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

M. FLORIAN
P. ROBILLARD

Programmation hyperbolique en variables bivalentes

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 5, n° V1 (1971), p. 3-9.

<http://www.numdam.org/item?id=RO_1971__5_1_3_0>

R.I.R.O.
(5e année, V-1, 1971, p. 3-9)

PROGRAMMATION HYPERBOLIQUE EN VARIABLES BIVALENTES

par M. FLORIAN et P. ROBILLARD (1) (2)

Résumé. — *Plusieurs problèmes économiques peuvent être avantageusement formulés dans le contexte de la programmation hyperbolique pseudo-Booléenne à variables bivalentes. Dans cet article une approche nouvelle pour résoudre ces problèmes est décrite alors que l'on montre que la solution optimale peut être obtenue après avoir résolu un nombre fini de problèmes qui consistent à minimiser une fonction linéaire sous les contraintes du problème original. Un algorithme est décrit et des résultats numériques concernant son utilisation sont discutés.*

1. INTRODUCTION

Comme l'ont noté Hammer et Rudeanu [4] plusieurs problèmes économiques peuvent être avantageusement formulés dans le contexte de la programmation hyperbolique pseudo-booléenne. Un exemple de ces problèmes est de minimiser le rapport entre le coût total de production, qui est habituellement représenté par une fonction linéaire, et la quantité d'objets produits. Un autre exemple est de vouloir maximiser par un investissement judicieux le pourcentage des gains en respectant certaines contraintes. Des méthodes mathématiques ont été présentées en [6] et [4] pour résoudre ces problèmes pour le cas où il n'y a pas de contraintes et celui où les contraintes sont exprimées grâce à des inégalités sur des fonctions croissantes pseudo-booléennes.

Le problème de programmation hyperbolique qui nous intéresse ici peut être formulé comme suit :

$$\text{(1)} \qquad \text{minimisez } F(X) = \frac{a_0 + \sum_{i=1}^{n} a_i x_i}{b_0 + \sum_{i=1}^{n} b_i x_i}$$

(1) Département d'informatique, Université de Montréal, mai 1970.

(2) Cette étude a été effectuée grâce à une subvention du Conseil National des Recherches du Canada.

avec les contraintes

$$x_i = \{0, 1\}, i = 1, 2, ..., n \tag{2}$$

$$H_j(X) \leqslant d_j, j = 1, 2, ..., m, \tag{3}$$

où les H_j sont des fonctions pseudo-booléennes; on impose que

$$b_i > 0, i = 0, 1, 2, ... n. \tag{4}$$

Nous désignerons à l'avenir le problème (1), (2) et (3) comme le problème P et nous dirons que le vecteur X^* est une solution optimale de P s'il minimise (1) et satisfait les contraintes (2) et (3). Notons que le problème P étudié ici est un peu plus général que le problème étudié en [4] et [6] où on spécifie en plus que $a_i \geqslant 0, i = 1, 2, ..., n$.

Dans la première partie de cet exposé nous présenterons une approche nouvelle pour résoudre ce problème et nous montrerons que la solution optimale peut être obtenue après avoir résolu un nombre fini de problèmes qui consistent à minimiser une fonction linéaire sous les contraintes originales (2) et (3). Nous décrirons ensuite un algorithme pour résoudre le problème P et nous examinerons le cas sans contraintes. Nous terminerons cette étude en présentant des résultats et des observations numériques concernant l'utilisation des algorithmes proposés.

2. LINEARISATIONS SUCCESSIVES DU PROBLEME

Dénotons par C^0 l'ensemble des vecteurs X qui satisfont aux inégalités données en (3) i.e.

$$C^0 = \{ X \mid H_j(X) \leqslant d_j, j = 1, 2, ..., \text{m} \}.$$

Si $C^0 = \varnothing$ nous disons que le problème P est irréalisable. Dans le cas contraire trouvons le vecteur X^0 qui maximise $F(X)$ sans contraintes. Ceci peut être fait très efficacement en utilisant un des algorithmes suggérés en [4] ou en [6]. Le maximum de $F(X)$ peut aussi se calculer en utilisant l'algorithme suggéré ici, dans le cas particulier où il n'y a pas de contraintes comme indiqué en section 3.

