Méthodes de décomposition pour la programmation mathématique

Philippe Mahey

▶ **To cite this version:**

Philippe Mahey. Méthodes de décomposition pour la programmation mathématique. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1990. tel-00337842

HAL Id: tel-00337842
https://theses.hal.science/tel-00337842

Submitted on 10 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HABILITATION

A DIRIGER DES RECHERCHES

présentée à

L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

par

Philippe MAHEY

METHODES DE DECOMPOSITION POUR LA PROGRAMMATION MATHEMATIQUE

Soutenue le 14 septembre 1990 devant la commission d'examen :

P.J. LAURENT	(Président)
J. FONLUPT	(Rapporteur)
B. LEMAIRE	
M. MINOUX	
V.H. NGUYEN	
A. TITLI	(Rapporteur)
P. TOLLA	(Rapporteur)

TABLE DES MATIERES

INTRODUCTION

L'ensemble des travaux présentés ci-après a pour objet l'étude de problèmes de Programmation Mathématique de grande dimension sous divers aspects. On a focalisé tout particulièrement les méthodes de décomposition et certaines implications de ces méthodes tant du point de vue théorique que du point de vue de leurs mises en œuvre et de leurs applications. Ces travaux ont été regroupés en cinq chapitres qui définissent les principales lignes de recherche qui ont guidé nos recherches depuis 1978.

Le premier chapitre concerne les modèles de planification de production. Un peu à l'écart des autres thèmes, par son caractère très appliqué et sa forte liaison avec notre travail de thèse de 3ème cycle, ce thème a fortement influencé notre réflexion ultérieure sur les aspects hiérarchisés des problèmes de prise de décisions optimales comme il nous a fourni un cadre idéal pour l'application des méthodes de décomposition.

Le deuxième chapitre concerne la décentralisation des décisions locales dans les problèmes d'optimisation comportant plusieurs sous-systèmes interconnectés. Ce thème aux nombreuses implications d'ordre organisationnelles mais aussi numériques a été une des premières motivations pour la recherche d'algorithmes de décomposition originaux.

Le troisième chapitre est dédié aux méthodes de sous-gradients et, principalement, à leur rôle dans les problèmes issus de programmes linéaires de très grande dimension ou d'origine combinatoire. Ces méthodes, réputées lentes et peu fiables, s'avèrent très utiles pour préparer le terrain à d'autres approches, plus efficaces mais difficiles à initialiser.

Le quatrième chapitre est centré sur la description et la validation d'une méthode de décomposition nouvelle de type primal-dual initialement destinée à la Programmation Linéaire. Cette méthode conduit à une décentralisation complète des décisions et cela sans le concours d'un niveau supérieur de coordination.

Le dernier thème abordé est celui des méthodes proximales dans le cadre de l'optimisation convexe. Ces travaux, plus récents, sont également motivés par la recherche de méthodes de décomposition donnant lieu à des sous-problèmes plus réguliers et plus stables numériquement. Ils s'étendent à des études sur les opérateurs monotones et la minimisation des fonctions dc.

Finalement, nous exposons brièvement les perspectives futures de nos recherches et discutons l'évolution de l'intérêt pour les problèmes d'optimisation de grande dimension et les méthodes de décomposition.

CHAPITRE 1

MODELES DE PLANIFICATION DES SYSTEMES DE PRODUCTION

Les modèles que nous avons considérés concernent la planification à moyen terme des systèmes de production à fort degré d'automatisation. On entend par moyen terme l'horizon de travail du décideur qui cherche à optimiser les niveaux de production des différents postes de travail ainsi que les niveaux de stockage des différents produits en transit dans l'atelier en fonction de la demande, des capacités et des coûts de fabrication et de stockage. Cet horizon moyen terme peut varier de la semaine à quelques mois suivant les cas. Il s'oppose au long terme qui détermine la politique globale de production dans un environnement incertain et au court terme qui doit garantir la faisabilité des ordres de production en temps réel. Il s'agit donc de problèmes d'optimisation déterministe. Les modèles étudiés ont de plus les caractéristiques suivantes :

- la linéarité des fonctions en présence.
- la discrétisation du caractère dynamique du problème.

Ces approximations mènent à un modèle général de Programmation Linéaire de grande dimension fortement structuré qui a fait l'objet de la thèse de 3ème cycle :

[1] "*Etude de la planification à moyen terme d'une unité de fabrication en vue de sa gestion intégrée* ", Thèse de 3ème cycle, Université Paul Sabatier, Toulouse, fev. 1978.

Ce travail s'intègre dans une action thématique sur "l'*Automatisation Intégrée des Systèmes de Production* " démarrée au LAAS, Toulouse, en 1974. La thèse est centrée sur la justification du modèle linéaire pour la planification à moyen terme d'un atelier de fabrication de circuits intégrés de la RadioTechnique-Compelec à Evreux. Ce modèle a tout d'abord été traîté par le logiciel MPSX d'IBM, aujourd'hui encore l'outil le plus performant pour résoudre les programmes linéaires de très grande taille. Un générateur de matrices a été mis au point ainsi qu'un programme interactif en langage MPS. La double structure multi- produits / multi-périodes a conduit à une première analyse de l'emploi de méthodes de décomposition comme approche alternative pour ce modèle. Une méthode de décomposition par les prix du type Dantzig-Wolfe fut donc proposée et sa mise en œuvre fit l'objet de la thèse de 3ème cycle de J.B. Lasserre (1979).

En parallèle, l'équipe 'Ordonnancement' du LAAS développait une ligne de recherche sur les approches par modèles agrégés de la production. Les méthodes d'agrégation sont complémentaires des méthodes de décomposition. Ce thème de recherche a été repris plus tard après de

stimulantes discussions sur les méthodes d'agrégation itérative avec G. Cohen. Un article en portugais sur les liens entre agrégation itérative, cohérence et les méthodes de projections ou multigrilles en Analyse Numérique a été publié dans les annales du 6ème Congrès Brésilien d'Automatique en 1987.

Les aspects hiérarchisés de la planification de production ainsi que la validation du modèle linéaire proposé pour le niveau moyen terme ont été résumés dans un article présenté au IIème Congrès Brésilien d' Automatique à Florianopolis en 1978 :

[2] "*An original linear model for average-term planning* ", II° Congresso Brasileiro de Automatica, Florianopolis, 1978, Proc. pp. 510- 519.

Citons enfin dans cette ligne une étude effectuée plus récemment en collaboration avec J.B. Lasserre sur les réseaux de Petri :

[3] "*Using linear programming in Petri nets analysis* ", RAIRO-Recherche Opérationnelle 23, 1, 1989, pp. 43-50, en collaboration avec J.B. Lasserre.

Dans cet article, on utilise des techniques d'algèbre linéaire pour l'analyse de certaines propriétés des réseaux de Petri. On montre que pour démontrer ces propriétés (invariants, réseau borné, réseau vivant, places implicites), la Programmation Linéaire peut être utilisée à la place de la Programmation Entière, trop coûteuse, et que le calcul d'une base d'invariants n'est jamais nécessaire. La taille du réseau n'est donc pas un obstacle pour l'analyse de ces propriétés.

II Congresso Brasileiro de Automática - Florianópolis,UFSC,
Set., 1978

AN ORIGINAL LINEAR MODEL FOR AVERAGE-TERM PRODUCTION PLANNING

Dr. Philippe Mahey

Departamento de Engenharia Elétrica
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 209, ZC-20
Rio de Janeiro, RJ, 20.000, Brasil

Abstract: In this paper, we show how to modelize a jobshop production planning on an average-term horizon assuming the following two objectives: (1) the necessity of integration of all the management functions of a production system and (2) the construction of a data-processing system well fitted to a real case in order to simulate and justify it. Linear decision rules are outlined and the choice of an original "average-term policy" leads to a relevant multilinear cost function. By the mean of a large-scale specific linear-programming optimizer,on line simulations of production plans have been made basing on industrial real data - Limits and extensions of this study are exposed in the conclusion.

Resumo: UM MODELO LINEAR ORIGINAL PARA PLANEJAMENTO DE PRODUÇÃO A PRAZO MÉDIO: Neste artigo nós mostramos como modelar, a prazo médio, o planejamento de produção duma empresa supondo os seguintes objetivos: (1) A necessidade de integração de todas as funções gerenciais dum sistema de produção e (2) a construção dum sistema de processamento de dados bem ajustado a um

caso real, para fins de simulação e justificação. Regras de decisão lineares são descritas e a escolha duma "política original a prazo médio" leva a uma função de custo multilinear relevante. Por meio dum otimizador específico de programação linear de grande porte, simulações "on-line" do planejamento da produção foram feitos com bases em dados reais de indústria. Limitações e extensões deste estudo são apresentadas nas conclusões.

I - Introduction

Production planning is concerned with specifying how the production resources of the firm are to be employed over some future time period. The composition of the planning function, such as the techniques used, the number of people involved, and the degree of detail depends in part on the planning horizon or the length of the period for which plans are to be made. Long-term planning, for a period of two or more years, would consider suchthings as plant expansion, capital budgeting and job-shop design for example. Short-term planning, for periods of one day to one month, organize the daily scheduling of the jobshop.between these two extremes, we consider what we call average (or intermediate) term planning which role is to specify the resource requirements for each of a series of time increments, resources including factors contributing to production capacity, inventory control or production quality. These various plans are connected in a decision structure and information flow as shown in fig. 1. According to a real industrial example, we are going to define the characteristics and the mathematical structure of the average term planning model.

II -Structure hypothesis

If seems irrealistic to try to generate an universal model All information we used was taken from a jobshop manufacturing integrated circuits. We have to point out the main caracteristic of the production as follows:

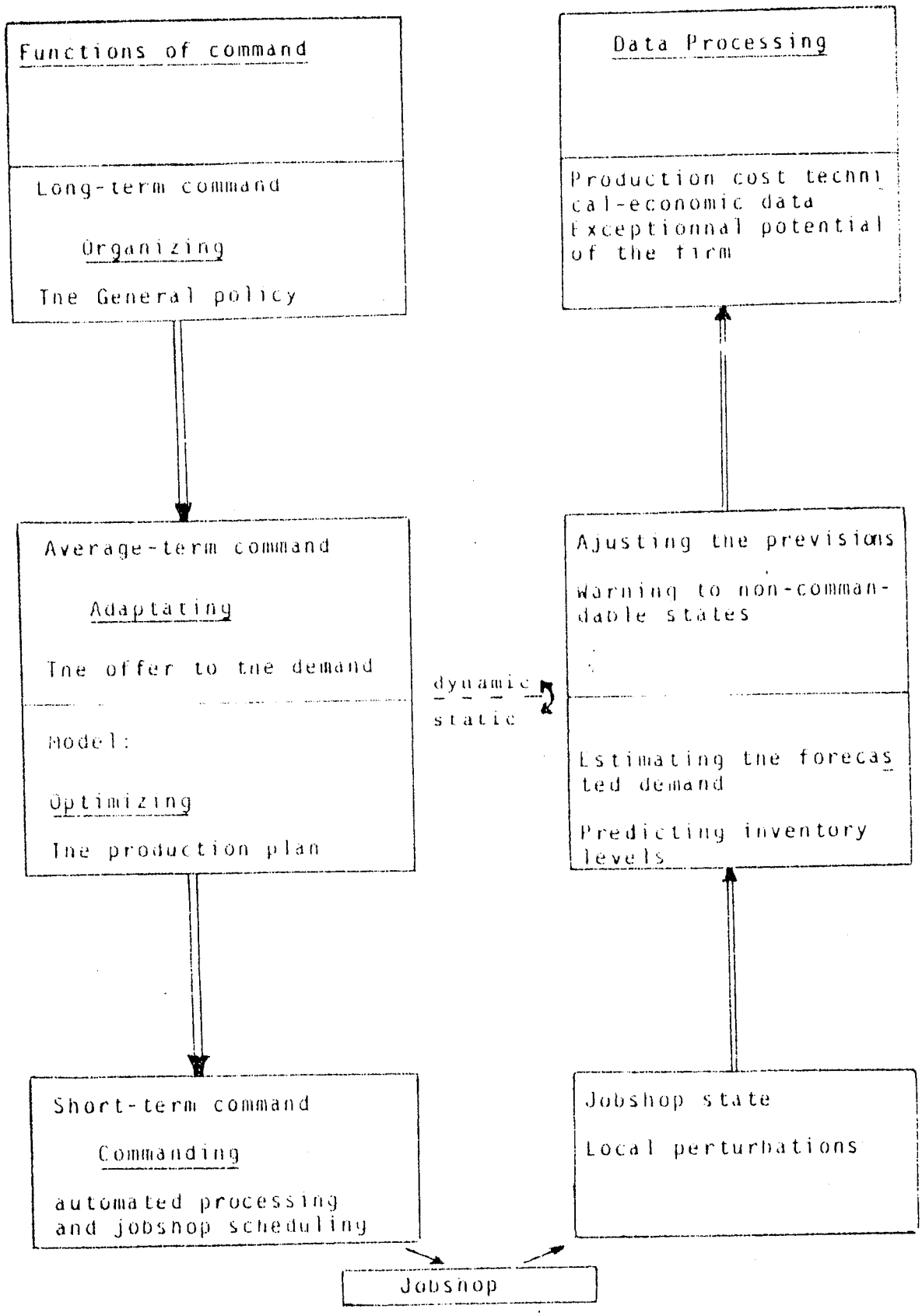

Functions of command
Long-term command
Organizing
The General policy
Data Processing
Production cost technical-economic data
Exceptionnal potential of the firm
Average-term command
Adaptating
The offer to the demand
Model:
Optimizing
The production plan
dynamic
static
Ajusting the previsions
Warning to non-commandable states
Estimating the forecasted demand
Predicting inventory levels
Short-term command
Commanding
automated processing and jobshop scheduling
Jobshop state
Local perturbations
Jobshop

Fig.1

i) A non-repetitive production, specified by each demand. A manufacturing technically not stabilized and Homoge - nous.

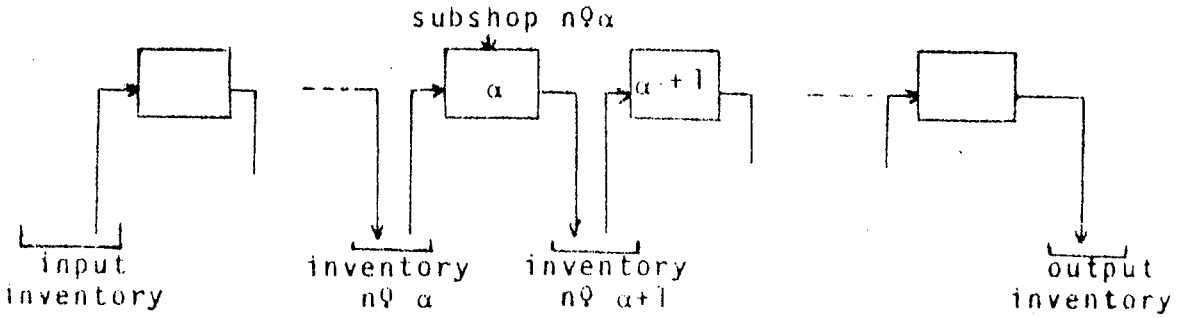
ii) The jobshop can be represented by fixed input and out - put of products, between which one and only one route is assigned to each product (unique routing function).

iii) The average-term horizon must concord with the average total operating-time of a product (in that case one month).

III-Model equations

3.1. Discretizing space and time.

The jobshop, considered as a line between two fixed points is discretized in smaller parts, "subshop" con nected by work-in-process inventories (fig. 2)



Rule: that discretization is unique =>the work-in-process inventories S_α could be the nodes of the modelized jobshop.

Then a discretization of the time-horizon appears as a con sequence as shown on fig. 3.

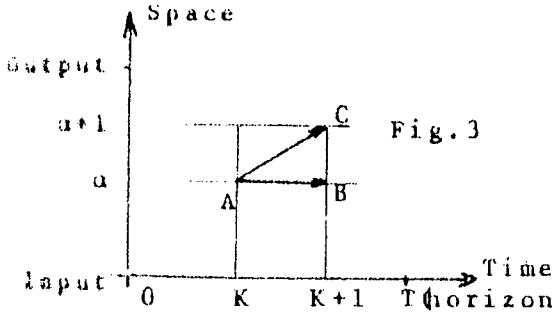


Fig. 3

Suppose a lot of product nº i in the stock S_α at time K(A)only two possibilities occur:

. this product is processed during the K^{th} period and arrived in stock α+1 at time K+1→ AC

or

. this product stay in stock α→ AB

Let us go back to our objective: to optimize the planning of the production during the horizon $[o,T]$. That is, to fix the optimal quantity of each product to be processed during each period (for instance each week) in each "subshop". Let us call the quantity of product nº i shipped from stock S_α at the K^{th} period: $Y(\alpha,i,K)$ and $X(\alpha,i,K)$ the quantity of product nº i that stay in the stock S_α during the K^{th} period.

The conservative equations of the production are for each α, each K and each i:

$$K(\alpha,i,K) + Y(\alpha,i,K) = X(\alpha,i,K-1) + \rho(\alpha-1,i)\ Y(\alpha-1,i,K-1) \quad (1)$$

$$\text{with } X(\alpha,i,K) > 0 \quad \text{and } Y(\alpha,i,K) > 0$$

$\rho(\alpha,i)$ is a statistical parameter, average value of the proportion of good items of product i when processed in subshop α. If must be commanded by the Production quality controller.

Initial conditions are: $X(\alpha,i,o) + Y(\alpha,i,o) = E_o(\alpha,i)$ (2)

3.2. Capacity restrictions:

To diminish the number of variables, we decided to consider a subset of the set of facilities. In that subset, we managed to enter the most critical production resources, which are selected according to the highest operating costs or the location of the principal backlogs.

Let $\{ m_\beta \}$ = subset of critical machines

$\{ A_\alpha \}$ = set of subshops

$\{ P_i \}$ = set of products-in-process

To the triplet (α,i,β) let us associate a parameter $F(\alpha,i,\beta)$ so that: - If a unit of P_i is to be processed on the machine m_β when routing from stock S_α to $S_{\alpha+1}$, the operating time is $F(\alpha,i,\beta)$. If that unit does not utilize m_β then

$F(\alpha,i,\beta) = 0$

Remarks: 1) as long as a simple application exists between α and β we could simplify this and call operating time $F_{i\beta}$

2) Note the statistical origin of the parameter $F_{i\beta}$

Let $CAP(\beta,K)$ = the maximum time of employment of machine M_β during the K^{th} period.

then, for $m_\beta \in A_\alpha$: $\sum_i F_{i\beta}\, Y(\alpha,i,K) \leq CAP(\beta,K)$ (3)

IV - Choice of the relevant costs.

To evaluate the efficiency of the average-term policy, we have to determine which costs are relevant in front of this policy and to understand their behavior with changes in the planning variables.

Let us define the three main aspects that we selected:

4.1. Satisfaction of the customer demand.

Let $C_{11}(i,D)$ be the cost per unit of product P_i lacking in the output stock at the due date D and $C_{12}(i,D)$ the cost per unit of product P_i staying in the output inventory.

Let $XC(i,D)$ = total demand of item i at date D ($D \in [o,T]$)

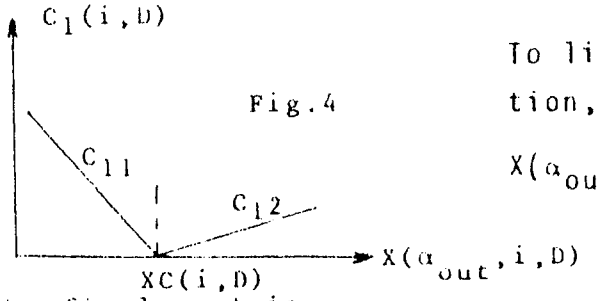


Fig.4

To linearize this cost function, we can write:

$$X(\alpha_{out},i,D) = XC(i,D) + \varepsilon_1 - \varepsilon_2$$

The final cost is:

$$C_1 = \sum_{i,D} \left[C_{11}(i,D)\varepsilon_1(i,D) + C_{12}(i,D)\varepsilon_2(i,D)\right] \quad (4)$$

4.2. Best facility utilization

We have to force the system to load each machine at its maximum employment level. Let the corresponding cost be:

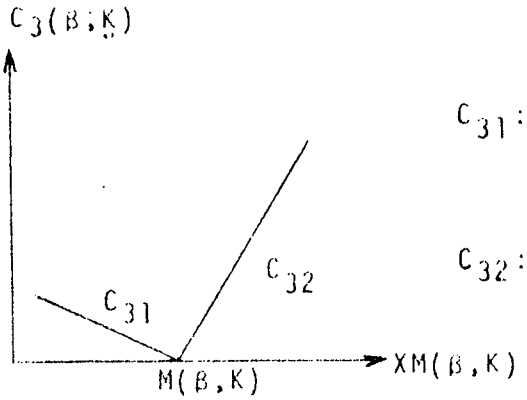
$$C_2(\beta,K) = c_2(\beta,K)\left[\,CAP(\beta,K) - \sum_i F_{i\beta}\, Y(\alpha,i,K)\right] \quad (5)$$

$$C_2 = \sum_{\beta,K} C_2(\beta,K)$$

4.3. Work-in-process inventory control

The preceding cost assumes the risk of unemployment level costs. We now point out the risk of inventory shortages. The idea of an optimum inventory level is relevant. This level (positive) could be the result of leveled production and could be able to meet unexpected changes in demand or any pertubations in the flow process.

Let $M(\beta,K)$ = optimal employment level let before the machine M_β during the K^{th} period as a security buffer.



C_{31}: Cost of risk of running out of stock per unit of employed time.

C_{32}: Cost per unit of unemployed time (work-in-process cost)

Fig.5

with $XM(\beta,K) = \sum_i F_{i\beta}\, X(\alpha, i, K)$

The linearisation is similar to 4-1.

Thus, the total cost is:

$$C_3 = \sum_{\beta,K}\left[C_{31}(\beta,K)\varepsilon_1(\beta,K) + C_{32}(\beta,K)\varepsilon_2(\beta,K)\right]$$

We have defined a three-terms cost function, each term being a linear combination of the variables of the system.

According to the linear structure of the model equations, we are tempted to use Linear Programming techniques to solve the problem that is optimize a cost function $C = C_1 + C_2 + C_3$.

V - Computing a solution - Perfomances and limits.

As a direct application of this original model, we used real data issued from a jobshop which led us to the following problem:

- Number of different items manufactured at the same period in the jobshop : about 800, that we simplify by considering only 12 " families" of very close products.
- Number of subshops: the jobshop was cut in 4 parts.
- Number of "critical" machines: 15 critical machines were kept at the end of the procedure.
- Number of periods of the optimizing horizon: 8 weeks.

After simplifying the model as much as possible, we reached a problem of 720 rows and 880 variables (assuming that only the Y variables have been kept by the use of inequality constraints).

The solution of this large-scale LP problem was computed by the mean of an MPSX/370 package from IBM, well adapted to the internal constitution of the matrice. (cf 1 & 3)

On fig. 6 is shown the comparative performatices for two real jobshops:

	Rows	Columns	Elements	Computing Region	CPU time (on IBM 370)
Jobshop 1	716	880	12500	266K	65"
Jobshop 2	445	520	8400	178K	48"

All the details of the computing procedure and performance are to be found in a larger report[6].

It seems anyway that again for an optimization problem, "the curse of dimensionality" rapidly inhibits large restrictions. Nevertheless, the model we built has allowed us to get closer to the vulnerable points of a real manufacturing process. Solutions of the program have been fructuosly compared to the operating plans at a reasonable cost.

VI - Conclusion

Such a model of the production system needs always real simulations to justify it. The access to industrial data is often diplomatic and boring work but it is necessary to work close to the real problems. The control of the production orders during an average-term horizon by using of fictive work-in-process inventories is a relative easy to formulate technique, but this lead us to big problems of adaptation to the manager's objectives. This model could be easily inserted in a global structure of an "integrated production control system". Next problems to solve are connections with the short-term control, that is the scheduling of the operations of the workshop[2] and, too, the control[5] of the production quality which needs a probabilistic approach of the system.

Referenciâs

1 DANTZIG GB. "Linear programming and extensions", Princeton U. Press 1963

2 ERSCHLER J. & ROUBELLAT F. "A decision-making process for the real time control of a production unit" Int. J.Prod. Res. 1976, vol.14 nº 2.

3 GALE J. "The theory of linear economics model" Mac-Graw-Hill 1960

4 GAVETT J.W. "Production and Operations management" Harcourt, Brace & Wed 1968

5 KING JR. "Production planning and control" Pergamon Int. Lib. 1975

6 MAHEY P. "Planification d'une unitē de fabrication en vue de sa gestion intēgrée" Thēse 3ème cycle. Toulouse 1978

7 SOLAEN R. "Contribuition ā l'ētude des systēmes de conduite en temps rēal en vue de la commande d'unitēs de fabrications" Thēse d'ētat - LILLE 1977

Recherche opérationnelle/Operations Research
(vol. 23, n° 1, 1989, p. 43 à 50)

USING LINEAR PROGRAMMING IN PETRI NET ANALYSIS (*)

by Jean B. LASSERRE (1) and Philippe MAHEY (2)

Abstract. — *The algebraic representation of polyhedral sets is an alternative tool for the analysis of structural and local properties of Petri nets. Some aspects of the issues of reachability, boundedness and liveness of a net are analyzed and characterized by the means of linear system of inequalities. In most cases, linear programming instead of integer programming can be used to check these properties therefore enabling the validation of very large nets.*

Keywords : Petri nets; linear programming.

Résumé. — *On utilise des techniques simples d'Algèbre linéaire pour l'analyse de certaines propriétés des réseaux de Pétri. On montre que ces propriétés (invariants, réseau borné, réseau vivant, places implicites) peuvent être analysées en utilisant la Programmation Linéaire (de complexité polynomiale) et non la Programmation en nombres entiers. Pour les propriétés citées, le calcul (en général prohibitif) d'une base d'invariants n'est jamais nécessaire. La taille du réseau n'est donc pas un obstacle pour l'analyse de ces propriétés.*

Mots clés : Réseaux de Pétri; programmation linéaire.

