REVUE FRANÇAISE D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. SÉRIE VERTE

V. GINSBURGH
A. VAN PEETERSSEN

Un algorithme de programmation quadratique en variables binaires

Revue française d'informatique et de recherche opérationnelle. Série verte, tome 3, n° V2 (1969), p. 57-73

<http://www.numdam.org/item?id=RO_1969__3_2_57_0>

R.I.R.O.
(3e année, no V-2, 1969, p. 57-74)

UN ALGORITHME DE PROGRAMMATION QUADRATIQUE EN VARIABLES BINAIRES (1)

par V. Ginsburgh et A. Van Peeterssen (2)

Résumé. — *L'algorithme présenté généralise celui de Balas au cas quadratique, moyennant certaines hypothèses restrictives sur les coefficients. Néanmoins, l'algorithme s'applique à une classe intéressante de problèmes dans lesquels les termes quadratiques (ou d'ordre plus élevé) sont assimilables à des intersections d'ensembles. C'est le cas, par exemple, des « duplications » de lecture dans les problèmes d'optimisation d'audience publicitaire.*

L'expérimentation de l'algorithme donne des résultats satisfaisants bien que le temps calcul augmente rapidement avec le nombre de variables. Ce temps calcul peut être sensiblement réduit si l'on se borne à chercher une solution admissible : on constate, en effet, que la première solution admissible trouvée est toujours très proche de la solution optimale, ce qui est le cas de la plupart des algorithmes « séparation et évaluation progressive » ou « branch and bound ». On peut généraliser l'algorithme aux cas de degré supérieur à deux. Cette généralisation alourdit fortement les calculs, et réduit la portée pratique. Les applications peuvent cependant dans la plupart des cas être ramenées à du quadratique.

1. INTRODUCTION

L'algorithme présenté ici est une généralisation de l'algorithme additif de Balas (3) au cas où les contraintes et la fonction d'objectif contiennent des termes quadratiques.

La section 2 donne la formulation du problème. Dans la section 3 seront présentées les hypothèses permettant l'utilisation de l'algorithme développé dans la quatrième section. Dans la section 5, nous envisagerons la généralisation de l'algorithme au cas plus que quadratique.

---

(1) Nous sommes reconnaissants au Pr J. Waelbroeck, Université Libre de Bruxelles et au Pr E. Balas, Carnegie Mellon University, pour leurs suggestions et leurs encouragements ; la responsabilité des erreurs reste nôtre.

(2) Département d'Économie Appliquée de l'Université Libre de Bruxelles.

(3) E. Balas, *Programmation bivalente par énumération implicite*, Thèse de doctorat, Bruxelles, 1967.

E. Balas, « An additive algorithm for solving linear programs with 0 — 1 variables » in *Operations Research*, vol. 13, 1965, pp. 517-546.

La sixième section présente quelques applications qu'il est possible de traiter par cet algorithme ; à la section 7 on trouvera quelques exemples de résolution.

2. FORMULATION DU PROBLEME

Le problème peut être formulé de la manière suivante : trouver le vecteur $x = (x_1, x_2 \ldots x_N)$ tel que :

$$\min \sum_{j=1}^{N} a_j^M x_j - 1/2 \sum_{j=1}^{N} \sum_{j'=1}^{N} a_{jj'}^M x_j x_{j'}$$

$$j \text{ et } j' = 1, 2 \ldots N \tag{2.1}$$

soumis aux contraintes

$$\sum_{j=1}^{N} a_j^i x_j - 1/2 \sum_{j=1}^{N} \sum_{j'=1}^{N} a_{jj'}^i x_j x_{j'} \geqslant b_i \qquad i = 1, 2 \ldots (M-1) \tag{2.2}$$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier} \tag{2.3}$$

Cette formulation permet d'engendrer toute une série de cas particuliers, notamment celui de la linéarité de la fonction d'objectif.

On appellera solution, tout vecteur $x = (x_1 \ldots x_N)$ satisfaisant (2.3). Une solution satisfaisant (2.3) et (2.2) sera appelée solution admissible. Enfin une solution admissible rendant minimum (2.1) sera appelée solution optimale.

On appellera pseudo-solution tout ensemble de variables Ψ_s auxquelles on a donné une valeur 0 ou 1 :

$$\Psi_s = \{ x_{j1} = \delta_{j1} \; ; \; x_{j2} = \delta_{j2} \; ; \; \ldots \; x_{jq} = \delta_{jq} \} \tag{2.4}$$

où δ_{jk} est le δ de Kronecker ($k = 1, 2 \ldots q$ et $1 \leqslant q \leqslant N$).

Notons par $\bar{N}_s$ cet ensemble d'indices et par N_s l'ensemble d'indices des variables « libres » (c'est-à-dire celles auxquelles on n'a donné aucune valeur ou encore celles qui ne figurent pas dans Ψ_s). On a $N = \bar{N}_s \cup N_s$.

L'ensemble $\bar{N}_s$ est constitué par deux sous-ensembles J_s^1 et J_s^0 tels que :

$$J_s^1 = \{ j \in N \mid \delta_j = 1 \} \tag{2.5}$$

$$J_s^0 = \{ j \in N \mid \delta_j = 0 \} \tag{2.6}$$

3. HYPOTHESES SUR LES COEFFICIENTS

H. 1 : Sans perte de généralité, on peut considérer que les matrices $A^i = \| a_{jj'}^i \|$ pour $i = 1, 2 \ldots M$ sont symétriques. (3.1)

H. 2 : Les termes des matrices $A^i = \| a_{jj'}^i \|$ pour $i = 1, 2 \ldots M$ sont positifs ou nuls. (3.2)

H. 3 : Les diagonales principales des matrices $A^i = \|a^i_{jj'}\|$ pour $i = 1, 2 \ldots M$ sont vides. Cette hypothèse n'est absolument pas restrictive : en effet, étant donné que $0 \leqslant x_j \leqslant 1$, x_j entier, les termes diagonaux a^i_{jj} peuvent être traités en même temps que les a^i_j.

