

P. MAHEY

Méthodes de décomposition et décentralisation en programmation linéaire

RAIRO. Recherche opérationnelle, tome 20, n° 4 (1986),
p. 287-306

http://www.numdam.org/item?id=RO_1986__20_4_287_0

© AFCET, 1986, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

MÉTHODES DE DÉCOMPOSITION ET DÉCENTRALISATION EN PROGRAMMATION LINÉAIRE (*)

par P. MAHEY ⁽¹⁾

Résumé. — Dans cet article, certains aspects des techniques de décomposition de programmes linéaires à matrices bloc-angulaires sont analysés. Il est montré en particulier comment la décentralisation des décisions optimales ne peut être atteinte avec les schémas de coordination classiques. Un nouvel algorithme de décomposition mixte complètement décentralisé est proposé. Une méthode de sous-gradient permet d'approcher la solution duale et devient un algorithme de point fixe dans une seconde phase pendant laquelle les sous-problèmes échangent directement les informations sans intervention d'un niveau coordinateur.

Mots clés : Décomposition; Programmation linéaire; Décentralisation.

Abstract. — In this paper, some aspects of decomposition techniques for block-angular linear programs are analyzed. In particular, we show how the decentralization of the optimal decisions cannot be performed with the classical coordination schemes. A new decentralized algorithm with mixed allocations is then introduced: a subgradient algorithm approximates the dual solution and is substituted by a fixed-point algorithm in a second step where the subproblems may exchange information directly without need of a coordination level.

Keywords: Decomposition; Linear programming; Decentralization.

1. INTRODUCTION

L'objectif principal des méthodes de décomposition en programmation mathématique est naturellement de réduire la taille du problème, permettant de substituer un gros moyen de calcul central par des calculateurs locaux de taille réduite travaillant en parallèle ou séquentiellement, ou même de manière

(*) Reçu juin 1984.

(¹) Pontificia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, Cx Postal 38063, Rio de Janeiro, Brasil.

interactive suivant le type de coordination adopté. Dans le cas de la programmation linéaire, il devient clair cependant que cet objectif seul ne justifie pas de poursuivre des recherches dans cette voie. Ceci est dû au fait que les techniques modernes de traitement des matrices creuses développées de concert avec la commercialisation intense des grands ordinateurs depuis les années 60 ont permis la construction de codes de programmation linéaire, comme par exemple, MPSX d'I.B.M., capables de résoudre « pratiquement tous » les problèmes. De tels efforts n'ont pas été fournis pour construire des codes de décomposition. Par exemple, bien que la méthode de Dantzig et Wolfe [6] date de 1960, le premier code de haut niveau n'a vu le jour qu'en 1981 [10].

On peut cependant avancer d'autres objectifs justifiant des recherches sur les méthodes de décomposition en programmation linéaire. C'est tout d'abord la *décentralisation* à la fois des informations et des décisions optimales qui doit permettre aux sous-systèmes de préserver leur autonomie jusqu'à la solution optimale globale et cela, sans l'aide d'un niveau de coordination centralisateur. C'est dans ce but que sera introduit dans la dernière partie un nouvel algorithme de décomposition. Mis à part le gain introduit par la simplification de la coordination, l'objectif de décentralisation a des conséquences intéressantes sur l'organisation du processus décisionnel, dans le cas des systèmes économiques notamment (cf. [1] et [11] par exemple).

Une autre justification importante de la décomposition est la possibilité de découpler des sous-systèmes de natures différentes ou même de poids différents dans le bilan global des contraintes du système.

Dans cet article, un certain nombre d'algorithmes types seront présentés en partant des schémas de décomposition classiques que sont la décomposition par les prix et la décomposition par les ressources pour aboutir à un algorithme original de décomposition mixte totalement décentralisé et dans lequel les sous-problèmes échangent directement les informations sans l'aide d'un niveau supérieur de coordination.

L'étude sera limitée au cas des modèles à structures bloc-angulaires. Le problème global s'écrit :

$$\begin{aligned} & \text{Minimiser } \sum_{i=1}^n c_i x_i \\ (\text{P0}) \quad & \left\{ \begin{array}{l} \text{sous les contraintes: } \sum_{i=1}^n A_i x_i = b_0 \\ x_i \in S_i = \{ x_i \in R^{n_i} \mid B_i x_i = b_i, x_i \geq 0 \}, \forall i = 1, \dots, n \end{array} \right. \end{aligned} \quad (1)$$

$$(2)$$

avec

$$\begin{aligned} x_i &\in R^{n_i}, & i = 1, \dots, n \\ b_0 &\in R^{m_0} \\ b_i &\in R^{m_i}, & i = 1, \dots, n. \end{aligned}$$

Hypothèses initiales :

- $\forall i = 1, \dots, n :$
 B_i est de rang plein.
 S_i est borné.
- La solution optimale de (P0), $x^* = [x_1^*, \dots, x_n^*]$ existe, est unique et non dégénérée.

Cette dernière hypothèse implique que la solution optimale duale de (P0) est unique et non dégénérée. On notera p^* le vecteur des multiplicateurs optimaux associés aux contraintes de couplage (1).

Notons encore que, par convention, les vecteurs associés aux coûts et aux multiplicateurs sont considérés comme des vecteurs lignes quand ils interviennent dans les produits scalaires avec des vecteurs de variables primales (pour éviter les notations surchargées).

2. DÉCOMPOSITION PAR LES PRIX

Ce type de décomposition crée un niveau coordinateur qui ajuste itérativement un vecteur de « prix » associé aux contraintes de couplage et introduit dans la fonction critère de chaque décideur local. En termes de programmation mathématique, c'est une méthode duale consistant à traiter les contraintes couplantes (1) par relaxation lagrangienne.

