j \in K^1(S)} c'_j x_j + \sum_{i \in I} c_i^0 b_i \right] + \sum_{j \in J(S)} c'_j x_j + \sum_{i \in I} c_i^0 s'_i$$

ce qui donne pour évaluation par défaut du sommet S :

$$\underline{z}(S) = z_0 + \sum_{j \in K^1(S)} c'_j x_j + \sum_{i \in I - I(S)} c_i^0 b_i$$

où $I(S)$ est l'ensemble des contraintes non saturées :

$$I(S) = \{ i/b_i = 0 \} .$$

La séparation du sommet S pourra ensuite se faire sur les variables naturelles $x_j(j \in J(S))$ ou bien sur des variables d'écart $s_i(i \in \bar{I}(S))$.

Pour cela, nous définissons une pénalité pour chaque x_j et une pour chaque s_i ; S sera séparé suivant la variable de plus forte pénalité :

— Pour chaque x_j tel que $j \in J(S)$, $c'_j = 0$, on considère les lignes i non saturées telles que $a_{ij} = 1$; on définit alors :

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

et

$$p_j(i) = \min_{k \in J_i(S) - \{j\}} c'_k .$$

La pénalité de x_j sera alors :

$$\boxed{p_j = \max_{\substack{i \in I(S) \\ a_{ij} = 1}} p_j(i) .}$$

— Pour chaque variable s_i la pénalité sera :

$$p_i = c_i^0 .$$

• Alors, si la pénalité maximale correspond à une variable naturelle x_j, le sommet S donnera naissance :

au sommet ($x_j = 1$) d'évaluation $\underline{z}(S)$

et

au sommet ($x_j = 0$) d'évaluation $\underline{z}(S) + p_j$.

• Si elle correspond à une variable d'écart s_i, S sera partagé en :

un sommet ($s_i = b_i - 1$) d'évaluation $\underline{z}(S)$

et en

un sommet ($s_i \geqslant b_i$) d'évaluation $\underline{z}(S) + p_i$.

REMARQUE 1

Les choix $s_i = b_i - 1$ transforment progressivement le problème de recouvrement initial en un problème de type partitionnement (b_i non nécessairement nuls comme dans le problème pur) plus contraint, donc plus facile à résoudre.

REMARQUE 2

En cas d'égalité pour les pénalités maximales et de choix entre une variable d'écart et une variable naturelle *on préférera la variable d'écart* car elle entraîne des implications plus fortes. Si le choix concerne des variables de même espèce, on retiendra celle qui est liée à la contrainte i, $i \in I(S)$ correspondant à :

$$\min_i \mid J_i(S) \mid .$$

Si on a encore le choix entre deux variables naturelles, on prendra celle qui est liée à :

$$\max_j \mid \{ i, a_{ij} = 1 \} \mid .$$

Avant de séparer à nouveau les sommets de l'arborescence, on prendra soin de réduire (paragraphe 2) le problème de recouvrement obtenu dans les cas où :

$$\exists i, i \in I(S) \qquad \text{tel que} \qquad \min_{j \in J_i(S)} c'_j$$

est non nul, et de tenir compte des implications liées au choix effectué.

5. IMPLICATIONS

Elles concernent : soit la valeur de la fonction économique (5.1), soit la satisfaction des contraintes (5.2).

5.1. Si $\bar{z}$ est la meilleure valeur connue pour une solution réalisable, la recherche d'une solution associée à un sommet S telle que $z(S) < \bar{z}$ entraîne l'élimination des variables telles que :

$$c'_k \geqslant \bar{z} - z_0(S)$$

ce qui permet de réduire progressivement la dimension du problème.

5.2. Les implications seront déduites des contraintes i, $i \notin I(S)$. Les implications proviennent de deux situations différentes :

— *Cas* 1

La valeur de la variable d'écart s_i associée a été précédemment fixée : $s_i = b_i - 1$ avec $b_i \geqslant 1$ connu. Si en outre :

$$\mid J_i \cap K^1(S) \mid = b_i ,$$

alors :

$$i \in J_i \cap J(S) \Rightarrow x_j = 0 .$$

Le cas $b_i = 1$ redonne par exemple les implications du problème de partitionnement.