Il suffit de poser

$$a^i_{j*} = a^i_j - a^i_{jj} \tag{3.3}$$

H. 4 : $a^i_j - 1/2 \sum_{j'=1}^{N} (a^i_{jj'} + a^i_{j'j}) \geqslant 0$ pour $i = 1, 2 \ldots M$ et $j, j' = 1, 2 \ldots N$ (3.4)

Cette hypothèse H. 4 semble très restrictive à première vue. Une classe intéressante de problèmes relève cependant de cette structure. En effet, supposons que le coefficient a_j affectant la variable x_j représente la valeur des éléments d'un ensemble. Les coefficients $1/2(a^i_{jj'} + a^i_{j'j})$ représentent alors la valeur de l'intersection de deux ensembles j et j'. A condition qu'il n'y ait pas d'intersections d'ordre plus élevé que 2, alors (3.4) est la valeur de l'ensemble j diminué de toutes les intersections possibles avec les autres ensembles. Il est clair que (3.4) est dès lors positif ou nul. Comme on le verra dans la section 6, certaines applications importantes peuvent être décrites en ces termes.

Les hypothèses (3.1) à (3.4) confèrent au problème une propriété intéressante de monotonie non décroissante. En effet, supposons

$$x_j = 1 \qquad \text{pour} \qquad j \in S_1 \text{ et } S_1 = \{ j = 1, 2 \ldots S_1 \}$$

Ceci donne pour un membre gauche d'une contrainte quelconque (2.2) ainsi que pour la fonction d'objectif (2.3) :

$$Q_1 = \sum_{j \in S_1} a^i_j - 1/2 \sum_{j, j' \in S_1} (a^i_{jj'} + a^i_{j'j}) \tag{3.5}$$

Ajoutons un élément k à cet ensemble ($k \notin S_1$; $x_k = 1$) et formons ainsi l'ensemble $S_2 = S_1 \cup \{ k \}$. Il vient :

$$Q_2 = Q_1 + a^i_k - 1/2 \sum_{j' \in S_1} (a^i_{kj'} + a^i_{j'k}) \tag{3.6}$$

D'après l'hypothèse (3.4), il est clair que l'on aura :

$$Q_2 \geqslant Q_1 \tag{3.7}$$

Ceci signifie que le passage d'une variable x_j de l'ensemble N_s à l'ensemble J^1_s ne peut jamais réduire la valeur du membre gauche d'une contrainte (2.2) ni la valeur de la fonction d'objectif (2.3).

La formulation est cependant insuffisante pour tenir compte des intersections de plus de deux ensembles. Nous verrons dans la section 5 comment y remédier.

4. ALGORITHME DE RESOLUTION DU CAS QUADRATIQUE

L'algorithme repose sur une modification de l'algorithme de Balas dans lequel on commence par une solution non-admissible

$$x_j = 0 \ (j = 1, 2 \ldots N)$$

et une valeur infinie de la fonction d'objectif $z^* = \infty$.

On introduit peu à peu les variables qui réduisent au mieux la violation des contraintes. Dès qu'une solution admissible est découverte, on calcule la fonction d'objectif et on procède à des échanges de variables basiques en essayant de trouver une meilleure solution.

Les variables sont normalement sorties de ou introduites dans J^1 ou J^0 une à une (étape de retour ou étape de génération régulière). L'algorithme fournit également un test permettant d'introduire dans J^1 ou J^0 plusieurs variables à la fois (étape de saut).

Dans le cas où des formes bilinéaires apparaissent dans les contraintes et la fonction d'objectif, l'introduction dans J^1 d'une variable x_j entraîne nécessairement certaines des formes bilinéaires dont le nombre est égal au nombre des variables $x_k \in J^1$ lors de l'étape précédente. Ainsi si $x_j = 1$ pour $j = 1, 2, 3$ et 5, l'introduction de $x_8 = 1$ entraîne $x_j x_8$ pour $j = 1, 2, 3$ et 5.

L'algorithme de Balas est applicable, dans ses grandes lignes, à condition d'associer à chaque nouvelle variable introduite dans J^1 ou J^0 non seulement le coefficient a_j, mais également les coefficients

$$1/2(a^i_{jj'} + a^i_{j'j}),\ j \in J^1$$

affectés du signe négatif, soit :

$$d_{j'} = a^i_{j'} - 1/2 \sum_{j \in J^1} (a^i_{jj'} + a^i_{j'j})$$

On forme ainsi une variable auxiliaire d_j, qui ne peut prendre que des valeurs positives ou nulles en vertu de l'hypothèse (3.4).

On commence par donner à z la valeur $z^* = \infty$ pour $J^1_0 = \varnothing$ et $J^0_0 = \varnothing$. On suppose qu'après s étapes on ait atteint la pseudo-solution Ψ_s à laquelle un certain nombre de tests seront appliqués.