Le lagrangien de (P0) s'écrit alors :

$$L(x, p) = \sum_{i=1}^n c_i x_i + p \left(\sum_{i=1}^n A_i x_i - b_0 \right). \quad (3)$$

Pour un vecteur de prix \bar{p} fixe, le problème se décompose en n sous-problèmes :

$$(P1)_i \left\{ \begin{array}{l} \text{Minimiser } (c_i + \bar{p} A_i) x_i \\ x_i \in S_i. \end{array} \right.$$

Soient $X_i(\bar{p})$ l'ensemble des solutions optimales de (P1)_i et $h_i(\bar{p})$ la valeur optimale du critère. Le coordonnateur doit donc rechercher le point-selle du lagrangien, c'est-à-dire résoudre le problème dual :

$$(P2) \left\{ \begin{array}{l} \text{Maximiser } h(p) = \sum_{i=1}^n h_i(p) - pb_0 \\ p \in R^{m_0}. \end{array} \right.$$

On montre facilement que h est une fonction concave, linéaire par morceaux et que le sous-différentiel de h au point p , noté $\partial h(p)$, est donné par :

$$\partial h(p) = \left\{ g \in R^{m_0} / g = \sum_{i=1}^n A_i x_i - b_0, x_i \in X_i(p), i=1, \dots, n \right\} \quad (4)$$

Il est bien connu que, en général, h n'est pas différentiable en p^* , solution optimale, et par conséquent, les sous-problèmes ne pourront être décentralisés. On entend ici par décentralisation la capacité de chaque sous-système de calculer et identifier sa propre décision optimale en résolvant (P1)_i avec $\bar{p} = p^*$.

On peut envisager deux classes d'algorithmes pour résoudre (P2) :

2.1. La méthode des sous-gradients (A 1)

A l'itération t , elle consiste à résoudre chaque sous-problème pour $p = p^t$. Le niveau coordonnateur calcule alors :

$$p^{t+1} = p^t + \varepsilon_t g^t / \|g^t\| \quad \text{où } g^t \in \partial h(p^t) \quad (5)$$

g^t n'est pas nécessairement une direction de montée pour h , mais c'est une direction de descente pour la fonction distance à l'optimum. Le choix du pas ε_t est assez délicat et on se reportera au livre de Minoux [18] pour une discussion complète sur ce sujet. En résumé, on peut espérer une convergence géométrique de l'algorithme (5) mais avec un taux généralement proche de l'unité.

2.2. L'algorithme des plans de coupe (A 2)

A l'itération t , le coordonnateur utilise les hyperplans supports associés aux sous-gradients calculés aux itérations précédentes pour approcher la

fonction h . Il résoud alors le programme linéaire suivant :

$$(P3) \left\{ \begin{array}{l} \text{Maximiser } \sigma \\ \sigma \leq h(p^l) + g^l \cdot (p - p^l), \quad l = 1, \dots, t \\ \text{où} \\ g_l \in \partial h(p^l). \end{array} \right.$$

Observons qu'il est nécessaire au début des itérations de borner p pour garantir une solution à distance finie de (P3).

Si on élimine à chaque itération les hyperplans inactifs, on retrouve sous une forme duale l'algorithme de décomposition de Dantzig-Wolfe [4]. L'algorithme converge en un nombre fini d'itérations vers la solution p^* et de plus, la solution duale optimale de (P3) fournit les poids des points extrêmes associés à chaque g^l dans une solution primale admissible pour (P0) donc permet de calculer la solution optimale x^* . Depuis sa publication, cette méthode a été appliquée à de nombreux problèmes réels, fournissant dans l'ensemble un bilan plutôt pessimiste quant à ses performances numériques (cf. [7], p. 84 par exemple).

On peut expliquer en partie la lenteur de la convergence par l'observation suivante : les solutions successives (σ^{t+1}, p^{t+1}) sont telles que σ^{t+1} décroît de façon monotone vers $h(p^*)$, mais p^{t+1} présente des oscillations instables autour de p^* , comme le montre la figure 1.

On peut éventuellement remédier à ce défaut soit :

- par sous-relaxation : Notons $p^{t+(1/2)}$ la solution de (P3). Alors :

$$p^{t+1} = \varepsilon p^{t+(1/2)} + (1 - \varepsilon) p^t$$

- soit en limitant p autour de p^t dans (P3). C'est l'idée utilisée par Charreton [4] ainsi que dans la méthode Boxstep [16].

2.3. Un algorithme composite (A 3)

Plus récemment, l'auteur a expérimenté un algorithme combinant A-1 et A-2 [15].

- Phase 1 : Algorithme A-1 pendant T_1 itérations.
- Phase 2 : Algorithme A-1 pendant T_2 itérations. Les sous-gradients g^l , $T_1 + 1 \leq l \leq T_2$, sont gardés en mémoire.
- Phase 3 : On monte et on résoud le problème (P3) de l'algorithme A2 avec les sous-gradients obtenus pendant la phase 2 et en bornant p autour de $p^{T_1+T_2}$.