— *Cas* 2

s_i doit simplement obéir à $s_i \geqslant b_i$. Alors :

a) si

$$| J_i \cap (K^1(S) \cup J(S)) | < b_i$$

le sous-problème correspondant au sommet S n'admet pas de solution (FIN des implications) ;

b) si

$$| J_i \cap (K^1(S) \cup J(S)) | = b_i$$

alors $\qquad \forall j, j \in J_i \cap J(S) \Rightarrow x_j = 1 .$

Quand les implications et les réductions précédentes sont faites sur chaque nouveau sommet S, on sépare le sommet dont l'évaluation $Z(S)$ est la plus faible.

Il est possible de reconstruire périodiquement une solution approchée $X_0(S)$ pour n'importe quel sommet pendant. On a choisi dans les calculs manuels de ne faire ce travail que pour certains sommets.

Ces sommets sont tels que :

$$I(S) \neq o$$
$$J(S) \neq o$$

et le dernier choix fait contredit, pour une variable au moins, la valeur retenue dans la dernière solution réalisable $X_0(S)$ calculée.

Exemple 1

L'ensemble des variables de coûts réduits nuls donne la solution X_0 :

$$x_2 = x_5 = x_1 = 1 \qquad \text{avec} \qquad \bar{z} = 16 ,$$

d'où l'implication par (3.4) : $x_4 = 0$ à l'optimum puisque $c'_4 = 8 > 16 - 10$ ce qui entraîne $x_2 = 1$.

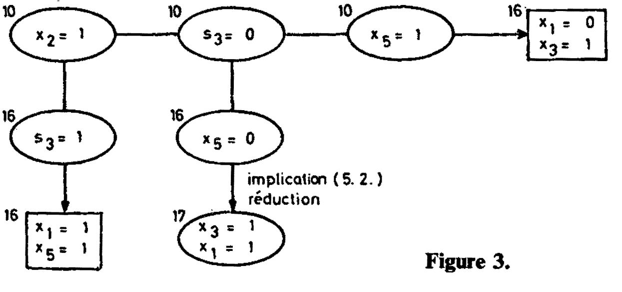

Figure 3.

D'où deux solutions optimales $x_1 = x_2 = x_5 = 1$ et $x_2 = x_3 = x_5 = 1$ avec $Z = 16$.

EXEMPLE 2

Solution initiale X_0 :

$$x_{25} = x_6 = x_{18} = x_{10} = x_8 = x_{19} = x_{11} = 1 \quad \text{et} \quad \bar{z} = 15 .$$

L'implication (3.4) entraîne donc l'élimination définitive des variables 12, 17, 22, 27, 29 et 30.

De même, on doit avoir pour toute solution meilleure que X_0 :

$$s_{10} = s_{13} = s_{14} = s_{15} = 0$$

puisque ces variables ont une pénalité de 2.

Toutes les variables naturelles restantes ont alors une pénalité nulle.

On arbitrera donc en premier lieu s_4, s_6, s_9, s_{11} de pénalités égales à 1.

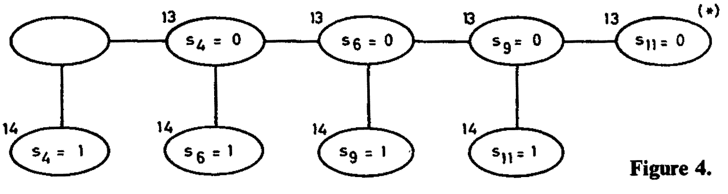

Figure 4.

De plus, toute solution de coût au plus 14 satisfait :

$$s_4 + s_6 + s_9 + s_{11} \leqslant 1 .$$

On se placera alors au sommet (*) en considérant successivement les lignes par n_i croissant (la première est L_7). L'arborescence finale est représentée par la figure de la page suivante, où, sans ambiguïté, pour les variables naturelles,

$$j \text{ signifie } x_j = 1 \quad \text{et} \quad \bar{j} \text{ signifie } x_j = 0 ;$$

et où les sommets sont numérotés en chiffres romains dans l'ordre dans lequel ils sont étudiés.

