

J.-P. LEGENDRE

M. MINOUX

**Une application de la notion de dualité en
programmation en nombres entiers : sélection et
affectation optimales d'une flotte d'avions**

RAIRO. Recherche opérationnelle, tome 11, n° 2 (1977),
p. 201-222

http://www.numdam.org/item?id=RO_1977__11_2_201_0

© AFCET, 1977, 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/>

UNE APPLICATION DE LA NOTION DE DUALITÉ EN PROGRAMMATION EN NOMBRES ENTIERS : SÉLECTION ET AFFECTATION OPTIMALES D'UNE FLOTTE D'AVIONS (*)

par J.-P. LEGENDRE ⁽¹⁾ et M. MINOUX ⁽²⁾

Résumé. — *Le choix de la composition optimale d'une flotte d'avions à moyen terme dans une compagnie aérienne conduit à la résolution de problèmes d'affectation généralisés de grandes dimensions. La méthode classique (programmation linéaire continue + arrondi) est trop coûteuse en temps de calcul et en volume de mémoire pour permettre la résolution interactive à partir d'un terminal temps-partagé. On propose ici une nouvelle approche du problème d'affectation généralisé, basé sur la notion de dualité en programmation discrète.*

Outre sa souplesse d'utilisation et la rapidité des calculs, la méthode fournit de nombreuses solutions entières quasi-optimales ou ϵ -admissibles très utiles en vue d'une analyse de sensibilité. Après avoir décrit les principales méthodes de résolution du problème dual, on présente des résultats de calcul sur un certain nombre d'exemples réels comportant jusqu'à 700 variables entières environ.

On étudie enfin le problème de la recherche des solutions admissibles, et on montre comment les méthodes développées dans cet article peuvent s'y appliquer.

I. INTRODUCTION

I.1. Position du problème

Air France se pose le problème important du renouvellement de sa flotte, donc de l'investissement en avions sur un horizon de 5 à 10 ans. Compte tenu de l'obsolescence commerciale des appareils anciens et de l'évolution des marchés des années à venir, on doit examiner régulièrement l'introduction de nouveaux avions. Ce choix dépend bien entendu du prix d'achat des avions, de la nature du réseau exploité (itinéraires retenus, trafic à prendre), de la composition de la flotte déjà acquise, et des possibilités d'acquisition d'avions nouveaux.

Le modèle d'affectation optimale des avions de la flotte moyen-courrier employé jusqu'ici à Air France [1] est exploité depuis 1972 sur ordinateur en batch. Il utilise l'algorithme du simplexe qui exige trop de mémoire et de temps calcul pour être exploité en temps partagé. La méthode qui fait l'objet de cet article a permis de développer un nouveau projet compatible avec les exigences du temps partagé. La gestion interactive de l'information permet

(*) Reçu décembre 1975.

(¹) Ingénieur à la Compagnie Air France.

(²) Conseiller scientifique du Centre national d'Études des Télécommunications.

de réaliser l'affectation optimale des avions dans de nombreuses hypothèses et de simuler l'évolution de la flotte sur plusieurs années avec un temps de réponse très rapide. De nombreuses modifications de données d'entrées peuvent être effectuées simplement à partir du terminal à la vue des résultats. Les données peuvent être affinées en particulier sur les hypothèses d'évolution de marché et sur la structure des itinéraires envisagés. Un grand nombre de variantes doivent, en effet, être étudiées et comparées entre elles pour étayer le programme d'investissement.

Comme dans [1], l'étude se développe en deux phases :

- la première envisage, pour chaque itinéraire à desservir, les différentes combinaisons d'avions qui permettent d'acheminer le trafic potentiel. On associe à chaque combinaison commercialement réalisable une consommation en avions et une fonction économique qui tient compte des coûts d'exploitation, des coûts d'amortissement et de financement des avions utilisés dans cette combinaison, et de la recette envisagée;

- la seconde phase choisit pour chaque itinéraire une combinaison (et une seule) de manière à respecter les contraintes de consommation maximale en avions pour chacune des flottes envisagées, et de manière à optimiser la fonction économique globale (maximiser le bénéfice ou minimiser les coûts). Il s'agit d'un programme linéaire en nombres entiers qui est résolu en utilisant la notion de dualité en programmation en nombres entiers (*cf.* § II).

I.2. Phase (1) : Génération des combinaisons d'avions pour chaque itinéraire et fonction économique associée

Soient :

$I = (1, \dots, n)$ l'ensemble des types avions en concurrence;

$J = (1, \dots, m)$ l'ensemble des itinéraires à desservir.

Avant de sélectionner la meilleure composition de flotte, on recherche pour chaque itinéraire $j \in J$ et chaque type avion en concurrence $i \in I$ le programme de desserte ij vérifiant les contraintes commerciales qui dégagent le bénéfice brut d'exploitation maximal.

Pour chaque itinéraire $j \in J$ il est possible d'estimer la demande potentielle D_j en passagers sur la période de référence (semaine caractéristique), en général celle de la pointe de l'été qui conditionne le dimensionnement de la flotte.

La part du trafic que l'on peut transporter dépend de l'offre en nombre de sièges sur l'itinéraire, du nombre de vols, ainsi que de la position des vols dans la semaine.

Prenons deux exemples pour illustrer ces contraintes :

- Considérons un itinéraire de trafic « affaire » de 1 000 passagers par semaine, et envisageons la desserte par un avion de 250 sièges : 4 vols hebdomadaires permettent d'acheminer le trafic, ce programme est cependant

peu réaliste, les passagers en effet attendent un service quotidien. On prendra en compte cette contrainte commerciale en imposant que le nombre de vols soit multiple de 7.

— Considérons maintenant un itinéraire à trafic « touristique », on peut estimer la demande D_{j_1} sur les jours ouvrables de la semaine et la demande D_{j_2} du week-end. Dans ce cas on imposera que D_{j_1} soit acheminée par un service quotidien sur les jours ouvrables, donc multiple de 5 vols, et D_{j_2} par un service multiple de 2 vols. Les 2 demandes potentielles D_{j_1} et D_{j_2} peuvent être considérées comme indépendantes et acheminées sur 2 itinéraires j_1 et j_2 distincts.

L'étude de la courbe de demande en passagers $D(t)$ en fonction du temps t sur la semaine permet de caractériser chacun des courants de trafics indépendants sur chaque itinéraire. Dans la suite de l'exposé, ils sont supposés acheminés par des itinéraires distincts.

(a) On donne $\forall j \in J$ le plus petit commun multiple p_j du nombre de vols commercialement réalisables pour acheminer la demande D_j .