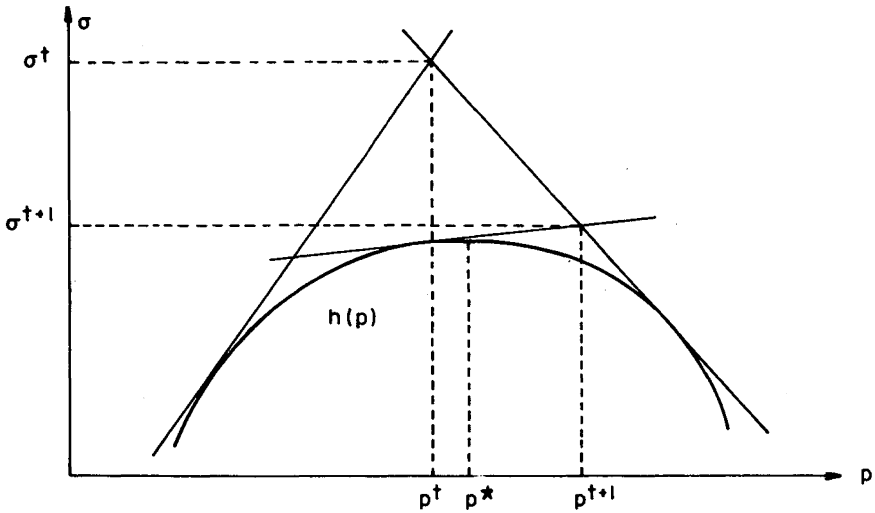


Figure 1. — Instabilité des solutions.
 $\sigma^{t+1} < \sigma^t$ mais $\|p^{t+1} - p^*\| > \|p^t - p^*\|$.

Les oscillations de l'algorithme A1 qui tendent à devenir incontrôlables quand on s'approche de l'optimum sont ici pleinement utilisées, car elles permettent d'identifier les points extrêmes actifs dans la solution optimale. Le nombre d'itérations du type Dantzig-Wolfe peut être ainsi considérablement réduit. Des valeurs typiques de T_1 et T_2 sont

$$T_1 = m_0 \quad \text{et} \quad T_2 = \frac{3m_0}{2}.$$

3. DÉCOMPOSITION PAR LES RESSOURCES

Ce type de décomposition met en jeu n vecteurs u_i , $i = 1, \dots, n$, représentant une allocation *a priori* des seconds membres des contraintes couplantes (1) aux sous-systèmes. L'admissibilité de l'allocation impose la contrainte :

$$\sum_{i=1}^n u_i = b_0. \quad (6)$$

Chaque sous-système doit alors résoudre :

$$(P4)_i \left\{ \begin{array}{l} \text{Minimiser } c_i x_i \\ A_i x_i = u_i \\ x_i \in S_i \end{array} \right.$$

Soit $v_i(u_i)$ la valeur optimale du coût et $\Pi_i(u_i)$ l'ensemble des multiplicateurs optimaux associés aux allocations u_i . Il est clair que le dual de $(P4)_i$ a toujours une solution [la solution duale optimale de $(P0)$]. Donc si $(P4)_i$ n'a pas de solution, $\Pi_i(u_i)$ sera le cône engendré par les rayons extrémaux optimaux dans le dual de $(P4)_i$ et $v_i(u_i) = +\infty$.

Posons $u = (u_1, \dots, u_n)$ et définissons le problème de coordination comme :

$$(P5) \left\{ \begin{array}{l} \text{Minimiser } v(u) = \sum_{i=1}^n v_i(u_i) \\ \sum_{i=1}^n u_i = b_0. \end{array} \right.$$

On montre facilement que v est une fonction convexe au sens large, linéaire par morceaux et que :

$$\partial v(u) = \{ \pi = (\pi_1, \dots, \pi_n) / -\pi_i \in \Pi_i(u_i), i = 1, \dots, n \} \quad (7)$$

pour u tel que $v(u)$ est finie.

Comme dans le cas précédent, v n'est, en général, pas différentiable au point $u^* = (u_1^*, \dots, u_n^*)$, où $u_i^* = A_i x_i^*$. Les sous-systèmes n'ont donc aucun moyen d'identifier la solution optimale et par conséquent, on ne peut décentraliser totalement les décisions par cette méthode.

Nous présentons ci-dessous deux types d'algorithmes notés $(A4)$ et $(A5)$ pour résoudre $(P5)$ qui sont analogues aux algorithmes $(A1)$ et $(A2)$ du paragraphe 2.

3.1. Algorithme de sous-gradient projeté $(A4)$

$$u' \in \Omega_0 = \left\{ u = (u_1 \dots u_n)^T \quad \sum_{i=1}^n u_i = b_0 \right\} \quad \text{et} \quad \pi' \in \partial v(u').$$

On doit projeter orthogonalement π' sur

$$\Omega = \left\{ u = (u_1 \dots u_n)^T \quad \sum_{i=1}^n u_i = 0 \right\}.$$

On remarque alors que le vecteur $\tilde{\pi}^t = \text{Proj}_\Omega \pi^t$, où Proj_Ω est l'opérateur de projection orthogonale sur Ω , s'écrit :

$$\tilde{\pi}_i^t = \pi_i^t - \frac{1}{n} \left(\sum_{i=1}^n \pi_i^t \right), \quad i = 1, \dots, n. \quad (8)$$

Par conséquent, $\tilde{\pi}^t$ est un sous-gradient de la fonction :

$$\varphi(u) = v(u) + \chi_{\Omega_0}(u)$$

où $\chi_{\Omega_0}(\cdot)$ est la fonction indicatrice de Ω_0 .

Il est donc naturel de choisir le pas de façon à converger vers le minimum de φ :

$$u^{t+1} = u^t - \varepsilon_t \frac{\tilde{\pi}^t}{\|\tilde{\pi}^t\|} \quad (9)$$

et $\varepsilon_t > 0$ satisfait des règles analogues à celles utilisées pour l'algorithme (A 1).