Les croix correspondent aux sommets éliminés par 5.2*a*).

Les flèches aux implications 5.1 ou 5.2.

On obtient donc ainsi une première solution optimale :

$$(\mathrm{X_{II}}) \quad \boxed{x_{25} = x_{19} = x_{18} = x_8 = x_6 = x_5 = 1} \quad \text{avec} \quad z^* = 14 .$$

Si l'on se contente d'une solution optimale, FIN.

Sinon, le sommet V, impliquant par L_2, $x_{23} = 1$ d'où $\overline{14}$, $\overline{18}$, $\overline{9}$ puis, $x_{26} = 1$ par L_3, procure la seule solution :

$$(\mathrm{X_V}) \quad \boxed{x_{25} = x_{26} = x_{23} = 1}$$

Le sommet VI, à son tour, donne la solution heuristique :

$$(X_{VI}) \qquad \boxed{x_7 = x_4 = x_6 = x_{10} = x_{11} = x_{19} = x_8 = x_{18} = 1}$$

avec $\bar{z} = 14$ qui est donc solution optimale et c'est la seule que l'on génère à partir de ce sommet ; c'était la solution obtenue dans [10].

Il reste pour épuiser toutes les possibilités à examiner les quatre sommets de type ($s_i = 1$). L'ensemble des inconnues est alors réduit à J^0 puisque $\underline{z} = 14$.

Les sommets sont éliminés (5.2*a*) à tour de rôle, d'où en fin de compte *trois* solutions optimales pour ce problème.

REMARQUE 3

L'heuristique H_3 permet ici d'obtenir directement les solutions (X_{II}) et/ou (X_{VI}) suivant que les contraintes sont ou non considérées dans l'ordre des n_i. On aurait pu enfin tout aussi bien définir un principe de séparation et des pénalités basés sur cette heuristique et les quantités $c_j / | I^0 \cap I_j |$.

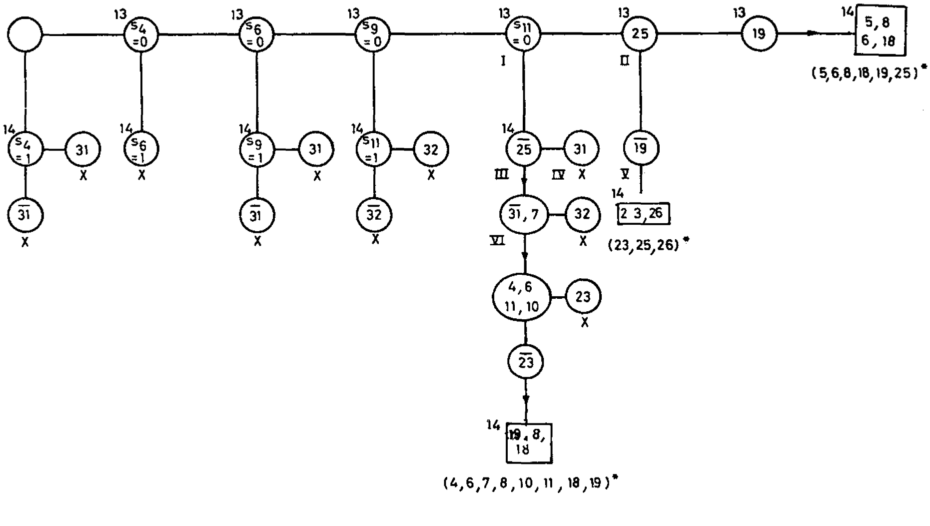

Figure 5.

6. PROBLÈMES DE RECOUVREMENT GÉNÉRALISÉ

La généralisation peut se faire sur la fonction économique et sur les contraintes.

Considérons d'abord le cas des contraintes.

L'adjonction des contraintes aura deux effets :

— accélération des implications,

— mais une moins bonne évaluation par défaut.