I. INTRODUCTION

Petri nets have become a widely used tool for modelling and analysing large, complex and discrete event systems. They appear in computer science as well as in operations research for modelling a quite large number of problems such as information processing, communication network design, scheduling and control of manufacturing processes. Here we focus on some aspects of the analysis of Petri nets. An important issue is for example to know whether it is able to realize and complete all the tasks for which it has been designed. Among the related properties which are commonly analyzed are the boundedness and the liveness of the net in order to certify that neither traps nor deadlocks may occur. These two properties are linked to the issue

(*) Received August 1988.
This work was realized while the first author was visiting PUC, Rio de Janeiro.
(1) Laboratoire d'Automatique et d'Analyse des Systèmes, C.N.R.S., Toulouse, France.
(2) Departamento de Engenharia Elétrica, P.U.C./R.J., Rio de Janeiro, Brazil.

Recherche opérationnelle/Operations Research, 0399-0559/89/01 43 08/$ 2.80
© AFCET-Gauthier-Villars

of reachability, which in turn is one of the key concept in mathematical system theory. In that sense it is quite natural to view the evolution of the marking on the places of the graph as a linear time invariant discrete system. If M_k is the vector of marks on the space of places of the net at time k, s_k is a control vector of firings on the space of transitions and C is the incidence matrix of the graph, then the dynamic state equation is:

$$M_{k+1} = M_k + C s_k$$

In fact, linear system theory is of limited help when dealing with Petri nets. Some partial characterization of controllability and reachability are given in Murata [6] but these results are not sufficient and cannot be exploited practically. On the other hand, linear algebra and in particular some duality results for systems of linear inequalities gave rise to a few interesting propositions about boundedness, liveness and reduction techniques (*cf.* [2, 5]). This approach constitutes a valid alternative to the classical analysis where we must build the reachability tree, *i. e.* the tree of all feasible firing sequences from a given initial state. However the computational cost remains high because the conditions have to be tested on the set of integers [8]. Peterson [7] has mentionned the possibility to relax the integrality condition and to use linear programming for the detection of semi-flows in the net. We now extend this statement, proving its validity for the conditions of boundedness, liveness and for reduction techniques. Duality is then used to yield some insights on the geometry of the reachable set.

II. LINEAR ALGEBRA APPROACH

2.1. Notation and key lemma

We use the same notation as in Brams [2].

A Petri net is a four-tuple $R = \langle P, T, \text{Pré}, \text{Post} \rangle$ where:

P is a finite set of places with $|P| = m$

T is a finite set of transitions with $|T| = n$,

Pré: $P \times T \to N$ is the forward incidence mapping

Post: $P \times T \to N$ is the backward incidence mapping.

A marked net is a couple $\langle R; M \rangle$ where R is a Petri Net and $M: P \to N$ is a marking mapping. We denote $M(p)$ the marking of place $p \in P$, which is also the number of tokens available on place p.

The incidence matrix C is defined as follows:

$$\forall (p, t) \in P \times T, \quad C(p, t) = \text{Post}(p, t) - \text{Pré}(p, t)$$

If M' is an accessible marking from M, then we have the fundamental relationship:

$$M' = M + Cs \tag{1}$$

where $s(t)$ is the number of times transition t has been fired. Petri net analysis using linear algebra is based on the above equation.

In the following, we consider some weighting or valuation function $f: P \to N$. In fact, it can be viewed as a linear functional defined on the primal space of marking vectors, N^m. Then f is a dual vector associated to the primal equation (1).

Before proceeding further, we need the two following lemma:

LEMMA 1: *Let C be any matrix with coefficients in Z. Then we have:*

$$\Omega_0 = \{f: f \in N^m, C^T f = 0\} \neq \{0\}$$
$$\Leftrightarrow \quad \Omega_0' = \{f: f \in R^m, f \geqq 0, C^T f = 0\} \neq \{0\}$$

Proof:

$\Rightarrow$ is trivial.

$\Leftarrow$ Ω_0' is a closed polyhedral convex cone. Hence,

$$\forall f \in \Omega_0', \quad f = \sum_{i=1}^{l} \lambda_i f^i, \lambda_i \geqq 0 \qquad \text{for} \quad i = 1, \ldots, l$$

where $\{f^i\}_{i=1}^{l}$ is a set of generators of the cone.

A set of generators can be found by solving all homogeneous linear systems of $m-1$ linearly independent equations with m unknowns built from the rows of $\begin{bmatrix} C^T \\ I \end{bmatrix}$ (*cf.* [3, 9]). Consequently, as C and I have integer coefficients, it is always possible to compute solutions in Q^m (elementary pivoting operations with integer coefficients). Multiplying the rational values by a common denominator, we can then always obtain f^i in N^m. Hence $f^i \in \Omega_0$ and $\Omega_0 \neq \{0\}$.

LEMMA 2: *Let C be any matrix with coefficients in Z. Then we have:*

$$\Omega_1 = \{f: f \in N^m, f > 0, C^T f \leqq 0\} \neq \varnothing$$
$$\Leftrightarrow \quad \Omega_1' = \{f: f \in R^m, f \geqq e, C^T f \leqq 0\} \neq \varnothing$$

where e is an m-vector of ones.

Proof:

$\Rightarrow$ is trivial.

$\Leftarrow$ $\Omega_1' \neq \varnothing$. Then Ω_1' is an unbounded convex polyhedron ($f \in \Omega_1'$, then $rf \in \Omega_1'$, $\forall r \geqq 1$). Hence, any vector f in Ω_1' can be written:

$$f = \sum_{i=1}^{l} \lambda_i f^i + \sum_{j=1}^{l'} r_j f'^j$$

with

$$\lambda_i \geqq 0,\ i = 1, \ldots, l, \qquad \sum_{i=1}^{l} \lambda_i = 1$$

$$r_j \geqq 0, \qquad j = 1, \ldots, l'$$

f^i, $i = 1, \ldots, l$, are the extreme points of Ω_1'.

f'^j, $j = 1, \ldots, l'$, are the extreme rays of Ω_1'.

Again by standard arguments on the computation of vertices and extreme rays of a polyhedron in R^n given by a set of inequalities with rational coefficients, we obtain that f^i and $f'^j \in Q^m$, $\forall i, j$.

We can then take any f^i and multiply it by the common denominator $\lambda_0 > 0$ to get $\lambda_0 f^i \in \Omega_1$ (in fact, $\lambda_0 > 1$, then $\lambda_0 f^i \in \Omega_1'$). Hence, $\Omega_1 \neq \varnothing$.

Remarks: (i) these results still hold if the matrix C has coefficients in Q.

(ii) to check if Ω_0 (respectively Ω_1) is empty one only need to check if Ω_0' (resp. Ω_1') is empty. This can be done by using Linear Programming techniques which can handle very large size problems.

2.2. Boundedness

Boundedness is an important desirable property for Petri nets. It means that the number of tokens in every place is bounded whatever happens. We know (*see* [2] for example) that:

$$R \text{ is bounded} \quad \Leftrightarrow \quad \exists f \in N^m,\ f > 0,\ C^T f \leqq 0$$

If such an f exists, then we have:

$$M(p) \leqq \frac{f^T M_0}{f(p)}, \qquad \forall p \in P$$

In view of lemma 2, we now have:

PROPOSITION 1:

$$R \text{ is bounded} \Leftrightarrow \exists f \in R^m,\ f \geqq e,\ C^T f \leqq 0.$$

COROLLARY: *A bound for the number of tokens in place p* is given by.

$$\begin{array}{l} \text{Min } M_0^T f \\ \left| \begin{array}{l} C^T f \leqq 0 \\ f \geqq e \\ f(p) = 1 \end{array} \right. \end{array}$$

which is a linear program.

2.3. *p*-semi-flows and liveness

p-semi-flows are also important in the analysis of Petri nets [5]. The set of semi-flows on a net R is precisely the set Ω_0 defined in Lemma 1. The purpose of this section is to illustrate the fact that any homogeneous linear relation that must be satisfied on the set of semi-flows can be tested equivalently on the set Ω_0'. Indeed, this is again because we are able to find a set of integer-valued generators for the cone Ω_0'.

For instance, the invariance property of semi-flows is valid on Ω_0': for any accessible marking M, we have:

$$\forall f \in \Omega_0',\quad f^T M = f^T M_0$$

and if $f(p) \neq 0$, $M(p) \leqq M_0^T f / f(p)$, $\forall f \in \Omega_0'$.

A bound on $M(p)$ can therefore be computed by solving the linear program:

$$\begin{array}{l} \text{Min } M_0^T f \\ \left| \begin{array}{l} C^T f = 0 \\ f(p) = 1 \\ f \geqq 0 \end{array} \right. \end{array}$$

A necessary condition for liveness can be simplified by rising the same argument:

PROPOSITION 2: *If* $\langle R;\ M \rangle$ *is live, then:*

$$\forall f \in \Omega_0',\quad f^T M \geqq f^T \text{pré}(.,\ t),\quad \forall t = 1,\ \ldots,\ n$$

To check this condition, we only need solve the n linear programs:

$$\left|\begin{array}{l}\text{Min } f^T(M - \text{pré}(.,t)), \quad t=1,\ldots,n \\ C^T f = 0 \\ f \geqq 0\end{array}\right.$$

Hence, if $\langle R; M \rangle$ is live, the optimal value of the above linear program is equal to zero for any t.

2.4. Réduction techniques

Reduction techniques are used to simplify large scale Petri nets into "equivalent" smaller nets. Simplification of implicit places is such a technique (*see* [1]).

p is an implicit place iff there exists $\bar{f}: p \to Z$ such that:

- $\bar{f}(p) > 0$ and $\bar{f}(q) \leqq 0, \quad \forall q \neq p.$
- $\forall M_0 \in \bar{M}_0, \bar{f}^T M_0 \geqq 0$
- $\forall t \in T, \bar{f}^T \text{Pré}(.,t) \leqq \text{Min}\{\bar{f}^T M_0, M_0 \in \bar{M}_0\}$
- $\exists c \in N^m, f^T C = c$

where $\bar{M}_0$ is a set of initial markings.

Again, it suffices to check that the system below has a solution:

$$\begin{array}{rl} f \in R^m \quad \text{and} & C^T f \geqq 0 \\ f(p) = 1 & \\ f(q) \leqq 0, & \forall q \neq p \\ f^T M_0 \geqq 0, & \forall M_0 \in \bar{M}_0 \\ f^T \text{Pré}(.,t) \leqq f^T M_0, & \forall M_0 \in \bar{M}_0, \quad \forall t \in T \end{array}$$

This can be done by standard Linear Programming.

III. A BOUND ON THE REACHABLE SET

In this section, we give an analytical expression of a set which always contains the reachable set after K iterations have been fired (for any K). When the reachable set is bounded, this set is also bounded and we retrieve the bound given in the previous section. When the reachable set is not bounded, it allows one to give bounds on the places after K transitions have been fired.

Let M be an accessible marking from M_0 after K transitions. Then, there exists $s \in N^m$ such that, $M - M_0 = Cs$, $e^T s = K$

Now, let $\Omega(K)$ be the convex set defined by:

$$\Omega(K) = \{M \in R^m;\ M \geqq 0, \text{ there exist } s \in R^n \text{ such that } Cs = M - M_0,\ e^T s = K,\ s \geqq 0\}$$

Introducing dual variables $f \in R^m$ and $v \in R$, we characterize $\Omega(K)$ by its bounding hyperplanes:

$$\Omega(K) = \{M \in R^m;\ M \geqq 0,\ f^{iT}(M - M_0) + Kv^i \geqq 0,\ i = 1, \ldots, r\}$$

where (f^i, v^i), $i = 1, \ldots, r$ are the extreme rays of the polyhedral cone:

$$\mathscr{C} = \left\{ \begin{pmatrix} f \\ v \end{pmatrix} \in R^{m+1};\ C^T f + ev \geqq 0 \right\}$$

$\Omega(K)$ always contains the reachable set after K transitions since the integrality constraint has been dropped.

To retrieve the boundedness results, we observe that $M(p)$ is bounded by the optimal value of the following *LP*:

$$\begin{array}{ll} \text{Max} & M(p) \\ & -M + Cs = -M_0 \\ & e^T s = K \\ & s \geqq 0 \\ & M \geqq 0 \end{array}$$

The dual problem is:

$$\begin{array}{ll} \text{Min} & M_0^T f - Kv \\ & C^T f + ev \leqq 0 \\ & f(p) \geqq 1 \\ & f \geqq 0 \end{array}$$

Let (f^i, v^i) be the vertices of the dual feasible set. Three cases must be considered:

(*a*) There exists an (f, v) dual feasible such that $v > 0$. Then, for large K, the optimal value of the dual problem becomes negative, which implies, because $M(p) \geqq 0$, that this value is $-\infty$ and the primal is infeasible. In other words the reachable set is empty.

(*b*) for all the vertices, $v^i < 0$.

Then, for K large enough, the optimal value is $M_0^T \bar{f} - K\bar{v}$ where $(\bar{f}, \bar{v})$ is a vertex such that $\bar{v} = \underset{i}{\text{Min}}\, v^i$. Therefore, $M(p)$ is not bounded when $K \to \infty$.

(c) for all the vertices, $v^i \leqq 0$ and for at least one, say v^j, $v^j = 0$. Then, for K large enough, the optimal value is $\min \{M_0^T f^j, j \text{ s. t. } v^j = 0\}$. Observe that as $f \geqq 0$, we have at the optimal solution $f(p) = 1$ and we retrieve the bound obtained in the precedent section.

IV. CONCLUSION

As we have just seen, all the properties we have investigated (boundedness, semi-flows, liveness and simplification of implicit places) can be checked by using standard Linear Programming instead of Integer Programming as used in some packages like Ovide [8] and without computing a basis of invariants. This means that for that kind of analysis, very large Petri Nets can be handled whereas the use of Integer Programming leads to serious limitations on the size of the net.

REFERENCES

[1] G. BERTHELOT and G. ROUCAIROL, *Reduction of Petri Nets*, Lecture Notes in Computer Science, vol. 45, 1976, pp. 202-209.

[2] G. W. BRAMS, *Reseaux de Petri: théorie et pratique*, Masson, 1983,

[3] J. B. LASSERRE, *Consistency of Linear System of Inequalities*, JOTA, vol. 49, N° 1, 1986, pp. 177-179.

[4] K. LAUTENBACH and H. SCHMID, *Use of Petri Nets for Proving Correctness of Concurrent Process Systems*, Proc. 1974 I.F.I.P. Congress, North Holland, 1974, pp. 187-191.

[5] G. MEMMI, *Applications of the Semiflow Notion to the Bound Dedness and Liveness Problems in Petri Net Theory*, Proc. 1978 Conf. on Information Sciences and Systems, John Hopkins University, 1978, pp. 505-509.

[6] T. MURATA, *State Equations, Controllability and Maximal Matchings of Petri Nets*, I.E.E.E. Trans. Automatic Control, AC-22, 3, 1977, pp. 412-416.

[7] J. L. PETERSON, *Petri Net Theory and the Modelling of Systems*, Prentice Hall, 1981.

[8] B. PRADIN, *Un outil graphique interactif pour la vérification des systèmes à évolutions parallèles décrits par réseaux de Petri*, Docteur-Ingénieur Thesis, Université Paul-Sabatier, Toulouse, 1979.

[9] M. SIMMONNARD, *Programmation linéaire*, Dunod, 1962.

CHAPITRE 2

DECOMPOSITION ET DECENTRALISATION

Les méthodes de décomposition en Programmation Mathématique consistent à transformer des problèmes d'optimisation de grande dimension en la résolution séquentielle ou en parallèle de sous-problèmes de taille plus petite. Par 'grande dimension', on sous-entend à la fois :

- le grand nombre de variables et/ou de contraintes
- l'existence d'une structure sous-jacente difficile à exploiter dans une approche globale.

Ce dernier point est important dans le cas de la Programmation Linéaire car les logiciels modernes basés sur la méthode du Simplexe (comme MPSX) ou sur la récente méthode de Karmarkar peuvent en principe traîter des problèmes de très grande dimension. L'effort considérable investi par les grandes compagnies informatiques dans les années 50 et 60 pour la construction de codes 'in-core' très performants et tournant sur de grosses machines a longtemps fait de l'ombre aux recherches sur les méthodes de décomposition. Il semble que deux facteurs jouent aujourd'hui en faveur de leur réhabilitation :

- l'intérêt pour certains problèmes 'durs' originaires de l'optimisation stochastique et de l'optimisation combinatoire
- l'évolution du matériel vers le *calcul parallèle*.

Ces observations montrent d'ailleurs que le cadre de la Programmation Linéaire est insuffisant pour rendre compte de l'impact de ces nouvelles données. De plus, les méthodes de décomposition peuvent être motivées par d'autres considérations que les difficultés introduites par la dimension. On peut citer en particulier :

- l'abordage de problèmes aux *données hétérogènes*
- la *décentralisation* des processus de décision.

Ce dernier aspect, aux implications importantes dans les modèles d'origine économique (Arrow et Hurwicz, 1960), a d'ailleurs motivé une partie de nos recherches comme il sera vu plus loin.

Une approche par décomposition comprend généralement deux phases:

- l'*analyse structurale* du problème dans le but d'identifier des *sous-systèmes* (le résultat de cette analyse n'étant pas unique, on choisit généralement la structure qui conduit au découplage maximum de ces sous-systèmes)
- la résolution (itérative ou pas) de ces sous-problèmes éventuellement supervisée par un niveau de *coordination* qui tient compte du couplage entre les sous-systèmes par l' intermédiaire d'un certain nombre de paramètres de coordination.

Schématiquement, êtant donné un problème (S) posé sur un espace

produit $X=X_1 x \ldots x X_p$, on remplace sa résolution par la résolution de p sous-problèmes (S_i), i=1,...,p, chacun d'eux posé sur un des espaces X_i et dépendant de paramètres a_i. Ces paramètres, dits de coordination, peuvent être des variables du problème (S) (on les appellera alors des variables de couplage) ou des indicateurs liés aux interactions entre les sous-systèmes. Ils sont réajustés au cours d'un processus itératif, soit par un niveau supérieur de coordination (on parlera alors de décomposition-coordination), soit directement par certains sous-problèmes qui les transmettent aux autres (ces méthodes seront dites à un niveau). Soient $\{a_i^t\}$ la suite des vecteurs de paramètres déterminés pour chaque i à chaque itération t et $\{x_i^t\}$ la suite des solutions calculées par les sous-problèmes. On est donc amené à poser les questions suivantes :

- Si la suite $\{x_i^t\}$ converge vers x_i^* pour chaque i, $(x_1^*,\ldots,x_p^*)$ est-il optimal pour (S)?
- Si la suite $\{a_i^t\}$ converge vers a_i^* pour chaque i et que les valeurs des paramètres sont fixées à leurs valeurs limites, est-ce que chaque sous-problème peut calculer une solution x_i^* telle que $(x_1^*,\ldots,x_p^*)$ soit optimal pour (S) et peut-il reconnaître cette optimalité?

On dira que le système est *décentralisé* si la réponse à cette dernière question est affirmative. On parlera de décentralisation partielle si la convergence des paramètres de coordination est assurée par un niveau supérieur de coordination qui, de plus, teste et garantit l'optimalité du problème (S).

[4] "*On the drawbacks of a linear model in the decentralization of a decision process* ", 3rd IFAC Conference on System Approach for Development, Rabat 1980, Proc. pp.1-7.

Dans cet article, on analyse le degré de décentralisation lié à l'applications des méthodes classiques de décomposition à la Programmation Linéaire. En particulier, la méthode de décomposition par les prix de Dantzig et Wolfe (1960) est partiellement décentralisée car les sous-problèmes, étant des programmes linéaires posés sur des polyhèdres fixes indépendants des paramètres de coordination, ne peuvent calculer que des sommets de ces polyhèdres.En fait, l'intérêt de cette analyse (la non décentralisation de la méthode de Dantzig-Wolfe est connue depuis Baumol et Fabian, 1964) réside dans la mise en évidence des causes de cette situation générale, causes liées à la non différentiabilité des fonctions de coordination induites par la décomposition. Cette non différentiabilité implique non seulement la non unicité des solutions de certains sous-problèmes (puisque chaque solution est associée à un sous-gradient de la fonction de coordination) mais aussi la dégénérescence des sous-problèmes dans le cas linéaire.

Ces observations suggèrent que ces non différentiabilités sont également responsables pour les maigres performances des méthodes de décomposition rapportées dans la littérature (cf Dirickx et Jennergren, 1979) presque toujours dans le cadre de la Programmation Linéaire. Cela nous amena à explorer trois directions de recherches distinctes :

- 1 : adapter les méthodes propres à l'optimisation non différentiable aux techniques de décomposition.
- 2 : rechercher une méthode de décomposition complètement décentralisée pour la Programmation Linéaire en éliminant les dégénérescences dans les sous-problèmes.
- 3 : forcer l'unicité des solutions des sous-problèmes par des techniques de régularisation.

[5] "*Price- and -resource directive allocation models in linear programming decomposition methods* " (en collaboration avec P.C. Marques Vieira), International Congress on Mathematical Programming, Rio de Janeiro, 1981.

Cet article est basé sur les résultats de la thèse de Mestrado de P.C.M. Vieira qui a analysé et comparé un grand nombre de méthodes de décomposition des programmes linéaires publiées dans la littérature spécialisée entre 1960 et 1980. Il résultait de cette étude que très peu de travaux allaient dans le sens des trois objectifs décrits plus haut. En fait, la ligne de recherche 1 avait motivé certaines implantations de méthodes de sous-gradients pour des problèmes de localisation et de multiflots (Kennington et Shalaby, 1977, Minoux et Serreault, 1981). Dans l'esprit de la ligne 2, Kydland (1975) proposait une phase d'allocations mixtes pour forcer la décentralisation des sous-problèmes et cela dans le cas d'un couplage dit hiérarchisé (i.e. bloc-triangulaire) et Jennergren (1973) répondait partiellement à la troisième question en introduisant des termes quadratiques dans le coût des sous-problèmes.

Copyright © IFAC 3, System Approach for Development
Rabat, Morocco, 1980

ON THE DRAWBACKS OF A LINEAR MODEL IN THE DECENTRALIZATION OF A DECISION PROCESS

P. Mahey

Department of Electrical Engineering, Pontifícia Universidade Católica, Rio de Janeiro, Brazil

Abstract: In the domain of multilevel analysis, the idea of decentralization has a clearly different meaning from that of reducing the size of an optimization problem by decomposition techniques. It is well known that prices only cannot usually be utilized to coordinate a linear economic system. This paper is an attempt to conceptualize the justifications of decomposition techniques against the powerful linear programming tools available on the market. Starting from the classical Dantzig-Wolfe's algorithm, the drawbacks of the "bang-bang" behaviour of linear models are analyzed and quantified, relatively to the desired autonomy of the subsystems and to the information volume treated at each cycle. The problem of justifying expensive and sophisticated algorithms yielding a coherent decentralization is then posed in consideration to their integration in a human decision-making environment.

Keywords. Decision theory; linear programming; multilevel hierarchical systems.

I. Introduction

We are going to discuss along this paper several aspects of the decomposition of optimization models for decentralized organizations. In particular we want to focus on the well-known "bang-bang" behaviour of the completely linear case when a price directive coordination is designed for both decomposition and decentralization purposes. One might be surprised to find here the subject of an extensively discussed polemic since the publication of Dantzig-Wolfe's algorithm in 1961. Very soon, economists have perceived an analogy between decomposition algorithms and multi-level organizational models and a lot of energy has been wasted in order to build algorithms as faithful reproductions of real situations. The auxiliary problems, termed subprograms, may correspond to individual decision-making units, each with its own objectives and resources. The master problem may be interpreted either as a central planning agency as in a socialist economy or as a competitive market as in a capitalist environment. Each iteration of the algorithm represents an information exchange between master and subprograms. Typically the master program would send down marginal prices of the corporate resources and the subproblems would use these prices to compute a local proposal sent back to the master program which may eventually reconsider its former decision in front of the new demand for corporate resources. Distinctions may be made between two interpretations of the iterative process (Kornai, 1973):

- a preparation of the optimal plan.
- a simulation of the price-adjustment process as the Walras's "tatonnement" process of the everyday control of an economic system does.

Besides all confusions which could have been made between these two interpretations, three principal critics have been adressed to the second:

- The first is that "traditional design models, based upon decomposition are highly abstract representations of reality and ignore vital aspect of real organizational considerations, such as the effects of uncertainty, risk sharing, time and informal communication" (Moore, 1979).
- Secondly, in a real-world market, the excess supply of t th period is added to initial stock of the (t+1) th period. The excess demand of the t th period may not be added fully to the demand of the (t+1) th period but can increase it. In most decomposition algorithms, this problem disappears entirely (Kornai, 1973).
- Thirst, the proposals suggesting decomposition algorithms as normative theories of planning are somewhat unrealistic, since no planning process would be organized along the lines of a mathematical program which could easily reach thousands of iterations. Some essays to simulate only few iterations as a suboptimal process have led to very poor results (Christensen and Obel, 1978).

The main argument which justifies the research of decomposition algorithms to solve socio-economic planning problems might be the increasing complexity of such problems and the supposed computational limitations of centralized optimization. Unfortunately, the interest of the big data processing societies in developing more and more sophisticated packages, principally in the field of linear programming, has led to direct methods which are actually quicker and simpler than any of the decomposition methods.

In presenting some disadvantages of a price-directive coordination of linear subprograms, we want to show how experiments and theoretic research on decomposition algorithms can shed light on the mechanisms of socio-economic processes and we hope that this type of system approach will contribute to the development of efficient decomposition programs. The objective of decentralizing the subsystems, as being one of the most relevant feature of modern organizations, will be the principal criterion for comparing and evaluating the performances of some decomposition algorithms.

The purpose of decentralization is to let the subsystems or divisions of an economy calculate their own optimal decisions such that the set of all the optimal decisions is a feasible optimal solution for the global problem. Arrow and Hurwicz (1960) introduced that property as the result of a perfect competition between the divisions (as in the Pareto optimality), i.e. individual profit maximization leads to the maximization of profits for the whole sector. We shall give some conditions of existence of a decentralized decomposition in the general framework of cooperative equilibriums and then go back to the linear case to study proposed solution of this problem found in the literature, beginning with the famous Dantzig-Wolfe's algorithm.

