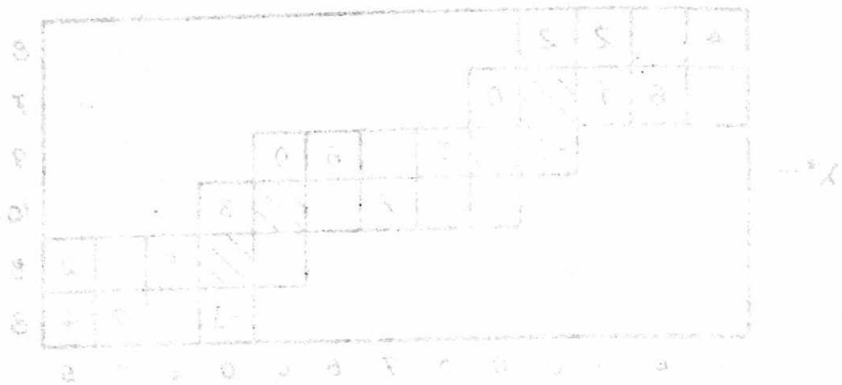




SUR UNE MÉTHODE DE DÉCOMPOSITION POUR LE PROBLÈME DE PROGRAMMATION MONOTONE

par

ȘTEFAN ȚIGAN

à Cluj

1. Du point de vue pratique il est souvent nécessaire de considérer une classe de problèmes de programmation mathématique ayant une structure spéciale, qui théoriquement peuvent se résoudre mais dont la résolution effective par les méthodes générales n'est pas possible à cause de difficultés, dont l'une des plus importantes est la dimension du problème. Ce fait a suggéré l'idée d'obtenir des méthodes pour la résolution des problèmes de programmation mathématique ayant une structure spéciale, qui n'utilisent pas simultanément toutes les données du problème.

Parmi ces méthodes il y a aussi le principe connu de décomposition de DANTZIG et WOLFE [2] pour le problème de programmation linéaire, où la matrice des coefficients des contraintes du problème a une forme spéciale. Ainsi qu'il est bien connu, ce principe réduit la résolution du problème initial à la résolution d'une suite de problèmes de programmation linéaire avec dimensions plus petites que celle du problème original.

Le problème considéré en [2] a la structure suivante. Maximiser la fonction

$$(1) \quad Z = \sum_{i=1}^r C_i X_i$$

dans les conditions

$$(2) \quad \sum_{i=1}^r A_i X_i = b_0$$

$$(3) \quad B_i X_i = b_i \quad (i = 1, 2, \dots, r)$$

$$(4) \quad X_i \geq 0 \quad (i = 1, 2, \dots, r),$$

où A_i et B_i sont des matrices de dimensions $(m_0 \times n_i)$ et $(m_i \times n_i)$ respectivement et $X_i = (x_{i1}, \dots, x_{in_i})$, $C_i = (c_{i1}, \dots, c_{in_i})$, $b_i = (b_{i1}, \dots, b_{in_i})$, ($i = 1, 2, \dots, r$).

Des méthodes de décomposition pour les problèmes de programmation nonlinéaire ont été considérées par plusieurs auteurs, parmi lesquels nous rappellerons ROSEN [5], WHINSTON [6], ZANGWILL [7], CHANDRA [1].

Dans ce travail nous considérerons une classe de problèmes de programmation monotone avec des contraintes linéaires de la forme (2)–(4) et pour cette classe de problèmes on obtient un principe de décomposition analogue au principe de décomposition Dantzig-Wolfe pour le problème de programmation linéaire. Ce principe de décomposition pour le problème de programmation monotone contient comme cas particuliers le principe de décomposition pour la programmation fractionnaire obtenu par S. CHANDRA [1] et le principe de Dantzig et Wolfe de décomposition pour la programmation linéaire.

2. Le problème de programmation monotone consiste en la détermination du maximum de la fonction monotone g

$$(5) \quad \max g(x)$$

dans les conditions

$$(6) \quad \sum_{j=1}^n P_j x_j = P_0$$

$$(7) \quad x_j \geq 0 \quad (j = 1, 2, \dots, n),$$

où $P_0 = (d_1, \dots, d_m)$ et $P_j = (a_{j1}, \dots, a_{jm})$ ($j = 1, 2, \dots, n$) sont des vecteurs donnés et $x = (x_1, \dots, x_n) \in R^n$.

On dit que la fonction g définie sur le domaine convexe $D \subset R^n$ est monotone sur D , si quelques soient les points x^1 et x^2 de D on a

$$(8) \quad \min \{g(x^1), g(x^2)\} \leq g(tx^1 + (1-t)x^2) \leq \max \{g(x^1), g(x^2)\},$$

quelque soit $t \in [0, 1]$.