Un exemple classique est le problème du voyageur de commerce. Pour ce problème, on peut définir deux ensembles de contraintes ; les contraintes d'affectation et les contraintes de connexité (cf. Dantzig, Fulkerson et Johnson [18], Held et Karp [19]). La réduction de la fonction économique par Little and all. [20] revient à n'utiliser que les contraintes d'affectation. C'est ce que nous proposons pour les problèmes de recouvrement généralisé.

Etudions maintenant plus en détail le cas d'une fonction économique généralisée (cf. Roy [14]).

Cas 1

$$\min z \text{ où } z = \max_{j \in J} c_j x_j . \tag{6.1}$$

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

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

avec :

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

$$c'_j = \max (0, c_j - z_0) . \tag{6.4}$$

On a alors trivialement :

Proposition 1

Une solution optimale d'un (PR) avec (6.1) comme fonction économique est obtenue en donnant la valeur 1 à l'ensemble des variables de coût réduit nul.

EXEMPLE 3

$z = \max (3 x_1, 7 x_2, 5 x_3, 8 x_4, 10 x_5, 4 x_6, 6 x_7, 9 x_8)$ sous les contraintes (1, 2), (3, 4, 5), (5, 6, 7), (7, 8) et (2, 4, 6) où (2, 4, 6) par exemple correspond à la contrainte $x_2 + x_4 + x_6 \geqslant 1$.

La réduction donne :

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

d'où la solution optimale : $x_1 = x_3 = x_6 = x_7 = 1$.

Dans le cas où il existe d'autres contraintes, les implications et les pénalités se font de la même façon que pour le (PR) simple.

EXEMPLE 4

Même problème que l'exemple 3 avec comme contraintes supplémentaires $x_1 . x_6 = 0$ et $x_2 . x_8 = 0$. Une solution réalisable sera X_0 :

$$x_1 = x_7 = x_4 = 1, \qquad \bar{z} = 8 .$$

Alors par (3.4), on a les implications $x_5 = x_8 = 0$. Puis (5.2*b*) entraîne $x_7 = 1$.

L'arborescence est alors la suivante :

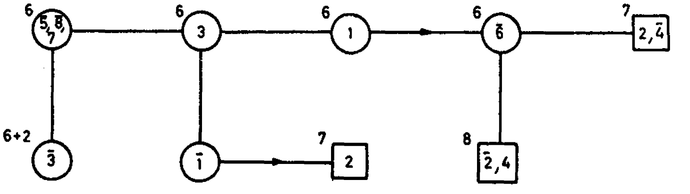

Figure 6.

d'où deux solutions optimales $x_2 = x_3 = x_7 = 1$ et $x_2 = x_1 = x_3 = x_7 = 1$ avec $z = 7$.

Cas 2

$$z = \min_{\substack{j \in J \\ x_j = 1}} c_j . \tag{6.5}$$

Le problème n'a alors que peu d'intérêt puisqu'il suffit de prendre $x_j = 1$ pour une des variables de coût réduit

$$\left(c''_j = c_j - \min_{i \in J} c_i \right)$$

nul et n'importe quoi pour les autres variables.

Cas 3

$$z = \max_{j \in J} c_j x_j - \min_{\substack{j \in J \\ x_j = 1}} c_j . \tag{6.6}$$

On transforme l'équation (6.6) de la façon suivante :

Soit $C = \max_{j \in J} c_j$, alors :

$$C + (c_j - C) x_j = \begin{cases} C & \text{si} \quad x_j = 0 \\ c_j & \text{si} \quad x_j = 1 \end{cases}$$

donc si le vecteur x n'est pas identiquement nul, on a :

$$\min_{\substack{j \in J \\ x_j = 1}} c_j = \min_{j \in J} \left[C + (c_j - C) x_j \right] .$$

Finalement pour un x non nul, et en posant $d_j = C - c_j$, (6.6) s'écrit :

$$z = \max_{j \in J} c_j x_j + \max_{j \in J} d_j x_j - C . \tag{6.7}$$

Alors en utilisant deux fois la réduction du cas 1, on trouve :

$$z = z_0 + \max_{j \in J} c'_j x_j + \max_{j \in J} d'_j x_j \tag{6.8}$$

avec :

$$\begin{aligned} z_0 &= z_c + z_d - C \\ z_c &= \max_{i \in I} \min_{j \in J_i} c_j \\ z_d &= \max_{i \in I} \min_{j \in J_i} d_j \\ c'_j &= \max (0, c_j - z_c) \geqslant 0 \\ d'_j &= \max (0, d_j - z_d) \geqslant 0 . \end{aligned}$$

On obtient ainsi une première évaluation par défaut z_0.