II. Existence of price-directive coordination.

The problem is that of a competitive market of corporate resources in a general decomposable economy.
Let us consider n centers of decision and denote by x_i, $A_i(x_i)$ and $J_i(x_i)$ respectively the variables of decision, the corporate resource consumption and the cost criterion associated with the ith center. We have supposed that the cost criterion is a function separable of the decision variables, so that the coupling between the n centers may be represented by the corporate resources constraints only:

$$\sum_{i=1}^{n} A_i(x_i) \leq 0$$

We are now able to define the so-called corporative equilibrium problem (Pareto optimality) as a mathematical program, or totally centralized problem (P):

$$(P) \quad \begin{cases} \text{Min} \sum_{i=1}^{n} J_i(x_i) \\ \sum_{i=1}^{n} A_i(x_i) \leq 0 \\ x_i \in X_i \end{cases}$$

where X_i is a technological constraints set associated with the ith center. ($X_i \subset E_i$, definition space of the functional J_i)
A_i is an application from E_i to ψ, normed vector space.

The most natural method to solve this problem by decomposition, is to consider it as a model of a manufacturing company consisting of headquarters and n division, the headquarters fixing iteratively a marginal price for the corporate resources consumption of each division. In mathematical programming, that price is generally the same for all the divisions and could be interpreted as a Lagrange multiplier parametrizing the cost functional associated which each division. Let $u \geq 0$ be such a price. Strictly speaking, $u \in \psi^*$, dual space of ψ.
We define now the ith subproblem, $SP_i(u)$, as:

$$SP_i(u) \quad \begin{cases} \text{Min } J_i(x_i) + \langle u, A_i(x_i) \rangle \\ x_i \in X_i \end{cases}$$

where $\langle \cdot, \cdot \rangle$ denotes duality product between ψ and ψ^* (See Bensoussan et al., 1972).

Let $R_i(u)$ be the set of optimal solution to the ith subproblem. The price-adjustment process or coordination task is effectuated at a superior level (master-program) which modified iteratively the value of u.

Definition 1: The price-adjustment process is called "well-defined" when the following properties are met (Malinvaud, 1967, p. 177).

i) There always exist solutions to the operations according to which the subproblems proposals, the prices and the plan can be determined.

ii) The plan generated by the coordination level is feasible.

Definition 2: The price-adjustment process is called "decentralized" when the following properties are met.

i) The process is well defined

ii) The process converges in a finite number of iterations.

iii) $\exists u^* \geq 0$ such that:

$$\forall x_i \in R_i(u^*), \langle u^*, \sum_{i=1}^{n} A_i(x_i^*) \rangle = 0$$

If these properties are met, it is clear that for $u = u^*$ the set of solutions of the subproblems $SP_i(u^*)$ is optimal for (P) and reversely if $(x_1^*, \ldots x_i^*, \ldots x_n^*)$ is optimal for (P), then $x_i^* \in R_i(u^*) \quad \forall i = 1, \ldots, n$.

In fact we want to characterize conditions of existence of such a decentralization property:

Assumptions: 1) $X = X_1 \times \ldots \times X_n$ is a closed, bounded subset

2) J_i and A_i are continuous on X_i.

Theorem 1: A necessary and sufficient condition for the price-coordination to be decentralized is that the functional $L(x_1,\ldots x_n, u) = \Sigma J_i(x_i) + \langle u, \Sigma A_i(x_i) \rangle$ has a saddle-point on $\{u \geq 0;\ x_i \in X_i,\ i=1,\ldots n\}$ at $(u^*, x_1^*,\ldots,x_n^*)$ and that the functional $L(x_1,x\ldots,x_n,u^*)$ has a unique minimum on X.

Proof: For the elements of the proof of this theorem, we refer to the results obtained by Potier (1972); p. 255). The addition of the condition of unicity of the minimum of $L(x,u^*)$ ensures the process to be well-defined.
We now define the dual function h(u) as:

$$h(u) = \operatorname{Min}_{x \in X} L(x,u)$$

Lemma: h(u) is a concave function of u (for the proof see, e.g., Lasdon, 1970). The condition of unicity means that h is differentiable at u^* with partial derivatives

$$\frac{\partial h}{\partial u}\bigg|_{u=u^*} = \sum_{i=1}^{n} A_i(x_i(u^*))$$

or, equivalently, that $\sum_i A_i(x_i)$ is constant over $R_i(u^*)$, i=1,...,n.
A number of sufficient conditions may be given to guarantee unique minima for L(x,u). The simplest is strict convexity.We can see, also,that strict convexity implies unicity of the solution of the $SP_i(u)$, $\forall u$, $\forall i$=1,...,n.

These results join the general and well-known observation that the complete decentralization will not generally be attained in the linear case (see e.g., Baumol and Fabian,1964; Charnes, Clower and Kortanek, 1967; Lasdon, 1970, p. 162).
This difficulty has led many authors to introduce new modes of coordination based or not on a price-directive iterative process.

III. Coordination procedures for linear decomposition algorithms.

3.1 - The Dantzig-Wolfe's algorithm.

The purpose is not here to develop the decomposition principal of Dantzig-Wolf (See e.g., Dantzig Wolfe, 1961) but to yield a basic framework for our analysis.
The model is that of classical activity analysis, J_i, A_i and X_i being defined linearly in function of the activity levels x_i. We arrive then at the following block-angular structure (fig. 1).

$$\operatorname{Min} \sum_{i=1}^{n} c_i' x_i$$

Subject to:

$$\sum_{i=1}^{n} A_i x_i \leq b_o,$$

$$B_i x_i \leq b_i,\ i=1,\ldots,n$$

$$x_i \geq 0,\ i=1,\ldots,n$$

x_i is an n_i-vector
A_i is a (m_o, n_i) matrix
b_o is an m_o-vector of corporate resources availabilities.
B_i is an (m_i, n_i) matrix
b_i is an m_i-vector

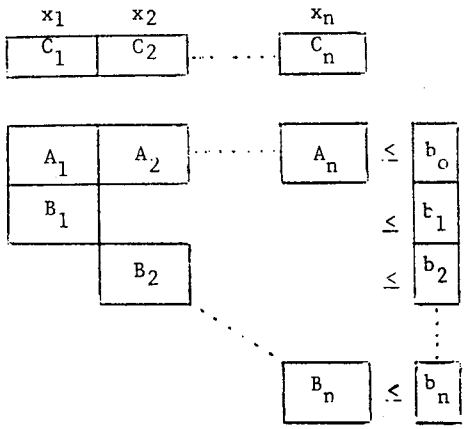


Fig.1-A block-angular decomposable system

Let π^k be the dual m_o-vector associated with the corporate constraints at the k th iteration. The subproblems are defined as

$$\operatorname{Min} (c_i' - \pi^{k'} A_i) x_i$$

Subject to: $B_i x_i \leq b_i$

$$x_i \geq 0$$

As dual variables may generally be identified as marginal prices, we see that the decomposition suggests here a price-directive planning procedure. However, if the subproblems are equivalent to that obtained in a classical tatonnement scheme, the superior level has a quite different task of coordination: instead of making a simple updating of prices as guided by the last observation on excess demands, the Dantzig-Wolfe's method accumulates all the information previously generated and so combines efficient production responses in order to find a possible better structure of prices. Thus the Dantzig-Wolfe's process requires much more work at the coordination level, but it is much more robust in the sense of assuming convergence to optimality.

Then, only the coordination level is able to find a feasible solution at each iteration combining eventually various solutions of the same subproblem. As indicated very soon by Baumol and Fabian (1964) the reason of it is that the solution of the subproblems must be an extreme point of the convex polyhedra $X_i = \{x_i \geq 0;\ B_i x_i \leq b_i\}$, and that the global solution will generally not be the union of such points.
Let us detail that last observation in the light of the theorem of section II.

Let us define the dual function h (u) as:

$$u \geq 0$$

$$h(u) = \underset{x_i \in X_i}{\text{Min}} \{\sum_i c_i' x_i + u'(\sum_i A_i x_i - b_o)\}$$

Suppose now that $\bar{u}$ is such that the n subproblems have a unique solution $x_i(\bar{u})$, extreme point of X_i. The dual function value is then:

$$h(\bar{u}) = \sum_i c_i' x_i(\bar{u}) + \bar{u}' (\sum_i A_i x_i(\bar{u}) - b_o)$$

for all such $\bar{u}$, which is the equation of a hyperplane in the (h,u) space. When u varies in one direction s, some relative cost factors of the nonbasic variables of each subsystem diminish linearly:

$$u = \bar{u} + \rho s, \quad \rho \geq 0$$

$\forall s \in R^{m_o}$, $\exists \rho_1 > 0$ such that: for all $\rho < \rho_1$, $x_i(u) = x_i(\bar{u})$; for $\rho = \rho_1$, one marginal cost of one nonbasic variable of one subsystem (SP_i) reaches zero and the linear subproblem has a nonunique solution; for $\rho > \rho_1$, a pivot operation has modified the solution of the ith subproblem, hence $x_i(u) \neq x_i(\bar{u})$ and $x_j(u) = x_j(\bar{u})$, for $j \neq i$.

Roughly speaking h(u) is a concave piecewise-linear function. Each hyperface of this concave polyhedra is bounded by edges which represent a pivot operation in the current basis of one of the subproblems. For $u = \bar{u} + \rho_1 s$, the solution of one subproblem is not unique and then h(u) is not differentiable.
*Observe that a good choice for s might be:

$$s_j = \text{Min}\{0, \sum_i A_{ij} x_i(\bar{u}) - b_{oj}\}, j = 1, \ldots, m_o.$$

The solution of the saddle-point problem corresponds to $\underset{u \geq o}{\text{Max}}\, h(u)$, hence to a vertex of the concave polyhedra. But an extreme-point of this polyhedra is the intersection of $m_o + 1$ hyperfaces, hence the price-adjustment process will reach the optimal solution with at most m_o simultaneous pivot operations on the basis of the subproblems. Observe that we need not have exactly m_o changes of basis since some of the hyperfaces supported by the hyperplanes $u_j = 0$ could contain the optimal solution.

Then a maximum number of m_o subproblems cannot be decentralized by the price-coordination method. Each time a marginal optimal price is zero, then this number diminishes of one unity (A zero price means a loose corporate constraint, hence a weaker coupling between the subsystems).

Let us go back to the Dantzig-Wolfe master-program with n convexity constraints (Lasdon, 1970, p. 153). A master basis has $(m_o + n)$ rows, hence no more than $m_o + n$ subproblems must be included in the solution, this means that m_o extreme points are to be distributed among the n subproblems. Therefore, a maximum number of m_o subproblems will have more than one extreme point in the global solution.
These two parallel results lead to the following assumption: The dual vector π^k generated by the Dantzig-Wolfe's master program corresponds to an extreme-point of the concave polyhedra of the global dual function h(u), at the optimal solution only.

To show the usefulness of the h(u) function, let us consider the following case. Suppose that there exists i such that $\eta_i \leq \eta$.

Then the linear system:

$$A_i' u + c_i = o$$

may have a nonnegative solution. This means that, if the procedure generates such a price-vector, the ith divisional decision is indifferent to the price allocation and the optimal solution will be interior to the constraint set X_i. Charnes et al. have observed this phenomenon (1967). Let us illustrate this result by a simple example:

Let us consider the following two-blocks decomposable structure:

$$\text{Min } -2x_1 - x_2 - x_3 - y_1 - \frac{1}{2} y_2$$

$$\begin{aligned}
\text{Subject to: } & x_1 + x_2 + 2x_3 + y_1 \leq 4 \\
& 3x_1 - x_3 + 2y_1 + 3y_2 \leq 9 \\
& x_1 + x_2 + x_3 \leq 3 \\
& 2x_1 - x_2 + x_3 \leq 2 \\
& y_1 + y_2 \leq 2 \\
& x_1, x_2, x_3, y_1, y_2 \geq 0
\end{aligned}$$

The polyhedra X_1 and X_2 associated with the x-block and the y-block are shown on Fig.2.a. and fig. 2.b. respectively.

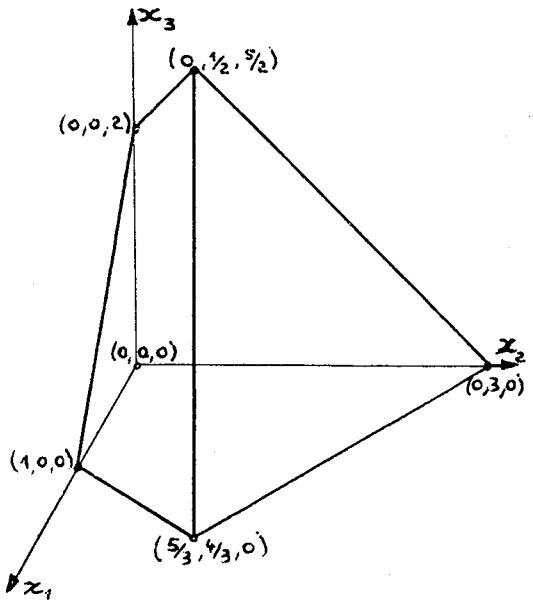


Fig. 2.a

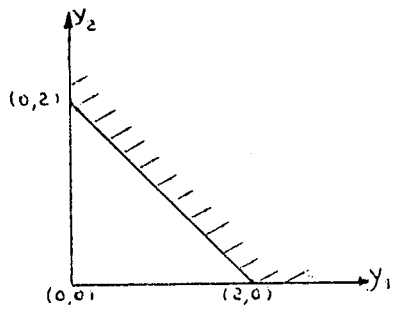


Fig. 2.b

For this simple example, we have:

$n = 2$
$n_0 = 2$
$n_1 = 3$
$n_2 = 2$

Following the notations of section II, the dual function h(u) is:

$$h(u) = \min_{x \in X_1} \{u_1+3u_2-2)x_1+(u_1-1)x_2+(2u_1-u_2-1)x_3\}+$$
$$+\min_{y \in X_2} \{(u_1+2u_2-1)y_1+(3u_2-\tfrac{1}{2})y_2\}-2\lambda_1-\alpha\lambda_2$$

In the plane (u_1,u_2), the frontiers of the regions for which the solutions of the subproblems remain unchanged are shown of fig.3. These solutions define for each region a function h(u):

Region	x_1	x_2	x_3	y_1	y_2	h(u)
1	5/3	4/3	0	2	0	$(5-\alpha)u_1$
2	0	1/2	5/2	2	0	$(\frac{15}{2}-\alpha)u_1 - \frac{15}{2}u_2$
3	5/3	4/3	0	0	0	$(3-\alpha)u_1 - 4u_2$
4	5/3	4/3	0	0	2	$(3-\alpha)u_1 + 2u_2$
5	0	3	0	0	0	$(3-\alpha)u_1 - 9u_2$
6	1	0	0	0	0	$(1-\alpha)u_1 - 6u_2$
7	1	0	0	0	2	$(1-)u_1$
8	0	1/2	5/2	0	0	$(\frac{11}{2}-\alpha)u_1 - \frac{23}{2}u_2$
9	0	0	0	0	0	$-\alpha u_1 - 9u_2$
10	0	0	2	0	0	$(4-\alpha)u_1 - 11u_2$

Now, if we choose $\alpha=2$, it is easy to observe that the maximization of h(u) in the regions 3,4,6 and 7 leads to the point A. As h(u) is concave, A is then the saddle-point solution of this problem.
At this corner-point, the solution of the first subproblem lies on the edge which links the extreme points $(\frac{5}{3}, \frac{4}{3}, 0)$ and (1,0,0), and the solution of the second subproblem lies on the edge (0,0)-(0,2). Hence both cannot be decentralized at the optimal solution.

If now we set $\alpha=4$, the saddle-point is in B, which is exactly the intersection of the three dotted lines. The optimal solution is then such that the first subproblem is well decentralized with $x^* = (\frac{5}{3}, \frac{4}{3}, 0)$, but the cost function of the second subproblem is zero and the solution may be interior to X_2. This occurs because $n_2 = n_0$.

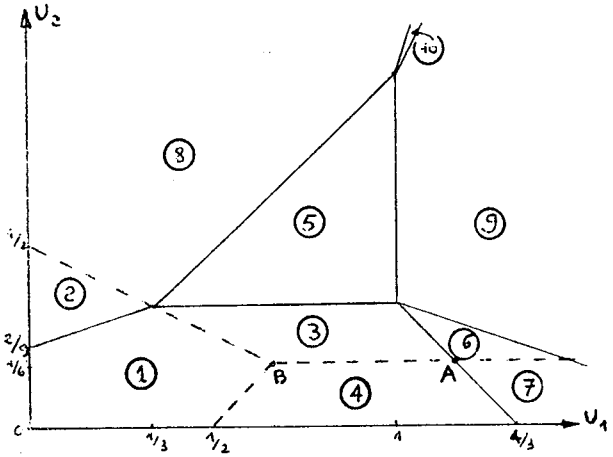


Fig. 3: The darkened lines limit the regions with x(u) constant; the dotted lines limit the regions with y(u) constant.

3.2 - Modifications of the price-adjustment process.

Baumol and Fabian (1964) concluded their analysis about Dantzig Wolfe's algorithm by suggesting the use of suitable nonlinearities in order to avoid the "bang-bang" behaviour of linear models. The price-schedules algorithm of Jennergren (1973) provides us a good example of the choise of these nonlinearities. Jennergren's idea was to associate to each corporate resource a linearly increasing price schedule rather than a constant price. The prices effectively considered by the subdivision manager will depend on the quantity of each resource purchased or delivered by his division. The subproblem can there be formulated as:

$$\text{Min } (C_i - ((\pi^k)' + r\, x_i' A_i')A_i)x_i \qquad (3.1)$$

subject to: $x_i \in X_i$

This problem has now a quadratic objective function and Jennergren proved that this procedure is well-defined and decentralized. The headquarters compute the new π^{k+1} as:

$$\pi_j^{k+1} = \text{Max}\{0; \pi_j^k + \alpha[\sum_i A_{ij}\, x_i(\pi^k) - b_{oj}]\} \qquad (3.2)$$

The process converges to an optimal solution in an infinite number of iterations and precious information on the lower bound of the cost function may be used to stop the procedure for ε-optimality.

The addition of this schedule to the normally constant prices may be interpreted as the addition of preemptive goals in the divisional decision task. Charnes et al. (1967) have introduced this notion in a fundamental paper about coherent decentralization. These goals could be added in the objective function or in the subproblem constraints. The former case leads to subproblems of the form:

$$\text{Min } (C_i - \pi^{k'} Ai)x_i + M||\alpha_i^k - A_i' x_i|| \qquad (3.3)$$

Subject to: $x_i \in X_i$

where α_i^k, i= ,...,n, is a partition of the resource vector b_o such that:

i) $x_i \in X_i$ such that $\alpha_i^k = A_i x_i$, for all i.

$$\text{ii) } \sum_{i=1}^{n} \alpha_i^k \leq b_o \qquad (3.4)$$

$$\text{iii) } \sum_{i=1}^{n} \pi^{k'} \alpha_i^k = \pi^{k'} b_o \qquad (3.5)$$

α_i^* is then an allocation of resource forced by the superior level. As Charnes et al. have observed, this allocation could be directly affected to the constraints of the subproblem, modifying the polyhedra X_i:

$$\min (C_i - \pi^{k'} A_i) x_i$$

subject to:

$$\begin{aligned} B_i x_i &\leq b_i \\ A_i x_i &= \alpha_i^k \\ x_i &\geq 0 \end{aligned} \qquad (3.6)$$

The task of the master-program will be now, not only the determination of the prices π^k, but too the computation of the allocation vector α_i^k. We may observe that, if nonunicity affects the solution of the subproblems, the constraint (3.6) ensures that the process is well-defined. Then, with the choice of "coherent" partions as defined by Charnes (p. 312), the process will be decentralized.

Various attempts have been made to yield coherent allocations combined with well-defined prices and maybe one of the most powerful is the algorithm built by Sengupta and Gruver (1974) which was truly applied to some educational planning problems. In this algorithm the subproblems have the form:

$$\text{Min } (C_i - \pi^{k'} A_i) x_i$$

$$\text{Subject to: } \begin{aligned} B_i x_i &\leq b_i \\ A_i x_i &\leq \alpha_i^k \\ x_i &\geq 0 \end{aligned} \qquad (3.7)$$

This approach is original in the sense that the master-program will be able to identify the subproblems which have a high potential for changing the convex combination of former proposals in a profitable way. Thus, if the greater positive value associated with the jth element of (3.7) (the jth corporate resource) occurs for subproblem i, then an arbitrary large quantity will be allocated to the jth element of α_i^k, allowing the ith subproblem to form a new proposal which uses a greater amount of the jth corporate resource.

We have seen then two general modes of modifying the coordination by prices only. But, as far as decentralized optimization is concerned, both "nonlinearization" of the cost functions and combined allocation of resource have a very similar meaning = considering the imposibility inherent to the linear structure to reach a completely decentralized optimum, the solution generally adopted was to give to the decision centers some additional information about the consequences of their own decisions. It is well known that resource allocation in the constraints is a dual method of transfer pricing. Geoffrion (1970) speaks of dual price-directive and primal resource-directive approaches. In the hierarchical control theory, the terms of unfeasible goal coordination and feasible model coordination are employed (Titli,1975). This is not our objective to look for the good choice between price and resource-allocation since no definite answer to that question could be assumed. Some resource-directive algorithms have seemed to work well in some special economic cases such as capital budgeting allocation method of Maier and Vanderweide (1976), but Moore (1979) has shown under some empirical experiments that the hypothese that budgeting would outperform transfer pricing is not confirmed. Finally, the study of all types of algorithms, primal or dual, which have yielded truly decentralized decision would not be complete without mentioning the min-max algorithms, which we may classify as primal-dual algorithms. These algorithms simply investigate the solution of the saddle-point problem defined in section II. Potier (1972) has developed an adaptation of Uzawa's min-max algorithm to the linear decomposed problem. It seems that the direction-finding procedure $u=u+\rho s$ would accelerate the Dantzig-Wolfe's iterative process in the early iterations. A modified steepest-ascent algorithm has been confronted with the models of Jennergren, Sengupta-Gruver and Dantzig-Wolfe in a large-scale production-inventory system. Unfortunately, the diversity and the heterogenous character of these results difficult their publication inside this paper.

IV. Concluding remarks

The laboratory experiments made by Moore(1979) are very useful in the sense that they show us how careful we must be in the search of algorithms which would lead to perfect decentralization in an on-line simulation. Since unmodellable factors such as organizational design and time pressure constraints strongly influence the decision making behaviour, it has little sense to search for an algorithm whose iterations are all better than the sequential decisions of the tatonnement process. Undoubtly, the decomposed algorithms work as an aid to the decision-makers and not as their substitutes. If the mathematical linear decomposed programs cannot lead to complete decentralization, it is not a failure inherent to their mechanisms, but in the contrary, they may give to the decision-makers much information besides the simple task of decentralizing. The study of the dual function

is an example of the powerful tools issued from mathematical programming which economists or manager could use to know better the structure of the systems in which they are integrated.

REFERENCES

Arrow, K. and L. Hurwicz (1960). Decentralization and computational in resource allocation. In R. Phouts (Ed.), Essays in Economics and Econometrics, University of North Carolina Press, Chapel Hill, pp. 34 - 104.

Baumol, W. J. and T.Fabian (1964). Decomposition, pricing for decentralization and external economics. Management Sci., 11, 1 - 32.

Bensoussan, A., J.L.Lions and R. Teman(1972). Sur les méthodes de décomposition, de décentralisation et de coordination et applications. Cahiers de l'IRIA, 11, nº 2, 5 - 189.

Charnes, A., R.W. Clower and K. O. Kortanek (1967). Effective control through coherent decentralization with preemptive goals. Econometrica, 35, 294 - 320.

Christensen, J. and B. Obel (1978).Simulation of decentralized planning in two danish organizations using mathematical programming decomposition. Management Sci., 24, 1658 - 1667.

Dantzig, G.B. and P. Wolfe (1961). The decomposition algorithm for linear programs. Econometrica, 29, 767 - 778.

Geoffrion, A.M. (1970). Primal resource-directive approaches for optimizing nonlinear decomposable systems. Oper.Res., 18, 375 - 403.

Jennergren, L.P. (1973). A price schedules decomposition algorithm for linear programming problems. Econometrica, 41, 965 - 980.

Kornai, J. (1973). Thoughts on multi-level planning systems. In L.M. Goreux and A. S. Manne (Eds.), Multi-level planning: Case Studies in México, North Holland, Amsterdam, pp. 521 - 548.

Lasdon, L.S. (1970). Optimization Theory for Large Systems. Mac Millan, New York.

Maier, S.F. and J.H. Vander Weide (1976) Capital budgeting in the decentralized firm. Management Sci., 23, 433 - 443.

Malinvaud, E. (1967). Decentralized procedures for planning. In E. Malinvaud and M.O.L. Bacharech (Eds.), Activity Analysis in the Theory of Growth and Planning, Mac Millan.

Moore, J.H. (1979). Effects of alternate information structures in a decomposed organization: a laboratory experiment. Management Sci., 25, 485 - 497.

Potier, D. (1972). Algorithmes de coordination.Application à la gestion d'unités de production interdépendentes. Cahier de l'IRIA, 11, nº 2, 241 - 375.

Sengupta, J.K. and G.W.Gruver (1974). On the two-level planning procedure under a Dantzig-Wolfe decomposition. Int. J. Syst. Sci., 5, 857 - 875.

Titli, A. (1975). Commande Hierarchisée et Optimisation des Systèmes Complexes. Dunod Automatique, Paris.

CHAPITRE 3

METHODES DE SOUS-GRADIENTS ET APPLICATIONS

Les fonctions de coordination expriment l'effet global des paramètres de coordination sur les sous-problèmes. Leur forme générale est donc, a étant un vecteur de paramètres de coordination :

$\phi(a) = \Sigma\, \phi_i(a_i)$ où chaque ϕ_i est de la forme :

$$\phi_i(a_i) = \underset{x_i}{\text{Min}} \{ f_i(x_i, a_i) \mid x_i \in S_i \}$$

Calculer ϕ_i en a_i revient donc à résoudre le sous-problème (S_i) avec les paramètres fixés aux valeurs a_i. Nous nous sommes intéressés tout d'abord au cas où ces fonctions sont convexes (ou concaves) et sous-différentiables. On suppose de plus que le coût de calcul d'un sous-gradient de ϕ_i est faible. C'est le cas, par exemple, des méthodes de décomposition par les prix pour les problèmes séparables avec contraintes de couplage :

$$(S) : \text{Minimiser} \sum_{i=1}^{p} f_{0i}(x_i)$$

$$\sum_{i=1}^{p} f_{ki}(x_i) \le 0 \ , \ k=1,\ldots,q$$

$$x_i \in S_i, \ i=1,\ldots,p$$

Les paramètres de coordination sont ici les variables duales associées aux q contraintes de couplage. On les notera a_k, k=1,...,q, et on observe que le même vecteur de paramètres est transmis à tous les sous-problèmes. On a alors les sous-problèmes :

$$(S_i) \quad : \phi_i(a) = \text{Min} \{ f_{0i}(x_i) + \sum_{k=1}^{q} a_k f_{ki}(x_i) \mid x_i \in S_i \}$$

La fonction ϕ est concave si chaque f_{ki}, k=0,...,q, i=1,...,p, est convexe et chaque S_i est un convexe fermé de X_i, i=1,...,p. Elle est en général sous-différentiable et si $x_i(a)$ est une solution optimale du sous-problème (S_i), alors $g = (g_1, \ldots, g_q)$ avec $g_k = \sum f_{ki}(x_i(a))$ est un sous-gradient de ϕ en a. On voit immédiatement que, si (S) est un programme linéaire, ϕ est concave et linéaire par morceaux donc presque

sûrement non différentiable en son point de maximum.