Test n° 1 : La solution est-elle admissible ? On calcule :

$$\hat{b}_i = b_i - \sum_{j \in J^1_s} a^i_j + 1/2 \sum_{jj' \in J^1_s} (a^i_{jj'} + a^i_{j'j}) \qquad i = 1, 2 \ldots (M-1) \quad (4.1)$$

si $\hat{b}_i \leqslant 0$ pour tout $i \in (M-1)$, la solution est admissible. On calcule :

$$z^* = \sum_{j \in J^1_s} a^M_j - 1/2 \sum_{jj' \in J^1_s} (a^M_{jj'} + a^M_{j'j}) \quad (4.2)$$

et l'on passe à l'étape de retour en remplaçant le z^* précédent par la valeur trouvée en (4.2). Il est en effet impossible de trouver une solution qui se rapproche davantage de l'optimum et qui contienne une variable $x_k \in N_s$ de façon que $J_{s+1}^1 = J_s^1 \cup \{ x_k \}$; en effet, en vertu de (3.7), la valeur de la fonction d'objectif ne peut décroître.

Si $\hat{b}_i > 0$ pour un i quelconque, on passe au test n° 2.

Test n° 2 : Examen des coefficients de la fonction d'objectif :

Désignons par z^* la valeur minimale qu'a atteint jusqu'à ce stade s la fonction d'objectif.

Il est clair que tout x_k ($k \in N_s$) tel que

$$a_k^M - 1/2 \sum_{j \in J_s^1} (a_{kj}^M + a_{jk}^M) + \sum_{j \in J_s^1} a_j^M - 1/2 \sum_{jj' \in J_s^1} (a_{jj'}^M + a_{j'j}^M) > z^*$$

ne peut être introduit dans l'ensemble J_s^1 sans augmenter la valeur de la fonction d'objectif. Dès lors, on ne peut que lui assigner la valeur nulle en l'introduisant dans l'ensemble J_s^0.

On testera

$$\hat{a}_k^M = a_k^M + \sum_{j \in J_s^1} a_j^M - 1/2 \sum_{j \in J_s^1} (a_{kj}^M + a_{jk}^M) - \sum_{jj' \in J_s^1} (a_{jj'}^M + a_{j'j}^M) - z^* \qquad \text{pour } K \in N_s \quad (4.3)$$

et on introduira dans J_s^0 tout k tel que $\hat{a}_k^M > 0$.

Si après cela $N_s = \varnothing$, on passe à l'étape de retour : on n'a plus de variables libres. Si $N_s \neq \varnothing$, on passe au test n° 3.

Test n° 3 : le test n° 2 montre que $N_s \neq \varnothing$; il y a donc encore des variables auxquelles on n'a assigné aucune valeur. Le test n° 3 vérifie s'il y a encore suffisamment de variables libres pour satisfaire chacune des contraintes pour lesquelles $\hat{b}_i > 0$.

De façon formelle on calcule :

$$\bar{b}_i = \hat{b}_i - \sum_{j' \in N_s} \left[a_{j'}^i - 1/2 \sum_{\epsilon J_s^1} (a_{jj'}^i + a_{j'j}^i) - 1/2 \sum_{j \in N_s} (a_{jj'}^i + a_{j'j}^i) \right] \qquad \text{pour tout } \hat{b}_i > 0. \quad (4.4)$$

Si $\bar{b}_i > 0$ pour un $\hat{b}_i > 0$, on passe à l'étape de retour ; il ne reste pas suffisamment de variables libres pour satisfaire les contraintes : on va chercher un autre ensemble d'indices.

Si $\bar{b}_i \leqslant 0$ pour tout $\hat{b}_i > 0$, on passe au test n° 4.

Test n° 4 : le test n° 3 montre que les contraintes peuvent toutes être satisfaites. Le test n° 4 va déterminer s'il y a des variables qui doivent nécessairement être introduites dans J^1 pour satisfaire les contraintes.

Si l'on n'en trouve pas, on engendre une nouvelle solution de façon régulière. S'il y en a, on passe au test n° 5.

Mathématiquement, l'on part des considérations suivantes : d'après (4.4) on peut écrire :

$$\hat{b}_i - \bar{b}_i = \sum_{j' \in N_s} \left[a^i_{j'} - 1/2 \sum_{j \in J^1_s} (a^i_{jj'} + a^i_{j'j}) - 1/2 \sum_{j \in N_s} (a^i_{jj'} + a^i_{j'j}) \right]$$

$$= \sum_{j' \in N_s | j' \neq k} \left[a^i_{j'} - 1/2 \sum_{j \in J^1_s} (a^i_{jj'} + a^i_{j'j}) - 1/2 \sum_{j \in N_s} (a^i_{jj'} + a^i_{j'j}) \right]$$

$$+ a^i_k - 1/2 \sum_{j \in J^1_s} (a^i_{jk} + a^i_{kj}) - 1/2 \sum_{j \in N_s} (a^i_{jk} + a^i_{kj}) \quad (4.5)$$

où $k \in N_s$

Si pour la variable $x_k \in N_s$ on peut établir la relation

$$- a^i_k + 1/2 \sum_{j \in J^1_s} (a^i_{jk} + a^i_{kj}) + 1/2 \sum_{j \in N_s} (a^i_{jk} + a^i_{kj}) < \bar{b}_i \quad (4.6)$$

il vient, en ajoutant (4.6) à (4.5)

$$\hat{b}_i > \sum_{j' \in N_s | j' \neq k} \left[a^i_{j'} - 1/2 \sum_{j \in J^1_s} (a^i_{jj'} + a^i_{j'j}) - 1/2 \sum_{j \in N_s} (a^i_{jj'} + a^i_{j'j}) \right] \quad (4.7)$$

(4.7) indique clairement que si $k \notin J^1_s$, la contrainte i n'est pas satisfaite.

Donc il faut que $k \in J^1_s$ pour que cette contrainte soit satisfaite.