Posons $F_j = c'_j + d'_j$ pour tout $j \in J$ et considérons la contrainte i. Puisque cette contrainte doit être vérifiée, on doit choisir un x_j non nul dans cette contrainte, donc z_0 sera pénalisé d'au moins la quantité :

$$P_i = \min_{j \in J_i} F_j .$$

Si on pose alors $P = \max_{i \in I} P_i$, on a la proposition suivante :

Proposition 2

$$z_1 = z_0 + P \text{ est une évaluation par défaut de } z . \tag{6.9}$$

Pour tout $j \in J$, posons maintenant

$$g_j = \max (c'_j, d'_j) ,$$

puis $Q_i = \min_{j \in J_i} g_j$.

On a alors la proposition suivante :

Proposition 3

Pour tout couple (i_1, i_2) tel que :

$$\begin{aligned} J_{i_1} \cap J_{i_2} &= \emptyset , \\ z_2 &= z_0 + Q_{i_1} + Q_{i_2} \end{aligned} \tag{6.10}$$

est une évaluation par défaut de z.

Pratiquement, on utilisera la proposition 3 en prenant pour i_1 la contrainte qui maximise Q_i, soit

$$Q_{i_1} = Q = \max_{i \in I} Q_i ,$$

et en prenant i_2 tel que :

$$Q_{i_2} = \max_{\substack{i \\ J_i \cap J_{i_1} = \emptyset}} Q_i .$$

Posons

$$c''_j = \max(0, c'_j - Q) \quad \text{et} \quad d''_j = \max(0, d'_j - Q),$$

on a alors :

Proposition 4

On obtient une solution réalisable d'un *PR* avec (6.6) en faisant $x_j = 1$ pour l'ensemble des variables tels que $c''_j = d''_j = 0$. La fonction économique de cette solution est alors :

$$\hat{z} = z_0 + 2\,Q . \tag{6.11}$$

La séparation se fait d'une manière un peu différente à celle du paragraphe 4. On choisit la ligne i_0 que maximise P_i ($P_{i_0} = P$) et on sépare la variable x_{j_0} telle que $P_{i_0} = f_{j_0}$.

EXEMPLE 5

Mêmes données que pour l'exemple 3 mais avec z de la forme (6.6).

La réduction (6.8) donne :

$$z = -1 + \max(x_2, 2\,x_4, 4\,x_5, 3\,x_8) + \max(4\,x_1, 2\,x_3, 3\,x_6, x_7) .$$

La proposition 2 donne $z_1 = -1 + 2 = 1$ et la proposition 3 donne

$$z_2 = -1 + 2 + 1 = 2 .$$

La proposition 4 donne la solution réalisable :

$$\hat{x}_2 = \hat{x}_3 = \hat{x}_4 = \hat{x}_7 = 1 \quad \text{et} \quad \hat{z} = 3 .$$

L'arborescence est alors la suivante :

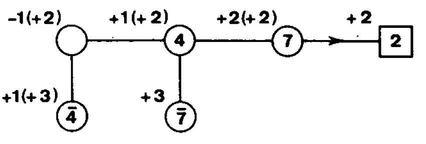

Figure 7.

Les termes entre parenthèses sont les évaluations par défaut z_2.

La solution optimale est donc $x_2 = x_4 = x_7 = 1$ avec $z = 2$.