Les méthodes de sous-gradient étudiées par des chercheurs russes de l'école de Kiev dans les années 60 consistent à mettre à jour les paramètres a_k dans la direction d'un sous-gradient de ϕ en effectuant des 'petits pas'. La théorie (cf Shor, 1979, ou Minoux, 1983) nous dit que ces pas doivent tendre vers zéro, mais pas trop vite.

Pourquoi choisir ces méthodes réputées lentes et incontrôlables ? Les raisons principales sont :

- le faible coût d'implémentation peu sensible à la dimension du problème de coordination
- la trajectoire en zig-zags de la suite des itérés de part et d'autre des surfaces de non différentiabilité de la fonction.

La première raison signifie que ces méthodes sont bien adaptées aux problèmes 'durs' dans lesquels le nombre de paramètres de couplage est très grand. C'est le cas, par exemple des problémes qui proviennent de relaxations linéaires de certains problèmes combinatoires

La deuxième raison, illustrée sur la figure 1, montre que, si les directions successives utilisées sont de mauvaise qualité, les petits pas effectués autour des points de non différentiabilité permettent d'identifier plusieurs points extrêmes du sous-différentiel et donc de reconstituer tout ou partie de l'information nécessaire à la convergence du processus. Cette idée suggère l'emploi des méthodes de sous-gradients combinées avec d'autres approches qui garantissent une convergence rapide (finie dans le cas linéaire) quand elles disposent d'une description complète du sous-différentiel, mais dépensent un grand nombre d'itérations à reconstituer ces informations.

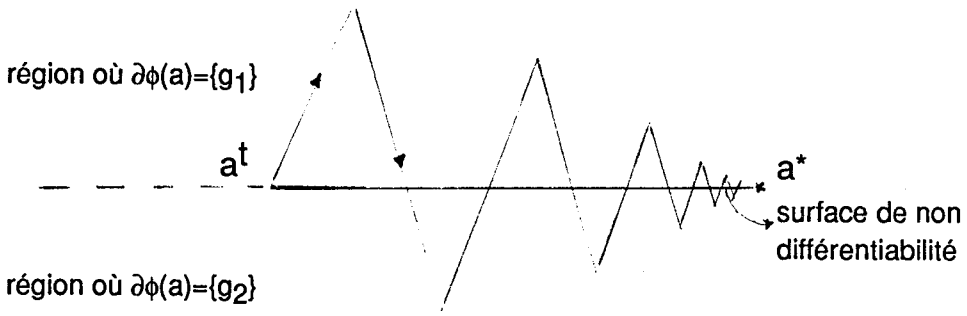


Fig. 1 Comportement typique d'une méthode de sous-gradients

[6] "*A subgradient algorithm for accelerating the Dantzig-Wolfe method* ", Methods of Operations Research, 53, 1986, pp. 697-707.

Dans ce travail, on applique une méthode de sous-gradients à la fonction duale ϕ décrite ci-dessus. Dans une première phase, cette méthode permet d'approcher la solution optimale duale a*. Dans une deuxième phase, les itérations prennent leur allure typique en zig-zags dans la région de l'optimum où le degré de non différentiabilité est dans

un sens maximum. Au cours de cette deuxième phase, on néglige la recherche de l'optimum ce qui permet de maintenir le pas de sous-gradient constant et on met en mémoire les différents sous-gradients calculés à chaque itération. Finalement, une dernière phase consiste à monter une base de départ du programme-maître de Dantzig-Wolfe avec les sous-gradients accumulés dans la phase 2. Cet algorithme est testé sur un modèle de gestion de production adapté de Potier (1972).

Un des points importants soulevés par cette étude est l'explication de la lenteur de convergence de la méthode de Dantzig-Wolfe. Cette méthode équivaut en fait à appliquer une technique de plans de coupe à la maximisation de la fonction ϕ qui consiste à remplacer ϕ par des fonctions ϕ_t, enveloppes inférieures des supports affines de ϕ associés aux sous-gradients générés jusqu'à l'itération t. On en déduit que :

- la valeur maximum de ϕ_t obtenue en a^t est un majorant de la valeur optimale $\phi(a^*)$. De plus, la suite de ces valeurs est monotone décroissante.
- la suite des distances $\{||a^t-a^*||\}$ tend vers zéro mais n'est pas monotone.

Ce dernier point rend la convergence très lente (de fait, sous-linéaire). On peut opposer ces deux propriétés aux deux caractéristiques des méthodes de sous-gradients :

- la suite des valeurs $\{\phi(a^t)\}$ tend vers $\phi(a^*)$, mais n'est pas monotone.
- la distance à l'optimum $||a^t-a^*||$ est monotone décroissante.

En résumé, il est possible d'améliorer les performances de ces deux approches complémentaires en les combinant de manière efficace. Des variantes d'autres méthodes classiques de décomposition ont fait également l'objet de tests numériques dans le cadre de travaux de fins d'études à l'Université Catholique de Rio de Janeiro.

Toute la difficulté d'implémentation des méthodes de sous-gradient réside dans le choix du pas de déplacement à chaque itération. L'efficacité de l'algorithme dépend en fait fortement de certaines caractéristiques géométriques de la fonction comme la condition qui estime le plus grand angle entre un sous-gradient et la direction optimale. Les choix possibles et leurs implications dans les applications aux problèmes d'origine combinatoire d'abord puis, aux approches par décomposition, ont été étudiés dans les deux publications suivantes :

[7] "*Subgradient techniques and combinatorial optimization* ", Working paper, Université de Bonn, WP 85397, 1985.

[8] "*Otimizaçao nao diferenciavel nos métodos de decomposiçao* ", SBA- Controle e Automaçao, 1, 2 (1987), pp. 154-161 (en portugais).

Le choix de pas resté le plus populaire dans les applications récentes de ces méthodes (du moins en occident) est celui préconisé par Held et

Karp (1971) dans un article pionnier sur la relaxation lagrangienne d'un modèle du problème du voyageur de commerce. L'idée revient à estimer la projection de la solution optimale sur la direction du sous-gradient grâce à une estimation de la valeur optimale (figure 2).

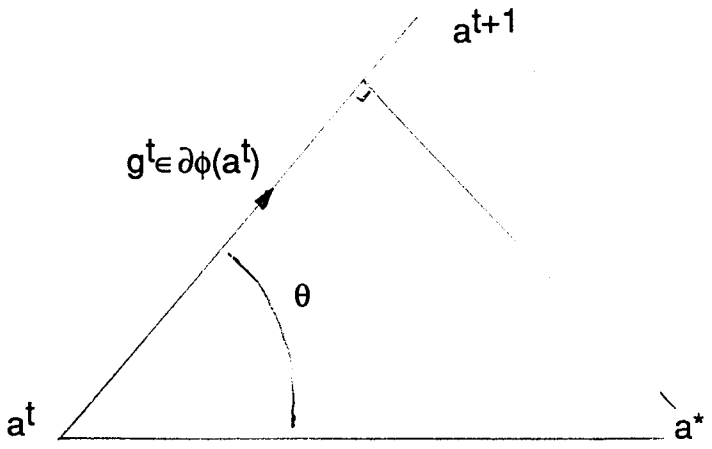


Fig. 2 Estimation du pas optimal :
$\phi(a^*) \leq \phi(a^t) + ||g^t||\ ||a^*-a^t||\cos\theta$
$\Rightarrow ||a^{t+1}-a^t|| \geq (\phi(a^*) - \phi(a^t)) / ||g^t||$

La valeur optimale $\phi(a^*)$ n'étant généralement pas connue, on doit la remplacer par une estimation. La dualité a la propriété fondamentale de fournir des bornes de la valeur optimale. Nous avons donc construit une nouvelle méthode de type primal-dual pour améliorer la stabilité numérique et les performances de la méthode de Held et Karp.

[9] "*Decomposition of large-scale linear programs by subgradient optimization* ", Mat. Aplic. Comput. 1, 2, 1982, pp. 121-134.

L'algorithme proposé dans cet article est composé de deux sous-algorithmes travaillant séparément sur les fonctions de coordination primale et duale. Chaque fonction est traîtée par une méthode de sous-gradient et les meilleurs bornes primale et duale obtenues au cours des itérations sont utilisées comme estimations de la valeur optimale dans le calcul des pas de sous-gradients. En plus de l' intérêt évident d'améliorer ces estimations au cours des itérations, ce qui accélère la convergence, l'écart entre ces deux bornes fournit un critère d'arrêt qui fait défaut à la plupart des méthodes de sous-gradient. Les résultats ont été très concluants malgré l'augmentation du coût de calcul dû à la double résolution des sous-problèmes avec des prix ou avec des ressources. De plus, une implémentation en parallèle (théorique, car nous ne disposions pas de machines multiprocesseurs) semble être la mieux adaptée à cette approche.

Finalement, les méthodes de sous-gradients ont l'avantage d'être très flexibles dans la mesure où elles s'intègrent facilement dans un algorithme existant et sont peu sensibles à la dimension du problème. Par contre, elles restent des méthodes très lentes (on ne peut espérer des

vitesses de convergence meilleures que des convergences linéaires de taux 0.7). L'investissement vers des méthodes plus performantes comme les méthodes de faisceaux introduites par Lemaréchal (1975) n'a été fait que tardivement dans le cas des techniques de décomposition avec la thèse de Medhi (1987) à l'Université de Wisconsin-Madison, USA. Un autre aspect que nous avons analysé du point de vue théorique est l'analogie entre les fonctions de type 'max', qui constituent le gros des fonctions non différentiables rencontrées dans les méthodes de décomposition, et la programmation non linéaire :

[10] "*About the generalized gradient of max-functions* ", Xème Congrès Brésilien de Mathématiques Appliquées, Gramado, 1987.

On y démontre que le sous-espace affine engendré par le sous-différentiel d'une fonction max est orthogonal au sous-espace tangent à la surface de non différentiabilité (figure 3). De plus, la projection de l'origine sur ce sous-espace affine (moins chère à calculer que la projection sur le sous-différentiel lui-même) est toujours une direction de descente pour la fonction. Ce résultat géométrique semble ne pas avoir été exploité pour la construction d'algorithmes de descente en optimisation non différentiable.

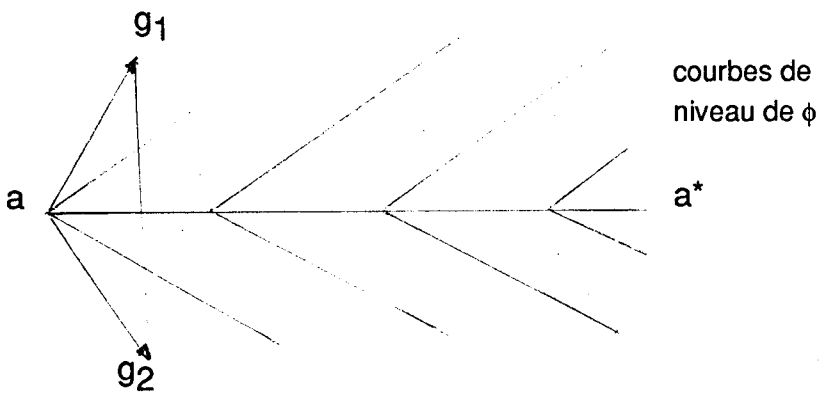


Fig. 3 Calcul d'une direction de descente par projection

A SUBGRADIENT ALGORITHM FOR ACCELERATING THE DANTZIG-WOLFE DECOMPOSITION METHOD

Philippe MAHEY, Rio de Janeiro, Brazil

ABSTRACT

In this paper, , we propose to accelerate the convergence of classical decomposition methods by a subgradient algorithm. The case of Dantzig-Wolfe's algorithm applied to large scale block-angular linear programs is analyzed and we show how the information brought by a chain of subgradients can improve its performance.

I. INTRODUCTION

Subgradient techniques have gained a large acceptance in the past fifteen years, mainly for the approximation of large scale combinatorial problems. Lagrangian relaxation is the main tool to transform large or hard problems in easier subproblems and the price to pay is the non differentiability of a typical dual function we have to maximize to get a lower bound or sometimes an exact solution to the original problem.

Since the early papers by Held and Karp ([1]) and Held, Wolfe and Crowder ([2]), , many implementations of subgradient algorithms have contributed successfully to the solution of problems hardly solvable by standard methods. Many people have used subgradient techniques as a basis for comparisons with heuristics or other more sophisticated algorithms and, curiously, few information is reported in the literature on its relative importance between other methods for large-scale problems (see for example, Gondran and Minoux [3] ,Shapiro [4] and Mahey [5]).

Decomposition methods for large-scale linear programming lead too to nonsmooth optimization problems when treated by Lagrangian relaxation or even by primal decomposition or partitio-

ning. Actually, some large-scale hard combinatorial problems possess an underlying block-separable structure which may be exploited by decomposition (this is the case for example of multicommodity maximum flow problems). Very few attempts were made to implement subgradient algorithms in the case of general block-angular linear programs (see for example Gondran and Minoux [3] and Mahey [5]). This is maybe a reflex of a kind of mistrust towards decomposition methods which have brought serious prejudice to classical methods as the Dantzig-Wolfe's methods since its publication in 1960 [7] to the benefit of direct techniques as sparse matrix methods, LU factorization , multiple pricing, all of these developed in the shadow of the industry for large computers.Consequently, it seems difficult to defend slow approximation schemes as are known to be the subgradient algorithms when the finitely convergent methods are known to have weak performance.

In this paper, we analyze the implementation of subgradient methods to large-scale block-angular linear programs and then discuss the inconvenients and slow behaviour of Dantzig - Wolfe's algorithm. We then propose a general scheme consisting in combining the two approaches in a sequential way. This idea is not new, though it has not clearly lead to successful implementations in the applications. It lies on some experimental observations of the subgradient method, which has generally shown to come close to the solution quite quickly independently of the scale of the problem and then slow down as the increasing nonsmoothness around the solution tends to increase its oscillating behaviour. The Dantzig-Wolfe's method have been often reported to be slow too when getting close to the solution, but we try to show how the first phase will bring some precious information to the last phase improving the global performance of the decomposition algorithm. Computational tests on come classical production planning problems have confirmed these results.

II. SUBGRADIENT ALGORITHMS AND DECOMPOSITION METHODS.

As we are mostly interested in accelerating the convergence of Dantzig-Wolfe's method, we shall limit our study to dual

decomposition schemes (known as price-diretive schemes in operations research). Let us then consider a block-angular linear program:

$$\text{Minimize} \quad \sum_{i=1}^{n} c_i^T x_i$$

$$(P) \quad \Big| \sum_{i=1}^{n} A_i x_i = b \qquad (1)$$

$$x_i \in S_i, \; i = 1,\ldots,n$$

where $x_i \in R^{n_i}$, $A_i(m_o \times n_i)$ and S_i is a bounded polyhedron in R^{n_i}:

$$S_i = \{x_i \in R^{n_i} \mid B_i x_i = b_i, \; x_i \geq 0\} \qquad (2)$$

To relax the coupling constraints (1), we introduce a vector of dual variable $u \in R^{m_o}$ and for each u, the Lagrangian relaxation of (P) splits into n sub-problems:

$$\text{Minimize} \; (c_i^T + u^T A_i) x_i$$

$$(SP_i) \quad \Big| \; x_i \in S_i$$

Let $X_i(u)$ be the set of optimal solutions in R^{n_i} of (SP_i) and $v_i(u)$ its optimal value. Then we may associate to (P) an equivalent dual problem:

$$(D) \qquad \text{Maximize} \quad v(u) = \sum_{i=1}^{n} v_i(u) - u^T b$$

It is well known that v(u) is a concave piecewise linear function and the set of subgradiets in each u, $\partial v(u)$, is easily obtained:

$$\partial v(u) = \{g \in R^{m_o} \mid g = \sum_{i=1}^{n} A_i x_i - b, \; x_i \in X_i(u)\} \qquad (3)$$

We shall analyze two specific approaches to solve (D), the subgradient method and the cutting-plane method, the latter leading to Dantzig-Wolfe's algorithm.

Subgradient algorithms have been first studied at the Cybernetics Institute of Kiev in the 60's where most theoretical

results appear in the works of Shor, Ermoliev, Polyak, Nurminski among others. The general algorithm consists at iteration k in picking up one subgradient g^k in $\partial v(u^k)$ (obtained easily with a particular optimal solution in each (SP_i)) and updating the dual vector by:

$$u^{k+1} = u^k + \lambda_k \, g^k / \| g^k \| \tag{4}$$

The main fact is that g^k is not necessarily an ascent direction for v and that we must consequently decrease the step size λ_k in order to converge. A general simple condition which ensures convergence to the optimal solution of (P) is given in Polyak [8] . This is:

$$\lambda_k \to 0 \text{ and } \Sigma_k \, \lambda_k \to + \infty \tag{5}$$

Unfortunately, this condition implies a very slow rate of convergence for (4), indeed never faster than the rate of decrease of λ_k which is not even geometric. The step size strategy in (4) is based mainly on the fact that g^k is a descent direction for the optimal distance, $d_k = \text{Inf}\{\| u - u^k \|, \; u \in V^*\}$, where V* is the set of optimal solutions for (D) (see [9] for more details).

The simplest choices which yield a geometric rate of convergence under some rather mild conditions are:

i) $\lambda_k = \lambda_o \, \rho^k$ (6)

where $\lambda_o > 0$ and $0 < \rho < 1$

ii) $\lambda_k = \omega_k \dfrac{\bar{v} - v(u^k)}{\| g^k \|}$ (7)

where $0 < \omega_k < 2$ and $\bar{v}$ is an estimate for the optimal value v*.

The formula (6) is known as Shor's method or the convergent serie method (in opposition to (5)) and (7) is known as the relaxation method. Goffin([10] , [11]) has studied the rates of convergence of these methods which depend critically on a condition number γ measuring the bad conditioning of the concave func

tion v(u).

$$\gamma = \text{Inf Min } \{ \frac{g^T(u^* - u)}{\|g\| \, \|u^*-u\|}, \; g \in \partial v(u), \; u^* \in V^* \} \qquad (8)$$

We resume below the main conclusion of Goffin's analysis:

- λ_o must be an estimate for $d_o \gamma$.
- ρ cannot be smaller than $\sqrt{1 - \gamma^2}$ when $\gamma \leq \sqrt{\frac{2}{2}}$ which is the practical situation.
- If $\bar{v}$ is an underestimate of v^*, then finite convergence to a point such $v(u) > \bar{v}$ is expected.
- If $\bar{v}$ is an overestimate of v^*, then we must decrease ω_k.

We refer to Minoux[9]for some additional comments on acceleration techniques based on information about successive subgradients.

As λ_o and γ are quite difficult to estimate, the relaxation strategy has long seemed best suited to the applications . In fact, as $\bar{v}$ is mostly an overestimate (for example the slowest value for the previously found feasible primal points), we must define ω_k as a convergent serie and the problem is the same as with Shor's methods.

III. A FIRST PHASE TO DANTZIG-WOLFE'S METHOD

The Dantzig-Wolfe's algorithm DW is equivalent to substitute (D) by a partial outer-linearization of v(u) using the past subgradients:

$$v(u) = \text{Inf } \{ v(\bar{u}) + g^T (u-\bar{u}) \mid \forall \bar{u} \in R^{m_o}, \; g \in \partial v(\bar{u}) \} \qquad (9)$$

Then we approximate (D) at iteration k by:

Maximize w

$$(DW) \mid w \leq v(u^t) + g_t^T (u - u^t), \; g_t \in \partial v(u^t)$$

$$, \; t = 1, \ldots\ldots, k-1$$

The optimal solution for (DW) is then u^k and the dual solution yields the weights of the convex combination of the ac

tive subgradients to get a primal feasible point. (DW) is then a kind of cutting-plane strategy. Hence, we can explain in part its slow behaviour as a consequence of the instability of suc - cessive solutions (Fig.1).

It is worth observing the complementary behaviour of the subgradient algorithm and of the DW algorithm:

	Subgradient	DW
Monotonicity of objective function	No	Yes
Monotonicity of optimal distance	Yes	No

Attempts to annihilate the lack of monotonicity as the ε-ascent subgradient, or bundle methods, ([12]) , for the first class and the Boxstep method ([13]) for the second class have given relativelyfew positive results because of their high cost of implementation.

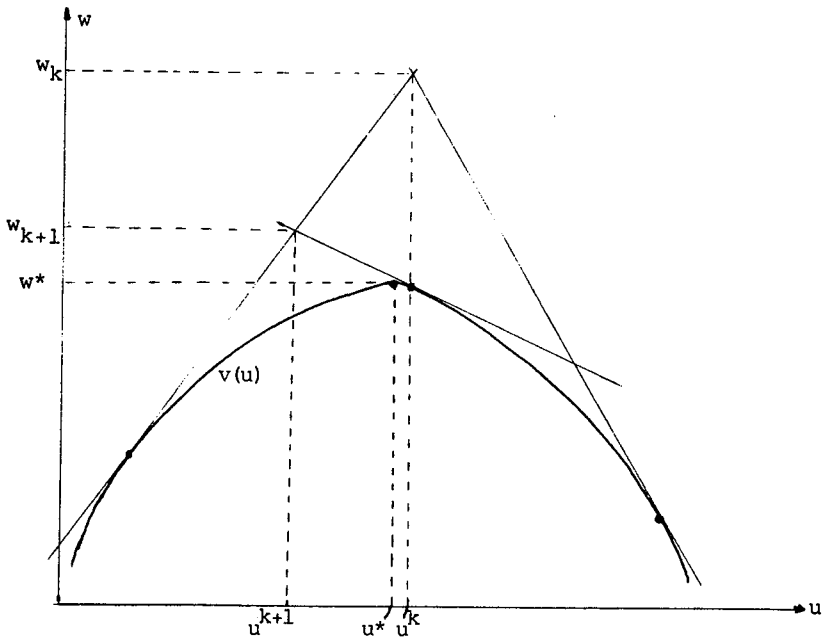


Fig. 1 - Non monotonicity of optimal distance with cutting-plane method.

We shall now get an insight into problem (P) and define the concept of active extreme points at the optimal solution. We mean by that the extreme points of each polyhedron S_i which have a strictly positive weight in the optimal convex combination that gives the optimal solutions x_i^* , i= 1,.....,n. If problem (P) is not degenerate, then the number of active extreme points is equal to m_o+ n. We observe that if these active extreme points are known, then the master program of the Dantzig-Wolfe's algorithm gives us the optimal weights , i.e.,the optimal solution. Now, if u=u*, the optimal dual solution , then each active extreme point is optimal in its respective subproblem (SPi). Furthermore, there exists a neighbourhood N* of u* such that, for each u ε N*, each subproblem (SPi) has an active extreme point as optimal solution. Then, the monotonicity of the optimal distance is a favourable property to get inside N* and identify the active extreme points. This is the reason why subgradient optimization is able to accelerate the Dantzig-Wolfe's algorithm.

The basic strategy of a combined algorithm is a follows:

- 1st. phase : let perform T iterations of the subgradient algorithm, where T is large enough in order to generate all active extreme points.
- 2nd. phase : let form a master program with the previously generated local solutions. If all active extreme points are included, then the optimal solution is obtained.

Of course, it is quite a hard matter to know which is the smallest T that permits the switching to the second phase. To overcome this difficulty, we have fixed a reasonably small number, K, of iterations in the first phase. If the optimal solution is not obtained in the second phase (i.e. if we failed to identify all active extreme points) , then we switch back to the first phase and perform K new subgradient iterations to get closer to the optimal dual solution u*. We repeat then the cycle until reaching optimality.

We may observe that, as the second phase yields a feasible primal solution, we may improve the upper bound $\bar{v}$ used in the stepsize update (7).

We can now sketch the combined algorithm:

1. Initialize v^o, $\bar{v}$, $\omega_o = 2$, $\rho = \sqrt{1-\gamma^2}$. J=1
2. First Phase: perform K iterations of the subgradient algorithm (4)-(7) with $\omega_k = \omega_o \rho^k$, k=(J-1)K + 1, to JK
   Store the local solutions of each SPi at each k.
3. Second phase: Solve the master program of (DW) including all the local solutions stored in the first phase and the optimal columns obtained in the second phase at cycle J-1.
   Let W^J be the optimal (feasible) value.
4. Test optimality. If not ,go to 5.
5. If $\bar{v} > W^J$, let $\bar{v} = W^J$. Go back to 2. J=J+1

IV COMPUTATIONAL RESULTS AND CONCLUSIONS.

The mixed algorithm described before have been tested on some typical production planning models. We resume below the main aspects of these models.

We have to plan n productions activities on T time periods using a limited production capacity. Let y_{it} be the production level for unit i in period t, x_{it} the inventory level for unit in period t, d_{it} the corresponding demand.