$\forall i \in I$, le programme de desserte Y_{ij} appartiendra au domaine \mathcal{D} défini par

$$f_{ij} = kp_j \quad \text{avec } k \text{ entier positif,}$$

où f_{ij} est le nombre de vols du programme de desserte Y_{ij} .

(b) On recherche pour chaque type avion $i \in I$ et itinéraire $j \in J$ le programme de desserte $Y_{ij} \in \mathcal{D}$ qui maximise le bénéfice brut d'exploitation.

Soient :

S_{ij} la capacité disponible de l'avion i en sièges sur l'itinéraire j on tient compte ici du coefficient de remplissage moyen sur l'itinéraire; c_{ij} le coût d'exploitation de l'avion i pour un vol sur l'itinéraire j ; r_j la recette unitaire sur l'itinéraire j ,

on détermine le niveau de fréquences (nombre de vols) f_{ij}^0 (arrondi à l'entier supérieur) qui peut acheminer la demande D_j :

$$(f_{ij}^0 - 1) S_{ij} < D_j \leq f_{ij}^0 S_{ij}.$$

Soit k^0 tel que

$$k^0 p_j \leq f_{ij}^0 \leq (k^0 + 1) p_j;$$

• pour $k \leq k^0$ le programme de desserte $Y_{ij}^k = kp_j$ ne peut transporter toute la demande d'après les formules ci-dessus le bénéfice brut d'exploitation sera

$$B_{ij}^k = kp_j(r_j S_{ij} - c_{ij});$$

• pour $k = k^0 + 1$:

$$B_{ij}^k = D_j r_j - c_{ij} \cdot kp_j;$$

• pour $k > k^0 + 1$; la recette $D_j r_j$ reste constante et le coût d'exploitation augmente avec k .

On retiendra pour chaque type avion $i \in I$, le programme Y_{ij} tel que

$$B_{ij} = \max(B_{ij}^k), \\ k \leq k^0 + 1.$$

I.3. Phase (2) : Problème d'affectation sous contraintes de Flotte

Pour résoudre le problème général, il faut retenir pour chaque itinéraire j une solution et une seule des $|I|$ solutions Y_{ij} (f_{ij} , B_{ij}) construites précédemment, de manière à vérifier les contraintes du nombre d'avions disponibles par type et à maximiser le bénéfice diminué des coûts d'amortissement et de financement des avions.

Soit la variable booléenne x_{ij} telle que

$$\begin{cases} x_{ij} = 1 & \text{si le programme de desserte } Y_{ij} \text{ est retenu,} \\ x_{ij} = 0 & \text{sinon;} \end{cases}$$

le choix d'une solution et une seule pour chaque itinéraire s'écrit alors :

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J.$$

La limitation en nombre d'avions s'exprime en volume d'heures disponibles à affecter sur les itinéraires :

- si h_{ij} est le nombre d'heures affectées (heures de vol + heures d'escale) du type avion i sur l'itinéraire j , la consommation en avions du programme Y_{ij} (f_{ij} , B_{ij}) sera

$$a_{ij} = f_{ij} \cdot h_{ij};$$

- si H_i est le nombre d'heures affectables par période pour l'avion i (heures de vols + temps passé en escale), et N_i le nombre d'avions de type i , la contrainte de limitation de flotte s'écrit :

$$\sum_{j \in J} a_{ij} x_{ij} \leq H_i N_i.$$

On calcule par ailleurs les frais financiers F_i à l'heure affectée d'un avion i d'après son prix d'achat. Ils sont obtenus en divisant l'annuité de coût avion par les heures affectées annuelles. L'annuité découle de la valeur sur le marché de l'avion, et de ses rechanges, calculée à partir d'un taux d'actualisation appliqué à une durée de vie économique (en supposant la valeur de revente réelle en fin de période d'utilisation).

En associant à chaque programme Y_{ij} (B_{ij} , f_{ij}) la fonction économique :

$$c_{ij} = (F_i f_{ij} h_{ij} - B_{ij}).$$

Le problème d'affectation de flotte se formalise de la façon suivante :

$$(P_0) \left\{ \begin{array}{l} \min \sum_{i \in I, j \in J} c_{ij} x_{ij}, \\ \sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \text{ ensemble des itinéraires,} \\ \sum_{j \in J} a_{ij} x_{ij} \leq H_i N_i, \quad \forall i \in I \text{ ensemble des types avions,} \\ x_{ij} \geq 0, \\ x_{ij} \text{ entier } 0 \text{ ou } 1. \end{array} \right.$$

II. FORMULATION MATHÉMATIQUE DU PROBLÈME

Le problème (P₀) appartient à la classe des programmes linéaires en nombres entiers du type :

$$(I) \left\{ \begin{array}{l} \text{Min } z = c \cdot x = \sum_{j=1}^n c_j \cdot x_j, \\ \text{sous les contraintes :} \\ A \cdot x \leq b, \tag{1} \\ \sum_{j \in J_p} x_j = 1 \quad (p = 1, \dots, P), \tag{2} \\ x_j \geq 0 \quad (j = 1, \dots, n), \tag{3} \\ x_j \text{ entier } 0 \text{ ou } 1, \tag{4} \end{array} \right.$$

où $A = (a_{ij})$ est une matrice réelle $m \times n$; $x = (x_1, x_2, \dots, x_n)^T$; $b = (b_1, b_2, \dots, b_m)^T$ seconds membres; $c = (c_1, c_2, \dots, c_n)$ coûts. $J = (1, 2, \dots, n)$ est partitionné en P sous-ensembles disjoints :

$$J = J_1 \cup J_2 \cup \dots \cup J_p.$$

N.B. : Le problème d'affectation classique rentre dans cette formulation. et par suite on peut considérer (I) comme un problème d'affectation généralisé (cf. aussi [18]).

(I) comporte typiquement $n \simeq 1\,000$ variables $P \simeq 100$ contraintes (2) (dites « de choix multiple ») et le nombre m de contraintes (1) est toujours assez faible (4 à 6 contraintes).

Dans ce qui suit, on appellera *solution* de (I) un vecteur x qui vérifie (2), (3) et (4). Une *solution réalisable* est une solution qui vérifie (1).

II.1. Relaxation des contraintes (1)

La difficulté du problème (I) tient essentiellement à la prise en compte des contraintes (1).