Il suffit dès lors de considérer l'ensemble d'indices F_s tel que :

$$F_s = \bigcup_{i \in M^*} \left\{ j' \in N_s \mid - a^i_{j'} + 1/2 \sum_{j \in J^1_s} (a^i_{jj'} + a^i_{j'j}) \right.$$

$$\left. + 1/2 \sum_{j \in N_s} (a^i_{j'} + a^i_{j'}) < \bar{b}_i \right\} \quad (4.8)$$

avec $M^* = \{ i \in (M - 1) \mid \hat{b}_i > 0 \}$

Si $F_s = \varnothing$, on constate qu'on ne peut déterminer aucune variable indispensable et l'on passe à l'étape de génération régulière.

Si $F_s \neq \varnothing$, on passe au test n° 5.

Test n° 5 : le test précédent montre que certaines variables libres sont indispensables à la satisfaction de certaines contraintes. Ce test-ci examine si l'on se rapproche ou non de l'optimum en incluant ces variables dans J^1.

Si l'optimum est amélioré, ces variables sont introduites dans J^1, sinon, il faut faire un autre choix d'indices.

On calculera :

$$\gamma = \sum_{k \in F_s \cup J_s^1} a_k^M - 1/2 \sum_{kk' \in F_s \cup J_s^1} (a_{kk'}^M + a_{k'k}^M) - z^* \qquad (4.9)$$

Si $\gamma > 0$, on ne peut améliorer la solution et l'on cherche un autre choix d'indices en passant à l'étape de retour.

Si $\gamma \leqslant 0$, l'introduction des indices F_s dans J_s^1 est intéressante : on passe à l'étape de saut.

Étape de génération régulière : la variable $x_{j'} \in N_s$ à introduire dans J^1 est celle qui réduit au mieux la somme des violations des différentes contraintes $i \in M^*$; ceci signifie que l'on calcule

$$V_{j'}^s = \sum_{i \mid \hat{b}_i - a_{j'}^i + 1/2 \sum_{j \in J_s^1} (a_{jj'}^i + a_{j'j}^i) > 0} (\hat{b}_i - a_{j'}^i + 1/2 \sum_{j \in J_s^1} (a_{jj'}^i + a_{j'j}^i)) \qquad (4.10)$$

et l'on choisit la variable j'_* telle que

$$V_{j'_*}^s = \min V_{j'}^s \qquad (4.11)$$

Si j'_* n'est pas unique, on choisit celui pour lequel $a_{j'}^M$ est minimum.

On construit alors une nouvelle solution définie par :

$$J_{s+1}^1 = J_s^1 \cup \{ j'_* \} \qquad \text{et} \qquad J_{s+1}^0 = J_s^0 \qquad (4.12)$$

et l'on retourne au test n° 1 (nouvelle itération).

Étape de saut : on construit une nouvelle solution définie par :

$$J_{s+1}^1 = J_s^1 \cup F_s \qquad \text{et} \qquad J_{s+1}^0 = J_s^0 \qquad (4.13)$$

et l'on retourne au test n° 1 (nouvelle itération).

Étape de retour : on retourne à la solution Ψ_r à partir de laquelle la solution Ψ_s a été engendrée. Si $s = 0$, soit la solution associée au z^* minimum est optimale, soit il n'y a pas de solution admissible ($z^* = \infty$).

Si Ψ_s a été engendré régulièrement, on introduit dans J^0 l'indice j défini par $\{ j \} = J_s^1 - J_r^1$ et on retourne au test n° 2 (le test n° 1 est inutile à recalculer : les résultats ont déjà été obtenus par ailleurs).

Si Ψ_s a été obtenu par une étape de saut, on recommence l'étape de retour ; l'étape de saut implique que l'on a trouvé des variables qui doivent nécessairement appartenir à J_s^1 ; il est donc exclu de trouver une autre solution dans laquelle ne figureraient pas obligatoirement les variables imposées par l'étape de saut.

Étant donné les restrictions imposées aux coefficients, le théorème de convergeance est exactement le même que celui démontré par Balas pour le cas linéaire.

5. GENERALISATION DE L'ALGORITHME

Dans cette section, nous envisageons une généralisation de l'algorithme présenté à la section 4 au cas où l'on doit tenir compte des intersections de plus de deux ensembles. Nous adopterons la simplification de notation suivante, en posant

$$T_l^i = \sum_{j_1=1}^{N} \sum_{j_2=2}^{N} \sum_{j_3=3}^{N} \cdots \sum_{j_1=l}^{N} a^i_{j_1 j_2 j_3 \ldots j_2} x_{j_1} x_{j_2} x_{j_3} \cdots x_{j_l} \tag{5.1}$$

avec $j_1 < j_2 < j_3 \ldots < j_l$

Ceci nous permet de noter par exemple pour les intersections de 4 ensembles 3 à 3 :

$$T_3 = a_{123} x_1 x_2 x_3 + a_{124} x_1 x_2 x_4 + a_{134} x_1 x_3 x_4 + a_{234} x_2 x_3 x_4$$

On constatera que le nombre de termes de T_l^i est égal à $\binom{N}{l}$ si N est le nombre de variables.

Il est clair aussi que l'on aura

$$T_1^i \geqslant T_2^i \geqslant T_3^i \ldots \geqslant T_{l-1}^i \geqslant T_l^i \geqslant T_{l+1}^i \ldots \geqslant T_N^i \tag{5.2}$$

en vertu des propriétés des intersections des ensembles : la partie commune (ou intersection) à $(l-1)$ ensembles est au moins égale à la partie commune à l'ensemble parmi lequel figurent les $(l-1)$ précédents.