Production cost are divided in production level costs and inventory costs. The coupling contraints are of the type :

$$\sum_{i=1}^{n} A_{it}\, y_{it} \leq S_t, \quad t = 1,\ldots,T.$$

where S_t is a vector (mostly a scalar) of capacities requirements in period t. Then the global model is, introducing technological bounds α_i, β_i, q_i, r_i on each variables:

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{t=1}^{T} (c_{1i}\, y_{it} + c_{2i}\, x_{it})$$

$$x_{it} - x_{i,t-1} - k_i\, y_{it} = - d_{it}$$

$$\alpha_i \leq x_{it} \leq \beta_i$$

$$q_i \leq y_{it} \leq r_i \qquad i = 1,\ldots,n \quad t = 1,\ldots,T$$

$$x_{io} \text{ given}$$

$$\sum_{i=1}^{n} A_{it}\, y_{it} \leq S_t, \quad t = 1,\ldots,T$$

We have tested different possible structures for the coupling blocks A_{it}. The biggest example tested was :

$$n = 7,\ T = 12,\ S_t \in R^2 \Longrightarrow m_o = 24,\ m_i = 36,\ n_i = 24$$

In table 2, we have reported some results illustrating the tests for one particular structure: for different values of K, we have quoted the values of the primal objective after each master iteration w^J and the values of the best dual value until iteration k= JxK, noted v^J. We have chosen ρ=.92, but another test has shown that ω may be held constant because of the decrease of the upper bound $\bar{v}$. The problem was solved for K=0 (pure Dantzig-Wolfe's algorithm) in 58 iterations. The case K=20 shows that with 60 subgradient iterations and only 3 DW iterations an optimal solution is obtained.

Our computational experience is of course quite limited, mainly by the rather small size of the examples. We expect in fact that the number K* of subgradient iterations to generate all the active extreme points at the optimal solution (in our test, K* = 80) will not increase too much above some typical threshold depending on the geometric properties of the block-angular problem as the condition number γ. Hence, for very large-scale problems, the best way to combine both approaches seems to be to iterate with the subgradient method a sufficient number of iterations in order to get all or quite all the information necessary to obtain the optimal solution in only one master iteration.

J =	1		2		3	
K	w^1	v^1	w^2	v^2	w^3	v^3
5	~	171322	182896	175550	179088	176872
10	183097	175428	179504	177115	179306	177922
20	180121	176420	179315	178049	178650*	178559
30	180043	177000	179124	178147	178650*	178618
40	179625	177848	178650*	178554		
50	179010	178147	178650*	178521		
60	178902	178521	178650*	178547		
70	178902	178589	178650	178589		
80	178650*	178589				

~: no feasible primal points.
*: an optimal solution was found.

Finally, we may observe that 80 subgradient iterations are necessary to generate all the active extreme points which seems to indicate that the subgradient algorithm is slower than (DW) (only 58 iterations). In fact the progress towards the solution in the first phase is much faster than with (DW). This fact has been observed for example in the case K=40: the proposed algorithm has performed 2K=80 iterations to get the optimal solutions, but we have verified that only 6 subgradient iterations are necessary in the second cycle which gives a total of 46 iterations, inferior to 58.

Anyway the computational cost is highly reduced because a subgradient iteration is cheaper than a (DW) iteration. We show in the last table the computational costs (in sec. of a CYBER-170/835) to get the optimal solution in function of K.

K	C.P.U. Time (Sec.)
0 (DW)	217
20	88
30	150
40	139
50	154
60	174
70	192
80	115

The proposed algorithm in its simple form has shown a good behaviour accelerating the Dantzig-Wolfe's method for a wide range of values of K. Some additional features in the step-size updating may ameliorate this performance. Another possibility is to decrease K at each cycle in order to reduce the total number of iterations.

BIBLIOGRAPHY:

[1] Held M. and R.M. Karp, "THe traveling-salesman problem and minimum spanning trees. Part. II", Math. Prog.1 (1971) 6-25.

[2] Held M., P. Wolfe and H.P. Crowder, "Validation of subgradient optimization", Math. Prog.6 (1974) 62-88.

[3] Gondran M. and M. Minoux, Graphes et Algorithmes, Eyrolles, Paris, 1979.

[4] Shapiro J.F., Mathematical Programming - Structures and Algorithms, John Wiley, 1979.

[5] Mahey P., "Subgradient techniques and Combinatorial Optimization", 1985, to appear.

[6] Mahey P., "Decomposition of large-scale linear programs by subgradient optimization", Mat. Aplic. Comp.1, (1982) 121-134.

[7] Dantzig G.B. and P. Wolfe, "The decomposition algorithm for linear programming", Op. Res. 8 (1960).

[8] Polyak B.T. , "Minimization of unsmooth functionals", USSR Comput. Math. and Math. Physics 9 (1969) 14-29.

[9] Minoux M., Programmation Mathématique - Théorie et Algorithmes, Dumond, Paris, 1983.

[10] Goffin J.L., "On convergence rate of subgradient optimization methods", Math. Prog. 13 (1977) 329-347.

[11] Goffin J.L., "The relaxation method for solving systems of linear inequalities", Math. of Op. Res. 5 (1980) 388-414.

[12] Lemaréchal C., "Bundle methods in nonsmooth optimization", in Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., IIASA Proc. Series, Pergamon Press, 1978.

[13] Marsten R.E., W. Hogan and J.W. Blankenship, "The boxstep method for large scale optimization", Oper. Res.23 (1975) 389-402.

Philippe Mahey
Dep.t Electrical Engineering,
Catholic University of Rio de Janeiro
Caixa Postal 38063 - Gávea
Brazil

Report No. 85397 – OR

SUBGRADIENT TECHNIQUES AND COMBINATORIAL OPTIMIZATION*)

by

Philippe Mahey**)

August 1985

*) A preliminary version of this work was presented at the School on Combinatorial Optimization, Rio de Janeiro, July 1985

**) Presently at the Dept. Electricial Engineering Catholic University, Rio de Janeiro, Brazil

I. Introduction

Since the pioneer work by Held and Karp in 1970 Lagrangean Relaxation has been one of most successful approach to solve a great variety of hard combinatorial optimization problems. There are basically two underlying justifications for this success: first, many "hard" problems yield Lagrangean subproblems for which "good" algorithms may be used. Second, the successive dual values yield lower bounds easily used by branch and bound techniques in integer programming. Undoubtly, the most popular algorithms to solve the dual problem generated by the relaxation are subgradient algorithms. Indeed, a linear or even convex combinatorial optimization problem will lead to a concave almost everywhere differentiable dual function, in most cases piecewise linear, and a subgradient of that function is readily given by each Lagrangean problem. Though subgradient algorithms may be considered slow for many monsmooth optimization problems , they take advantage of some particular aspects of combinatorial problems treated by Lagrangean Relaxation. These are for example:

- The large scale of the dual space when many constraints are relaxed as i.e. in the travelling salesman problem or in the generalized assignment problem. Generalized linear programming or descent algorithms would be very expansive to implement because of the high dimension of the subdifferential at breakpoints.

- The dual approach does not allow in general to obtain the optimal primal solution because of the existence of duality gaps. We need not then the exact dual solution but only an approximation which is anyway a lower bound for the primal optimal value. The subgradient algorithm is well appropriate because it is a cheap way to get these approximations.

Most of the theorical proofs and basic ideas that sustain the subgradient algorithms have born at the Cybernetic Institute of Kiev in the 60's. The russian litterature is very vast as are the individual works of Demyanov, Shor, Polyak, Ermoliew and Nurminski between others. The step-size determination remains today the main source of discussions and headaches. Fixed steps., over-and underrelaxation schemes , geometric series have been tempted successfuly or not and the applications show that little has been done to take advantage of the particular structures of some combinatorial problems.

An important feature which needs still some orien - ted studies is the influence of some condition numbers measuring the severeness of gradient discontinuities on the step-size choice. The work of Goffin is a good introduction to this matter. Finally, the research for more sophisticated algorithms must be discussed in the framework of combinato - rial applications. Conjugate subgradients, descent methods and second-order models investigated by Wolfe, Lemaréchal and Shor among others have lead to relatively few results when applied to large scale combinatorial optimization prob- lems.

II. Lagrangian relaxation for combinatorial optimization

Lagrangian relaxation is the most popular way to approach hard or large-scale convex optimization problems. The necessity of convexity is not a real obstacle as a lot has been done to device dual methods for nonconvex problems and most of the combinatorial optimization problems possess an underlying convex structure sufficient to produce sharp bounds in branch-and-bound algorithms. A general model for these problems is:

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ & g(x) \leq 0 \\ & x \in S \end{aligned} \tag{2.1}$$

where f is a convex function on R^n and g is a convex multi-valued mapping from R^n to R^p. S is a finite discrete subset of R^n. Let $u \in R^p$, $u \geq 0$, be a dual vector, introduced to relax the "hard" constraints. The main assumption which in fact motivates the relaxation is that we have at hand an efficient algorithm to solve the relaxed subproblem for each $u \geq 0$:

$$\begin{aligned} \text{Minimize} \quad & f(x) + u^T g(x) \\ & x \in S \end{aligned} \tag{2.2}$$

Let w(u) be the optimal value of (2.2) and X(u) the set of optimal solutions ($X(u) \subset S$).

Then w(u), the dual function, is a concave function and we get easily its subdifferential for each $u \geq 0$:

$$\partial w(u) = \text{Conv}\{y \in R^p \mid y=g(x),\ x \in X(u)\} \tag{2.3}$$

where conv denotes the convex hull in R^p.

Therefore, the main difficulty to solve the dual problem ,

$$\begin{aligned} &\text{Maximize} \quad w(u) \\ &u \geq 0 \end{aligned} \tag{2.4}$$

is to identify correctly the subdifferential $\partial w(u)$.

Let us, look for example at the particular case of the 0-1 Integer Programming problem. The model is:

$$\begin{aligned} &\text{Minimize} \quad c^T x \\ &Ax \leq b \\ &x_i \in \{0,1\}, \quad i=1,\ldots,n \end{aligned} \tag{2.5}$$

A is a (pxn) matriz whose columns are noted A_i, $i=1,\ldots,n$.

To solve the relaxed subproblem, we need the set:

$$I(u) = \{i \in \{1,\ldots,n\} \mid \bar{c}_i = c_i + u^T A_i = 0\}$$

Then, we get:

$$\partial w(u) = \text{conv}\{y \in R^p \mid y = \sum_{i=1}^{n} A_i x_i - b,\ x_i \in X(u)\}$$

$$= \{y \in R^p \mid y = \bar{y} + \sum_{i \in I(u)} \alpha_i A_i,\ \alpha_i \in [0,1],\ \sum_i \alpha_i = 1\}$$

where $\bar{y} = \sum_{i/\bar{c}_i < 0} A_i - b$

If $k_u = \text{card}(I(u))$, we see that $\partial w(u)$ is a p-dimensional box with 2^{k_u} vertices.

Unfortunately, it is almost always a burdensome task to gather all the information about the subdiffential at an interesting point, i.e., a breakpoint where w is nonsmooth. In most cases, only one subgradient is available, which is equivalent to say roughly that we are only able to generate differentiable points. This difficulty leads naturally to subgradient methods, but is seems to us convenient to complete this theoretic overview with some useful results on ascent properties of the subdifferential (2.3).

To characterize an ascent direction at u, we look at the directional derivative $w'(u;z)$, where $z \in R^p$. We have (Rockafellar, 1970):

$$w'(u;z) = \mathrm{Inf}\{y^T z \mid y \in \partial w(u)\}$$

$$= \mathrm{Min}\{g(x)^T z \mid x \in X(u)\} \qquad (2.6)$$

Then z is an ascent direction if $w'(u;z) > 0$

The steepest ascent direction $\tilde{z}$ is given by:

$$\underset{\|z\|\leq 1}{\mathrm{Max}} \; w'(u;z) = \underset{\|z\|\leq 1}{\mathrm{Max}} \; \underset{y\in\partial w(u)}{\mathrm{Min}} \{y^T z\}$$

$$= \underset{y\in\partial w(u)}{\mathrm{Min}} \|y\| = \|\tilde{z}\| \qquad (2.7)$$

$\tilde{z}$ is the subgradient with minimum norm ($\tilde{z}$= Nr $\partial w(u)$).

A necessary and sufficient condition for optimality of (2.4) is:

$$\tilde{z} = 0 \text{ or } 0 \in \partial w(u) \qquad (2.8)$$

The negative fact that a subgradient at u could as well be a descent direction for w is however compensated by the following result (we suppose for sake of simplicity that (2.4) has a unique solution):

Theorem (2.1) If u* is the unique optimal solution of (2.4)

and $d(u) = \|u-u^*\|$, $(u \neq u^*)$, then any subgradient at u is a descent direction for d.

Proof: Let $z \in \partial w(u)$. Then, $\forall v \geq 0$, we have:

$$w(v) \leq w(u) + z^T(v-u)$$

In particular, if $v=u^*$, $v-u = -\nabla(\frac{d^2}{2}(u))$ and:

$$0 \geq w(u) - w(u^*) \geq z^T \nabla(\frac{d^2}{2}(u))$$

As we shall see later, the convergence of subgradient algorithms depends on a condition number which measures the largest angle betweem a subgradient and the optimal direction u^*-u. Following Goffin (1977), we define for each $u \neq u^*$:

$$\gamma(u) = \min_{y \in \partial w(u)} \frac{g^T(u^*-u)}{\|g\| \, \|u^*-u\|}$$

$\gamma(u)$ is then the cosine of the angle α between the worst subgradient and the optimal direction (fig. 1). Theorem(2.1) asserts simply that this angle is smaller than 90^o.

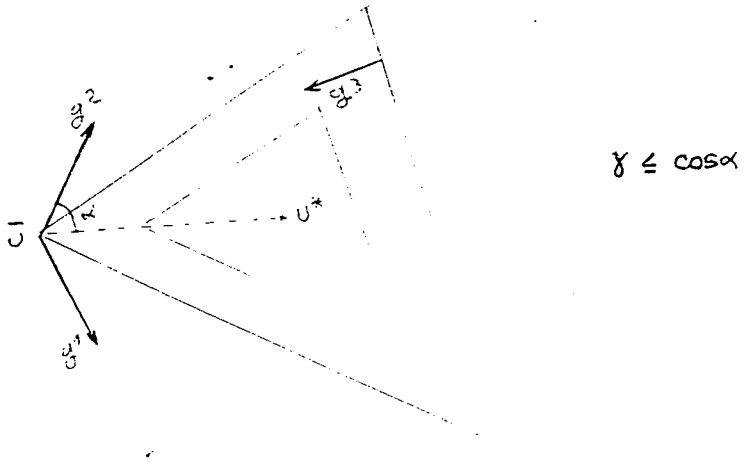


Figure 1 - Descent subgradient and condition number

We can define a condition number characterizing the global behaviour of the concave function w:

$$\gamma = \inf\{\gamma(u),\ u \geq 0\} \tag{2.9}$$

Then, we see that, the smallest is γ, the smallest is the probability to get an ascent subgradient. We can interpret this condition number observing that for a quadratic function, the convergence rate of the steepest descent algorithm is equal to $\sqrt{1-\gamma^2}$.

III. Subgradient methods - General results

The subgradient algorithm way be viewed as an ex - tension of the gradient method for nonsmooth but subdiffentiable functions. Looking at problem (2.4), we solve for $u=u^k \geq 0$ the subproblem (2.2) to obtain a subgradient $g^k \in \partial w(u^k)$ and then compute:

$$u^{k+1} = (u^k + \lambda_k \frac{g^k}{\|g^k\|})^+ \qquad (3.1)$$

where λ_k is a positive step.

$(z)^+$ denotes the orthogonal projection of z on the positive orthant of R^p.

This projection will not affect the convergence properties of the algorithms studied later. It will be omitted hereafter.

As g^k is not necessarily an ascent direction, it seems reasonable to choose a small step. On the other hand, it cannot be too small to avoid converging too early. Indeed, convergence of (3.1) to the solution of (2.4) is ensured by a quite simple characterization of the successive step - sizes:

Theorem 3.1 (Polyak, 1969)

If λ_k in (3.1) is chosen as to satisfy:

$$\lambda_k \to 0 \text{ and } \sum_{k=1}^{\infty} \lambda_k = +\infty \qquad (3.2)$$

then we have

$$\lim_{k\to+\infty} \sup\{w(u^k)\} = w^* = \operatorname{Max}_{u\geq 0} w(u)$$

We note that condition (3.2) does not depend on w and that it supplies an easy stopping criterion for the algorithm. On the other hand, it suggests a very slow rate of convergence, indeed never faster than the rate of decrease of λ_k, a divergent serie. The regulating function of the stepsize decrease becomes clear when we observe that condition (3.2) is the one obtained for stochastic gradient algorithms, confirming in some sense the Markov nature of the subgradient method.

Let us now look at some geometric properties which will prove to be useful to design algorithms with a linear convergence rate.

We suppose known the optimal distance $d_k = \|u^* - u^k\|$ of the current point to the optimal solution and let γ be the condition number.

From theorem (2.1) we have seen that g^k is a descent direction for the optimal distance d(u). Then the best step with respect to d is:

$$\hat{\lambda}_k = d_k \cos \alpha_k \quad d_{k+1} = d_k \sin \alpha_k$$

where α_k is the angle between g^k and (u^*-u^k). Fig. 2)

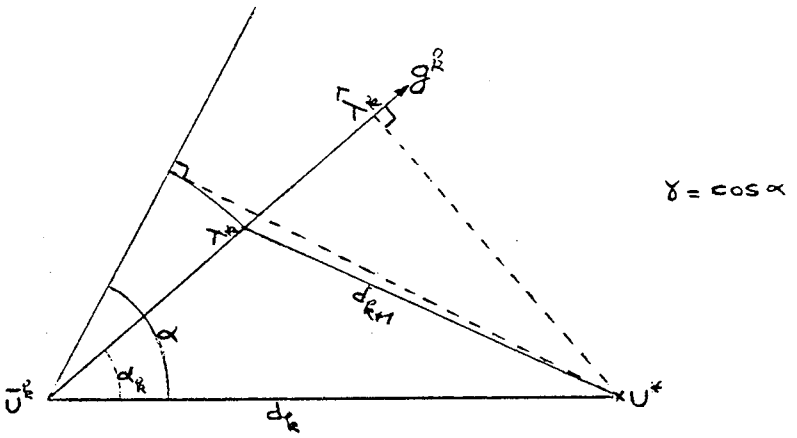


Fig. 2 - Approximative stepsize with condition number.

We can now draw two directions to approximate $\hat{\lambda}_k$:

i) Bounding $\cos\alpha_k$ by the condition number γ:

Indeed, we have $\cos\alpha_k \geq \gamma$.

This strategy leads to the following relation:

$$\lambda_k = d_k \gamma \leq \hat{\lambda}_k \tag{3.3}$$

$$= d_{k+1} \leq d_k \sqrt{1-\gamma^2} \tag{3.4}$$

which means that we expect a linear convergence at the rate $\sqrt{1-\gamma^2}$. The difficulty is of course to estimato d_0 and γ.

ii) Using the convexity of w:

As $g^k \in \partial w(u^k)$, we have:

$$w(u^*) \leq w(u^k) + g^{kT} (u^*-u^k)$$

But $$\cos \alpha_k = \frac{g^{kT}(u^*-u^k)}{\|g^k\| \, d_k}$$

Then $$\hat{\lambda}_k \geq \lambda_k = \frac{w(u^*)-w(u^k)}{\|g^k\|} \qquad (3.5)$$

To analyze the convergence of (3.1) with λ_k given by (3.5), we must make the following assumption: there exist a constant ℓ such that:

$$w(u^*)-w(u) \geq \ell \, g^T(u^*-u), \ \forall u \geq 0 \text{ and } g \in \partial w(u) \qquad (3.6)$$

and $0 < \ell < 1$

But $g^{kT}(u^*-u^k) \geq \gamma \, \|g^k\| \, d_k$

Then $\lambda_k \geq \ell \, \gamma \, d_k$

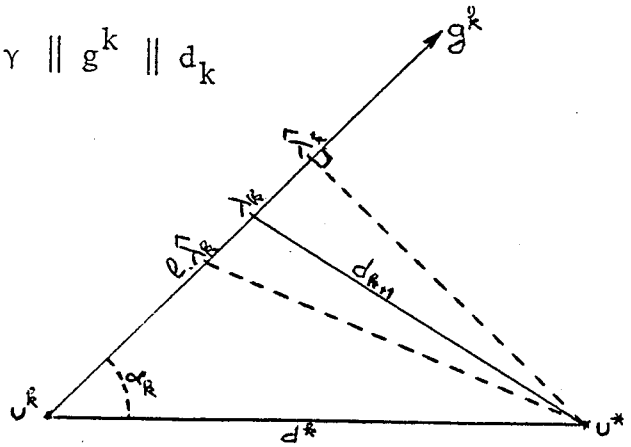


Fig. 3

$$d_{k+1}^2 \leq d_k^2 \sin^2 \alpha_k + d_k^2 (\cos\alpha - \ell\gamma)^2$$

$$d_{k+1}^2 \leq d_k^2 (1-2\ell\gamma \cos\alpha_k - \ell^2\gamma^2)$$

$$\Rightarrow d_{k+1}^2 \leq d_k^2 (1-2\ell\gamma^2 + \ell^2\gamma^2)$$

Then we obtain again a linear convergence and as $0<\ell<1$, we may estimate the rate of convergence as $\sqrt{1-2\ell\gamma^2+\ell^2\gamma^2}$ which is always greater (i.e. slower) than $\sqrt{1-\gamma^2}$, if we observe that $0<\ell\ (2-\ell)<1$.

The difficulty to reach this rate is of course to estimate $w(u^*)$.

We may now analyze the two principal methods based on this two cases. These are Shor's method for case i) and the relaxation method for case ii).

IV. Shor's method (or "convergent serie")

Shor's algorithm (1962) is based on estimate (3.3) which has lead to a geometric convergence of rate $\sqrt{1-\gamma^2}$. Let $\lambda_o > 0$ and $\rho \in (0,1)$, then the step is:

$$\lambda_k = \lambda_o \rho^k \qquad (4.1)$$

with $\lambda_o > 0$

$0 < \rho < 1$

The sucess of the method depends on how one succeeds in approximating the best λ_o, i.e. $\lambda_o = d_o\gamma$, and the best ρ, i.e. $\rho = \sqrt{1-\gamma^2}$. Before analyzing some practical ideas related with (4.1), let us briefly give a geometrical interpretation of the method. A detailed study may be found in the excellent paper by Goffin (1977). Starting from u^o, we suppose known an upper bound for d_o, say M_o. If $\gamma = \cos\alpha$, then u^* lies within the spherical sector of the sphere S_o centered in u^o, of ray M_o with an angle 2α apart the direction g^o (Fig. 4). We try now to determine the step λ_o in the direc - tion g^o such that u^* is contained in a new sphere S_1 smaller than S_o centered in u^1. As S_1 must be as small as possible, we distinguish two cases:

i) $\alpha \geq \frac{\pi}{4}$

Then the sphere S_1 centered in u^1, when $\lambda_o = M_o\gamma$, and with a ray equal to $M_o\sqrt{1-\gamma^2}$ contains the whole spherical sector and then contains u^*.

ii) $\alpha \leq \frac{\Pi}{4}$

The sphere described in i) becomes now too small to contain the hachured sector. The best idea is to choose λ_o such that the projection of u^1 on the extreme ray of the sector is exactly at a distance $\frac{M_o}{2}$ from u^o (Fig. 4.2). This means that:

$$\lambda_o = \frac{M_o}{2\cos\alpha} = \frac{M_o}{2\gamma} \tag{4.3}$$

and the new sphere S_1 has a ray equal to $\frac{M_o}{2\gamma} = M_1$.

We verify that $\alpha \leq \frac{\Pi}{4} = \gamma \geq \frac{\sqrt{2}}{2} = M_1 \leq \frac{M_o}{\sqrt{2}}$

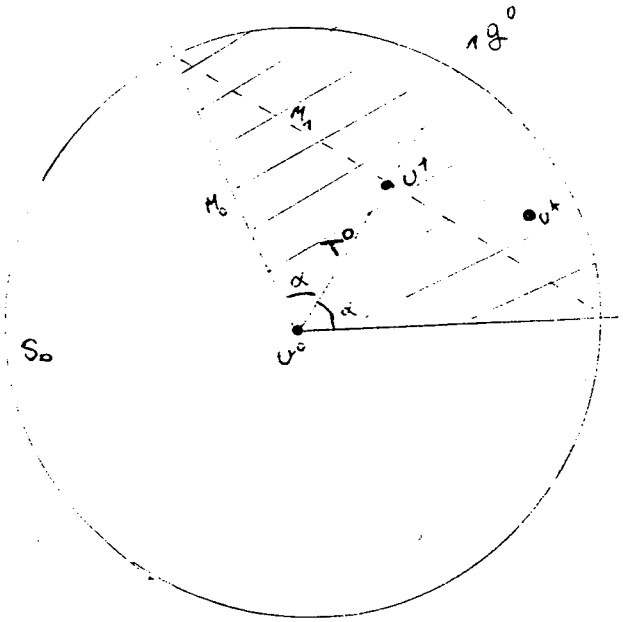


Fig 4.1 - $\alpha \geq \frac{\pi}{4}$

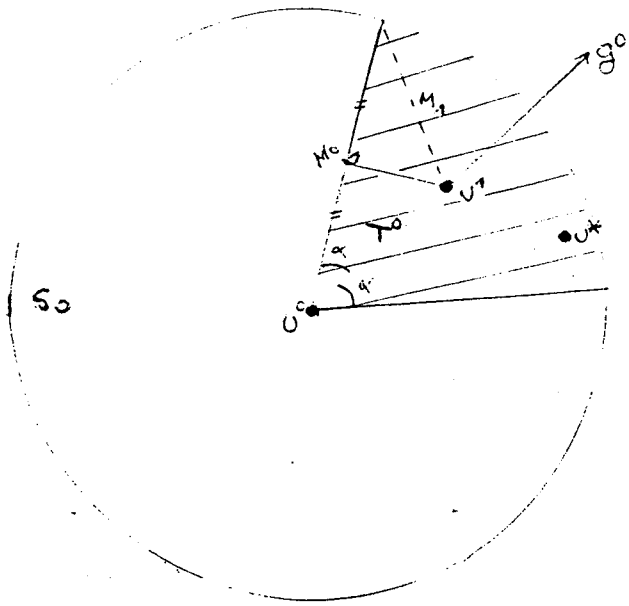


Fig. 4.2 - $\alpha \leq \frac{\pi}{4}$

We observe on the figures what might happen if we fail to approximate do and γ:

. First, as the sphere S_1 is constructed to be as small as possible, we cannot try to make a larger step than the one obtained before. Then, we have:

$$\gamma \leq \frac{\sqrt{2}}{2} \Rightarrow \quad \rho \geq \sqrt{1-\gamma^2}$$

$$\gamma \geq \frac{\sqrt{2}}{2} \Rightarrow \quad \rho \geq \frac{1}{2\gamma}$$

Fig. 5 shows what is the best geometric rate we may expect in function of γ.