En effet, en omettant les contraintes (1), le problème (I) se décompose en P sous-problèmes indépendants du type :

$$(S.P.) \left\{ \begin{array}{l} \text{Min } \sum_{j \in J_p} c_j \cdot x_j, \\ \text{avec } \sum_{j \in J_p} x_j = 1, \\ x_j \geq 0 \end{array} \right.$$

(pour $1 \leq p \leq P$) dont la solution est obtenue en recherchant l'indice j_0 , tel que

$$c_{j_0} = \text{Min} \{ c_j \}_{j \in J_p}$$

et en attribuant à la variable x_{j_0} la valeur 1 (toutes les autres variables x_j , $j \in J_p$ sont nulles).

N.B. : Comme la solution obtenue est toujours entière, la contrainte d'intégrité est inutile dans les sous-problèmes (S.P.).

Cette propriété du problème (I) sera exploitée systématiquement par la suite.

II.2. Nouvelle formulation du problème (I)

Notons X l'ensemble (fini) des points extrêmes du polyèdre convexe défini par les contraintes (2) et (3). D'après ce qui précède X est un ensemble de points de R^n à coordonnées entières (0 ou 1) : $X \subset \{0, 1\}^n$.

Le problème (I) peut alors se mettre sous la forme équivalente :

$$(P) \left\{ \begin{array}{l} \text{Min } z = c \cdot x \\ \text{sous les contraintes :} \\ A \cdot x \leq b \\ \text{et} \\ x \in X. \end{array} \right. \quad (1) \quad (5)$$

Un très grand nombre de problèmes en nombres entiers sont susceptibles d'une telle formulation; citons entre autres :

- recherche d'une borne inférieure de la longueur d'un cycle hamiltonien dans le problème du voyageur de commerce (cf. [2, 3]). Dans ce cas, X est un ensemble de 1-arbres (1-trees);
- problème d'affectation classique (cf. [3, 4]);
- problème du plus court chemin avec contraintes supplémentaires.

Dans ce cas, X est un ensemble de chemins (cf. [10]);

— programmes en nombres entiers généraux formulés à l'aide de la théorie des groupes (cf. [4]).

Tous ces problèmes partagent la particularité suivante : quel que soit le vecteur coût γ , on sait calculer rapidement :

$$\gamma \cdot \bar{x} = \text{Min}_{x \in X} \{ \gamma \cdot x \};$$

III. DUALITÉ EN PROGRAMMATION EN NOMBRES ENTIERS

Nous nous contenterons ici de rappeler les principales notions et les principaux résultats. On pourra se référer également à [4].

III.1. Définition du programme dual

En associant à chaque contrainte i ($1 \leq i \leq m$) de (1) une variable duale $\pi_i \geq 0$ (multiplicateur de Lagrange) on définit la fonction de Lagrange :

$$L(\pi) = -\pi \cdot b + \text{Min}_{x \in X} \{ cx + \pi \cdot A \cdot x \}.$$

La propriété $L(\pi) \leq z^* \forall \pi \in \mathbf{R}^{m+}$ est bien connue [en tout point π , $L(\pi)$ est un minorant de l'optimum z^* de (P)].

Le problème dual de (P) est précisément de trouver le meilleur des minorants de ce type, c'est-à-dire :

$$(D) \quad \begin{cases} L(\pi^*) = \text{Max } L(\pi), \\ \pi \in \mathbf{R}^{m+}. \end{cases}$$

On sait, d'autre part, que si (P) était un programme *convexe* on aurait : $L(\pi^*) = z^*$.

Ce serait le cas, en particulier, si (P) ne comportait pas de contrainte d'intégrité. (D) serait alors le dual d'un programme linéaire continu « ordinaire ».

Cependant, avec les contraintes d'intégrité, on aura en général :

$$L(\pi^*) < z^*.$$

La différence $z^* - L(\pi^*)$ est le « *saut primal-dual* » (« duality gap »).

III.2. Intérêt du programme dual

L'intérêt du programme dual (D) est multiple :

(a) Il est plus facile à résoudre que le primal.

(b) Sa résolution (exacte ou approchée) fournit des bornes inférieures de z^* .

Il peut donc être utilisé de façon systématique dans une méthode arborescente de type S.E.P. (Branch and bound).

(c) La résolution de (D) donne généralement comme résultats intermédiaires des solutions admissibles (entières) de (P) souvent proches de l'optimum, quelquefois optimales (cf. § III.5).

(d) La connaissance d'une solution primale-admissible (optimale ou suboptimale) et d'une solution duale-admissible (optimale ou suboptimale) permet d'encadrer z^* et, par suite, d'interrompre le calcul si l'on se trouve dans les limites d'une tolérance ε fixée.

Nous ne parlerons pas ici de l'utilisation du problème dual (D) au sein d'une procédure de recherche arborescente de z^* . La raison principale en est que le problème (I) est généralement de grandes dimensions (voir introduction) et la méthode S.E.P. serait, dans ce cas, beaucoup trop coûteuse en temps calcul et en encombrement mémoire : d'où impossibilité d'implanter le programme en temps partagé. Par ailleurs, la recherche (à grands frais) d'un optimum exact ne se justifie pas ici, étant donné que les seconds membres b des contraintes (1) sont toujours connus avec une certaine marge d'incertitude. Un bon encadrement (à 0,5 % près par exemple) de z^* est amplement suffisant.

L'approche que nous avons adoptée ici consiste à résoudre le dual (D) soit de façon exacte (cf. chap. V) soit de façon approchée (chap. IV) pour obtenir une borne inférieure de z^* . Par ailleurs, nous verrons que l'on obtient au cours du calcul, des solutions primales-admissibles (donc entières) et il suffit de retenir celle de moindre coût comme borne supérieure de z^* . La méthode fournira donc :

- une bonne solution admissible;
- un encadrement de z^* .

III.3. Propriétés du programme dual

Posons $|X| = K$ et $X = \{x^k\}_{k=1 \dots K}$ (dans le cas du problème (I) les x^k sont les points extrêmes du polyèdre convexe défini par (2) et (3).

Pour $1 \leq k \leq K$, soit

$$L(\pi, x^k) = -\pi \cdot b + c \cdot x^k + \pi \cdot A \cdot x^k.$$

En notant

$$\gamma^k = A \cdot x^k - b, \quad (k = 1, 2, \dots, K)$$

et

$$c^k = c \cdot x^k, \quad (k = 1, 2, \dots, K),$$

il vient

$$L(\pi, x^k) = c^k + \pi \cdot \gamma^k.$$

Et par suite

$$L(\pi) = \underset{k=1, \dots, K}{\text{Min}} \{L(\pi, x^k)\} = \underset{k=1, \dots, K}{\text{Min}} \{c^k + \pi \cdot \gamma^k\}. \quad (6)$$

On voit d'après (6) que la fonction $L(\pi)$ est, dans \mathbf{R}^{m+1} , l'enveloppe inférieure d'une famille (finie) de K hyperplans d'équations :

$$y = c^k + \pi \cdot \gamma^k \quad (k = 1, \dots, K).$$

Il en résulte que $L(\pi)$ est une fonction *concave* (donc continue) de $\mathbf{R}^m \rightarrow \mathbf{R}$.