5.1. Formulation du problème

Il faut trouver le vecteur $x = [x_1 \ldots x_N]$ tel que :

$$\min T_1^M - T_2^M + T_3^M - \ldots (-1)^{N-1} T_N^M \tag{5.3}$$

soumis aux contraintes :

$$T_1^i - T_2^i + T_3^i - \ldots (-1)^{N-1} T_N^i \geqslant b^i \qquad i = 1, 2 \ldots (M-1) \tag{5.4}$$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier} \tag{5.5}$$

5.2. Hypothèses sur les coefficients

H. 1 : Tous les coefficients sont positifs ou nuls (5.6)

$$\text{H.2 :} \quad a^i_{j_1} - \sum_{j_1 j_2} a^i_{j_1 j_2} + \sum_{j_1 j_2 j_3} a^i_{j_1 j_2 j_3} - \ldots (-1)^{N-1} \sum_{j_1 \ldots j_N} a^i_{j_1 j_2 \ldots j_n} \geqslant 0 \tag{5.7}$$

Ceci n'est que la généralisation de l'hypothèse (3.4) et les implications en sont les mêmes que celles tirées en (3.7).

5.3. Algorithme de résolution du cas général

Cas A : les intersections de plus de 2 ensembles sont rares.

Dans ce cas, il sera sans doute plus intéressant d'utiliser l'algorithme tel qu'il a été développé plus haut pour le cas quadratique : introduction systématique dans tous les tests et calculs des intersections d'ordre trois et plus.

Cas B : les intersections de plus de 2 ensembles sont nombreuses.

L'utilisation de l'algorithme complet risque d'allonger fortement le temps de calcul. Aussi vaut-il mieux perdre légèrement en efficacité ce que l'on gagne en temps calcul.

Seule une série de simulations permettra de dire quelle méthode est plus rapide. L'algorithme « court » présenté ici est basé sur les propriétés (5.2) et (5.7) de monotonie croissante : l'introduction d'une variable dans l'ensemble J^1 ne peut pas détériorer une contrainte et ne peut pas faire diminuer la valeur de la fonction d'objectif.

En outre, d'après (5.2) on a :

$$T^i_{l-1} - T^i_l \geqslant 0 \qquad i = 1, 2 \ldots (M-1) \tag{5.8}$$

Il s'ensuit que (5.3) et (5.4) peuvent s'écrire

— soit

$$\min\ (T^M_1 - T^M_2) + (T^M_3 - T^M_4) + \ldots \tag{5.9}$$

soumis à

$$(T^i_1 - T^i_2) + (T^i_3 - T^i_4) + \ldots \qquad i = 1, 2 \ldots (M-1) \tag{5.10}$$

— soit

$$\min\ T^M_1 - (T^M_2 - T^M_3) - (T^M_4 - T^M_5) - \ldots \tag{5.11}$$

soumis à

$$T^i_1 - (T^i_2 - T^i_3) - (T^i_4 - T^i_5) - \ldots \quad i = 1, 2 \ldots (M-1) \tag{5.12}$$

Dans chacun de ces cas, les termes entre parenthèses sont positifs.

Nous définirons encore $T^i_l(J^1_s)$ pour $i = 1, 2 \ldots M$, qui représente la partie de T^i_l pour laquelle $x_j \in J^1_s$. Si (5.8) est vrai, l'on peut aussi écrire (5.13) :

$$T^i_{l-1}(J^1_s) - T^i_l(J^1_s) \geqslant 0 \tag{5.13}$$

On commence par donner à z la valeur $z^* = \infty$ pour $J^1_0 = \varnothing$ et $J^0_0 = \varnothing$. On suppose qu'après un certain nombre d'étapes s, on ait atteint la

solution Ψ_s à laquelle un certain nombre de tests seront appliqués.

Test n° 1 : calculer

$$\hat{b}_{1i} = b_i - T_1^i(J_s^1) \qquad i = 1, 2 \ldots (M-1)$$

Si $\hat{b}_{1i} > 0$ pour un i quelconque, passer au test n° 2.

Si $\hat{b}_{1i} \leqslant 0$ pour tout i, calculer :

$$\hat{b}_{2i} = \hat{b}_{1i} + T_2^i(J_s^1) \qquad i = 1, 2 \ldots (M-1)$$

Si $\hat{b}_{2i} \leqslant 0$ pour tout i, on a découvert une solution admissible : on calcule

$$z^* = T_1^M(J_s^1) - T_2^M(J_s^1) + \ldots (-1)^{N-1} T_N^M(J_s^1) \tag{5.14}$$

et on passe à l'étape de retour.

Si $\hat{b}_{2i} > 0$ pour un i quelconque, calculer

$$\hat{b}_{3i} = \hat{b}_{2i} - T_3^i(J_s^1) \qquad i = 1, 2 \ldots (M-1)$$

Si $\hat{b}_{3i} > 0$ pour un i quelconque, passer au test n° 2.

Si $\hat{b}_{3i} \leqslant 0$ pour tout i, introduire $T_4^i(J_s^1)$, etc.

De façon générale, le test peut s'écrire

$$\hat{b}_{2k-1,i} = b_{2k-2,i} - T_{2k-1}^i(J_s^1) \qquad i = 1, 2 \ldots (M-1) \qquad k = 1, 2 \ldots N \in J_s^1 \tag{5.15.1}$$

Si $\hat{b}_{2k-1,i} > 0$ pour un i quelconque, passer au test n° 2.