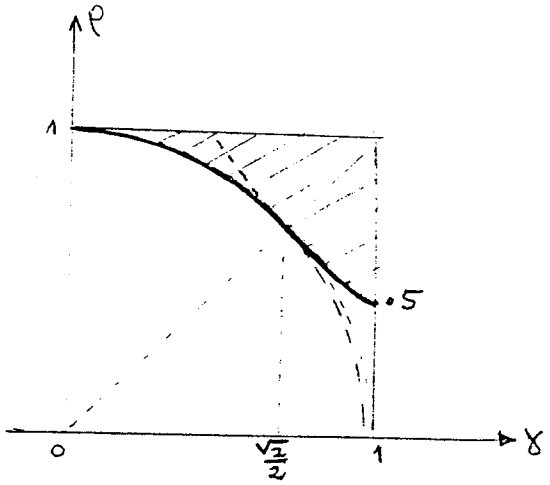


Fig. 5 - Best expected rate

. The second aspect is that λ_o cannot be chosen too small. It seems logical to have $M_o \geq d_o$, which implies $d_o \leq \frac{\lambda_o}{\sqrt{1-\rho^2}}$ or $d_o \leq \frac{\lambda_o}{\sqrt{1-\rho^2}}$ in the case i) for example.
We may sharpen this bound if we suppose $\gamma \geq \sqrt{1-\rho^2}$
Then we have $d_o \leq \lambda_o \frac{\gamma+\sqrt{\gamma^2-(1-\rho^2)}}{1-\rho^2}$ which is the bound obtained by Goffin (1977).

What about now the practical implementation of this method?

1. It seems natural that the condition number does not depend on the dimension of the problem nor on the special data but much more on the particular structure of the problem. Most problems checked yield estimates lower than .5 which means that $\rho \geq .87$. Bad conditioning appeared for instance with the travelling salesman problem (Ribeiro, 1983). Then ρ could have to be superior than .95. A future research might be to obtain some trust ranges of ρ for each class of combinatorial problem.

2. To estimate the best λ_o, it is common to use initial approximations of (3.5). Ribeiro (1983) suggests the following scheme:

Given $\hat{w}$ an upper bound for $w(u^*)$, let

$$\lambda_o = 2^{\beta} \frac{\hat{w}-w(u^o)}{\|g^o\|}$$

and test some typical values for β: $\beta=1,0,-1,-2,-3,\ldots$. To do that, we may try first a large value ($\beta=1$) and iterate the k_1 first iterations until $w(u^{k1}) > w(u^o)$. If k_1 is large and ρ is small, then bad conditioning is likely to happen. We have then to augment ρ and halve λ_o. If k_1 is small and ρ is large, the opposite situation, we must diminish ρ. The other cases must be checked with another criterion as the best value obtained after a given large number of iterations.

V. The relaxation method

This method related with estimate (3.5) is very close with relaxation techniques introduced to solve large systems of linear inequations (Agmon, 1954 and Motzkin , Schoenberg, 1954).

We put:

$$\lambda_k = \rho_k \frac{\bar{w}-w(u^k)}{\|g^k\|} \tag{5.1}$$

where $\rho_k \in (0,2)$ and $\bar{w}$ is:

i) an overestimate: $\bar{w} > w(u^*)$ (5.1.1)

ii) an underestimate : $\bar{w} < w(u^*)$ (5.1.2)

iii) the exact optimal value: $\bar{w} = w(u^*)$ (5.1.3)

Goffin (1978) has shown the exact relation between the relaxation method for systems of linear equations and the subgradient algorithm with the step chosen by (5.1). Case i) seems to be the most interesting in combinatorial optimization. The overestimate can be obtained when feasible primal solutions are available. Held, Wolfe and Crowder (1974) in a remarkable paper have studied the convergence of (5.1.1). As $\bar{w} > w(u^k)$, $\forall k$, we have to choose ρ_k such that:

$$\rho_k \to 0 \text{ when } k \to +\infty \tag{5.2}$$

Held et al suggest the following empirical strategy:

$\rho_k=2$ during $2n$ iterations, then halve ρ_k after n iterations, then after $\frac{n}{2}$ iterations, $\frac{n}{4}$,..., until we reach the iteration number z. Then ρ_k is halved every z iterations until the resulting λ_k becomes sufficiently small.

This method was tested on some large-scale traveling salesman problems and some multi-commodity flow problems. Comparisons with a previous work by Held and Karp (1971) showed that linear convergence happened in most cases. In the early paper by Held and Karp, the step-size is chosen as:

$$\lambda_k = \bar{\lambda} \ \|g^k\| \tag{5.3}$$

Curiously, this apparently naive choice of a constant step in the subgradient direction gave quite nice results. The reason is indeed closely related with the study of the relaxation method. Held and Karp has shown that with step (5.3), the sequence $\{w(u^k)\}$ satisfies:

$$\text{Sup } w(u^k) \geq w(u^*) - \frac{1}{2} \bar{\lambda} \lim_{k\to\infty} \sup \|g^k\|^2 \tag{5.4}$$

They verify then that, for the traveling-salesman problem, $\|g^k\|$ tends to go to zero as $k\to+\infty$

VI. Acceleration techniques

All the algorithms previously presented seem to yield very poor rates of convergence. This of course is the price to pay for limiting our first-order information about w to only one subgradient. The use of other subgra - dients to get better directions may have two distinct objectives:

i) To produce an ascent direction.

ii) To approximate second-order information.

We shall not describe in detail the algorithms available in the first class. This is mainly, because very few were applicated to large-scale combinatorial optimization prob - lems. Fist, we note that, if $\gamma \geq \frac{\sqrt{2}}{2}$ ($\alpha \leq \frac{\Pi}{4}$), then every subgradient is an ascent direction. Experience has shown that unfortunately this case is very rare. The other point is that an ascent algorithm needs a line-search procedure. Wolfe(1975) and Minoux (1980) have designed some simple algorithms which proved to be effective. The major problem is then to approximate the subdifferential at breakpoints in order to compute an ascent direction from (2.7). As we have already mentioned, it is quite unlikely to know more than one sub - gradients in one point. We have then to look for different ones in the neighbourhood, this strategy resulting in esti- mating an ε-subdifferential. An ε-subgradient at $\bar{u}$ (or a subgradient computed in the neighbourhood of $\bar{u}$) can be defined by:

$$g \in \partial_\varepsilon w(\bar{u}) \Rightarrow w(\bar{u}) \leq w(\bar{u}) + g^T(u-\bar{u}) + \varepsilon \qquad (6.1)$$

where $\varepsilon > 0$

Conjugate subgradient methods, ε-ascent algorithms and bundle methods have been proposed (Wolfe, 1975) or Lemarèchal, 1978). These methods are mainly based on successive projections on convex combinations of the successive sub - gradients. As the number of extreme points of the subdiffe - rential generally increases exponentially with n, these projections turn to be computationnally disastrous for combinatorial problems. The second category contains proba - bly more hope to ameliorate subgradient techniques. Polyak (1978) has given a survey of soviet research in that direc - tion. Shor, who can be considered the father of subgradient techniques, proposed to modify the subgradient direction by a space dilatation operator based on the difference of two successive subgradients (1970,1971). The method, which may be considered as a variable metric form of subgradient algorithms, has been reported to be efficient on some typical small nonsmooth problems (Lemarèchal, 1980) but two principal drawbacks turn its implementation for combina - torial optimization difficult: it is first the lack of global convergence and second, the necessity to update large matrices at each iteration. We resume the algorithm as the result of a linear transformation of the variables in a direction depending on the last subgradients.

Let $u^k \geq 0$ and $g^k \in \partial w(u^k)$

Let B_k an (nxn) non singular matrix

Then the linear transformation $u=B_k v$ leads to the algorithm:

$$u^{k+1} = u^k + \lambda_k B_k B_k^T g^k \tag{6.2}$$

B_k is update by a dilatation operator $R_\alpha(d_k)$:

$$R_{\alpha_k}(d_k) = I + (\alpha_k - 1) d_k d_k^T \tag{6.3}$$

where d_k and α_k may be chosen as:

i) $$d_k = \frac{r_k}{\|r_k\|} \text{ with } r_k = B_k^T (g^{k+1} - g^k) \tag{6.3.1}$$

$$\alpha_k = \alpha, \quad 0 < \alpha < 1$$

ii) $$d_k = \frac{r_k}{\|r_k\|} \quad \text{with } r_k = B_k^T g^k \tag{6.3.2}$$

$$\alpha_k = \alpha < 1$$

Shor suggested to realize a line-search in case (6.3.1) when $B_k B_k^T g^k$ is an ascent direction. If it is not, $\lambda_k = 0$, and (6.3.1) is substituted by $r_k = B_k^T (g^k - g^{k-1})$.

Algorithm (6.3.2) has some interesting consequences when implemented in the following manner, known as the ellipsoid method (Shor, 1977 and Khachian, 1979):

Let M be an upper bound for $\|u_o - u^*\|$. Then let choose:

$$\cdot\ \alpha_k = \alpha = \sqrt{\frac{n-1}{n+1}}$$

$$\cdot\ d_k = \frac{B_k^T\ g^k}{\|B_k^T\ g^k\|}$$

$$\cdot\ \lambda_k = \frac{M}{n+1}\ \frac{\beta^k}{\|B_k^T\ g^k\|} \quad \text{where } \beta = \frac{n}{\sqrt{n^2-1}} \qquad (6.4)$$

The geometric interpretation of the method is very close to the one described in section 4. The spheres are substituted by ellipsoids, which successive volumes converge to o at a geometric rate equal to $q = \alpha\ \beta^n$ (cf. Shor (1983),Grötschel, Lovasz and Schrijver,(1981), for related studies).

VIII. Subgradient techniques in the applications

Held and Karp (1971) have been the first to implement a version of the relaxation method for large-scale traveling-salesman problems. But the landmark paper by Held, Wolfe and Crowder (1974) has definitely clarified the potential of subgradient optimization for large-scale problems. A great variety of problems have been tested with their specific stepsize strategy. The technique has proved to be one of the most effective for:

- Generalized Assignment Problems
- Traveling-Salesman Problems
- Multicommodity Maximum Flow Problems.

Beside these models and their variations, the method has been implemented in other problems where it served more as a base for computational comparisons. These are for example set partitioning problems, scheduling problems, hydrothermal unit commitment problems and engineering design. Unfortunately few efforts were made to modify the step-size strategy or if it was done, this particular computational experience does not appear clearly in the biblio - graphy. Minoux (1979) suggested some new schemes and emphasized the link between the determination of the decrease rate of coefficient ρ_k in (5.2) and the convergence of Shor's algorithm. Indeed, halving ρ_k every five iterations is approximately a geometric serie with rate: $\rho=(\frac{1}{2})^{1/5} \cong .87$ which is the best rate of convergence of Shor's method when the condition number γ equals .5, an experimental

typical value. This fact is important and shows that the computational behaviour of subgradient optimization is closely related with some geometric parameters depending on the structure of the problem and not on its scale. We may now expect that subgradient optimization will give its best performance with very large-scale problems.

A typical but not exhaustive list of references in the spirit of relaxation method is given below:

Kennington and Shalaby (1977)
Cornuejols et al. (1977)
Legendre and Minoux (1977)
Gondran and Minoux (1979)
Bitran et al. (1981)
Neebe and Rao (1982)
Christofides and Beasley (1983)
Murtagh and Soliman (1983)
Ribeiro (1983)

Acceleration techniques have lead to very few applications. A simple formula is proposed by Camerini et al (1975):

Substitute g^k by s^k in (3.1) and use:

$$\left| \begin{array}{l} s^o = g^o \\ s^{k+1} = g^{k+1} + \beta_k s^k \end{array} \right. \qquad (7.1)$$

β_k may be chosen as:

$$\beta_k = \text{Max}\ \{0, -\gamma\ \frac{s^k\ T\ g^{k+1}}{\|s^k\|^2}\} \qquad (7.2)$$

An empirical choice for γ they have tested lead to

$$\beta_k = \frac{\|g^{k+1}\|}{\|s^k\|} \qquad (7.3)$$

which seems quite difficult to justify.

Crowder (1976) has proposed a formula similar to (7.1)but interpreting β_k as an exponential smoothing in the successive directions. Mulvey and Crowder (1979)and Arditti and Minoux (1983) reported some successful results with this idea in a set-partioning problem. Though (7.1) looks very much like a conjugate subgradient method, it is worth observing that the central idea of conjugacy must lead to an ascent algorithm where the new subgradient is orthogonal to the subspace of the past information. Bundle methods as described by Wolfe (1975) and Lemarechal (1975,1978) have been seen to have this property. Unfortunately these last methods have not been applied with success to large-scale combinatorial problems.

Finally a limited computational experience on Shor's method may be found in Minoux and Serreault (1981),

Ribeiro(1983) and Minoux (1984). Lemarèchal (1980) has applied the space extension operator (6.3) to the 48 cities TSP and reported some very good performance. Second - order algorithms will certainly permit to improve the rather poor convergence rates of subgradient algorithms and it seems that the main actual research in nonsmooth optimization is aiming to clarify this problem. Another interesting idea is to combine subgradient optimization with a faster but more expensive algorithm in order to avoid zigzagging when getting close to the dual optimum and eventually compute the exact optimal solution (Minoux, 1984, and Mahey, 1985).

BIBLIOGRAPHY:

1. Lagrangian Relaxation

Geoffrion A.M., "Lagrangian relaxation and its uses in integer programming", Math. Prog. Study 2 (1974) 82-114.

Fisher M.L.,W.D. Northup and J.F. Shapiro, "Using duality to solve discrete optimization problems: Theory and computational experience", Math. Prog. Study3(1975) 56-94.

Shapiro J.F., "A survey of Lagrangian techniques for discrete optimization", Annals of Discrete Mathematic 5 (1979) 113-138.

Shapiro J.F., Mathematical Programming-Structures and Algo - rithms, John Wiley, 1979.

Fisher M.L., "The Lagrangian relaxation method for solving integer programming problems", Man. Sci. 27(1981) 1-18.

Minoux M., Programmation mathématique-Théorie et Algorithmes , Dunod, 1983.

2. Non smooth optimization.

Rockafellar R.T., Convex Analysis, Princeton University Press, 1970.

Demyanov V.F., "On the maximization of a certain nondifferentiable function", JOTA (1971)75-89.

Polyak B.T., "Nonsmooth optimization: a survey of soviet research", in Non Smooth Optimization, C.Lemaréchal and R. Miffin, eds., Proc.IIASA Series, Pergamon Press, 1978.

Lemaréchal C., Extensions Diverses des Méthodes de Gradient et Applications, Doctorat d'Etat Thesis, Paris VI , 1980.

3. Subgradient methods

Agmon S., T. Motzkin and I. Schonberg, "The relaxation method for linear inequalities", Canad. J. Math. 6(1954) 382-404.

Shor N.Z., "The rate of convergence of the generalized gradient descent method", Cybernetics 4(1968)79-80.

Polyak B.T., "Minimization of unsmooth functionals", USSR Comput. Math. and Math. Physics 9(1969)14-29.

Shor N.Z., "Convergence rate of the gradient descent method with dilatation of space", Cybernetics 6(1970) 102-108.

Shor N.Z. and Z. Zhurbenko, "A minimization method using space dilatation in the direction of the difference of two successive gradients", Cybernetics 7(1971) 450-459.

Grinold R.C., "Steepest ascent for large-scale linear programs", SIAM Review 14(1972)447-464.

Held M., P. Wolfe and H.P. Crowder, "Validation of subgradient optimization", Math. Prog. 6(1974)62-88.

Camerini P.M., L. Fratta and Maffioli, "On improving relaxation methods by modified gradient techniques", Math. Prog. Study 3(1975)26-34.

Lemaréchal C., "An extension of Davidon's methods to nondifferentiable problems", Math. Prog. Study 3(1975)95-109.

Wolfe P., "A method of conjugate subgradients for minimizing non differentiable functions", Math. Prog. Study (1975)145-173.

Crowder H.P., "Computational improvements for subgradient optimization Symposia Mathematica 19(1976)357-372.

Goffin J.L. "On convergence rates of subgradient optimization methods" Math. Prog. 13(1977)329-347.

Shor N.Z.,"Cut-off method with space extension in convex programming problems", Cybernetics 13(1977)94-96.

Lemaréchal C., "Bundle methods", in Non Smooth Optimization , C. Lemaréchal and R. Miffin, eds. Proc. IIASA Series, Pergamon Press, 1978.

Goffin J.L., "The relaxation method for solving systems of linear inequalities", Math. Oper. Res. 5(1980) pp. 388-414.

Minoux M.,"Subgradient optimization and Benders decomposition for large-scale programming", in Mathematical Pro - gramming, R.W. Cottle, M.L. Kelmanson and B. Korte, eds., North Holland, 1984.

Mahey P., "A subgradient algorithm for accelerating the Dantzig-Wolfe decomposition algorithm", X Symposium in operations Research, München, 1985.

4. Applications

Held M. and R.M. Karp, "The traveling-salesman problem and minimum spanning trees. II", Math. Prog. 1(1971) 6-25.

Legendre J.P. and M. Minoux, "Une application de la notion de dualité en programmation en nombres entiers: sélec - tion et affectation optimale d'une d'avions", RAIRO Op. Res. 11(1977)201-222.

Kennington J. and M. Shalaby, "An effective subgradient procedure for minimal cost multicommodity flow problems", Man. Sci. 23(1977)994-1004.

Cornuejols G., M.L. Fisher and G.L. Nemhauser, "Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms", Man. Sci. 23 (1977)789-810.

Gondran M. and M. Minoux. Graphes et Algorithmes, Eyrolles , Paris, 1979.

Mulvey J.M. and H.P. Crowder, "Cluster analysis: an application of lagrangian relaxation", Man. Sci. 25(1979)239 - 340.

Bitran G., V. Chandru, D.E. Sempolinski and J.F. Shapiro , "Inverse optimization: an application to the capacitated plant location problems", Man. Sci, 27 (1981)1120-1141.

Minoux M. and J.Y. Serreault, "Subgradient optimization and large-scale programming: an application of optimum multicommodity network synthesis with security constraints", RAIRO 15(1981)185-203.

Neebe A.W. and M.R. Rao, "Sequencing capacity expansion projects in continuous time", WP82-3, Univ. Sounth Carolina, (1982).

Murtagh B.A. and F. Soliman, "Subgradient optimization applied to a discrete nonlinear problems in engi - neering design", Math. Prog. 25(1983)1.12.

Christofides N. and J.E. Beasley, "Extensions to a lagrangean relaxation approach for the capacitated warehouse location problem", Euro. J.O.R. 12(1983)19 - 28.

Ribeiro C.C., Algorithmes de Recherche de Plus Courts Chemins avec Contraintes: Etude Théorique, Implémentation et Parallelisation; Doct. Ing. Thesis, Paris, 1983.

Arditti D. and M. Minoux, "Un algorithme de détermination de partition utilisant la dualité lagrangienne", Note Tech. CNET, 1983.

Grötschel M., L. Lovasz and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", Combinatorica, 1(1981) 169-197.

Shor N.Z., "Generalized gradient methods of nondifferentiable optimization employing space dilation operations" , in Mathematical Programming - The state of the art- Bonn 1982, A. Bachem, M. Grotschel and B. Korte , eds., Springer V., (1983)501-529.

SBA: Controle de Automação, Vol. 1, Nº 2, pp 154-161

OTIMIZAÇÃO NÃO DIFERENCIÁVEL NOS MÉTODOS DE DECOMPOSIÇÃO

Philippe Mahey
Deptº Eng. Elétrica
PUC/RJ

Resumo

A classe das funções-Max constitui um caso particular bem conhecido em otimização não diferenciável. Os métodos de decomposição em programação matemática geram funções deste tipo. Apresenta-se as principais propriedades dessas funções e os algoritmos de subgradientes para minimizá-las. Mostra-se como esses algoritmos podem ser implementados para acelerar a convergência dos métodos clássicos para decomposição de programas lineares.

Nonsmooth optimization for decomposition methods.

Abstract

The class of Max-functions is a well studied case in nonsmooth optimization. We present here general results for these funtions and for the subgradient methods. We show then how to use these techniques to accelerate the convergence of some classical decomposition methods in linear programming.

1. INTRODUÇÃO

A otimização não diferenciável atraiu a atenção dos pesquisadores em programação matemática quando Held e Karp em 1970 propuseram um algoritmo eficiente para solução do problema do caixeiro viajante. Este algoritmo usava explicitamente como direção de busca um subgradiente de uma função linear por partes, logo não sempre diferenciável. De fato, a não diferenciabilidade em foco era compensada pela subdiferenciabilidade das funções, ou seja os resultados teóricos da analise convexa podiam sustentar a construção de algoritmos. Deste ponto de vista, a escola de Kiev, liderada por Shor, Polyak , Ermolev e outros, já vinha desenvolvendo pesquisas teóricas e aplicadas desde os anos 60.

Os trabalhos de Clarke (1975) contribuiram a partir de 1975 para delimitar o domínio prático da otimização não diferenciável sem restringir ao caso exclusivamente convexo. Com efeito, todas as funções encontradas na prática são localmente Lipschitz, ou seja, funções $f:R^n \to R$ tais que, em qualquer subconjunto limitado S, satisfazem:

$$|f(x_1)-f(x_2)| \leq K \, \| x_1-x_2 \| \, , \quad \forall x_1,x_2 \in S \qquad (1)$$

Um velho teorema de Rademacher afirma que tais funções possuem um gradiente em quase toda parte. Isso levou Clarke a definir o gradiente generalizado de uma função localmente Lipschitz em x como:
a casca convexa de todos os limites dos gradientes de f calculados em sequências de pontos convergentes para x.

O gradiente generalizado é denotado $\partial f(x)$ e tem as seguintes propriedades:

i) É um conjunto convexo, compacto e não vazio. Logo, pode-se definir uma função suporte notada:

$$f^o(x;v) = \text{Max}\{v^T g, \; g \in \partial f(x)\} \qquad (2)$$

Clarke a chama de derivada direcional generalizada.

ii) Se f é continuamente diferenciável em x então $\partial f(x)=\{\nabla f(x)\}$ o gradiente de f em x.

iii) Se f é convexa, então f é Lipschitz e $\partial f(x)$ é o subdiferencial de f em x, ou seja o conjunto de todos os subgradientes, os vetores $g \in R^n$ tais que:

$$f(y) \geq f(x) + g^T (y-x), \quad \forall y$$

Na tentativa de buscar aplicações práticas, Womersley (1980) introduziu a subclasse das funções diferenciáveis por parte. Dentro dessa classe estão as chamadas funções Max , do tipo:

$$f(x) = \text{Max}\{h(x,y), \; y \in Y\} \qquad (3)$$

onde Y é compacto e h é geralmente conjuntamente diferenciável em relação a x e y.

Pode-se verificar que uma função-Max é localmente Lipschitz e possui derivadas direcionais $f'(x;v)$ em todas as direções. Além disso $f'(x;v)=f^{o}(x;v)$ e f' é convexa em relação a v (propriedade de convexidade tangencial segundo Hiriart-Urruty, 1982).

Observamos que as funções Max contêm todas as funções convexas e s.c.i., as funções de penalidades não diferenciáveis ℓ_1 ou ℓ_∞ recentemente introduzidas em programação não linear restrita e todas as funções duais associadas a técnicas de relaxação lagrangeana.

Finalmente, pode-se observar também que a dificuldade de minimizar uma função Max é equivalente a resolver o seguinte problede programação não linear restrita:

Minimizar v

$$v \geq h(x,y), \quad \forall y \in Y \tag{4}$$

O conjunto Y é muitas vezes simples ou discreto. A dificuldade de resolver (4) vem do grande número de restrições, o que exige o uso de métodos enumerativos ou de geração de linhas.

Nosso propósito neste artigo é, na impossibilidade de retratar todos os aspectos desta relativamente nova área de pesquisa, para osquais o leitor pode se referir às monografias de Lemaréchal (1982), Shor (1985) e Kiwiel (1985), enfatizar o uso das técnicas de subgradientes nos modelos de decomposição de programas lineares de grande porte.

2. OS ALGORITMOS DE SUBGRADIENTES

2.1. A direção de maior descida

Começamos com alguns resultados teóricos que mostrarão as dificuldades encontradas para desenhar algoritmos de descida para a minimização de uma função do tipo (3).

Nota-se Y_x o subconjunto não vazio de Y dos y que minimizam $h(x,y)$ para um dado x (suponha-se $x \in R^n$ e $y \in R^m$)

O primeiro resultado clássico é:

$$\partial f(x)=\text{conv}\ \{\nabla_x h(x,y),\ y \in Y_x\} \tag{5}$$

A direção de maior descida em x é oposta ao vetor de norma mínima em $\partial f(x)$. Com efeito, a derivada direcional de f em x na direção v é dada por:

$$f'(x;v) = \text{Max}\ \{g^T v;\ g \in \partial f(x)\}$$

Logo:

$$\text{Min } f'(x;v) = \underset{\|v\| \leq 1}{\text{Min}}\ \underset{g\in\partial f(x)}{\text{Max}}\ g^T v$$

$$= \underset{g\in\partial f(x)}{\text{Max}}\ \underset{\|v\|\leq 1}{\text{Min}}\ g^T v = \underset{g\in\partial f(x)}{\text{Max}}\ (-\|g\|) =$$

$$= -\underset{g\in\partial f(x)}{\text{Min}}\ \|g\| = -\|\tilde{g}\| \tag{6}$$

Observa-se ainda que qualquer direção de descida satisfaz $f'(x;v)<0$, logo satisfaz:

$$g^T v < 0, \quad \forall g \in \partial f(x) \tag{7}$$

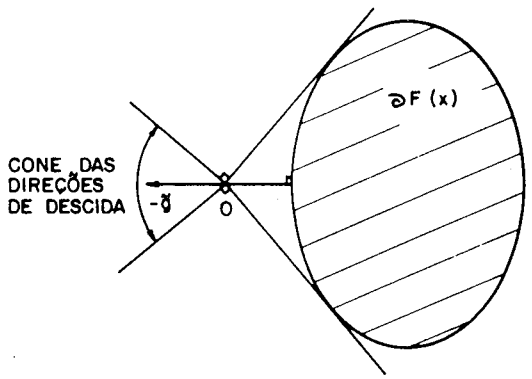


FIG. 1

Logo, uma condição necessária de otimalidade é que não existem direções de descida, ou seja:

$$\tilde{g} = 0, \text{ isto é } 0 \in \partial f(x) \tag{8}$$

Wolfe (1975) foi um dos primeiros a mostrar exemplos de funções-Max para as quais um algoritmo do tipo maior descida efetuando buscas unidimensionais exatas não converge. Além disso, mesmo nos casos favoráveis, o cálculo de $\tilde{g}$ pode ser muito caro nos pontos de não diferenciabilidade ("bicos") da função. Lemaréchal observa que, na prática, a função é sempre diferenciável em pontos possivelmente muito próximos de bicos. Isto significa que tem-se acesso a um subgradiente apenas e não ao conjunto $\partial f(x)$ inteiro. A abordagem de Lemaréchal(cf.(1978) por exemplo) é ampliar o subdiferencial para incluir os subgradientes da função nos pontos calculados anteriormente até uma certa memória (cadeia) gerando "feixes" de gradientes. Estes métodos geram algoritmos de descida, mas ainda não foram plenamente testados nos modelos de decomposição. Passaremos agora a descrever os métodos ditos de subgradientes. A característica principal destes últimos é que a função de descida associada não é a função objetiva, mas a função distância a solução ótima.