Cas 4

Un très grand nombre d'autres fonctions économiques peuvent être choisies (cf. Roy [4] et Auzet [5]) ; par exemple :

$$z = \frac{\sum_{j \in J} c_j x_j}{\sum_{j \in J} x_j} \tag{6.12}$$

$$z = \sum_{j \in J} c_j x_j + \sum_{i \in I} \min_{\substack{j \\ a_{ij} x_j = 1}} \Delta_{ij} . \tag{6.13}$$

L'algorithme décrit aux paragraphes précédents ne s'applique plus dans ces cas (la fonction économique (6.12) n'est même pas monotone par rapport aux variables et la fonction (6.13) l'est très rarement : cas où $c_j \geqslant \sum_i \Delta_{ij}$).

BIBLIOGRAPHIE

[1] AGARD J., ARABEYRE J. P., VAUTIER J., Génération automatique des rotations d'équipages, *R.I.R.O.*, n° 6, 1967.

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

[3] GARFINKEL R. S. and NEMHAUSER G. L., Optimal Political Distriucting by Implicit Enumeration Techniques, *Management Science*, 16, B 495-B 508.

[4] ROY B., An algorithm for General Constrained Set Covering Problem, in *Graph Theory and Computing*, Academic Press Inc., p. 267-283.

[5] AUZET C., Un modèle de recouvrement sous contraintes : synthèse des principales applications possibles, Direction Scientifique de *METRA*, note de travail n° 184, janvier 1973.

[6] MARTIN G., An accelerated Euclidean Algorithm for Integer Linear Programming, in *Recent Advances in Mathematical Programming* (Graves and Wolfe, ed.), McGraw-Hill, New York, 1963.

[7] SALKIN H. M. and KONCAL R. D., Set Covering by an All Integer Algorithm : Computational Experience, *Journal of the Association for Computing Machinery*, 1973, vol. 20, n° 2, p. 189-193.

[8] GONDRAN M., Un algorithme de coupes efficace par la méthode des congruences décroissantes, note EDF, HI 1234/02 du 11 juillet 1973.

[9] DELORME J., Thèse de troisième cycle, à paraître.

[10] LEMKE C., SALKIN H. and SPIELBERG K., Set Covering by single branch enumeration with linear programmiňg subproblems, *Oper. Res.*, 19 (1971), p. 998-1022.

[11] PIERCE J. F. and LASKY, All-zero-one Integer Programming Problems, *Management Science* 19, n° 5 (1973), p. 528-543.

[12] THIRIEZ H., Airline Crew Scheduling : a group theoric Approach, 1969, Rep. R-67, Flight Transportation Laboratory, Massachusetts Institute of Technology.

[13] HOUSE R. W., NELSON L. D. and RADO J., Computer Studies of a Certain Class of Linear Integer Problems, in *Recent Advances in Optimization Techniques* (Lavi and Vogl ed.), John Wiley and Sons, 1966.

[14] BELLMORE M. and RATLIFF H. D., Set Covering and Involutory Bases, *Management Science* 18 (1971), p. 194-206.

[15] ROY B., Algèbre linéaire et théorie des graphes, Tome 2, Chap. 10, Dunod, 1970.

[16] GONDRAN M. et LAURIÈRE J. L., Un algorithme pour le problème de partitionnement, *R.A.I.R.O.*, n° V-1, 1974, p. 25-38.

[17] GARFINKEL R. S. and NEMHAUSER G. L., Integer Programming, chapitre 8, John Wiley and Sons, 1972.

[18] DANTZIG G. B., FULKERSON D. R. and JOHNSON S. M., Solution of a large Scale Travelling Salesman Problem, *Opns. Res.* 2 (1954), p. 393-410.

[19] HELD M. and KARP R. M., The traveling-Salesman Problem and Minimum Spanning Trees, *Opns. Res. 18* (1970), p. 1138-1162.

[20] LITTLE J., MURTY K., SWEENEY D. and KAREL C., An algorithm for the Traveling Salesman Problem, *Opns. Res. 11* (1963), p. 979-989.

[21] HAMMER P. L., Booleau procedures for bivalent programming, in Mathematical programming theory and applications, North Holland, 1974.

[22] BALAS E. et PADBERG, M. W., On the set covering problem II. *Opns. Res.* 1974.