3.2. L'algorithme des plans de coupe (A 5)

Là encore, le coordonnateur résout un programme linéaire où on a rajouté la contrainte d'admissibilité :

$$\begin{aligned} & \text{Min } w \\ & w \geq \sum_{i=1}^n [v_i(u_i^l) + \pi_i^l \cdot (u_i - u_i^l)], \\ \text{(P 6)} \quad & l = 1, \dots, t, \quad \pi^l \in \partial v(u^l) \\ & \sum_{i=1}^n u_i = b_0. \end{aligned}$$

C'est la forme duale de l'algorithme de Tenkate [24].

On peut faire ici les mêmes observations qu'au paragraphe 2. 2.

3.3. Un algorithme de sous-gradient primal-dual (A 6)

L'auteur a construit un algorithme couplant les algorithmes de sous-gradients A 1 et A 4 [14].

On définit deux suites de pas $\{\varepsilon_{1t}\}$ et $\{\varepsilon_{2t}\}$:

$$\left. \begin{array}{l} \text{où} \\ \varepsilon_{1t} = \rho_{1t} \frac{\bar{h} - h(p^t)}{\|g^t\|} \\ 0 < \rho_{1t} < 2, \quad g^t \in \partial h(p^t) \\ \varepsilon_{2t} = \rho_{2t} \frac{\bar{v} - v(u^t)}{\|\tilde{\pi}^t\|} \\ \text{où} \\ 0 < \rho_{1t} < 2, \quad \tilde{\pi}^t = \text{Proj}_{\Omega} \pi^t \quad \text{et} \quad \pi^t \in \partial v(u^t) \end{array} \right\} \quad (10)$$

On choisira

$$\bar{h} = \min_{l \leq t} v(u^l)$$

et

$$\bar{v} = \max_{l \leq t} h(p^l).$$

A chaque itération on doit, en principe (de nombreuses variantes simplificatrices sont indiquées dans [14]), résoudre pour chaque sous-système les deux sous-problèmes (P1) et (P4). \bar{h} et \bar{v} sont respectivement des approximations par excès et par défaut de la valeur optimale $v(u^*) = h(p^*)$. Si $|\bar{h} - \bar{v}| \rightarrow 0$, on n'a pas à diminuer les coefficients ρ_1 et ρ_2 comme il est conseillé dans [9]. Toutefois la convergence de cet algorithme n'a pu être formellement établie que si $\bar{h} = \bar{v} = v(u^*) = h(p^*)$, c'est-à-dire dans le cas d'une méthode dite de relaxation (cf. [9] par exemple).

Observation : Tous les algorithmes décrits aux paragraphes 2 et 3 peuvent être transposés dans le cadre du problème dual de (P0). On aurait alors une matrice de contraintes bloc-angulaire avec variables de couplage. On obtiendrait des algorithmes souvent classés comme méthodes de partitionnement. L'algorithme de Dantzig-Wolfe (A2) deviendrait alors la méthode de Benders [2] qui a fait l'objet d'un grand nombre d'études et applications dont nous ne nous servons pas ici.

4. UN ALGORITHME DE DÉCOMPOSITION DÉCENTRALISÉ

4.1. Analyse de la dégénérescence des sous-problèmes

On a vu que les deux types de coordination précédents ne pouvaient permettre la décentralisation totale des décisions optimales locales. Montrons comment ce fait est lié à des dégénérescences introduites dans les sous-problèmes par la décomposition.

Avec les hypothèses faites au départ, on sait que la solution optimale contient exactement $\left(m_0 + \sum_{i=1}^n m_i\right)$ variables de base strictement positives.

Prenons alors les sous-problèmes $(P1)_i$ associés à la décomposition par les prix. Leur dimension étant m_i , la solution de base constituée par les n solutions locales contiendra au plus $\sum_{i=1}^n m_i$ variables positives. D'autre part, comme la solution optimale globale de $(P0)$ satisfait les conditions d'optimalité locales de chaque $(P1)_i$, on peut affirmer qu'une partie des sous-problèmes (au moins un et au plus m_0) possèdent une infinité de solutions et par conséquent ne peuvent être décentralisés. On est donc en présence d'une dégénérescence duale.

Dans le cas des sous-problèmes $(P4)_i$, la dimension de la solution de base constituée par les n solutions locales est

$$\sum_{i=1}^n (m_i + m_0) = \sum_{i=1}^n m_i + nm_0.$$

A l'optimum, il y aura donc nécessairement $(n-1)m_0$ variables dégénérées (en base et nulles) et cette dégénérescence portera sur au moins $n-1$ sous-problèmes.

On s'aperçoit donc que, si l'on veut maintenir les sous-problèmes linéaires ⁽²⁾, un algorithme décentralisé devra allouer exactement m_0 contraintes dans les sous-problèmes. C'est dans ce but que nous allons introduire une partition des contraintes de couplage en n sous-ensembles (un certain nombre de ces sous-ensembles peuvent être vides). L'allocation consistera alors en une bijection entre ces n sous-ensembles disjoints et les sous-problèmes. Chaque bloc A_i

⁽²⁾ Jennergren [11] a proposé un algorithme décentralisé où des perturbations quadratiques sont introduites dans les coûts des sous-problèmes pour assurer la convexité stricte du lagrangien.

est partitionné en n sous-blocs, de même pour le second membre b_0 :

$$A_i = \begin{bmatrix} A_{1i} \\ \vdots \\ A_{ii} \\ \vdots \\ A_{ni} \end{bmatrix}, \quad b_0 = \begin{bmatrix} b_{01} \\ \vdots \\ b_{0i} \\ \vdots \\ b_{0n} \end{bmatrix}$$

Les contraintes $\sum_{k=1}^n A_{ik} x_k = b_{0i}$ seront donc allouées au sous-problème $n^\circ i$, qui recevra des allocations de prix associées aux $(n-1)$ blocs de contraintes restants.