2.2. Convergência teórica

A seguir, f é uma função convexa própria no R^n. O algoritmo de subgradiente supõe que disponhamos de um subgradiente da função f no ponto iterado x_k, $g_k \in \partial f(x_k)$, e:

$$x_{k+1} = x_k - \alpha_k\, g_k / \|g_k\| \tag{9}$$

Na figura 2 são representadas as curvas de nível da função no $R^2(x=[\xi_1\ \xi_2]')$:

$$f(\xi_1,\xi_2)= \max\{|\xi_1+3\xi_2|,|\xi_1-3\xi_2|\}$$

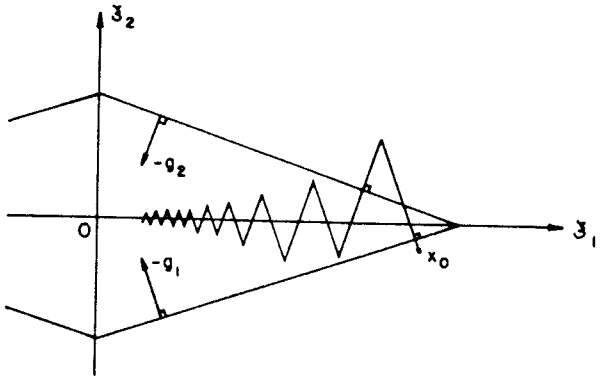


FIG. 2

Constata-se que no eixo $(0\xi_1)$ a função f não é diferenciável e que as direções opostas aos subgradientes extremos $-g_1$ e $-g_2$ (que são de fato as direções acessíveis na prática) não são direções de descida. Toda a dificuldade de implementação do algoritmo reside na escolha do passo α_k já que nenhuma busca unidimensional pode servir. Polyak (1969) mostrou que o passo α_k deve tender para 0, mas não muito rápido.

Teorema 1 (Polyak, 1969)

Supondo que o conjunto dos pontos minimizando f, X^*, é não vazio e limitado, e que:

$$\alpha_k \to 0 \quad \sum_{k=0}^{\infty} \alpha_k = +\infty \qquad (10)$$

então para qualquer ponto inicial x_o, existe uma subsequência de $\{x_k\}$, x_{k+1} dado por (9), que converge para um $x^* \in X^*$.

Prova: Ver Polyak para os detalhes técnicos. Observa-se apenas que como:

$$\|x_o - x_k\| \leq \|x_o - x_1\| + \ldots + \|x_{k-1} - x_k\| =$$

$$= t_0 + t_1 + \ldots + t_{k-1}$$

a condição $\Sigma t_k = +\infty$ significa que pode-se escolher um ponto x_o tão afastado de x^* quanto quiser.

A condição $t_k \to 0$ é necessária porque, no caso não diferenciável, a norma do gradiente não vai para zero em uma sequência convergente para x^*.

A condição (10), embora muito simples e de implementação imediata, sugere uma taxa de convergência muito lenta, aliás sublinear.

Teorema 2

Seja $d(x)= \frac{1}{2} \| x - x^* \|^2$ e f convexa e finita em $x \neq x^*$. Então qualquer direção $v=-g$, onde $g \in \partial f(x)$, é uma direção de descida para d.

Prova: Seja $g \in \partial f(x)$. Então

$$f(x^*) \geq f(x) + g^T(x^* - x)$$

Como $f(x^*)-f(x)<0$ e $x^*-x=-\nabla d(x)$, temos

$$0 > - g^T \nabla d(x)$$

Observa-se que o angulo θ entre v e $x^* - x$ (Fig.3) é sempre menor que 90^o e quando é próximo deste valor,

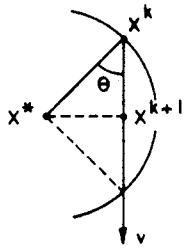


FIG. 3

o passo correspondente é menor. Seguindo esta idéia, Goffin (1977) define a condição da função f:

$$\gamma = \inf_{x \neq x^*} \{ \min_{g \in \partial f(x)} \frac{g^T (x - x^*)}{\|g\| \ \|x^* - x\|} \} \qquad (11)$$

ou seja, é o cosine do maior angulo θ encontrado. Logo, para garantir a descida da função d, o passo α_k deve satisfazer:

$$\alpha_k < 2\gamma \ \|x^* - x_k\| \qquad (12)$$

2.3. Convergência linear

Observamos que (12) depende do conhecimento de γ e da distância a solução ótima, parâmetros estes que podem geralmente ser aproximados por limites apenas muito grosseiros.

Shor propõe para o passo α_k uma série convergente:

$$\alpha_k = \alpha_o \rho^k \qquad (13)$$

onde α_o aproxima (12) no meio do arco:

$$\alpha_o \sim \gamma \ \| x^* - x_o \| \qquad (14)$$

e ρ aproxima $\dfrac{\| x^* - x_{k+1} \|}{\| x^* - x_k \|}$ ou seja:

$$\rho \sim \sqrt{1-\gamma^2} \qquad (15)$$

Goffin (1977) observa que, quando $\gamma > \frac{\sqrt{2}}{2}$ $(\theta < \frac{\pi}{4})$, a taxa de convergência ρ

pode ser aproximada por $\frac{1}{2}$ no lugar de (15) acelerando a convergência. Infelizmente, a prática mostra que este caso é raro e por outro lado, ele implica que qualquer v=-g, e g ε ∂f(x), é uma direção de descida para f. Em resumo, o algoritmo do subgradiente (9) - (13) é útil quando as direções de descida são difíceis de ser computadas ($\gamma \geq \frac{1}{4}$) e gera uma taxa de convergência $\rho \geq \sqrt{1-\frac{1}{2}} = 0.7$, ou seja ainda bastante lenta. Os testes com os modelos de decomposição em programação linear mostraram condições menores do que 0.5 levando a taxas de convergência entre 0.87 e 0.93.

Held, Wolfe e Crowder (1974) mostraram que, usando a convexidade, pode-se estimar a condição de f:

$$g \in \partial f(x) \Rightarrow f(x^*) \geq f(x) + g^T(x^*-x)$$

Logo:

$$\gamma \sim \frac{-f(x^*) + f(x)}{\|g\| \ \|x^* - x\|} \qquad (16)$$

É então natural de escolher α_k usando:

$$\alpha_k = t_k \frac{f(x) - f(x^*)}{\|g_k\|} \qquad (17)$$

e (12) implica que t_k satisfaz

$$0 < t_k < 2$$

Infelizmente, não se conhecendo f(x*), deve-se usar uma aproximação $\bar{f}$ que, em geral, é um limite inferior de f(x*) (se f é uma função do tipo dual, seria o valor de uma solução primal viável obtida anteriormente). Neste caso a convergência teórica pode ser problemática e Held, Wolfe e Crowder sugerem diminuir t_k para zero. Eles até propõem dividir t_k por 2 a cada 5 iterações, o que corresponde a uma taxa de convergência $\rho = (\frac{1}{2})^{1/5} = .87$, equivalente à série convergente de Shor quando $\gamma = .5$.

Goffin mostrou que o algoritmo de subgradiente (9)-(17) é equivalente ao método de relaxação para sistemas de equações lineares. A fórmula (17) é que foi a mais escolhida nas aplicações em otimização combinatória (ver, por exemplo, Mahey (1985)).

2.4. Caso restrito

Completamos a apresentação dos métodos de subgradientes introduzindo restrições. O problema é formulado como:

Minimizar f(x)

$$x \in \Omega \qquad (18)$$

e Ω é geralmente um conjunto convexo sim - ples como veremos nas aplicações aos modelos de decomposição.

A tendência natural é então projetar o iterado (9) sobre Ω a cada iteração. No entanto, o passo verdadeiro $\|x_{k+1}-x_k\|$ pode se tornar muito pequeno após a projeção o que leva a seguinte modificação: o problema (18) pode ser escrito:

Minimizar $F(x)=f(x) + \chi_\Omega(x)$

onde $\chi_\Omega(x)$ é a função indicadora de Ω ($\chi_\Omega(x)=0$ se $x \in \Omega$ e $=+\infty$ senão).

Tem-se:

$$\partial F(x) = \partial f(x) + N_\Omega(x) \qquad (19)$$

onde $N_\Omega(x)$ é o cone das normais a Ω em x.

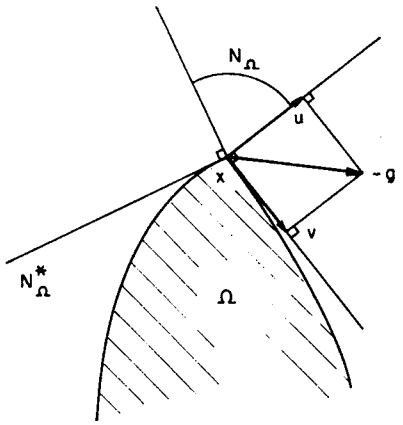


FIG. 4

Na fig 4, obtém-se:

$$\forall g \in \partial f(x), \quad -g=u+v$$

onde $u \in N_\Omega(x)$ e $v \in N_\Omega^*(x)$ (o cone polar de N_Ω ou cone das direções viáveis)

Logo - $v \in \partial F(x)$ e o algoritmo do subgradiente projetado deve ser:

$$x_{k+1} = x_k + \alpha_k \frac{v}{\|v\|} \qquad (20)$$

onde

$$v = \text{Proj}/_{N_\Omega^*}(-g) \text{ para um } g \in \partial f(x)$$

O algoritmo (20) significa que a escolha do passo, segundo uma estratégia do tipo (13) ou (17), deve ser feita na direção do subgradiente projetado e não na direção do subgradiente seguido da projeção do iterado.

3. APLICAÇÃO À DECOMPOSIÇÃO DE PROGRAMAS LINEARES

Consideremos agora um modelo linear com uma estrutura bloco-angular:

Minimizar $\sum_{i=1}^{n} c_i^T x_i$

sujeito a:

$$\sum_{i=1}^{n} A_i x_i \leq b_o$$

$$x_i \in S_i = \{x_i \in R^{n_i} \mid B_i x_i = b_i, x_i \geq 0\}$$

$$i=1,\ldots,n \qquad (21)$$

onde: $x_i \in R^{n_i}$

$c_i \in R^{n_i}$

A_i $(m_o \times n_i)$

B_i $(m_i \times n_i)$

$b_o \in R^{m_o}$

$b_i \in R^{m_i}$

Hipóteses:

- Os poliedros S_i, i=1,...,n, são limitados
- A solução ótima de (21), notada $x^* = [x_1^* \ldots x_n^*]^T$, é única e não degenerada. Notaremos p^* o vetor dos multiplicadores ótimos associados as m_o restrições de acoplamento ($p^* \in (R^{m_o})^+$).

3.1. Decomposição pelos preços

Esta técnica de decomposição consiste na separação do problema em n subproblemas cujos critérios são modificados por um vetor de "preços" associados às restrições de acoplamento (recursos comuns), vetor este ajustado iterativamente por um nível superior de coordenação até obter o equilíbrio entre a oferta e a demanda (destes recursos). Em programação matemática, é um método dual associado à relaxação lagrangeana das restrições de acoplamento.

O Lagrangeano é definido sobre $R^{n1} \times R^{n2} \times \ldots \times R^{nn} \times (R^{m_o})^+$

$$L(x,p) = \sum_{i=1}^{n} c_i^T x_i + p^T \left(\sum_{i=1}^{n} A_i x_i - b_o\right)$$

$$= \sum_{i=1}^{n} (c_i^T + p^T A_i) x_i - p^T b_o$$

Para um vetor de preços $p \geq 0$ dado, decompõe-se o problema nos n subproblemas:

$$h_i(p) = \min_{x_i \in S_i} (c_i^T + p^T A_i) x_i \qquad (22)$$

Seja $X_i(p)$ o conjunto das soluções ótimas de (22). O nível coordenador deve então arbitrar as soluções propostas pelos decisores locais resolvendo o problema dual:

$$\text{Minimizar } h(p) = \sum_{i=1}^{n} h_i(p) - p^T b_o \qquad (22)$$

$$\text{sujeito a } p \geq 0$$

Verifica-se que -h é uma função-Max e aplicando (5) obtém-se:

$$\partial h(p) = \{g \in R^{m_o} \mid g = \sum_{i=1}^{n} A_i x_i - b_o, x_i \in X_i(p)\} \qquad (24)$$

(23) é então um problema de otimização não diferencial restrito e pode-se resolvê-lo por um método de subgradiente do tipo (20). No entanto existem algoritmos que resolvem (21) em um número finito de passos. Tipicamente, é o algoritmo de Dantzig-Wolfe(1960) ou na versão dual equivalente, o algoritmo do "cutting-plane" de Kelley (1960). Este último resolve a versão transformada de (23) (cf.(4)):

$$v^t = \text{Max } v$$

$$v \leq h(p^\ell) + g_\ell^T (p - p^\ell), \quad \ell=1,\ldots,t \qquad (25)$$

$$p \geq 0$$

onde $g_\ell \in \partial h(p^\ell)$

As restrições são geradas iterativamente, ou seja p^{t+1} é a solução de (25) e gera uma nova restrição resolvendo os subproblemas (22) para $p=p^{t+1}$ para obter g_{t+1}.

Este algoritmo já foi bastante analizado e criticado (ver,por exemplo, Dirickx e Jennergren,1979) e as suas principais características são:

i) Convergência sublinear

ii) Monotonia da sequência $\{v^t\}$

iii)Não há monotonia da sequência $\{\|p^t - p^*\|\}$.

Essas propriedades chamam os comentários seguintes: por convergência sublinear entende-se a taxa de convergência de uma sequência gerada pelo algoritmo aplicado a uma função convexa geral. Se a convergência é finita como no caso presente, isso implica que o número de iterações pode se tornar muito elevado nas aplicações práticas que são de grande porte. As propriedades i) e ii) são complementares das propriedades observadas no algoritmo de subgradiente.

Na Tab.5 são representados quatro gráficos mostrando a convergência das sequências de custo e da distância ótima na comparação dos algoritmos do subgradiente (9-13) e Dantzig-Wolfe (25).

Obs: O modelo utilizado para a comparação é um modelo de planejamento da produção de uma oficina com vários ítens e vários meios de produção interligados sobre um horizonte fixo. O problema global (21) tem 336 variáveis e 216 restrições, das quais 48 são de acoplamento. O número de subproblemas é igual a 7. Cada iteração representa a resolução dos 7 subproblemas e também no caso de Dantzig-Wolfe a resolução do programa-mestre (25).

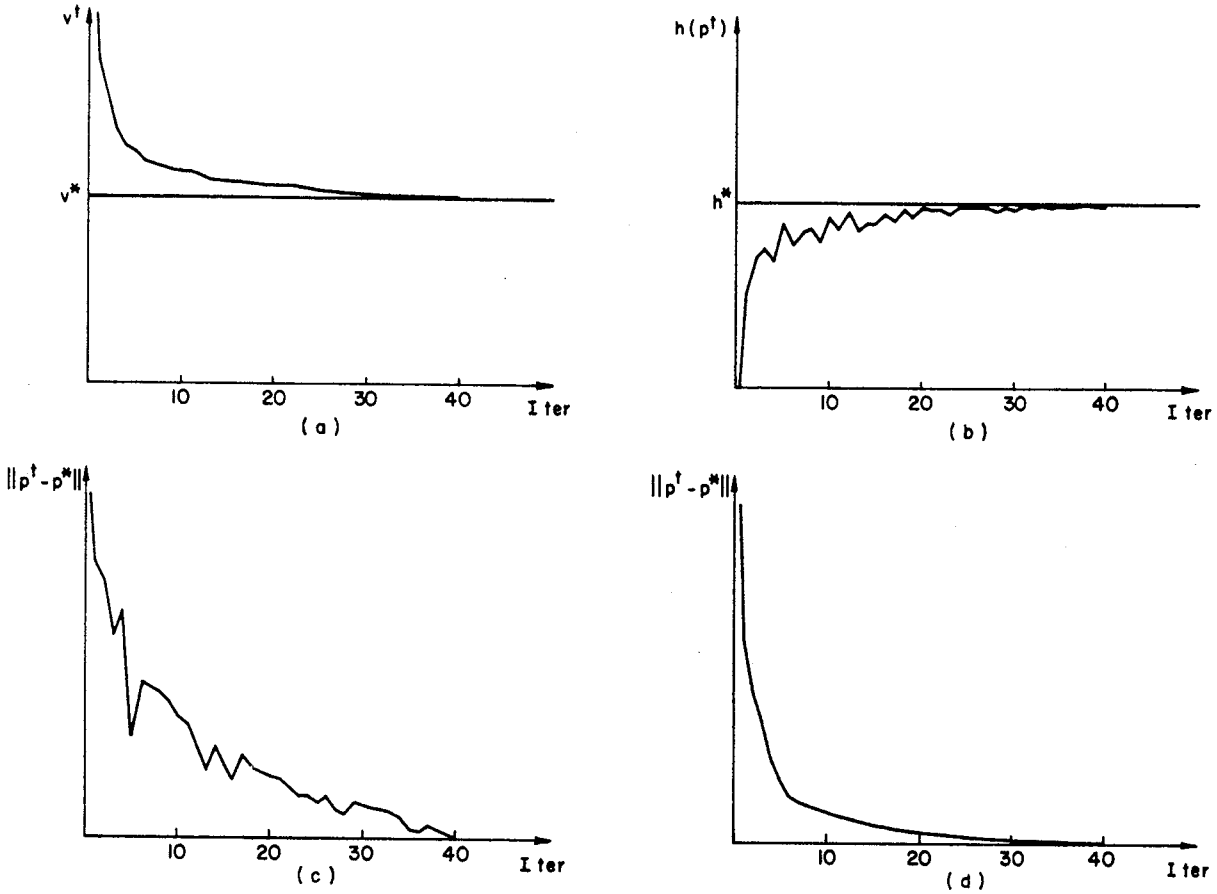


TABELA 5

VALOR ÓTIMO	(a) e (b)
DISTÂNCIA ÓTIMA	(c) e (d)
DANTZIG-WOLFE	(a) e (c)
SUBGRADIENTE	(b) e (d)

3.2. Um Algoritmo composto

As propriedades complementares dos algoritmos de Dantzig-Wolfe e do subgradiente sugerem que eles podem ser combinados para melhorar os seus desempenhos respectivos que, como vimos, são fracos.

O problema central recai finalmente sempre sobre o fato de que, para encontrar a solução ótima, é preciso conhecer todos os subgradientes ativos neste ponto, ou seja, o cume da função poliedral só é encontrado conhecendo todas as faces adjacentes. A instabilidade das soluções p^{t+1} no algoritmo de Dantzig-Wolfe torna essa busca muito lenta. Por outro lado, as oscilações com passos cada vez menores do método do subgradiente tendem a fornecer os subgradientes desejados, mas o algoritmo é incapaz de terminar na solução ótima.

Outro ponto de importância é o custo das iterações: o método do subgradiente é sempre mais barato porque não há programa-mestre a otimizar.

Essas deduções motivaram a construção de um algoritmo composto da seguinte maneira:

1. Fase 1: Algoritmo de subgradiente durante T_1 iterações. Esta fase serve para aproximar a solução ótima p* a baixo custo.

2. Fase 2: O Algoritmo de subgradiente é modificado de maneira a armazenar os subgradientes gerados, parando na iteração T_2.

3. Fase 3: Monta-se o programa-mestre de Dantzig-Wolfe com os subgradientes gerados na fase 2 e prosegue-se com este algoritmo até a solução ótima.

Os resultados relatados em Mahey(1986a) mostram uma nítida aceleração do método de Dantzig-Wolfe. Inclusive, se T_1 é suficientemente grande ($T_1 = 2m_o$ e $T_2 = 3m_o$), é comum observar que a fase 3 é reduzida a uma única iteração, o que significa que todo o subdiferencial na solução ótima foi gerado na fase 2.

3.3. Outros modelos de decomposição

No método de decomposição pelas quotas

é efetuada uma alocação a priori dos recursos comuns b_o aos subproblemas sob a forma de vetores $u_i \in R^{m_i}$ tais que $\sum_{i=1}^{n} u_i \leq b_o$.

Cada subproblema resolve então:

$$\text{Minimizar } c_i^T x_i$$

$$A_i x_i = u_i \qquad (26)$$

$$x_i \in S_i$$

Se $v_i(u_i)$ é o custo ótimo de (26) e $\Pi_i(u_i)$ é o conjunto dos multiplicadores ótimos, então (21) equivale a:

$$\text{Minimizar } v(u) = \sum_{i=1}^{n} v_i(u_i)$$

$$\sum_{i=1}^{n} u_i = b_o \qquad (27)$$

v(u) é uma função-Max (escreva,por exemplo, o dual de (26)) e:

$$\partial v(u) = \{ (\pi_1 \ldots \pi_n) \mid -\pi_i \in \Pi_i(u_i) \} \qquad (28)$$

Obtém-se de novo um problema de otimização não diferenciável restrito (Ω é aqui uma variedade linear).

Em Mahey (1982), a decomposição dual (25) é combinada com a decomposição primal (27). Os passos nos espaços duais e primais são escolhidos segundo a estratégia (17). Os valores dos custos dual e primal são usados como aproximações do custo ótimo em (17) e a redução do "gap" entre os dois valores fornece um critério de parada adequado. Esta idéia é explorada também em Sen e Sherali (1986).

Em Mahey (1986b), o método do subgradiente é usado como fase 1 de um algoritmo de decomposição misto onde as não diferenciabilidades encontradas (soluções não únicas em (22)) são eliminadas pela alocação de recursos no subproblema. No fim, cada subproblema recebe preços e recursos e o nível de coordenação (programa-mestre) é eliminado garantindo a descentralização do processo de decisão.

4. CONCLUSÃO

As técnicas de subgradientes fornecem algoritmos de baixo custo e bem adaptados aos modelos de decomposição em programação linear. A convergência não depende da qualidade das informações locais usadas como direções de busca mas de parâmetros associados à geometria da função como a condição. Os estudos de J.L. Goffin (1977 e 1980) são fundamentais e precisam ser aprofundados apesar da aparente dificuldade de aproximar corretamente essas informações globais. Se isso fosse possível, teríamos algoritmos talvez lentos (taxas ligeiramente inferiores a 1), mas cujo número de iterações para obter uma dada redução do passo não depende da dimensão do problema.

A convergência finita pode ser obtida em seguida pelos métodos clássicos à partir da solução dada pelo método do subgradiente.

REFERÊNCIAS BIBLIOGRÁFICAS

Clarke F.H. (1975). "Generalized gradients and applications", Trans. Amer. Math. Soc., 205, pp . 247-262.

Dantzig G.B. & P. Wolfe (1960). "The decomposition algorithm for linear programs", Econometrica, 29,4, pp. 767-778.

Dirickx Y. & L.P. Jennergren (1979). System Analysis by Multilevel Methods,J.Wiley.

Goffin J.L. (1977). "On convergence rates of subgradient optimization methods" , Math. Prog., 13, pp. 329-347.

Goffin J.L. (1980). "The relaxation method for solving systems of linear inequalities". Math. Oper. Res.,5, pp. 388-414.

Held M. & R.M. Karp (1971). "The traveling-salesman problem and minimum spanning trees". Math. Prog., 1, pp. 6-25.

Held M.,P. Wolfe & H.P. Crowder (1984) . "Validation of subgradient optimization", Math. Prog., 6, pp. 62-88.

Hiriart-Urruty J.B. (1982). "Approximating a second-order directional derivative for nonsmooth convex functions", SIAM J. Control and Optimization, 20,6, pp. 783-807.

Kelley J.E. (1960). "The cutting plane method for solving convex programs" , J. SIAM, 8, pp. 703-712.

Kiwiel K.C. (1985). Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics, 1133, Springer V.

Lemaréchal C. (1978). "Bundle methods", in Non Smooth Optimization, C. Lemaréchal & R. Mifflin, eds., Proc. IIASA, Pergamon Press.

Lemaréchal C. (1982). Basic Theory in Non-differentiable Optimization, Rap. de Recherche nº 181, INRIA.

Mahey P. (1982). "Decomposition of large-scale linear programs by subgradient optimization", Mat. Aplic. Comp. 1, 2 , pp. 121-134.

Mahey P. (1985). "Subgradient techniques and combinatorial optimization", W.P. rep. nº 85397, Univ. Bonn, RFA.

Mahey P. (1986a). "A subgradient algorithm for accelerating the Dantzig-Wolfe decomposition method", Methods of Oper. Res., 53, pp. 697-707.

Mahey P. (1986b). "Methodes de décomposition et décentralisation en programmation linéaire", RAIRO-Recherche Opérationnelle, 20,4.

Polyak B.T. (1969). "Minimization of unsmooth functionals", USSR Compt. Math. and Math. Physics, 9, pp. 14-29.

Sen S. & H.D. Sherali (1986). "A class of convergent primal dual subgradient algorithms for decomposable convex programs", Math. Prog., 35,3, pp. 279-297.

Shor N.Z. (1985). Minimization Methods for Nondifferentiable Functions, Springer V.

Wolfe P. (1975). "A method of conjugate subgradients for minimizing nondiffentiable convex functions", Math. Prog. Study 3, pp. 145-173.