Pour $\pi \in \mathbf{R}^{m+}$ donné, notons $V(\pi) \subset \{1, 2, \dots, K\}$ l'ensemble des indices l ($1 \leq l \leq K$) tels que

$$c^l + \pi \cdot \gamma^l = \text{Min}_{k=1, \dots, K} \{c^k + \pi \cdot \gamma^k\}.$$

Lorsque $|V(\pi)| = 1$, $L(\pi)$ est *différentiable* au point π et si $V(\pi) = \{k\}$ le vecteur γ^k est le gradient de L en π (noté : $\nabla L(\pi)$).

Lorsque $|V(\pi)| > 1$ la fonction L n'est pas différentiable en π . On généralise alors pour les fonctions concaves la notion de gradient de la façon suivante : on dit que $\gamma \in \mathbf{R}^m$ est un *sous-gradient* de L en π si et seulement si

$$L(\pi') - L(\pi) \leq (\pi' - \pi) \gamma \quad (\forall \pi' \in \mathbf{R}^m).$$

L'ensemble de tous les sous-gradients en π est un ensemble convexe fermé noté $\partial L(\pi)$ et appelé le *sous-différentiel* en π (cf. [11] ou [12]). En tout point π , la fonction $L(\pi)$ possède des *dérivées directionnelles* :

$$L'(\pi, d) = \text{Lim}_{\lambda \rightarrow 0^+} \frac{L(\pi + \lambda d) - L(\pi)}{\lambda} \quad (\forall d \in \mathbf{R}^m)$$

et on peut démontrer la relation fondamentale entre dérivées directionnelles et sous-gradients :

$$L'(\pi, d) = \text{Min}_{\gamma \in \partial L(\pi)} \{d \cdot \gamma\}.$$

On démontre enfin que les γ^l , $l \in V(\pi)$ sont les points extrêmes (générateurs) de $\partial L(\pi)$.

La résolution du programme dual (D) se ramène donc à la recherche du maximum d'une fonction concave non partout différentiable. De nombreuses recherches sont actuellement consacrées à l'optimisation des fonctionnelles non différentiables (cf. [8] et [9] par exemple). Nous verrons d'abord comment on peut généraliser, d'une certaine manière, les méthodes de gradient classiques pour résoudre (D) de façon approchée : méthodes de sous-gradient (§ IV). Au § V, enfin, on exposera une méthode de résolution exacte fortement apparentée au principe de décomposition de Dantzig et Wolfe en programmation linéaire.

IV. RÉOLUTION DU PROBLÈME DUAL PAR DES MÉTHODES DE SOUS-GRADIENT

IV.1. Principe de la méthode

En théorie, seule la connaissance complète de l'ensemble $\partial L(\pi)$ des sous-gradients en π , ($\pi \neq \pi^*$), permettrait de définir une direction de déplacement

augmentante (cf. [13] par exemple). Cependant, dans la plupart des problèmes pratiques (comme le problème (I)), la détermination de $\partial L(\pi)$ – ou, ce qui revient au même, de l'ensemble $V(\pi)$ – serait beaucoup trop coûteuse en temps de calcul. Tout ce que l'on sait faire, en général, c'est calculer un sous-gradient unique $\gamma(\pi) \in \partial L(\pi)$ en chaque $\pi \in \mathbf{R}^{m+}$.

Dans ces conditions, bien sûr, l'utilisation de la direction de déplacement $\gamma(\pi)$ ne garantit pas l'augmentation de L . Cette caractéristique (commune à tous les problèmes « non différentiables ») fait qu'il est également inutile d'optimiser $L(\pi)$ dans la direction du sous-gradient $\gamma(\pi)$. A chaque étape, la valeur du déplacement dans la direction $\gamma(\pi)$ est donc choisie *a priori*.

IV.2. L'algorithme

- (A 1) $\left\{ \begin{array}{l} (a) \text{ A l'étape } 0, \text{ partir d'un point } \pi^0 \in \mathbf{R}^{m+} \text{ et définir une suite de} \\ \text{ nombres } \lambda_j (j = 1, 2, \dots) \text{ (longueurs des déplacements à chaque} \\ \text{ itération } j) \text{ telle que } \lambda_j \rightarrow 0 (j \rightarrow +\infty). \\ (b) \text{ A l'étape } j \text{ déterminer un sous-gradient } \gamma(\pi^j) \in \partial L(\pi^j) \text{ au} \\ \text{ point } \pi^j. \\ (c) \text{ Définir } \pi^{j+1} \text{ par} \\ \qquad \qquad \qquad \pi^{j+1} = \pi^j + \lambda_j \cdot \gamma(\pi^j) \\ \text{ (si } \pi^{j+1} \notin \mathbf{R}^{m+}, \text{ projeter } \pi^{j+1} \text{ sur } \mathbf{R}^{m+}, \text{ retourner en (b).} \end{array} \right.$

Dans l'exemple qui nous intéresse [problème (I)] le sous-gradient $\gamma(\pi^j)$ au point π^j est obtenu en calculant successivement :

- les coûts modifiés $\tilde{c} = c + \pi^j \cdot A$;
- puis, par la méthode exposée en (II.2) :

$$c \cdot x^k = \text{Min}_{x \in X}(\tilde{c} \cdot x);$$

- le sous-gradient est alors

$$\gamma(\pi^j) = \gamma^k = A \cdot x^k - b.$$

Il s'agit de calculs matriciels élémentaires qui sont extrêmement rapides à effectuer sur calculateur électronique. La principale difficulté est plutôt d'ordre théorique : comment choisir les déplacements λ_j de façon à obtenir une convergence satisfaisante ?

IV.3. Étude des conditions de convergence

Polyak [14] a montré que la suite $L(\pi^j)$ converge vers $L(\pi^*)$ sous les seules conditions :

$$\left\{ \begin{array}{l} \lambda_j \rightarrow 0 \quad (j \rightarrow +\infty), \\ \sum_{j=1}^{\infty} \lambda_j = +\infty. \end{array} \right.$$

Cependant, on ne peut rien dire, sous ces seules conditions, sur la vitesse de convergence.

Si on choisit à chaque étape j des λ_j définis par

$$\lambda_j = \rho_j \frac{L(\pi^*) - L(\pi^j)}{\|\gamma(\pi^j)\|^2}, \quad (7)$$

où le coefficient de relaxation ρ_j vérifie

$$0 < \rho_j \leq 2 \quad (\forall j),$$

alors la convergence est géométrique (cf. [15]).