Nous écrivons que $F(X^0) = \lambda^0$ et

$$F(X^*) \leqslant F(X^0) = \lambda^0 \tag{5}$$

Utilisant la dernière inégalité et les conditions sur les b_i on déduit que

$$g(X, \lambda^0) = (a_0 - \lambda^0 b_0) + \Sigma(a_i - \lambda^0 b_i)x_i \leqslant 0 \text{ pour tout } X \in C^0. \tag{6}$$

Nous définissons maintenant le premier problème linéaire P^1 qui découle du problème P :

$$\text{minimisez } g(X, \lambda^0) \text{ et } X \in C^0. \tag{7}$$

Soit X^1 la solution de P^1 et $\lambda^1 = F(X^1)$. On a évidemment que $F(X^1) \leqslant F(X^0)$.

Si $g(X^1, \lambda^0) = 0$ on conclut utilisant (6) que X^1 est une solution optimale du problème P.

Dans le cas où $g(X^1, \lambda^0) < 0$ on sait que $\lambda^1 = F(X^1) < \lambda^0$ et que

$$F(X^*) \leqslant F(X^1).$$

On a donc

$$g(X^*, \lambda^1) = (a_0 - \lambda^1 b_0) + \sum_{i=1}^{n} (a_i - \lambda^1 b_i) x_i^* \leqslant 0 \tag{8}$$

et on définit le deuxième problème linéaire P^2 comme suit :

$$\text{minimisez } g(X, \lambda^1) \text{ et } X \in C^0. \tag{9}$$

Soit X^2 la solution de P^2 et $\lambda^2 = F(X^2)$. On a immédiatement $\lambda^2 \leqslant \lambda^1$ et si $g(X^2, \lambda^1) = 0$ alors X^2 est une solution optimale du problème P. Dans le cas où $g(X^2, \lambda^1) < 0$ on a $\lambda^2 < \lambda^1$ et $F(X^*) \leqslant F(X^2)$. Nous définissions comme précédemment un problème linéaire P^3.

On obtient de cette façon une suite de vecteurs distincts $X^1, X^2, \ldots$ tels que

$$X^i \in C^0 \text{ et } F(X^i) > F(X^{i+1}) \; i = 1, 2, \ldots$$

Comme C^0 contient un nombre fini de vecteurs la suite obtenue est finie et le dernier vecteur obtenu est une solution optimale du problème P. La méthode décrite ici est analogue à celle de Abadie et Williams [2] et Isbell et Marlow [5] pour le cas où les contraintes sont linéaires et les variables sont réelles quelconques. Bien entendu elle est ici adaptée au cas bivalent.

3. ALGORITHME POUR LE PROBLEME P

Nous formalisons l'algorithme pour résoudre le problème P. Comme auparavant nous écrivons, $g(X, \lambda) = (a_0 - \lambda b_0) + \Sigma(a_i - \lambda b_i)x_i$. Nous supposons aussi que $C^0 \neq \varnothing$.

(1) Trouvez X^0 qui maximise $F(X)$ (voir [6] et [4]); soit $\lambda = F(X^0)$;

(2) trouvez X^+ qui minimise $g(X, \lambda)$ et $X \in C^0$;

(3) si $g(X^+, \lambda) = 0$, X^+ est la solution optimale, arrêtez; sinon posez $\lambda = F(X^+)$ et retournez à (2).

La solution du problème P^{k-1} est une solution de départ pour le problème P^k.

Nous faisons maintenant quelques remarques concernant certains cas particuliers du problème.

Dans le cas où il n'y a pas de contraintes on doit modifier l'étape (1) de la façon suivante :

(1)* Pour un X^0 quelconque soit $\lambda = F(X^0)$;

si on prend $X^0 = 0$ (le vecteur nul), l'algorithme est alors identique à l'algorithme I décrit page 153 en [4].

Si on connait à l'avance un vecteur $X \in C^0$ on peut accélérer la procédure en remplaçant l'étape (1) par (1)** $\lambda = F(X)$;

On note qu'il peut exister plusieurs vecteurs de C^0 qui soient solutions optimales du problème; l'algorithme décrit ici ne produira qu'un seul de ces vecteurs solutions.