La preuve de l'existence d'une partition qui mène à des solutions locales non dégénérées dépend du résultat suivant :

LEMME 1 : *Étant donnée une matrice carrée non singulière et une partition quelconque des colonnes (resp. des lignes) de cette matrice, il existe au moins une partition des lignes (resp. des colonnes) faisant apparaître des blocs diagonaux non singuliers.*

Démonstration : Nous allons démontrer ce résultat pour le cas d'une partition en deux sous-ensembles, le cas général se déduisant immédiatement par induction. Soit B une matrice carrée régulière d'ordre n .

On a :

$$\det B = \sum_{i \in I} (-1)^{\sigma_i} b_1^i \dots b_n^i$$

ou I est l'ensemble des permutations des lignes de B , σ_i la signature de la permutation i et b_j^i le j -ième élément de la diagonale de la matrice B^i obtenue par la permutation i . Étant donnée une partition des colonnes de B , par exemple les k premières colonnes et $n-k$ dernières, on peut reconstituer toutes les permutations des lignes de B en séparant toutes les partitions distinctes en k lignes et $n-k$ lignes et pour chaque partition en énumérant toutes les permutations des k premières lignes d'une part et des $n-k$ dernières d'autre part. Pour chaque partition r , on obtient les produits $\det B_1^r \cdot \det B_2^r$ (voir fig. 2). On a donc :

$$\det B = \sum_{r \in R} (-1)^{\sigma_r} \det B_1^r \cdot \det B_2^r$$

ou R est l'ensemble des partitions des lignes de B en k lignes et $n-k$ lignes et σ_r est la signature de la permutation associée à B_1^r et B_2^r . En conséquence,

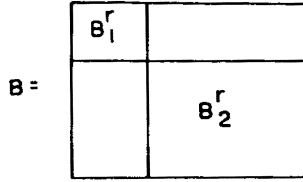


Figure 2

si $\det B \neq 0$, il existe au moins une partition r telle que $\det B_1^r \neq 0$ et $\det B_2^r \neq 0$.

C.Q.F.D.

4.2. Décomposition décentralisée

Étant donnée une partition des contraintes en n sous-ensembles, réécrivons le problème (P0) en rajoutant les variables $u = (u_1, \dots, u_n)$ telles que $u \in R^{m_0}$ et u_i a la même dimension que b_{0i} :

$$(P7) \left\{ \begin{array}{l} \text{Minimiser } \sum_{i=1}^n c_i x_i \\ A_{ii} x_i - u_i = 0 \\ u_i + \sum_{k \neq i} A_{ik} x_k = b_{0i} \quad i = 1, \dots, n \\ x_i \in S_i \end{array} \right.$$

On effectue alors une relaxation lagrangienne des contraintes de couplage :

$$u_i + \sum_{k \neq i} A_{ik} x_k = b_{0i} \quad i = 1, \dots, n.$$

On obtient alors un lagrangien en x, p et u :

$$L(x, p, u) = \sum_{i=1}^n [c_i x_i + p_i (\sum_{k \neq i} A_{ik} x_k + u_i - b_{0i})]$$

Pour un couple (\bar{p}, \bar{u}) , on a les sous-problèmes :

$$(P8)_i \left\{ \begin{array}{l} w_i(\bar{p}, \bar{u}_i) = \min [c_i x_i + \sum_{k \neq i} \bar{p}_k A_{ki} x_k] \\ A_{ii} x_i = \bar{u}_i \\ x_i \in S_i \end{array} \right.$$

Le coordinateur doit donc rechercher un point-selle de la fonction :

$$w(p, u) = \sum_{i=1}^n [w_i(p, u_i) + p_i(u_i - b_{0i})]$$

$w(p, u)$ est convexe en u et concave en p . A l'intérieur du domaine de w , on peut définir le sous-différentiel de w en (p, u) (Rockafellar, p. 374, [23]) :

$$\partial w(p, u) = \partial_p w(p, u) \times \partial_u w(p, u).$$

Soient $X_i(p, u)$ et $\Pi_i(p, u)$, les ensembles des solutions optimales primales et duales de $(P8)_i$. On a alors :

$$\left. \begin{aligned} \partial_{u_i} w(p, u) &= \{ -\pi_i + p_i \mid \pi_i \in \Pi_i(p, u) \} \\ \partial_{p_i} w(p, u) &= \{ \sum_{k \neq i} A_{ik} x_k + u_i - b_{0i} \mid x_k \in X_k(p, u) \}. \end{aligned} \right\} \quad (11)$$

On a vu que la possibilité de décentralisation requiert la non dégénérescence des sous-problèmes. Montrons qu'il existe une décomposition telle que chaque sous-problème $(P8)_i$ a une solution unique non dégénérée au voisinage de l'optimum.