(N.B. : $\|\cdot\|$ désigne la norme euclidienne.)

Évidemment, cela suppose que l'on connaisse $L(\pi^*)$ qui est précisément le résultat cherché !

Si on remplace dans (7) $L(\pi^*)$ par une estimation par défaut $\underline{L} \leq L(\pi^*)$ on montre (cf. [2] et [15]) que la séquence $L(\pi^j)$ converge vers \underline{L} , ou bien on obtient (au bout d'un nombre fini d'étapes) un point π^j vérifiant

$$\underline{L} \leq L(\pi^j) \leq L(\pi^*).$$

Ce dernier résultat découle directement des travaux de Agmon-Motzkin-Schoenberg ([5] et [6]) sur les systèmes d'inéquations linéaires.

Dans [3] Held, Wolfe et Crowder montrent que l'on peut remplacer $L(\pi^*)$ dans (7) par une estimation par excès $\bar{L} > L(\pi^*)$ sans que la convergence de l'algorithme en soit pratiquement affectée, évidemment, dans ce cas, la condition $\lambda_j \rightarrow 0$ ($j \rightarrow +\infty$) oblige à choisir $\rho_j \rightarrow 0^+$ ($j \rightarrow +\infty$).

En pratique, on choisira comme valeur \bar{L} la valeur \bar{z} de z atteinte par la meilleure solution primale réalisable rencontrée dans les étapes précédentes du calcul [c'est évidemment une borne supérieure de z^* , donc de $L(\pi^*)$].

IV.4. Choix des coefficients de relaxation

Étant donnée l'absence de résultats théoriques à ce sujet, on rentre ici dans le domaine de l'heuristique. Held et Karp [2] rapportent une expérience satisfaisante avec la règle suivante :

- $\rho_j = 2$ pendant $2n$ itérations (n est une bonne mesure de la taille du problème);
- puis à chaque étape, diviser par 2 la valeur de ρ_j et le nombre maximal d'itérations jusqu'à atteindre une limite inférieure q fixée du nombre d'itérations;
- diviser par 2 la valeur de ρ_j toutes les q itérations jusqu'à ce que les λ_j soient suffisamment petits ($< \varepsilon$ fixé).

On peut aussi déterminer les coefficients ρ_j de façon *dynamique* en tenant compte à chaque itération, de la progression de la fonction économique $L(\pi^j)$. L'idée est la suivante :

- si $L(\pi^j) > L(\pi^{j-1})$ la fonction a augmenté et on peut dire en un certain sens que le coefficient ρ_j est bien choisi. On utilisera donc la même valeur $\rho_{j+1} = \rho_j$ à l'étape suivante;
- si $L(\pi^j) < L(\pi^{j-1})$ la fonction a diminué; le coefficient de relaxation ρ_j est vraisemblablement trop grand, et l'on diminuera sa valeur pour l'itération suivante (par exemple, on fera $\rho_{j+1} = \rho_j/2$).

Cette règle a donné de bons résultats sur tous les exemples pratiques que nous avons traités.

IV.5. Exemples et résultats

Les développements qui précèdent seront d'abord illustrés sur l'exemple suivant :

$$n = 17, \quad m = 3, \quad p = 4.$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
c =	10	12	18	17	20	22	23	25	29	6	7	8	9	10	27	30	35		
	40	35	30	32	28	95	85	90	80	20	18	15	13	11	100	85	75	≤	b
	75	60	65	55	50	100	110	95	90	35	32	30	28	24	130	105	90	≤	11
	60	56	52	45	48	50	47	42	40	25	24	20	15	16	90	80	60	≤	12
	J ₁				J ₂				J ₃				J ₄						

Pour $\pi^0 = 0$, la solution est : $L(0) = 65$ et correspond a l'optimum de (P) sans contraintes :

$$(x_1 = 1, x_6 = 1, x_{10} = 1, x_{15} = 1).$$

N'ayant pas de solution réalisable au départ, on a pris comme estimation de $L(\pi^*)$ une valeur de 20 % supérieure soit : $\bar{L} = 65 \times 1,2 = 78$. (N.B. : ce choix est principalement guidé par l'expérience.)

Chaque fois qu'une solution réalisable de coût $v < \bar{L}$ est rencontrée, on donne à \bar{L} la nouvelle valeur v (estimation par excès). A partir de π^0 , la suite de points π^j est définie par

$$\pi^{j+1} = \pi^j + \rho_j \frac{\bar{L} - L(\pi^j)}{\|\gamma(\pi^j)\|^2} \gamma(\pi^j),$$

où $\gamma(\pi^j) \in \partial L(\pi^j)$ et où ρ_j est le facteur de relaxation.

Le premier groupe de résultats a été obtenu en faisant

$$\rho_j = 1 \quad \text{pour } 1 \leq j \leq 5, \quad \rho_{j+1} = \rho_j/2 \quad \text{pour } 5 < j \leq 20.$$

On obtient successivement les résultats du tableau I.

T A B L E A U I

Iterations	$L(\pi)$	Solution obtenue	Valeur de \bar{z}	realisable ?	\bar{L}
1	65	x_1, x_6, x_{10}, x_{15}	65	NON	78
2	72,43	x_2, x_6, x_{13}, x_{17}	78	OUI	78
33	69,9	x_1, x_7, x_{10}, x_{15}	66	NON	78
4	73,3	x_2, x_7, x_{10}, x_{17}	76	OUI	76
5	68,6	x_1, x_7, x_{10}, x_{15}	66	NON	76
6	72,86	x_2, x_6, x_{10}, x_{16}	70	NON	76
7	72,89	x_2, x_7, x_{13}, x_{17}	79	OUI	76
8	73,42		71	NON	76
9	72,99	x_2, x_7, x_{13}, x_{17}	79	OUI	76
10	73,59	x_2, x_7, x_{13}, x_{17}	79	OUI	76
11	73,72		71	NON	76
12	73,57	x_2, x_7, x_{13}, x_{17}	79	OUI	76
13	73,68	x_2, x_7, x_{10}, x_{17}	76	OUI	76
14	73,76		71	NON	76
15	73,73	x_2, x_7, x_{10}, x_{17}	76	OUI	76
16	73,77		76	OUI	76
17	73,76		71	NON	76
18	73,77		71	NON	76
19	73,77	x_2, x_7, x_{10}, x_{17}	76	OUI	76
20	73,78		71	NON	76