Le problème qui consiste à maximiser $F(X)$ avec les contraintes (2) et (3) peut être résolu avec l'algorithme décrit plus haut où l'on aura interchangé les mots « maximiser » et « minimiser ».

Exemple numérique

Dans le cas où les contraintes du problème sont linéaires on peut utiliser pour l'étape (2) de l'algorithme décrit plus haut des méthodes connues pour la solution de problèmes linéaires pseudo-booléen. Ainsi utilisant l'algorithme de Goeffrion [3], dont le code pour ordinateur est décrit en [2], nous avons mis au point un programme pour résoudre les problèmes hyperboliques à contraintes linéaires et nous décrivons maintenant quelques essais faits sur un ordinateur CDC 6400 à l'Université de Montréal.

Problème 1

Ce problème est celui décrit par Hammer et Rudeanu [4, p. 157]. Il faut minimiser la fonction

$$F = \frac{3 + 2x_1 + 4x_2 + x_3 + 2x_4 + 9x_5 + 6x_6 + 12x_7 + 8x_8 + 2x_9 + 3x_{10} + 3x_{11} + x_{12}}{6 + x_1 + 8x_2 + 3x_3 + 5x_4 + 15x_5 + 10x_6 + 25x_7 + 18x_8 + 6x_9 + 3x_{10} + 7x_{11}}$$

avec les contraintes

$$\begin{aligned}
&x_1 - 3x_2 + 12x_3 + x_5 - 7x_6 + x_7 - 3x_{10} + 5x_{11} + x_{12} - 6 \geqslant 0,\\
&-3x_1 + 7x_2 - x_4 - 6x_5 + 1 \geqslant 0,\\
&-11x_1 - x_3 + 7x_4 + x_6 - 2x_7 - x_8 + 5x_9 - 9x_{11} - 4 \geqslant 0,\\
&-5x_2 - 6x_3 + 12x_5 - 7x_6 - 3x_8 - x_9 + 8x_{10} - 5x_{12} + 8 \geqslant 0,\\
&7x_1 + x_2 + 5x_3 - 3x_4 - x_5 + 8x_6 + 2x_8 - 7x_9 - x_{10} + 7x_{12} - 7 \geqslant 0,\\
&2x_1 + 4x_4 + 3x_7 + 5x_8 + x_9 - x_{11} - x_{12} - 4 \geqslant 0;
\end{aligned}$$

alors que $x_i \in \{0, 1\}$ $i = 1, 2, \ldots, 12$.

Dans une première étape nous trouvons

$$X_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1)$$

et $$\lambda = F(X^0) = .900$$

La fonction à minimiser est maintenant

$$g(X, 0.9) = 1.1x_1 - 3.2x_2 - 1.7x_3 - 2.5x_4 - 4.5x_5 - 3x_6 - 10.5x_7 - 8.2x_8 - 3.4x_9 + 0.3x_{10} - 3.3x_{11} + x_{12}$$

Le minimum de $g(X, 0.9)$ est atteint à

$$X = (0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1)$$

et $$\lambda = F(x) = .516$$

La nouvelle fonction à minimiser est

$$g(X, 0.516) = 1.48x_1 - 0.13x_2 - 0.54x_3 - 0.58x_4 + 1.25x_5 + 0.83x_6 - .90x_7 - 1.29x_8 - 1.09x_9 + 1.45x_{10} - 0.62x_{11} + x_{12}$$

et le minimum est atteint à

$$X = (0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1)$$

où $$\lambda = F(X) = .509.$$

On doit minimiser maintenant $g(X, 0.509)$ et son minimum est 0. La procédure se termine et la solution optimale est le dernier vecteur X. Le temps de calcul sur ordinateur a été de 3.2 secondes pour ce problème.

Problème 2

Le problème consiste ici à minimiser la fonction F décrite lors du problème précédent soumis aux trois premières contraintes du groupe de six contraintes décrites plus haut.

La procédure débute de la même façon qu'auparavant soit en trouvant le vecteur X^0 où la fonction F atteint son maximum.