Si $\hat{b}_{2k-1,i} \leqslant 0$ pour tout i calculer

$$\hat{b}_{2k,i} = \hat{b}_{2k-1,i} + T_{2k}^i(J_s^1) \qquad i = 1, 2 \ldots (M-1) \qquad k = 1, 2 \ldots N \in J_s^1 \tag{5.15.2}$$

Si $\hat{b}_{2k,i} \leqslant 0$ pour tout i, calculer (5.14) et passer à l'étape de retour.

Si $\hat{b}_{2k,i} > 0$ pour un i quelconque, augmenter k de une unité et passer en (5.15.1).

Test n° 2 : Calculer :

$$\hat{a}_h^M = T_1^M(J_s^1 \cup \{h\}) - T_2^M(J_s^1 \cup \{h\}) + \ldots (-1)^{N-1} T_N^M(J_s^1 \cup \{h\}) - z^* \text{ pour } h \in N_s \tag{5.16}$$

et introduire dans J_s^0 tout h tel que $\hat{a}_h^M > 0$

Si, après cela $N_s = \varnothing$, passer à l'étape de retour.

Si $N_s \neq \varnothing$, passer au test n° 3.

Test n° 3 : Pour tout $\hat{b}_{2k-1,i} > 0$ calculer :

$$\bar{b}_{1i} = \hat{b}_{2k-1,i} - T_1^i(N_s \cup J_s^1)$$

Si $\bar{b}_{1i} > 0$, passer à l'étape de retour.

Si $\bar{b}_{1i} \leqslant 0$, calculer

$$\bar{b}_{2i} = \bar{b}_{1i} + T_2^i(N_s \cup J_s^1) + T_2^i(N_s)$$

Si $\bar{b}_{2i} \leqslant 0$, passer à l'étape de génération régulière.

Si $\bar{b}_{2i} > 0$, calculer

$$\bar{b}_{3i} = \bar{b}_{2i} - T_3^i(N_s \cup J_s^1) - T_3^i(N_s)$$

Si $\bar{b}_{3i} > 0$, passer à l'étape de retour.

Si $\bar{b}_{3i} \leqslant 0$, introduire $T_4^i(N_s \cup J_s^1)$ etc.

De façon générale, le test peut s'écrire

$$\bar{b}_{2l-1,i} = \bar{b}_{2l,i} - T_{2l-1}^i(N_s \cup J_s^1) - T_{2l-1}(N_s) \qquad (5.17.1)$$

Si $\bar{b}_{2l-1,i} > 0$, passer à l'étape de retour.

Si $\bar{b}_{2l-1,i} \leqslant 0$, calculer

$$\bar{b}_{2l,i} = \bar{b}_{2l-1,i} + T_{2l}^i(N_s \cup J_s^1) + T_{2l}^i(N_s) \qquad (5.17.2)$$

Si $\bar{b}_{2l,1} \leqslant 0$, passer à l'étape de génération régulière.

Si $\bar{b}_{2l,i} > 0$, augmenter l de une unité et passer en (5.17.1).

Étape de génération régulière : Afin de réduire les calculs, on utilisera la même formulation que dans le cas quadratique ; l'hypothèse implicite est que

$$T_3^i - T_4^i + \ldots < T_1^i - T_2^i$$

ce qui sera généralement le cas.

On calculera dès lors :

$$V_{j'}^s = \sum_{i \mid \hat{b}_i - a_{j'}^i + \sum_{\substack{j \in J_s^1 \\ j<j'}} a_{jj'} > 0} \left(\hat{b}_i - a_{j'}^i + \sum_{\substack{j \in J_1 \\ j<j'}} a_{jj'} \right) \qquad (5.18)$$

et l'on choisira j'_* tel que $\varphi_{j*}^{s'} = \min \varphi_{j'}^s$ (5.19)

Si j'_* n'est pas unique, on choisit celui pour lequel $a_{j'}^M$ est minimum.

On construit alors une nouvelle solution définie par :

$$J_{s+1}^1 = J_s^1 \cup \{ j'_* \} \quad \text{et} \quad J_{s+1}^0 = J_s^0 \qquad (5.20)$$

et l'on retourne au test n° 1 (nouvelle itération).

Étape de retour : On retourne à la solution Ψ_r à partir de laquelle la solution Ψ_s a été engendrée. Si $s = 0$, soit la solution associée à z^* minimum est optimale soit il n'y a pas de solution admissible ($z^* = \infty$).

On introduit dans J_r^0 l'indice j défini par $\{j\} = J_s^1 - J_r^1$ et on retourne au test n° 2.

REMARQUES : Cet algorithme risque de demander un temps de calcul assez long, bien que la décomposition des 1er et 3e tests en sous-tests permet (à condition de n'être pas trop près d'une solution admissible) d'aller assez vite.

L'étape de saut (tests 4 et 5 du cas quadratique) est supprimée : les calculs seraient trop longs et intuitivement, le temps calcul sans étape de saut semble plus rapide : l'algorithme est moins efficace, mais peut-être plus rapide.

6. APPLICATIONS

Pour simplifier l'exposé des applications, on supposera que les intersections de plus de deux ensembles sont nulles ou négligeables. En général d'ailleurs, on ne les connaît pas ou l'on ne peut les établir que difficilement.

6.1. Sélection d'un ensemble de revues et journaux optimisant la dépense publicitaire

Cette application est présentée en premier lieu, parce que c'est le calcul d'une couverture nette d'audience publicitaire qui est à l'origine du développement de l'algorithme.

Le problème soulevé par cette application est celui de la lecture de plusieurs supports publicitaires par le même lecteur.