On voit que la procédure fournit souvent des solutions réalisables. La meilleure trouvée est

$$\left. \begin{array}{l} x_2 = 1 \\ x_7 = 1 \\ x_{10} = 1 \\ x_{17} = 1 \end{array} \right\} \text{de coût} = 76.$$

Par ailleurs, la plus grande valeur de $L(\pi)$ est obtenue à la dernière itération : 73,78 pour

$$\pi_1 = 0,202, \quad \pi_2 = 0,034, \quad \pi_3 = 0,123.$$

L'approximation est excellente, puisque l'optimum calculé par programmation linéaire, est

$$L(\pi^*) = 73,81 \quad (0,05\%).$$

L'encadrement de l'optimum en nombres entiers est donc

$$74 \leq z^* \leq 76 \quad (2,5\%).$$

En réalité, 76 est la solution optimale, et on voit que le problème a un saut primal-dual de

$$z^* - L(\pi^*) = 76 - 73,81 = 2,19.$$

Le second groupe de résultats a été obtenu en déterminant les facteurs de relaxation ρ_j à chaque étape j suivant la règle (dynamique) :

$$\begin{array}{ll} \rho_{j+1} = \rho_j & \text{si } L(\pi^j) \geq L(\pi^{j-1}), \\ \rho_{j+1} = \rho_j/2 & \text{si } L(\pi^j) < L(\pi^{j-1}). \end{array}$$

Au départ on a $\rho_0 = 2$. Les résultats sont consignés dans le tableau II.

TABLEAU I I

Iteration J	$L(\bar{\pi})$	Solution obtenue	Valeur de z	Réalisable ?	ρ_{J+1}
1	65		65	NON	2
2	64,34	x_2, x_9, x_{14}, x_{17}	86	OUI	1
3	73,56		71	NON	1
4	64,94	x_4, x_9, x_{13}, x_{17}	90	OUI	0,5
5	69,71	x_2, x_7, x_{13}, x_{17}	79	OUI	0,5
6	73,12	x_1, x_7, x_{13}, x_{17}	77	OUI	0,5
7	71,17		66	NON	0,25
8	72,35		69	NON	0,25
9	73,51		69	NON	0,25
10	73,53	x_2, x_7, x_{10}, x_{17}	76	OUI	0,25
11	73,08		69	NON	0,12
12	73,44		69	NON	0,12
13	73,75		71	NON	0,12
14	73,61	x_2, x_7, x_{10}, x_{17}	76	OUI	0,06
15	73,76	x_2, x_7, x_{10}, x_{17}	76	OUI	0,06
16	73,62		69	NON	0,03
17	73,70		69	NON	0,03
18	73,77		69	NON	0,03
19	73,78		74	NON	0,03
20	73,74	x_2, x_7, x_{10}, x_{17}	76	OUI	0,01

On voit que l'on obtient dès la troisième itération, une excellente borne inférieure de l'optimum.

Les deux méthodes précédentes se sont avérées à peu près équivalentes à l'usage.

Le tableau III regroupe, par ailleurs, un certain nombre de résultats obtenus sur des problèmes d'affectation de flotte comportant un grand nombre de variables. On remarquera, en particulier, que l'encadrement de l'optimum est généralement excellent.

T A B L E A U III

Nombre de variables	Nombre de contraintes	Nombre d'itérations	Encadrement de l'optimum	Temps de calcul (sec)*
588	147 + 4	14	0.2 %	14
588	143 + 2	40	0.01 %	17
531	147 + 3	35	0.2 %	18
580	145 + 4	35	0.2 %	19
588	147 + 4	25	0.4 %	16
510	102 + 5	35	0.5 %	19
510	102 + 5	30	0.2 %	16
680	102 + 5	50	0.1 %	34
680	102 + 5	29	0.1 %	23
* L'ordinateur utilisé est un IBM 370/155				

IV.6. Solutions ε -admissibles et analyse de sensibilité

Une caractéristique intéressante des méthodes de sous-gradient est de fournir en cours de calcul, des solutions presque admissibles ou ε -admissibles.

On dit que x est une solution ε -admissible de (P) si

$$A_i \cdot x - b_i < \varepsilon |b_i| \quad (\forall i = 1, \dots, m)$$

(A_i désigne la ligne i de A).

Dans l'exemple précédent, la solution :

$$\left. \begin{array}{l} x_2 = 1 \\ x_7 = 1 \\ x_{10} = 1 \\ x_{16} = 1 \end{array} \right\} \text{ de coût } 71$$

rencontrée à l'itération 8 est telle que

$$\varepsilon = \text{Max}_i \frac{A_i x - b_i}{b_i} = \frac{13}{195} = 7\%.$$

De même la solution :

$$\left. \begin{array}{l} x_1 = 1 \\ x_7 = 1 \\ x_{10} = 1 \\ x_{17} = 1 \end{array} \right\} \text{ de coût } 74$$

est telle que

$$\varepsilon = \text{Max}_i \frac{A_i x - b_i}{b_i} = \frac{5}{305} = 1,6\%.$$

La connaissance d'un certain nombre de solutions ε -admissibles permet de mieux étudier la sensibilité de la fonction z^* aux seconds membres des contraintes (1). D'autre part, lorsque les seconds membres sont connus avec une certaine marge d'incertitude, ceci permet d'explorer un ensemble plus vaste de solutions optimales « potentielles ». C'est généralement le cas dans les problèmes d'affectation de flottes.

V. RÉOLUTION DU DUAL PAR PROGRAMMATION LINÉAIRE : MÉTHODE DE DÉCOMPOSITION

Nous donnons ici une méthode de résolution exacte du programme dual (D) par programmation linéaire.

On montre que la méthode n'est autre que l'application du principe de décomposition de Dantzig et Wolfe [16] au programme primal (I) où la contrainte d'intégrité a été relaxée.

V.1. Formulation de (D) en programmation linéaire

D'après (5) le problème dual s'écrit :

$$\text{Max}_{\pi \in \mathbb{R}^{m^+}} L(\pi) = \text{Max}_{\pi \in \mathbb{R}^{m^+}} \text{Min}_{k=1, \dots, K} \{c^k + \pi \cdot \gamma^k\}$$

ce qui équivaut au programme linéaire :

$$(II) \left\{ \begin{array}{l} \text{Max } w \\ \text{sous les contraintes} \\ c^k + \pi \gamma^k \geq w \quad (k = 1, \dots, K). \end{array} \right.$$

(II) possède m variables seulement, mais un nombre énorme de contraintes.