Pour la résolution du problème de programmation monotone (5)–(7) KUCHER [4], montre que l'on peut obtenir en certaines hypothèses de différentiabilité de la fonction g , un algorithme du type simplex. Cet algorithme pour la programmation monotone diffère de l'algorithme simplex pour la programmation linéaire seulement par le fait que à chaque itération on prend comme fonction à optimiser la fonction linéaire

$$(9) \quad g_k(x) = \sum_{i=1}^n \frac{\partial g(x^k)}{\partial x_i} x_i$$

où x^k est la solution de base obtenue jusqu'à l'itération d'ordre k .

Des théorèmes A et B de l'annexe il résulte immédiatement la remarque suivante.

Remarque 1. Si le domaine K défini par les conditions (6), (7), est borné, la fonction g est pseudo-concave* sur le domaine K et alors l'algorithme du type simplex qui a comme fonction à optimiser, à chaque itération la fonction (9), obtient (en cas d'absence de la dégénérescence) une solution optimale pour le problème de programmation monotone (5)–(7) après un nombre fini d'itérations.

3. Nous considérons le problème suivant de programmation monotone.

Maximiser la fonction monotone

$$(10) \quad f(X) = f(X_1, \dots, X_r)$$

dans les conditions (2)–(4). Nous faisons aussi l'hypothèse que les ensembles

$$K_i = \{X_i \geq 0 \mid B_i X_i = b_i\} \quad (i = 1, 2, \dots, r)$$

sont bornés.

Par la suite, nous montrerons que pour le problème de programmation monotone (10), (2)–(4), on peut obtenir un principe de décomposition analogue au principe de décomposition Dantzig-Wolfe pour la programmation linéaire.

Soit X_i^h ($h = 1, 2, \dots, h_i$) les points extrémaux de l'ensemble convexe K_i ($i = 1, 2, \dots, r$). Du fait que l'ensemble K_i est un polyèdre convexe (borné), $X_i \in K_i$ si et seulement s'il existe les nombres

$$(11) \quad s_{ik} \geq 0 \quad (k = 1, 2, \dots, h_i)$$

tels que

$$(12) \quad \sum_{k=1}^{h_i} s_{ik} = 1$$

et

$$(13) \quad X_i = \sum_{k=1}^{h_i} s_{ik} X_i^k$$

La formule (13) peut encore s'écrire

$$(14) \quad x_{ij} = \sum_{k=1}^{h_i} s_{ik} x_{ij}^k$$

*) La fonction différentiable $g: R^n \rightarrow R$ est pseudo-concave sur K si pour tout $x, x' \in K$, $\nabla g(x)(x' - x) \leq 0$ implique $g(x') \leq g(x)$, où

$$\nabla g(x) = \left(\frac{\partial g(x)}{\partial x_1}, \dots, \frac{\partial g(x)}{\partial x_n} \right).$$

où

$$X_i^k = (x_{i1}^k, \dots, x_{in_i}^k), \quad (i = 1, 2, \dots, r; k = 1, 2, \dots, h_i).$$

Des formules (11), (12), (13) il s'ensuit que le problème (10), (2)–(4) est équivalent au problème suivant.

Maximiser la fonction

$$(15) \quad w(S) = w(S_1, S_2, \dots, S_r) = f\left(\sum_{k=1}^{h_1} s_{1k} X_1^k, \dots, \sum_{k=1}^{h_r} s_{rk} X_r^k\right)$$

dans les conditions

$$(16) \quad \begin{aligned} \sum_{i=1}^r \sum_{k=1}^{h_i} s_{ik} A_i X_i^k &= b_0, \\ \sum_{k=1}^{h_i} s_{ik} &= 1 \quad (i = 1, 2, \dots, r) \\ s_{ik} &\geq 0 \quad (i = 1, 2, \dots, r; k = 1, 2, \dots, h_i), \end{aligned}$$

où

$$S = (S_1, \dots, S_r) = (s_{11}, \dots, s_{1h_1}, \dots, s_{r1}, \dots, s_{rh_r}).$$

Si nous employons les notations

$$(17) \quad \begin{aligned} p_{ik} &= A_i X_i^k \quad (i = 1, 2, \dots, r; k = 1, 2, \dots, h_i) \\ q_{ik} &= (p_{ik}, e_i) \quad (i = 1, 2, \dots, r; k = 1, 2, \dots, h_i) \\ b &= (b_0, e), \end{aligned}$$

où e_i est le vecteur $(0, \dots, 0, 1, 0, \dots, 0)$ (1 à la $i^{\text{ème}}$ place) et e est le vecteur $(1, \dots, 1)$ qui a r coordonnées, alors le problème (15), (16) peut se transcrire de la manière suivante.