La solution du premier problème linéaire permet de trouver un vecteur

$$X = (0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0)$$

pour lequel la fonction F prend la valeur 0.477. Ceci nous amène à définir un second problème linéaire dont la solution est

$$X = (0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0)$$

pour lequel F prend la valeur 0.421. Le troisième problème linéaire produit enfin la solution optimale du problème original soit

$$X = (0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0)$$

pour lequel F prend la valeur 0.4. Le temps de calcul pour résoudre ce problème a été de 0.76 secondes.

Problème 3

Le dernier problème que nous résolvons consiste à miniser la fonction

$$F = \frac{3 + 18x_1 + 42x_2 + 12x_3 + x_4 + 10x_5 + 6x_6 + 8x_7 + 55x_8 + 111x_9 + 18x_{10}}{1 + 9x_1 + 42x_2 + 7x_3 + 12x_4 + 18x_5 + 9x_6 + 14x_7 + 60x_8 + 500x_9 + 4x_{10}}$$

avec les contraintes

$$\begin{aligned}
&3x_1 + 12x_2 - 8x_3 - x_4 - 7x_9 + 2x_{10} + 8 \geqslant 0,\\
&-x_2 - 10x_3 - 5x_5 + x_6 - 7x_7 - x_8 + 13 \geqslant 0,\\
&5x_1 + 3x_2 - x_3 + 2x_8 + x_{10} - 6 = 0,\\
&-4x_3 + 2x_4 - 5x_6 - x_7 + 9x_8 - 2x_9 + 8 \geqslant 0,\\
&-9x_2 + 12x_4 - 7x_5 + 6x_6 - 2x_8 - 15x_9 - 3x_{10} + 12 \geqslant 0,\\
&-8x_1 - 5x_2 + 2x_3 + 7x_4 - x_5 + 5x_7 + 16 \geqslant 0,
\end{aligned}$$

alors que $x_i \in \{0, 1\}$, $i = 1, 2, ..., 10$.

Les vecteurs successifs produits par l'algorithme sont :

$$\begin{aligned}
X^0 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 1) \quad \text{et} \quad F(X^0) = 4.2\\
X^1 &= (0, 1, 0, 1, 0, 1, 1, 1, 1, 1) \quad \text{et} \quad F(X^1) = 0.38\\
X^2 &= (1, 0, 0, 1, 0, 0, 0, 0, 1, 1) \quad \text{et} \quad F(X^2) = 0.287.
\end{aligned}$$

Le vecteur X^2 est la solution optimale. Le temps de calcul pour ce problème est de 0.95 secondes.

CONCLUSION

L'approche suggérée ici pour résoudre les problèmes de programmation hyperboliques pseudo-booléen avec contraintes consiste à résoudre une suite de problèmes dans lesquels on minimise une fonction linéaire tout en conservant les contraintes originales. L'efficacité de l'algorithme proposé dépend évidemment de l'efficacité des algorithmes utilisés pour résoudre les problèmes intermédiaires. Dans le cas où les contraintes sont linéaires, les essais tentés jusqu'ici sur ordinateur semblent indiquer l'utilité de l'algorithme proposé. Notons que les trois problèmes présentés plus haut sont de petites tailles et nous devrons dans l'avenir évaluer l'efficacité de notre algorithme à résoudre des problèmes de grande taille.

REFERENCES

[1] J. M. ABADIE et A. C. WILLIAMS, *Dual and Parametric Methods in Decomposition*, in : Recent Advances in Mathematical Programming, Graves and Wolfe, eds, 1963, pp. 149-185, McGraw-Hill.

[2] A. M. GEOFFRION and A. B. NELSON, *User's Instruction for 0-1 Integer Linear Programming Code RIP30C*, Rand Memorandum, RM-5627-PR, May 1968.

[3] A. M. GEOFFRION, *An Improved Implicit Enumeration Approach for Integer Programming*, Opns. Res., 1969, 17, 437-454.

[4] P. L. HAMMER et S. RUDEANU, Boolean Methods in Operations Research and Related Areas, Springer-Verlag, 1968.

[5] J. L. ISBELL et W. H. MARLOW, *Attrition Games*, Nav. Res. Log. Quart., 1956, vol. 3, N° 1 and 2.

[6] P. ROBILLARD, *(0,1) Hyperbolic Programming Problems*, Publication Département d'informatique 19, Université de Montréal, 1970.