THÉORÈME 1 : *Il existe une décomposition des contraintes de couplage telles que chaque sous-problème $(P8)_i$ a une solution unique non dégénérée pour $p_k = p_k^*$, $\forall k \neq i$ et $u_i = u_i^*$ où $u_i^* = b_{0i} - \sum_{k \neq i} A_{ik} x_k^*$. Cette solution est égale à (x_i^*, p_i^*) et (p^*, u^*) est l'unique point-selle de $w(p, u)$.*

Démonstration : Considérons la base optimale du problème (P0). Appliquons le lemme 1 et observons que la partition i des lignes doit nécessairement inclure les lignes de B_i . Soit $\Delta_i = \begin{bmatrix} \tilde{A}_{ii} \\ \tilde{B}_i \end{bmatrix}$ le bloc diagonal n° i de la partition choisie telle que Δ_i régulier. Donc x_i^* est un point extrême non dégénéré (par construction de la partition) du polyèdre défini par $(P8)_i$. Il est clair que x_i^* satisfait les conditions d'optimalité de $(P8)_i$. De plus, si on considère le dual de $(P8)_i$, on peut observer que (p_i^*, π_i^*) est un point extrême non dégénéré optimal, ce qui garantit l'unicité de la solution primale et réciproquement.

On ne pourra donc parler de décentralisation que dans un voisinage de l'optimum où il est possible de caractériser une partition des contraintes

favorable. On doit donc se poser deux problèmes distincts :

- (i) Comment obtenir une telle partition ?
- (ii) Une fois connue une partition favorable, comment obtenir le point selle (p^*, u^*) ?

Répondons en premier lieu à la deuxième question : la forme particulièrement symétrique des sous-problèmes $(P 8)_i$ suggère le schéma itératif suivant :

A l'itération t , on résout les sous-problèmes $(P 8)_i$ séquentiellement. Appelons x_i^t la solution optimale primale et π_i^t le vecteur des multiplicateurs optimaux associé à l'allocation \bar{u}_i du sous-problème n° i . Ces informations sont transmises aux sous-problèmes suivants qui modifient leurs allocations primales et duales par :

$$\text{Sous-problème } \underbrace{\quad}_{i+1} \left. \begin{array}{l} \bar{p}_k = \pi_k^t, \quad k = 1, \dots, i \\ \bar{p}_k = \pi_k^{t-1}, \quad k = i+2, \dots, n \\ \bar{u}_{i+1} = b_{0, i+1} - \sum_{k=1}^i A_{i+1, k} x_k^t \\ - \sum_{k=i+2}^n A_{i+1, k} x_k^{t-1} \end{array} \right\} \quad (12)$$

Observons que ce schéma itératif est équivalent à résoudre $\partial w(p, u) = 0$ par un algorithme de point-fixe. Plus précisément, si on considère la base optimale de $(P 0)$ dont les lignes sont arrangées de manière à faire apparaître les bases optimales de chaque sous-problème $(P 8)_i$, on peut en déduire que l'algorithme proposé équivaut à appliquer une méthode de Gauss-Seidel par blocs à cette matrice. Sur la figure 3, on a noté \tilde{A}_{ij} et \tilde{B}_i les parties basiques des sous-matrices A_{ij} et B_i .

Par conséquent, l'algorithme converge vers le point-fixe (p^*, u^*) si le rayon spectral de la matrice $(D - E)^{-1} F$ est inférieur à l'unité.

Le principal avantage de cette décomposition est donc l'élimination du niveau coordonnateur dans le voisinage de la solution optimale où la partition des contraintes favorable a pu être identifiée. Cette méthode se rapproche en particulier de la méthode de décomposition par « prédiction des interactions » introduite par Mesarovic *et al.* [17] et également étudiée par Cohen dans le cas linéaire-quadratique [5]. On peut également l'insérer dans le cadre des méthodes d'optimisation par relaxation ([3, 13]).

Le problème de la recherche d'une partition des contraintes favorable non seulement à l'application du théorème 1 mais aussi à la convergence des

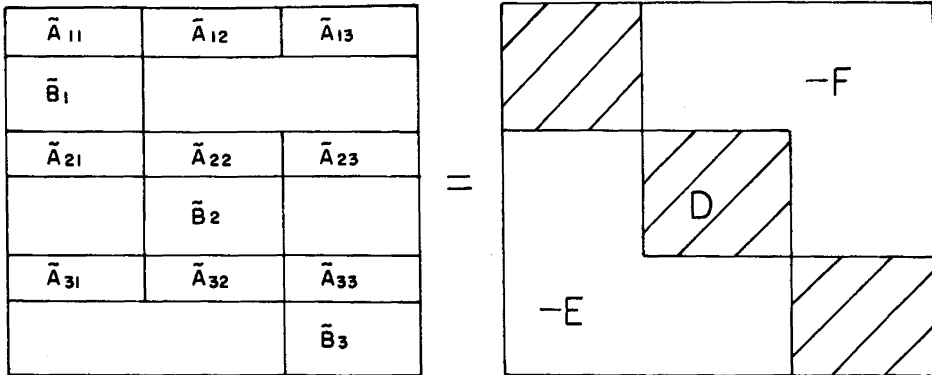


Figure 3. — Forme bloc-diagonale de la base optimale.

itérations de point fixe est crucial pour construire des algorithmes compétitifs avec les méthodes classiques. Nous y répondons partiellement dans la section suivante avec l'introduction de quelques règles heuristiques qui se sont avérées suffisantes sur les exemples numériques traités.