Maximiser la fonction (15) dans les conditions

$$(18) \quad \begin{aligned} \sum_{i=1}^r \sum_{k=1}^{h_i} s_{ik} q_{ik} &= b \\ s_{ik} &\geq 0 \quad (i = 1, 2, \dots, r; k = 1, 2, \dots, h_i). \end{aligned}$$

On voit que le problème (15), (18) a $m_0 + r$ contraintes pendant que le problème (10), (2)–(4) a $\sum_{i=0}^r m_i$ contraintes de sorte qu'en général le problème (15), (18) requiert une base plus petite que le problème (10), (2)–(4).

Le théorème suivant montrera que le problème (15), (18) est un problème de programmation monotone.

THÉORÈME 1. *Si la fonction f est monotone et pseudo-concave sur H (défini par (3)–(4)), alors la fonction w est monotone et pseudo-concave sur G défini par les conditions (18).*

Démonstration: Il faut montrer que quelques soient S' et S'' de G , on a

$$(19) \quad \min \{w(S'), w(S'')\} \leq w(tS' + (1-t)S'') \leq \max \{w(S'), w(S'')\}$$

quelque soit $t \in [0, 1]$.

En effet, si nous supposons, par exemple, que $w(S') \leq w(S'')$, alors il résulte de (15) et de la monotonie de la fonction f que

$$\begin{aligned} w(tS' + (1-t)S'') &= f\left(t \sum_{k=1}^{h_1} s'_{1k} X_1^k + (1-t) \sum_{k=1}^{h_1} s''_{1k} X_1^k, \dots, \right. \\ &\quad \left. t \sum_{k=1}^{h_r} s'_{rk} X_r^k + (1-t) \sum_{k=1}^{h_r} s''_{rk} X_r^k\right) \geq f\left(\sum_{k=1}^{h_1} s'_{1k} X_1^k, \dots, \sum_{k=1}^{h_r} s'_{rk} X_r^k\right) = \\ &= w(S') = \min \{w(S'), w(S'')\}. \end{aligned}$$

De la même manière, on peut démontrer l'autre inégalité de (19).

La pseudo-concavité de la fonction w résulte facilement de (15) et du fait que la fonction f est pseudo-concave.

4. Par la suite, en partant de la remarque 1 et du théorème 1 nous décrirons une méthode de décomposition pour le problème de programmation monotone (10), (2)–(4), qui est analogue à la méthode de décomposition Dantzig-Wolfe pour la programmation linéaire. Cette méthode de décomposition réduit la résolution du problème de programmation monotone (10), (2)–(4) à la résolution d'un problème de programmation monotone qui requiert une base plus petite que le problème (10), (2)–(4) et d'une suite de problèmes de programmation linéaire avec dimensions plus petites que celle du problème (10), (2)–(4).

Soit B une matrice de base (d'ordre $m_0 + r$) pour le problème (15), (18). Alors le vecteur des valeurs des variables de base est

$$(s_{i_1 k_1}^B, \dots, s_{i_{m_0+r} k_{m_0+r}}^B) = B^{-1}b.$$

Nous désignerons par S^B la solution admissible de base du problème (15), (18) qui a les composantes

$$s_{i_v k_v} = s_{i_v k_v}^B \quad (v = 1, 2, \dots, m_0 + r),$$

les autres composantes étant nulles et par X^B la solution admissible du problème (10), (2)–(4), obtenu de S^B par les formules (12).

Nous employons également les notations:

$$\nabla_i f(X^B) = \left(\frac{\partial f(X^B)}{\partial x_{i1}}, \dots, \frac{\partial f(X^B)}{\partial x_{in_i}} \right) \quad (i = 1, 2, \dots, r)$$

$$(20) \quad w_{ik} = \frac{\partial w(S^B)}{\partial s_{ik}} = \nabla_i f(X^B) X_i^k$$

$$w_B = (w_{i_1 k_1}, \dots, w_{i_{m_0+r} k_{m_0+r}}).$$

Soit $u = (u_1, u_2)$, où u_1 contient les premières m_0 composantes de u et u_2 les dernières r composantes de u . Alors de (17) et (20) on déduit

$$(21) \quad w_{ik} - z_{ik} = w_{ik} - w_B B^{-1} q_{ik} = w_{ik} - u_1 p_{ik} - u_{2i} =$$

$$= (\nabla_i f(X^B) - u_1 A_i) X_i - u_{2i},$$

où u_{2i} est la i -ème composante du vecteur u .