Pour le résoudre, il est classique de passer au dual en associant à chaque contrainte k une variable $\lambda_k \geq 0$:

$$\left\{ \begin{array}{l} \text{Min } \sum_{k=1}^K c^k \lambda_k, \\ - \sum_{k=1}^K \gamma^k \cdot \lambda_k \geq 0, \\ \sum_{k=1}^K \lambda_k = 1, \\ \lambda_k \geq 0 \quad (k = 1, \dots, K). \end{array} \right.$$

Dans le cas du problème (I) on a : $\gamma^k = A \cdot x^k - b$ et par suite ceci équivaut à

$$\text{(III)} \left\{ \begin{array}{l} \text{Min } \sum_{k=1}^K c^k \cdot \lambda_k, \\ - \sum_{k=1}^K (A \cdot x^k) \lambda_k \geq -b, \\ \sum_{k=1}^K \lambda_k = 1, \\ \lambda_k \geq 0 \quad (k = 1, \dots, K) \end{array} \right.$$

qui est exactement le maître-programme associé à (I) (sans les contraintes d'intégrité) dans la méthode de décomposition de Dantzig et Wolfe [16]. Cette propriété est vérifiée chaque fois que dans le problème (P) X est l'ensemble des points extrêmes d'un polyèdre convexe.

V.2. Résolution de (III)

De façon classique, (III) est résolu par *génération de colonnes* (cf. [12, 16]). On travaille alors sur un maître-programme restreint à un petit nombre r de colonnes : $1 \leq k \leq r$.

A une étape donnée, soit $(\bar{\pi}, \bar{\omega})$ la solution duale optimale du maître-programme restreint. On recherche alors la variable λ_{k_0} de coût réduit :

$$c^{k_0} + \bar{\pi} \cdot A \cdot x^{k_0} - \bar{\omega}$$

minimal ce qui revient à calculer la fonction de Lagrange $L(\pi)$; si $\bar{\pi}$ est l'optimum, alors $(c + \bar{\pi} A) x^{k_0} - \bar{\omega} \geq 0$. Sinon on ajoute dans le programme maître restreint, la colonne $A x^{k_0}$ comme $(r+1)$ -ième colonne.

V.3. Intérêt de la méthode pour la résolution de (P)

Lorsque l'optimum n'est pas encore atteint, on a

$$(c + \bar{\pi} \cdot A) x^{k_0} - \bar{\omega} < 0$$

tandis que

$$(c + \bar{\pi} \cdot A) x^k - \bar{\omega} \geq 0, \quad \forall k = (1, \dots, r).$$

Ceci montre que le point $x^{k_0} \in X$ est différent des r points x^1, x^2, \dots, x^r générés jusqu'ici (correspondant aux r colonnes du maître programme restreint).

Ceci est extrêmement intéressant, car contrairement aux méthodes de sous-gradient (§ IV) on ne rencontre jamais deux fois la même solution (entière) de (P).

En particulier, on est certain de ne jamais rencontrer deux fois la même solution admissible (entière) de (P).

Par conséquent l'utilisation de la méthode de décomposition permet une exploration beaucoup plus systématique de l'ensemble des solutions.

Comme dans les méthodes de sous-gradient, la valeur de $L(\bar{\pi})$ à chaque étape est un minorant de l'optimum $L(\pi^*)$ cherché. (on rappelle que ces minorants ne sont pas nécessairement croissants au cours des itérations, cf. [12] par exemple.)

On remarquera aussi que la seule différence avec une méthode de sous-gradient réside dans le calcul de π^j à l'itération j connaissant π^{j-1} à l'itération $(j-1)$: au lieu d'utiliser une formule itérative du type

$$\pi^{j+1} = \pi^j + \lambda_j \cdot \gamma(\pi^j),$$

on calcule π^{j+1} comme solution (duale), d'un programme linéaire (maître-programme restreint).

Dans le cas du problème d'affectation de flotte, ce calcul est extrêmement rapide puisque le maître-programme ne comporte généralement que 4 à 5 contraintes seulement. Comme dans les méthodes de sous-gradient, de nombreuses solutions ε -admissibles sont rencontrées au cours du calcul.

V.4. Exemple

La méthode de décomposition sera illustrée ici sur l'exemple du paragraphe IV.5.

Il est intéressant d'amorcer le processus itératif en commençant par quelques itérations d'une méthode de sous-gradient. Une dizaine d'itérations suffisent ici. On retient seulement les solutions différentes, dans ce cas :

- $(x_1, x_6, x_{10}, x_{15})$ de coût 65;
- $(x_2, x_6, x_{13}, x_{17})$ de coût 78 (réalisable);
- $(x_1, x_6, x_{10}, x_{16})$ de coût 68;
- $(x_2, x_7, x_{10}, x_{16})$ de coût 71;
- $(x_2, x_7, x_{13}, x_{17})$ de coût 79 (réalisable);
- $(x_2, x_7, x_{10}, x_{17})$ de coût 76 (réalisable).

Le détail des itérations est donné dans le tableau suivant :

Itération	$\bar{\pi} \begin{cases} \pi_1 \\ \pi_2 \\ \pi_3 \end{cases}$	\bar{w}	$L(\bar{\pi})$	Solution obtenue	Coût	Réalisable ?
1	0.108 0. 0.195	74,08	73,4	$x_1 \ x_7 \ x_{10} \ x_{16}$	69	NON
2	0.015 0. 0.282	73,94	73,1	$x_1 \ x_6 \ x_{10} \ x_{17}$	73	NON
3	0.090 0. 0.235	73,90	73,3	$x_1 \ x_7 \ x_{10} \ x_{17}$	74	NON
4	0.109 0.055 0.153	73,81	73,81 Optimum continu	$x_1 \ x_6 \ x_{10} \ x_{16}$	68	NON

On observe :

- 1° que la convergence est très rapide;
- 2° que la suite des $L(\bar{\pi})$ n'est pas monotone croissante.
- 3° que les $L(\bar{\pi})$ fournissent cependant des bornes inférieures de qualité (utilisables dans une procédure arborescente par exemple, sans qu'il soit nécessaire que la convergence ait eu lieu).
- 4° Que toutes les solutions rencontrées sont différentes (sauf à la dernière itération, ce qui, d'ailleurs, fournit un test d'arrêt).

L'emploi de la méthode de décomposition semble particulièrement recommandé dans le cas du problème de sélection et d'affectation de flotte où le nombre m de contraintes (1) est très réduit (on n'a à résoudre à chaque itération qu'un programme linéaire de très petite taille).

VI. RECHERCHE DE SOLUTIONS ADMISSIBLES

Il peut arriver que le problème (P) n'ait pas de solution admissible (entière). Généralement, ce sera aussi le cas lorsqu'on relaxe les contraintes d'intégrité de (P). On risque alors de faire un grand nombre d'itérations inutiles avant de s'en apercevoir.