Les coefficients $1/2\ (a_{jj'} + a_{j'j})$ jouent ici le rôle des duplications ; les coefficients a_j représentent le nombre de lecteurs du support j et les variables x_j le choix ou l'ignorance du support j.

L'algorithme permet de traiter deux cas :

Cas 1 : rendre minimum la dépense publicitaire, étant donné la fixation de certains objectifs quant à l'audience :

$$\min \sum_j c_j x_j \qquad j = 1, 2 \ldots N$$

Soumis à

$$\sum_j a_j^i x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'} \geqslant b_i \qquad i = 1, 2 \ldots M \quad (1)$$

$$0 \leqslant x_j \leqslant 1 \ ; \ x_j \text{ entier}$$

avec c_j = coût d'insertion dans le support j, a_j = nombre de lecteurs du support j, $a_{jj'}$ = nombre de lecteurs lisant les supports j et j' et b_i = objectif de lecture i (par exemple nombre de lecteurs à toucher, dont l'âge se situe entre 25 et 35 ans).

(1) Le terme 1/2, qui affecte les coefficients, peut avoir n'importe quelle valeur si on considère que les duplications peuvent avoir un effet non nul.

Cas 2 : rendre maximum la couverture nette, étant donné un budget D :

$$\max \sum_j a_j x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'}$$

soumis à $\sum_j c_j x_j \leqslant D$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier.}$$

6.2. Sélection d'emplacements de dépôts d'approvisionnement ou de succursales

Ici aussi, l'on peut envisager plusieurs cas ; définissons par a_j le « rayon d'action » (par exemple nombre de consommateurs possible, ou nombre de clients à livrer) du siège j, par c_j le coût d'installation de ce siège, par D le budget global, par B la clientèle potentielle à toucher et par $a_{jj'}$ la clientèle commune à deux rayons.

Cas 1 :

$$\min \sum_j \sum_{j'} a_{jj'} x_j x_{j'}$$

soumis à

$$\sum_j c_j x_j \leqslant D$$

$$\sum_j a_j x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'} \geqslant B$$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier.}$$

Cas 2 :

$$\text{Max} \sum_j a_j x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'}$$

soumis à :

$$\sum_j c_j x_j \leqslant D$$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier.}$$

Cas 3 :

$$\min \sum_j c_j x_j$$

soumis à :

$$\sum_j a_j x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'} \geqslant B.$$

6.3. Recherche de gisements miniers ou pétrolifères

On suppose que la recherche d'un gisement donne lieu à un certain nombre de forages. A chaque forage on peut attribuer une probabilité *a priori* de tomber sur un filon, et l'on espère qu'après un certain nombre de forages, le filon sera découvert. En outre, à chaque forage on peut attribuer des probabilités conditionnelles *a priori*, donnant la probabilité de trouver le filon, étant donné un forage voisin.

Définissons par x_j un forage envisagé ; si $x_j = 0$ le forage n'est pas réalisé ; $x_j = 1$ dans le cas où l'on procède au forage. Dès lors,

$$E[x_j = 1] = a_j$$

$$E[x_j = 1 \mid x_{j'} = 1] = E[x_{j'} = 1 \mid x_j = 1] = 1/2(a_{jj'} + a_{j'j})$$

Le programme peut s'écrire :

$$\min \sum_j c_j x_j$$

soumis à :

$$\sum_j a_j x_j - 1/2 \sum_j \sum_{j'} a_{jj'} x_j x_{j'} \geqslant 1$$

$$0 \leqslant x_j \leqslant 1 \; ; \; x_j \text{ entier}$$

où c_j indique le coût du forage j.

Il est clair que l'on peut introduire d'autres contraintes, telle que la contrainte de temps maximum disponible B, soit :

$$\sum_j b_j x_j \leqslant B$$

où b_j représente le temps de forage du puits j.

7. RESULTATS D'EXPERIMENTATION

Les calculs ont été faits sur un calculateur IBM 7040, disposant de 50 disques ; il est clair que le traitement de ce genre de problèmes pour lesquels les données sont très nombreuses, exige des mémoires à accès rapide. Le programme a été écrit en Fortan IV.

7.1. Optimisation d'un plan publicitaire de presse (1)

Le problème traité est celui décrit dans la section 6.1, cas 1 : sélectionner parmi 19 journaux belges, un sous-ensemble de supports rendant minimale la dépense totale en couvrant 60 à 70 % des audiences suivantes :

(1) population féminine de 18 à 35 ans
(2) population masculine de 18 à 25 ans
(3) population féminine totale
(4) à (7) population par région (on a considéré 4 régions)
(8) population des grandes agglomérations
(9) population des classes socio-professionnelles *A-B*.