Pour déterminer si la solution S^B est optimale il faut calculer

$$z = \max_{i,k} (w_{ik} - z_{ik}).$$

De (21) on déduit que le $\max_k (w_{ik} - z_{ik})$ est atteint dans un point extrémal du polyèdre convexe K_i , d'où il résulte que pour déterminer z il suffit de résoudre les problèmes de programmation linéaire suivants,

$$(22) \quad \max_{X_i \in K_i} Z_i(X_i) \quad (i = 1, 2, \dots, r),$$

où

$$Z_i(X_i) = (\nabla_i f(X^B) - u_1 A_i) X_i \quad (i = 1, 2, \dots, r).$$

Supposons que les problèmes de programmation linéaire (22) aient les solutions optimales X'_i ($i = 1, 2, \dots, r$) et soit

$$(23) \quad Z'_i = Z_i(X'_i) \quad (i = 1, 2, \dots, r).$$

Alors de (21) et (23), on déduit

$$(24) \quad z = \max_i (Z'_i - u_{2i}) = Z'_\alpha - u_{2\alpha}.$$

Si $z \leq 0$, alors la solution de base S^B est une solution optimale pour le problème (15), (18), donc X^B est une solution optimale pour le problème (10), (2)-(4).

Si $z > 0$, alors S^B et X^B ne sont pas des solutions optimales.

Par la suite, pour passer à l'itération suivante, on introduit (conformément à la technique simplex) dans la matrice de base du problème (15), (18) un vecteur $q_{\alpha\nu} = (p_{\alpha\nu}, e_\alpha)$ où $p_{\alpha\nu} = A_\alpha X_\alpha^\nu$.

Ainsi au lieu de la matrice de base B nous aurons une nouvelle matrice de base $\hat{B}$. Puis on calculera $\hat{B}^{-1}$, $S^{\hat{B}}$, $X^{\hat{B}}$, $w_{\hat{B}}$, $\hat{u}$ et on commencera l'itération suivante avec ces éléments.

On a le théorème suivant:

THÉORÈME 2. Si la fonction monotone f est pseudo-concave sur le domaine H , défini par les conditions (2)-(4), alors la méthode de décomposition pour le problème (10), (2)-(4) décrite ci-dessus détermine après un nombre fini d'itérations une solution optimale.

Si l'on remarque que la méthode de décomposition pour la programmation monotone décrite ci-dessus ne diffère pas de la méthode de décomposition Dantzig-Wolfe pour la programmation linéaire que par le fait qu'à chaque itération on change de fonction à optimiser, en prenant

$$(25) \quad F(X) = \sum_{i=1}^r \sum_{j=1}^{n_j} \frac{\partial f(\bar{X})}{\partial x_{ij}} x_{ij}$$

($\bar{X}$ est la solution obtenue jusqu'à l'itération considérée), alors le théorème 2 résulte immédiatement du fait que l'algorithme pour la programmation monotone et l'algorithme simplex pour la programmation linéaire déterminent une solution optimale après un nombre fini de pas, et du théorème 1.

Remarque 2. Nous faisons la remarque que de la méthode de décomposition pour la programmation monotone il résulte facilement, en particulier, la méthode de décomposition pour la programmation fractionnaire obtenue par S. CHANDRA en [1]. Ceci a lieu parce que le problème de programmation fractionnaire est un cas particulier du problème de programmation monotone (voir [8]) et le critère d'optimisation utilisé en [1] pour le problème de programmation fractionnaire peut s'obtenir facilement par l'application du critère d'optimisation (28) (de la programmation monotone) au problème de programmation fractionnaire.

5. Annexe. Soit $x' = (x'_1, \dots, x'_m, 0, \dots, 0)$, $x'_i > 0$ ($i = 1, 2, \dots, m$), une solution admissible de base du problème (5), (6), (7). Alors on a

$$P_0 = \sum_{i=1}^m x'_i P_i.$$

Alors on a les théorèmes suivants (voir [9], [4], [10]).