4.3. Un algorithme d'allocations mixtes ⁽³⁾ (A 7)

Dans cet algorithme, l'allocation des contraintes aux sous-problèmes est réalisée progressivement dans une première phase, l'algorithme (A 1) est appliqué à partir d'un vecteur de prix initial p^0 . Le but de cette première phase est d'une part d'approcher la solution duale p^* et d'autre part de permettre l'identification des contraintes à allouer à certains sous-problèmes dans une deuxième phase. Comme on l'a déjà observé pour l'algorithme (A 3), les petits pas dans la direction des sous-gradients successifs permettent d'identifier les points extrêmes actifs dans la solution optimale :

Supposons par exemple que le sous-problème (P 1)_i oscille entre les deux points extrêmes x_i^1 et x_i^2 , les solutions \bar{x}_k , $k \neq i$, des autres sous-problèmes restant fixes et soient g^1 et g^2 les sous-gradients de h avant et après le pivotage. S'il existe une contrainte $r \in \{1, \dots, m\}$ telle que :

$$g_r^1 g_r^2 < 0 \quad (14)$$

⁽³⁾ Une première version de cet algorithme a été présentée par l'auteur au Journées d'Optimisation de Montréal, 1983.

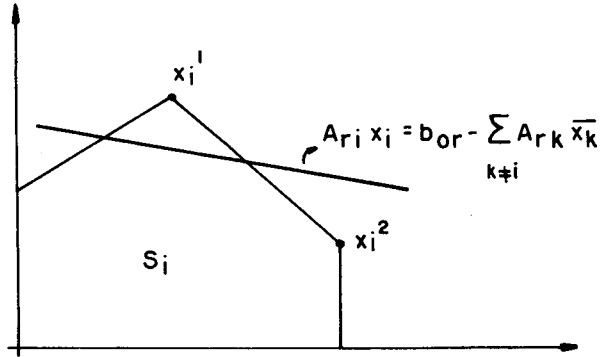


Figure 4

alors, l'hyperplan d'équation

$$A_{ri} x_i = b_{0r} - \sum_{k \neq i} A_{rk} \bar{x}_k$$

sépare strictement x_i^1 et x_i^2 (fig. 4).

L'allocation de la contrainte r au sous-problème i permet donc d'éliminer la dégénérescence détectée par les pivotages successifs. La dimension de l'espace des variables duales traitées par l'algorithme (A 1) diminue jusqu'à ce que toutes les contraintes soient allouées. Dans le cas où plusieurs contraintes satisfont (13), on a choisi la règle heuristique suivante :

Allouer la contrainte r telle que $\min g_s^1 g_s^2 = g_r^1 g_r^2 < 0$. On montrera dans un prochain travail comment cette règle permet d'approcher la notion de bloc-diagonale dominante (cf. [8] et [22]) suffisante pour la convergence de la dernière phase. Nous donnerons ici quelques résultats numériques qui montrent le bon comportement de l'algorithme en ce qui concerne la convergence de la phase finale, dite de décentralisation, où les sous-systèmes échangent séquentiellement les informations primales et duales sans l'aide d'un niveau coordinateur.

4.4. Résultats numériques

L'algorithme (A 7) a été testé sur un modèle de gestion d'atelier avec stocks sur un horizon discrétisé, adapté de [21]. Ce modèle consiste en 7 sous-systèmes contenant les équations de conservation de chaque produit fabriqué tout au long de l'horizon ainsi que les limites sur les niveaux de stocks et les variables de lancements en fabrication. Le couplage consiste en 24 contraintes

inégalités couplant les différents produits. Ces contraintes peuvent être modifiées suivant plusieurs schémas affectant l'intensité du couplage à l'optimum. Nous rapporterons ici trois problèmes-tests correspondant aux couplages suivants :

Problèmes	Nombre de contraintes saturées à l'optimum
1.	4
2.	10
3.	16

L'algorithme (A 7) sera comparé à l'algorithme (A 3) décrit dans [15] qui comporte également trois phases dont nous rappelons l'essentiel ci-dessous :

	A 3	A 7
Phase 1.	Algorithme A 1	Algorithme A 1
Phase 2.	Mise en mémoire des solutions locales	Allocation progressive des contraintes
Phase 3.	Algorithme A 2 (Dantzig-Wolfe)	Itérations de point fixe

L'algorithme A 1 a été implémenté suivant la méthode de Shor (*cf.* [18] par exemple), c'est-à-dire que les pas ε_t dans la direction des sous-gradients successifs sont choisis par :

$$\varepsilon_t = \varepsilon \rho^t$$

avec $\varepsilon = 200$ (estimation de la distance initiale de l'optimum p^* multipliée par X , où X est la condition de la fonction h)
et $\rho = .92$ (estimation de $\sqrt{1 - X^2}$).

On rapporte sur la figure 5 les résultats comparés en nombre d'itérations (une itération correspond à résoudre séquentiellement les sept sous-problèmes). L'algorithme des sous-gradients A 1 est interrompu après T itérations et on a choisi pour chaque problème deux valeurs types de T :

- (i) $T = T_1 = m_0$ où m_0 est le nombre de contraintes saturées à l'optimum;
- (ii) $T = 2 T_1$.

Les itérations des phases 2 et 3 de l'algorithme A 3 sont sommées directement sur la figure 5, l'arrêt correspondant à l'obtention de la solution optimale. La phase 3 de l'algorithme A 7 est interrompue quand la solution

Problemes	Nombre d'Iterations			
	Phase 1	A3 Phase 2 + 3	A7 Phase 2	A7 Phase 3
1	4	6	9	1*
	8	4	5	1*
2	10	18	22	4
	20	11	12	6
3	16	30	20	10
	32	26	18	6

Figure 5

optimale est obtenue (*) ou une précision de la valeur optimale inférieure à 0,5 %.