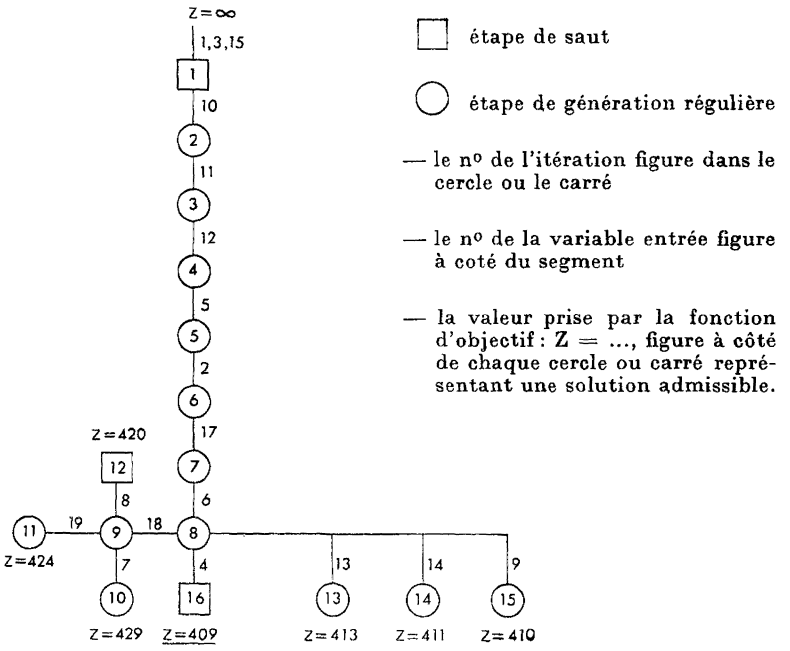
L'on définit ainsi 9 contraintes quadratiques : la partie linéaire représente dans chaque cas le nombre de lecteurs d'un journal déterminé, dans la strate de population visée ; les duplications de lecture sont représentées par la partie quadratique (2).

(1) Cas réel destiné à calculer le plan optimal de publicité dans la presse pour un lancement de cigarettes blondes s'adressant aux jeunes et plutôt à la population féminine.
(2) Source des statistiques utilisées : « Étude sur les lecteurs de la presse belge et luxembourgeoise », Centre d'Étude Belge des Supports de Publicité, Bruxelles, 1965.

La fonction d'objectif est bien entendu linéaire, les éléments donnant les coûts d'insertion par journal.

Le coût d'une annonce-type dans l'ensemble des 19 journaux s'élève à 545 000 FB. L'algorithme sélectionne un sous-ensemble optimal de 11 journaux pour un coût minimal de 409 000 FB.

L'arborescence explorée au cours de l'algorithme est représentée :



La solution optimale comprend les variables 1, 2, 3, 4, 5, 6, 10, 11, 12, 15 et 17.

Figure 1

Arborescence explorée par l'algorithme

7.2. Autres résultats

L'exemple détaillé ci-dessus comportait des matrices assez creuses ; en effet, dans le domaine de la presse, certains types de duplications sont rares : journaux de partis différents, de langues différentes, journaux locaux, etc.

Pour estimer le temps-calcul nécessité par les problèmes où les matrices ne seraient pas creuses, l'algorithme a été appliqué à 7 cas dans lesquels toutes les matrices étaient intégralement pleines. De plus, les fonctions d'objectif étaient également quadratiques.

Ces 7 cas ont été construits par simulation, les nombres utilisés étant tirés d'une distribution aléatoire (0 — 1), de façon à respecter les hypothèses (3.1) à (3.4).

TABLEAU 1. — *Résultats*

N° DU PROBLEME	NOMBRE DE VARIABLES	NOMBRE DE CONTRAINTES	TEMPS-CALCUL TOTAL (1)	NOMBRE TOTAL D'ITERATIONS	PREMIERE SOLUTION ADMISSIBLE		SOLUTION OPTIMALE		MAXIMUM DE LA FONCTION D'OBJECTIF
					Nombre de sommets explorés	Valeur de la fonction d'object.	Nombre de sommets explorés	Valeur de la fonction d'object.	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	19	9	3′30″	30	10	429,0	16	409,0	545,0
2	20	10	6′	35	15	107,3	16	104,8	210,0
3	20	20	9′30″	35	12	95,6	13	93,9	213,0
4	20	25	4′	14	7	120,0	7	120,0	231,1
5	30	10	18′	55	17	191,8	17	191,8	470,8
6	30	20	30′	52	20	195,9	24	195,9	458,1
7	30	25	33′	47	21	222,6	23	222,6	465,0
8	40	20	66′	69	30	390,6	34	390,6	806,0

(1) Avec un ordinateur IBM 7040.

7.3. Temps de résolution

Le tableau 1 donne une vue des temps-calcul exigés ; ceux-ci croissent très vite avec le nombre de variables (ceci est évident étant donné que l'on travaille avec des formes quadratiques) ; par contre, la relation temps-calcul nombre de contraintes apparaît moins clairement.

Seul le problème n° 1 a une signification économique : c'est le cas de l'optimisation du plan publicitaire presse. Les autres sont des constructions simulées.

Le tableau 1 donne pour chaque problème le nombre de variables et de contraintes, ainsi que le temps-calcul nécessaire au nombre total d'itérations. Étant donné la structure de l'algorithme, ces itérations peuvent être complètes (cas de l'étape de saut ou de génération régulière) ou incomplètes (cas de l'étape de retour, où l'on ne passe pas nécessairement tous les tests).

Dans les colonnes (6) et (7) figure le nombre de sommets explorés (ce nombre correspond à celui des itérations complètes) et la valeur de la fonction d'objectif pour la première solution admissible. Les colonnes (8) et (9) reprennent les mêmes éléments pour la solution optimale.

L'on peut tirer de ces chiffres deux conclusions intéressantes :

a) la première solution admissible ne s'écarte dans aucun cas de plus de 5 % de la solution optimale ;

b) le nombre total d'itérations est au moins égal au double du nombre de sommets explorés pour atteindre la première solution admissible.

Ces constatations sont importantes dans la mesure où l'on obtient une solution quasi-optimale (1) longtemps avant la fin des calculs (2), ce qui permet d'économiser du temps-machine, si cela s'avère nécessaire.

---

(1) Compte tendu de l'imprécision affectant généralement les données, il est souvent utopique de chercher le vrai optimum.

(2) C'est d'ailleurs une caractéristique commune à beaucoup d'algorithmes du type « branch and bound ».