THÉORÈME A. Si la fonction monotone g est pseudo-concave sur le domaine K (défini par (6), (7)) et s'il y a un indice k ($m < k \leq n$) tel que

$$(26) \quad \frac{\partial g(x')}{\partial x_k} - \sum_{i=1}^m \frac{\partial g(x')}{\partial x_i} y_{ik} > 0$$

alors

$$g(x'') > g(x').$$

où

$$(27) \quad x'' = (x'_1 - t_k y_{1k}, \dots, x'_m - t_k y_{mk}, 0, \dots, 0, t_k, 0, \dots, 0)$$

et

$$t_k = \min_{t_{ik} > 0} \frac{x'_i}{y_{ik}}.$$

Si pour un indice k ($m < k \leq n$) la relation (12) a lieu et les inégalités

$$y_{ik} \leq 0 \quad (i = 1, 2, \dots, m)$$

sont satisfaites, alors

$$\lim_{t_k \rightarrow \infty} g(x'') > g(x'),$$

où $t_k > 0$ est un nombre arbitraire en (27).

THÉORÈME B. Si

- a) la fonction monotone g est pseudo-concave sur le domaine K ,
- b) la solution admissible de base x' satisfait à la condition suivante

$$(28) \quad \frac{\partial g(x')}{\partial x_k} - \sum_{i=1}^m \frac{\partial g(x')}{\partial x_i} y_{ik} \leq 0, \quad k = 1, 2, \dots, n,$$

alors x' est une solution optimale pour le problème de programmation monotone (5), (6), (7).

BIBLIOGRAPHIE

- [1] Chandra S., *Decomposition principle for linear fractional functional programs*. Revue française d'Informatique et de Recherche Opérationnelle **10**-R2, 63-71 (1968).
- [2] Dantzig C. B., Wolfe P., *Decomposition principle for Linear Programs*, Operations Research, **8**, 11-111 (1960).
- [3] Kasianiuk S. A., Kucher B. M., *Pro monotone programuvanja*, Dopovidi Akademii Nauk Ukrain. R.S.R., s.A, **8** (1967).
- [4] Kucher B. M., *Pro odin algoritm rešenja zadaci monotonno programuvanja*, Dopovidi Akademii Nauk Ukrain. R.S.R., s.A, **10** (1967).
- [5] Rosen J. B., *Convex partition Programming*, Recent Advances in Math. Programming (1963).
- [6] Whinston A., *A dual decomposition algorithm for Quadratic Programming*, Cahiers du centre d'études de Recherche Opérationnelle, **6**, 188-201 (1964).
- [7] Zangwill W., *A decomposable nonlinear programming approach*, Operations Research, **15**, 16, 1068-1087 (1967).
- [8] Swarup K., *Some properties of fractional Programming*, Cahiers du Centre d'Études de Recherche Opérationnelle, **9**, 2 (1967).
- [9] Martos B., *The direct Power of adjacent vertex programming methods*, Mang. Sc. vol. **12**, 3 (1965).
- [10] Zangwill W., *Nonlinear Programming, A Unified Approach*, (1969)

ADRESELE AUTORILOR

АДРЕСЫ АВТОРОВ

LES ADRESSES DES AUTEURS

- Wolfgang W. Breckner, Universitatea „Babeș-Bolyai“, str. Kogălniceanu nr. 1, Cluj.
- Bruno Brosowski, Institut für Numerische und Angewandte Mathematik der Universität, 34 Göttingen Bürger-Str. 32, Bundesrepublik Deutschland.
- Gh. Cimoca, Institutul de calcul, Cluj, str. Republicii nr. 37, Cluj.
- Azmi Hanna, Department of Mathematics American University of Beirut, Beirut, Republic of Lebanon.
- A. Kovács, Institutul de calcul, Str. Republicii nr. 37, Cluj.
- Jean Meinguet, Université de Louvain, Belgique.
- Maria Mihoc, Institutul de calcul, Str. Republicii nr. 37, Cluj.
- Petru T. Mocanu, Universitatea „Babeș-Bolyai“, str. Kogălniceanu nr. 1, Cluj.
- A. B. Németh, Institutul de calcul, str. Republicii nr. 37, Cluj.
- L. Némethi, Institutul de calcul, str. Republicii nr. 37, Cluj.
- Lilly Jeanne Nicolescu, Institutul de matematică, Calea Grivitei nr. 21. București, 12.
- Tiberiu Popoviciu, Institutul de calcul, str. Republicii nr. 37, Cluj.
- Maxwell O. Reade, University of Michigan, Ann Arbor, Michigan, U.S.A.
- Eligiusz J. Zlotkiewicz, Marie Curie - Skłodowska University, Lublin, Poland.
- Dumitru Ripianu, Institutul de calcul, str. Republicii nr. 37, Cluj.
- M. H. Schwartz, 37 rue Purre Nicole, Parin 5^{ème}, France.
- Włodzimierz Szwarz, Technical Univ., Wrocław, Poland.
- Ștefan Țigan, Institutul de calcul, str. Republicii nr. 37, Cluj.