Il est important de noter que l'algorithme A 7 a convergé dans chacun des cas traités et que le nombre d'itérations effectuées est proche de celui obtenu avec A 3. L'avantage est bien sûr l'élimination du niveau coordinateur dans la phase 3, le coût du programme-maître dans l'algorithme de Dantzig-Wolfe étant généralement élevé. Par contre, c'est un algorithme de prédiction, ce qui signifie que, en dehors de certains cas où le couplage est faible, les solutions obtenues ne sont pas admissibles.

5. CONCLUSIONS

Ainsi, l'idée de décentralisation des décisions peut conduire à une analyse critique de certains algorithmes de décomposition. Une manière naturelle de l'obtenir est d'allouer aux sous-problèmes le nombre exact de contraintes de couplage. A la décomposition horizontale s'ajoute donc une décomposition verticale. Les algorithmes se transforment en un algorithme de décomposition à un niveau et certains aspects de la convergence ont été étudiés par Lhote et Miellou [13] et par Cohen [5] dans le cas de certains problèmes de commande optimale. Dans le cadre de la programmation linéaire, on trouve cette idée dans les travaux de Kydland [12] mais avec une décomposition des contraintes définie à l'avance par la structure particulière du couplage. Enfin Obel [20] et Nurminski [19] ont également proposé des algorithmes d'allocations mixtes en programmation linéaire.

REMERCIEMENTS

L'auteur tient à remercier Michel Minoux dont les remarques et suggestions ont permis d'améliorer les versions précédentes de cet article.

BIBLIOGRAPHIE

1. D. ATKINS, *Managerial Decentralisation and Decomposition in Mathematical Programming*, Op. Res. Quart., vol. 25, n° 4, 1974, p. 615-624.
2. J. F. BENDERS, *Partitioning Procedures for Solving Mixed Variables Programming Problems*, Num. Math., vol. 4, 1962, p. 238-252.
3. J. CEA et R. GLOWINSKI, *Sur des méthodes d'optimisation par relaxation*, RAIRO, R-3, 1973, p. 5-32.
4. R. CHARRETON, *La décentralisation des choix économiques à travers une méthode de résolution de programmes linéaires par décomposition*, RAIRO, R-3, 1973, p. 53-76.
5. G. COHEN, *Décomposition et Coordination en Optimisation Déterministe, Différentiable et Non-différentiable*, Thèse d'État, Paris, 1984.
6. G. B. DANTZIG et P. WOLFE, *The Decomposition Algorithm for Linear Programs*, Econometrica, vol. 29, n° 4, 1960, p. 767-778.
7. Y. DIRICKX et L. P. JENNERGREN, *System Analysis by Multilevel Methods*, J. Wiley, 1979.
8. D. FEINGOLD et R. S. VARGA, *Block Diagonally Dominant Matrices and Generalization of the Gershgorin Circle Theorem*, Pac. J. of Math., vol. 12, 1962, p. 1241-1249.
9. M. HELD, P. WOLFE et H. P. CROWDER, *Validation of Subgradient Optimization*, Math. Prog., vol. 6, 1974, p. 62-88.
10. J. K. HO et E. LOUPE, *An Advanced Implementation of the Dantzig-Wolfe decomposition algorithm for linear programming*, Math. Prog., vol. 20, 1981, p. 303-326.
11. L. P. JENNERGREN, *A Price-schedules Decomposition Algorithm for Linear Programming Problems*, Econometrica, vol. 41, 1973, p. 965-980.
12. F. KYDLAND, *Hierarchical Decomposition in Linear Economic Models*, Man. Sci., vol. 21, n° 9, 1975, p. 1020-1039.
13. F. LHOTE et J. C. MIELLOU, *Algorithmes de décentralisation et de coordination par relaxation en commande optimale*, dans *Analyse et Commande des Systèmes Complexes*, A. TITLI, éd., AFCET, Cepadues éditions, 1979.
14. P. MAHEY, *Decomposition of Large Scale Linear Programs by Subgradient Optimization*, Mat. Aplic. Comp., vol. 1, n° 2, 1982, p. 121-134.
15. P. MAHEY, *A Subgradient Algorithm for Accelerating the Dantzig-Wolfe Decomposition Method*, X Symp. Operations Research, Munich, 1985 (to appear).
16. R. F. MARSTEN, W. HOGAN et J. W. BLANKENSHIP, *The Boxstep Method for Large-scale Optimization*, Op. Res., vol. 23, n° 3, 1975, p. 389-405.
17. M. D. MESAROVIC, D. MACKO et Y. TAKAHARA, *Theory of Hierarchical Multilevel Systems*, A. Press, 1970.
18. M. MINOUX, *Programmation Mathématique-Théorie et Algorithmes*, Dunod, Paris, 1983.
19. E. A. NURMINSKI, *On a Decomposition of Structured Problems*, W.P. 81-31, IIASA, 1981.

20. B. OBEL, *A Note on Mixed Procedures for Decomposing Linear Programming Problems*, Math. Operations Forsch. Statist. Ser. Optimization, vol. 9, n° 4, 1978, p. 537-544.
21. D. POTIER, *Algorithmes de coordination — Applications à la gestion d'unités de production interdépendantes*, Méthodes Numériques d'Analyse des Systèmes, tome 2, Cahiers de l'I.R.I.A. n° 11, 1972.
22. F. ROBERT, *Blocs-H matrices et convergence des méthodes itératives classiques par blocs*, Linear Algebra and its Appl., vol. 2, 1969, p. 223-265.
23. R. T. ROCKAFELLAR, *Convex Analysis*, Princeton U. Press, 1970.
24. A. TENKATE, *Decomposition of Linear Programs by Direct Distribution*, Econometrica, vol. 40, n° 5, 1972, p. 883-898.