Par conséquent, il est essentiel de développer des méthodes de recherche de solutions admissibles permettant de détecter rapidement le caractère infaisable du problème.

Fondamentalement il s'agit de savoir si le système :

$$(IV) \left\{ \begin{array}{l} \sum_{k=1}^K \gamma^k \cdot \lambda_k \leq 0, \\ \sum_{k=1}^K \lambda_k = 1, \\ \lambda_k \geq 0 \quad (\forall k = 1, \dots, K) \end{array} \right.$$

a une solution ou non.

Pour cela, nous utiliserons le lemme de Farkas et Minkowsky [17].

Associons aux contraintes ci-dessus des multiplicateurs $\pi_1, \pi_2, \dots, \pi_m \geq 0$ et μ de signe quelconque.

Alors une condition nécessaire et suffisante pour que (IV) ait une solution est que

$$\forall \pi \in \mathbf{R}^{m^+}, \quad \pi \cdot \gamma^k + \mu \geq 0 \quad (\forall k = 1, \dots, K) \Rightarrow \mu \geq 0.$$

La relation :

$$\mu \geq -\pi \cdot \gamma^k \quad (k = 1, \dots, K)$$

est vérifiée, pour γ donné, si et seulement si

$$\mu \geq - \text{Min}_{k=1, \dots, K} \{ \pi \cdot \gamma^k \}$$

alors la condition précédente est équivalente à

$$\forall \pi \in \mathbf{R}^{m^+}, \quad \text{Min}_{k=1, \dots, K} (\pi \cdot \gamma^k) \leq 0.$$

Par suite, une condition nécessaire et suffisante pour que (IV) ait une solution est que

$$\text{Max}_{\pi \in \mathbf{R}^{m^+}} \text{Min}_{k=1, \dots, K} \{ \pi \cdot \gamma^k \} \leq 0.$$

On se ramène de la sorte à la recherche du maximum de la fonction « d'admissibilité » $F(\pi)$ définie par

$$F(\pi) = \text{Min}_{k=1, \dots, K} \{ \pi \cdot \gamma^k \}.$$

On voit que $F(\pi)$ est tout à fait analogue au lagrangien $L(\pi)$ (simplement on ne tient pas compte des coûts).

Les propriétés de $F(\pi)$ sont les mêmes que celles de $L(\pi)$: elle est concave non différentiable. Le maximum $F(\pi^*)$ peut donc être recherché en utilisant une des méthodes précédemment exposées. Par exemple une méthode de sous-gradient.

Pour le calcul des déplacements λ_j (cf. IV.3) il est intéressant de prendre la valeur $\bar{L} = 0$ comme estimation de $F(\pi^*)$ (c'est une estimation par *excès* si le problème est admissible et par défaut si le problème ne l'est pas). Les λ_j sont donc calculés par

$$\lambda_j = \rho_j \frac{-F(\pi^j)}{\|\gamma(\pi^j)\|^2}, \quad 0 < \rho_j \leq 2$$

et

$$\rho_j \rightarrow 0 \quad (j \rightarrow +\infty).$$

Généralement, cette méthode fournit une solution réalisable (entière) de (P) s'il en existe ou alors met en évidence un $\bar{\pi}$ tel que $F(\bar{\pi}) > 0$ ce qui permet d'affirmer que le problème n'est pas faisable. Comme $F(\pi^*) < 0$ n'implique pas nécessairement qu'il existe des solutions entières, dans un certain nombre de cas, il sera difficile de conclure. Néanmoins, la méthode fournira alors des solutions entières ε -admissibles qui pourront s'avérer très intéressantes d'un point de vue pratique pour le problème étudié.

BIBLIOGRAPHIE

1. J. AGARD, J. SUDAROVICH et F. HEMMER, *Sélection et affectation optimales d'une flotte d'avions*, R.A.I.R.O., 7^e année, V-2, mai 1973, p. 3-26.
2. M. HELD et R. M. KARP, *The Traveling Salesman Problem and Minimum Spanning Trees*, Mathematical Programming, vol. 1, n° 1, 1971, p. 6-25.
3. M. HELD, P. WOLFE et H. P. CROWDER, *Validation of Subgradient Optimization*, Mathematical Programming, vol. 6, n° 1, 1974, p. 62-88.
4. M. L. FISCHER et J. F. SHAPIRO, *Constructive Duality in Integer Programming* Working paper O. R. 008-72 Operation Research Center M.I.T., Cambridge, Mass., avril 1972.
5. S. AGMON, *The Relaxation Method for Linear Inequalities*, Canad. J. Math., vol. 6, 1974, p. 382-392.
6. T. MOTZKIN et I. J. SCHOENBERG, *The Relaxation Method for Linear Inequalities*, Canad. J. Math., vol. 6, 1974, p. 393-404.
7. P. WOLFE, M. HELD et R. M. KARP, *Large scale optimization and the relaxation method*, in: Proceedings of the 25th Nat. A.C.M. meeting Boston, Massachusetts, 1972.
8. C. LEMARÉCHAL, *Méthodes de sous-gradients*, Bulletin de la Direction des Études et Recherches EDF, série C, n° 2, 1974, p. 5-14.
9. P. WOLFE, *A Method of Conjugate Subgradients for Minimizing Non Differentiable Functions*, Mathematical Programming (à paraître).
10. M. MINOUX, *Plus court chemin avec contraintes supplémentaires*, Annales des Télécommunications, t. 30, n°s 11-12, 1975.
11. R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, 1970.
12. L. S. LASDON, *Optimization Theory for Large Systems*, Macmillan series for Ops. Res., 1970.
13. R. C. GRINOLD, *Steepest Ascent for Large Scale Linear Programs*, S.I.A.M. Rev., vol. 14, 1972, p. 447-464.
14. B. T. POLJAK, *A General Method of Solving Extremum Problems*, Sov. Math Doklady, vol. 8, 1967, p. 593-597.
15. B. T. POLJAK, *Minimization of Unsmooth Functionals*, U.S.S.R. Computational Math. and Math. Physics, 1969.
16. G. B. DANTZIG et P. WOLFE, *Decomposition Principle for Linear Programs*, J. ORSA, vol. 8, n° 1, 1960, p. 101-111.
17. J. FARKAS, *Über die Theorie der einfachen Ungleichungen*, Journal für die reine und angewandte Mathematik, vol. 124, 1901, p. 1-27.
18. G. T. ROSS et R. M. SOLAND, *A Branch and Bound Algorithm for the Generalized Assignment Problem*, Mathematical Programming, vol. 8, 1975, p. 91-103.