Womersley R.S. (1982). "Optimality conditions for piecewise smooth funtions", Math. Prog. Study 17, pp. 13-27.

Mat. Aplic. Comp., V. 1, nº 2, pp. 121 a 134, 1982

*DECOMPOSITION OF LARGE-SCALE LINEAR PROGRAMS BY SUBGRADIENT OPTIMIZATION**

P. MAHEY
Deptº de Engenharia Elétrica – PUC/RJ
R. Marquês de S. Vicente 209
22453 Rio de Janeiro – RJ – Brasil

ABSTRACT

The decomposition of large-scale linear programs by a price - or resource - coordination scheme leads to the optimization of piecewise linear functions. The large number of breakpoints of these function turns the convergence of direct methods hopeless. In this paper a Russian method for subgradient optimization is used to solve the resource allocation problem. As the step size depends on some lower bound of the primal function, a dual decomposition is performed and an approximation of this bound is obtained again by a subgradient optimization.

RESUMO

A decomposição de programas lineares de grande porte por alocação de preços ou de recursos gera funções lineares por partes cuja otimização dificilmente será realizada através dos métodos diretos. Neste artigo o problema é resolvido usando-se um algoritmo de subotimização cuja convergência foi estudada por Polyak [8] entre outros. Uma decomposição primal e uma decomposição dual são realizadas em paralelo para apertar a solução ótima entre limites inferior e superior que permitem o cálculo do passo na direção do subgradiente, problema fundamental para essa categoria de técnicas.

*Part of this work was presented at the IV Congresso Nacional de Matemática Aplicada e Computacional, Rio de Janeiro, September 1981.

1. INTRODUCTION – DECOMPOSITION BY RESOURCE ALLOCATION

Let us consider the following linear program (P):

$$(P)\quad \left|\begin{array}{ll} \text{Min } \sum_{i=1}^{n} c_i x_i & \\ \sum_{i=1}^{n} A_i x_i \leq b & (1) \\ B_i x_i \leq d_i & i=1,\ldots,n \\ x_i \geq 0 & \end{array}\right.$$

with $x_i \in R^{n_i}$, $b \in R^m$, $d_i \in R^{m_i}$.

Let X and f^* be respectively the set of optimal solutions and the optimal value of (P).

A way to solve this problem is to decompose the coupling constraints (1) by a priori fixing the quantity of corporate resources allocated to each subsystem.

Let $y_i \in R^m$ be the resource vector allocated to the subsystem number i.

We can now define the subproblems (SP_i) as:

$$(SP_i)\quad \left|\begin{array}{ll} \text{Min } C_i x_i & \\ A_i x_i \leq y_i & (2) \\ B_i x_i \leq d_i & \\ x_i \geq 0 & \end{array}\right.$$

The allocation y_i must be feasible in two ways: first each subproblem(SP_i) must be feasible. Let Y_i be the set of such y_i:

$$Y_i = \{y_i \in R^m | x_i \in R^{n_i} \text{ such that } A_i x_i \leq y_i,\ B_i x_i \leq d_i,\ x_i \geq 0\}\ .$$

We also must have:

$$\sum_{i=1}^{n} y_i = b \quad .$$

Let now $v_i(y_i)$ be the optimal value of (SP_i), and $v(y_1,\ldots,y_n)=\sum_{i=1}^{n} v_i(y_i)$.

Clearly, to solve (P) we must minimize the function v for feasible allocation y_i. The transformed problem (TP) is then:

$$(TP) \quad \left| \begin{array}{ll} \text{Min } v(y) & \\ \sum_{i=1}^{n} y_i = b & (3) \\ y_i \in Y_i \quad , \quad i=1,\ldots,n & (4) \end{array} \right.$$

In an early paper, Kornai and Liptāk [5] have attempted to solve (TP) by a simple adjustment process that used an average information on the dual variables associated with the constraints (2) of (SP_i) to modify y_i. In fact their algorithm did not converge.

A serious study of this problem may be found in Lasdon [6] and an algorithm for the linear case has been developed by Grinold [2]. The main results for whose proofs we refer to [6] may be summarized in:

1. v is a piecewise linear convex function. It has a finite number of breaklines where v is not differentiable.
2. Let $u_i \in R^m$ be an optimal multiplier associated with the constraints (2) in (SP_i). Then the vector $u=(u_1,\ldots,u_n)$ is a subgradient of v at y. This means that if y' are feasible allocations we have:

$$v(y') - v(y) \geq u(y'-y) \quad .$$

In his paper, Grinold [1] presents a feasible descent algorithm and the computation of the step size is such that only breakpoints are generated.

Minoux (in [7]) reports some successful experiments on a multicommodity flow problem based on a subgradient method to solve the decomposed problem. This is also the approach that we have chosen.

2. THE SUBGRADIENT METHOD

Held and Karp [3] have introduced first a subgradient algorithm in the linear case for large-scale transportation problems. Held, Wolfe and Crowder [4] justified then the use of Soviet methods to achieve convergence. Let us summarize the characteristics of these methods.

We consider the problem: $\underset{x \in X}{\text{Min}}\ f(x)$

where f is convex, defined on Ω and has a subdifferential at each point of Ω. Let g be a subgradient of f at x. Then

$$\forall\ y \in \Omega\ ,\quad f(y) - f(x) \geq g(y-x)\ .$$

In Fig. 1, we show geometrically that the directions $d = -g$ may not be a descent direction for f at a nondifferentiable point. However $d = -g$ is always a descent direction for the distance of x to the optimal point x^*.

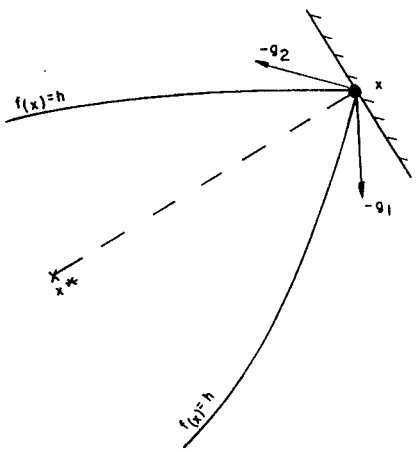


Fig. 1

We can observe that the smallest the scalar product $\sigma(x) = \dfrac{-g_1(x^*-x)}{\|g_1\|\,\|x^*-x\|}$, the worse is the direction $-g_1$ with respect to f. $\chi = \min \sigma(x)$ is the condition

number of the function f and it is a useful information to compute the step size in the direction -g and to determine the rate of convergence of the subgradient algorithm (see e.g. Goffin [1]).

Polyak [8] has proved that the following algorithm converges:

$$x^{k+1} = x^k - \lambda_k g^k$$

where $g^k \in \partial f(x^k)$, being $\partial f(x^k)$ the subdifferential at $x=x_k$,

$$\lambda_k > 0 \text{ is such that } \begin{cases} \lambda_k \to 0 \\ \sum_k \lambda_k \to +\infty \end{cases} \tag{5}$$

If λ_k is computed by:

$$\lambda_k = \rho^k \frac{f(x_k)-f^*}{\| g^k \|^2} \tag{6}$$

where $f^*=f(x^*)$ and $0<\varepsilon_1 \leq \rho^k \leq 2-\varepsilon_2$, $\varepsilon_2>0$ and if in addition f satisfies the following conditions:

(i) $\| g^k \| \leq C$ on $S = \{x \mid \| x-x^0 \| \leq \| x^*-x^0 \|\}$

(ii) $f(x)-f^* \geq m\|x-x^*\|$, $m>0$, for every x such that $\| x-x^* \| \leq \| x^0-x^* \|$

then the algorithm converges weakly to x^* with a geometric rate.

3. A DECOMPOSITION ALGORITHM USING SUBGRADIENTS

We are going now to apply the subgradient algorithm described above to the solution of problem (TP).

The implementation of this technique to the resource-allocation problem has led us to face two principal problems: to fulfill the convergence conditions of (5)-(6) and to keep the allocations feasible.

CONVERGENCE OF THE ALGORITHM. In practice, we don't know the optimal value V^* of the primal cost function. We may use an approximation $\overline{V}$ of V^* to compute the step size λ_k by (6).

If $\overline{V}>V^*$, the algorithm converges to $\overline{V}$ or generates a k such that $V_k<\overline{V}$ (see e.g. [8]). But in general, it is quite improbable that we may be able to find such an approximation.

If $\overline{V}<V^*$, which is the practical case, we have two possibilities yielding two distinct algorithms:

(i) $\overline{V}$ is a good approximation. In order to satisfy the condition $\lambda_k \to 0$, we must have $\rho_k \to 0$. Held, Wolfe and Crowder [4] have reported a satisfactory experimental way to compute the parameters ρ_k: "$\rho_k=2$ for 2m iterations and then successively halve ρ_k and the number of iterations until the number of iterations reaches some threshold value z; then halve ρ_k every z iterations". Although that rule violates the condition $\sum_k \lambda_k \to \infty$, it has yielded good results.

(ii) We may choose $\overline{V}$ such that:

$$\overline{V} = \operatorname{Max}_t M(\pi^t)$$

where $\operatorname{Max}_{\pi \geq 0} M(\pi)$ is the dual problem associated to (P) by dualization of the coupling constraints (1),and t are some iterations of an algorithm maximizing M in parallel to the present algorithm:

$$M(\pi) = \sum_{i=1}^{n} \operatorname{Min}\{(C_i+\pi A_i)x_i / B_i x_i = b_i, x_i \geq 0\} - \pi b \qquad (7)$$

$$\pi \geq 0 \quad , \quad \pi \in R^m \quad .$$

As $M(\pi)$ is a concave piecewise linear function, we can use a similar subgradient optimization. Indeed, for a fixed $\pi \geq 0$, the dual problem separates in n subproblems:

$$\begin{aligned} &\operatorname{Min} (C_i+\pi A_i)x_i \\ &B_i x_i = b_i \qquad (8) \\ &x_i \geq 0 \end{aligned}$$

If $x_i(\pi)$ is an optimal solution of (7), i=1,...,n, then $g=\sum_{i=1}^{n} A_i x_i(\pi)-b$ is a subgradient of M at π.

The algorithm modifies the prices π by:

$$\pi^{t+1} = \pi^t + \mu_t\, g^t \tag{9}$$

where

$$\mu_t = \rho_t \frac{\overline{M}-M(\pi^t)}{\| g^t \|^2} \tag{10}$$

and $\overline{M}$ is an upper bound of M^*. We can use the best value generated by the primal algorithm:

$$\overline{M} = \min_k V(y^k)$$

where k are the passed iterations of the primal algorithm.

The sequences $\{\mu_t\}$ and $\{\lambda_k\}$ have the same characteristics and in practice, we shall modify respectively $\overline{V}$ and $\overline{M}$ after a fixed number of iterations in each primal and dual decomposition algorithms.

FEASIBILITY OF THE ALLOCATIONS. The master problem (TP) is a constrained problem. We have used a projected subgradient on constraints (3) which are very easy to handle. Indeed, if u^k is a subgradient of V(y) at $y=y^k$, then

$$z^k = (z_1^k,\ldots,z_n^k) \quad \text{with} \quad z_i^k = y_i^k + \lambda_k\, u_i^k$$

is projected on (3) yielding:

$$y^{k+1} = (y_1^{k+1},\ldots,y_n^{k+1}) \quad \text{with} \quad y_i^{k+1} = y_i^k + \lambda_k \left(u_i^k - \frac{\sum_{i=1}^{n} u_i^k}{n} \right). \tag{11}$$

As the constraints (3) form a linear manifold which contains the optimal solution set, we have:

$$\| y^{k+1}-y^* \| \leq \| z^k-y^* \| \leq \| y^k-y^* \|$$

if λ_k satisfies the convergence conditions (5). This means that the rate of convergence of the projected subgradient algorithm is never worse than the rate of convergence of (5).

On the other hand, constraints (4) are very numerous and unknown. If some subproblem (SP_i) is infeasible for an allocation y_i, then its corresponding dual

is unbounded. If that occurs, we may extract a subgradient u_i from an optimal extreme ray of the dual subproblem.

SUMMARY OF THE ALGORITHMS. The central idea of these algorithms is the optimization of both primal - and dual - decomposition problems by a subgradient method. This will be done sequentially for case (i) and in parallel for case (ii).

We present first the two distinct routines, DUAL for the dual problem and PRIMAL for (TP).

DUAL:

0. Initialize $\overline{M}$, π^0, ρ^0
   $t = 0$

1. Solve (7), i=1,...,n, for $\pi = \pi^t$
   Compute $M(\pi^t)$ by (6).

2. Compute a subgradient:

$$g^t = \sum_{i=1}^{n} A_i x_i(\pi^t) - b$$

3. Compute the step:

$$\mu^t = \rho^t \frac{\overline{M}-M(\pi^t)}{\| g^t \|^2}$$

   Set $\pi^{t+1} = \pi^t + \mu^t g^t$

4. Stopping rule*
   $t = t+1$, go back to 1.

PRIMAL:

0. Initialize $\overline{V}$, y^0, ρ^0 .
   $k = 0$

1. Solve the subproblems (SP_i), i=1,...,n, for $y_i = y_i^k$.

---

*By this, we mean that we check $\left|\frac{\overline{M}-M(\pi^t)}{\overline{M}}\right|$ or ρ^t, or the iteration number t to decide whether we stop (and eventually activate PRIMAL) or modify ρ^t and continue.

2. If (SP_i) is infeasible for some i, get an optimal extreme ray and set r_i equal to the m-dimensional part of that ray associated to constraints (2). Then compute $\bar{y}_i^k = y_i^k - \alpha r_i$, $\alpha>0$, and solve again (SP_i) for $y_i = \bar{y}_i^k$. Repeat until feasibility is obtained.

3. Set U_i^k equal to the m-dimensional part of the dual optimal solution in (SP_i). Compute $V(y^k)$.

4. Compute the step:

$$\lambda_k = \rho_k \frac{V(y^k)-\bar{V}}{\| U^k \|^2}$$

and set $y_i^{k+1} = y_i^k - \lambda_k \left(U_i^k - \frac{\sum_{i=1}^{n} U_i^k}{n} \right)$, $i=1,\ldots,n$.

5. Stopping rule (see DUAL)

$k = k+1$ and go back to 1.

We summarize now the two composite algorithms we have tested:

A1. This algorithm corresponds to case (i). DUAL is used to give a good lower bound $\bar{V}$ for PRIMAL. But as DUAL must use an upper bound $\bar{M}$ to compute the step size, then we must adjust ρ_t as in (i). When ρ_t becomes inferior to a fixed ε_1, then we switch to DUAL setting $\bar{V}=\max_t M(\pi^t)$. In the second phase, we adjust ρ_k in the same way and stop the algorithm when $\left|\frac{\bar{V}-V(y^k)}{\bar{V}}\right|$ becomes inferior to a fixed ε_2.

A2. This algorithm corresponds to case (ii). DUAL and PRIMAL are set up in parallel and after every ℓ iterations the lower and upper bounds are actualized by:

$$\bar{V} = \max\{\max_t M(\pi^t), \bar{V}\} \quad \text{and} \quad \bar{M} = \min\{\min_k V(y^k), \bar{M}\}$$

4. RESULTS

We have tested the algorithms A1 and A2, with a production-inventory planning model, as described below.

We have n activities that have to be planned on T periods:

Let x_{it} be the stock level of the ith activity at the end of period t.

Let u_{it} be the production level of the ith activity for the tth period.

Let d_{it} be the known demand of product i for the tth period, and b_t a resource limitation in period t.

The constraints on each activity are:

$$x_{it} = x_{i,t-1} + \alpha_i u_{it} - d_{it} \quad , \quad t=1,\ldots,T$$

$$m_i \leq x_{it} \leq M_i$$

$$0 \leq u_{it} \leq F_i$$

and the coupling constraints are:

$$\sum_{i=1}^{n} u_{it} \leq b_t \quad , \quad t=1,\ldots,T \ .$$

With n=7, T=12, problem (P) has 96 constraints (not included the bounds on each variable) and 168 variables. A family of typical problems was generated by random perturbations on the right-hand side b.

Figure 2 and Table 1 show the evolution of $M(\pi^t)$ and $V(y^k)$ for two tests on algorithm A1. The first (A11) has used ε_1=1 and the second (A12) has used ε_1=0.3.

Fig. 3 and Table 2 show the evolution of $M(\pi^t)$ and $V(y^k)$ for the same problem solved by algorithm A2.

We observe first that in Fig. 2, the first phase DUAL of A1 behaves well, although the chosen upper bound $\bar{M}$=200000 was 12% above the maximal value. This fact confirms earlier results by Held et al. [4] We may add that there is no gain in the total iteration number when we use A12 which has performed 11 DUAL more iterations than A11 to compute a better lower bound for PRIMAL. But, as the PRIMAL iterations are more "expensive" (the subproblems are larger and some extra work to keep them feasible is often needed), the algorithm A12 is considered better (the computing time on the IBM-370 of the Catholic University of Rio de Janeiro was 1 minute 40 seconds for A12 and 2 minutes 20 seconds for A11).

The behaviour of A2 shows that if we ignore the first cycle (the 10 first iterations of each routine) where the poor initial bounds amplify the oscillating tendency of the subgradient algorithm, the "tail" of the iterations

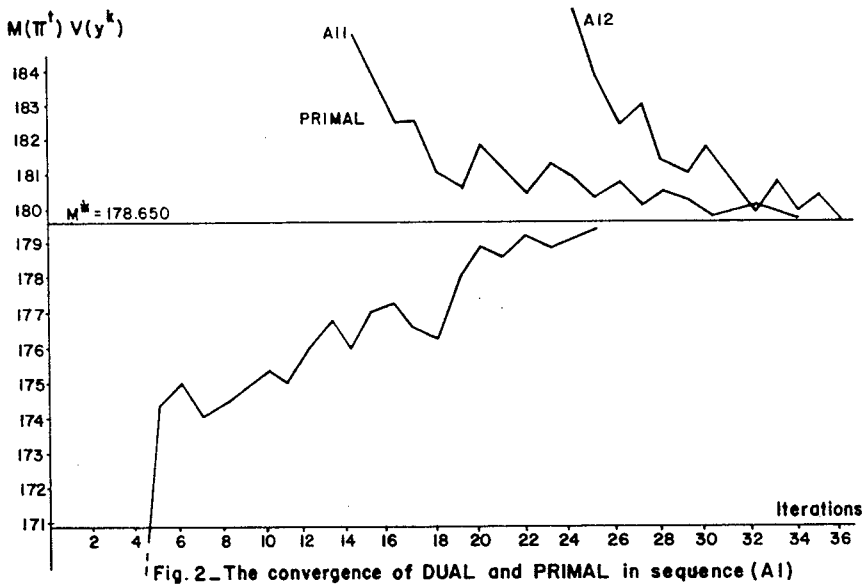


Fig. 2_The convergence of DUAL and PRIMAL in sequence (A1)

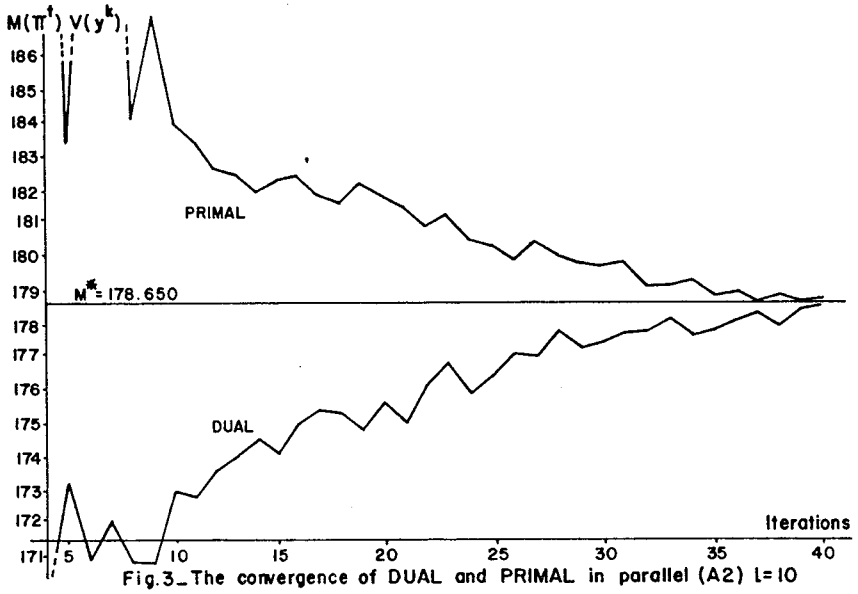


Fig.3_The convergence of DUAL and PRIMAL in parallel (A2) L=10

		DUAL	PRIMAL					
t	ρ^t	$M(\pi^t)$	$V(y^k)$					
1	2	101228						
2	2	159316						
3	2	163110						
4	2	169971						
5	2	173420						
6	2	174085						
7	2	173133						
8	2	173489						
9	2	173902						
10	2	174446	A11					
11	2	174060						
12	2	175124 →	$\bar{V}$=175124	k				
13	1	175872	191200	1				
14	1	175018	184185	2				
15	1	176009	182860	3				
16	1	176377	181473	4				
17	1	175632	181512	5				
18	1	175325	180010	6				
19	.5	177121	179619	7				
20	.5	177948	180883	8				
21	.5	177550	180227	9				
22	.5	178304	179416	10	→	A12		
23	.5	177912	180301	11		$\bar{V}$=178304	k	
24	.5	178281	179997	12		185500	1	
25	.25	178441	179325	13		182887	2	
			179771	14		181379	3	
			179111	15		182043	4	
			179508	16		180926	5	
			179284	17		180012	6	
			178887	18		180819	7	
			178931	19		179750	8	
			179146	20		179012	9	
			178804	21		179418	10	
			178711	22		178902	11	
						179100	12	
						178698	13	

Table 1: A1

	DUAL			PRIMAL	
t	$M(\pi^t)$	$\overline{M}$	$\overline{V}$	$V(y^k)$	k
1	101228	300000	0	198650	1
2	166410	-	-	184107	2
3	172098	-	-	185843	3
4	168425	-	-	191219	4
5	173175	-	-	183605	5
6	171000	-	-	188221	6
7	172117	-	-	215000	7
8	170881	-	-	184322	8
9	170919	-	-	187130	9
10	173043	-	-	184052	10
11	172815	183605	173043	183508	11
12	173620	-	-	182766	12
13	174033	-	-	182519	13
14	174568	-	-	182041	14
15	174119	-	-	182347	15
16	175094	-	-	182494	16
17	175428	-	-	181880	17
18	175317	-	-	181612	18
19	174816	-	-	182228	19
20	175660	181612	175660	181845	20
21	175032	-	-	181533	21
22	176190	-	-	180946	22
23	176833	-	-	181310	23
24	175904	-	-	180572	24
25	176421	-	-	180413	25
26	177126	-	-	179995	26
27	177044	-	-	180505	27
28	177801	-	-	180107	28
29	177245	-	-	179829	29
30	177479	-	-	179773	30
31	177750	179773	177801	179881	31
32	177796	-	-	179124	32
33	178124	-	-	179156	33
34	177645	-	-	179338	34
35	177828	-	-	178846	35
36	178124	-	-	179021	36
37	178383	-	-	178664	37
38	177990	-	-	178922	38
39	178482	-	-	178718	39
40	178571	-	-	178750	40
		178571	178664		

$M^*=V^*=178650$

Table 2. A2

gives us a way to strengthen the optimal value between two convergent sequences.

5. REFERENCES

[1] GOFFIN, J.L. - "On the convergence rates of subgradient optimization methods", *Math. Prog.*, vol. 13, nº 3, pp. 329-347, 1977.

[2] GRINOLD, R.C. - "Steepest ascent for large-scale linear programs", *SIAM Review*, vol. 14, nº 3, pp. 447-464, 1972.

[3] HELD, M. and KARP, R.M. - "The traveling salesman and minimum spanning trees", *Math. Prog.*, vol. 1, nº 1, pp. 6-25, 1971.

[4] HELD, M.; WOLFE, P. and CROWDER, H.P. - "Validation of subgradient optimization", *Math. Prog.*, vol. 6, nº 1, pp. 62-88, 1974.

[5] KORNAI, J. and LIPTÁK, J. - "Two-level planning", *Econometrica*, vol. 33, pp. 141-169, 1965.

[6] LASDON, L.S. - *Optimization Theory for Large Systems*, MacMillan, 1970.

[7] MINOUX, M. and GORDRAN, M. - *Graphes et Algorithmes*, Eyrolles, 1979.

[8] POLYAK, B.T. - "Minimization of unsmooth functionals", *USSR Computational Math. and Math. Physics*, vol. 9, nº 3, pp. 509-521, 1969.

Recebido em 11/XI/81; Revisado em 16/VI/82.

CHAPITRE 4

UNE METHODE DE DECOMPOSITION MIXTE

On s'attaque ici au problème central de la décentralisation associé à une approche par décomposition. L'algorithme proposé ci-dessous a été conçu pour le cas de la Programmation Linéaire et étendu par la suite au cas convexe. En fait, c'est l'analyse de la dégénérescence introduite dans les sous-problèmes par la décomposition qui en constitue le point de départ.

En effet, si on analyse les solutions de base optimales des sous-problèmes associées aux décompositions primales et duales, on constate :

- la méthode primale affecte chaque contrainte de couplage dans chaque sous-problème. Le nombre de contraintes associé à l'ensemble des sous-problèmes a donc augmenté de (p-1)q, si p est le nombre de sous-problèmes et q est le nombre de contraintes de couplage.
- la méthode duale relaxe toutes les contraintes de couplage en leur affectant un prix dans les fonctions objectifs des sous-problèmes. Le nombre total de contraintes a donc diminué de q.

D'autre part, dans les deux cas, la réunion des conditions d'optimalité des sous-problèmes quand les paramètres de coordination ont leurs valeurs optimales équivaut aux conditions d'optimalité du problème de départ. On en déduit que :

- dans la méthode primale, (p-1)q variables de base sont nulles à l'optimum et, par conséquent, au moins p-1 sous-problèmes sont dégénérés.
- dans la méthode duale, la solution optimale ne peut être formée par les solutions de base des sous-problèmes. Donc, au moins un sous-problème possède des solutions optimales non uniques, c'est-à-dire que son dual est dégénéré.

Pour éliminer ces dégénérescences et, par suite, pour garantir la décentralisation, on doit affecter chaque contrainte à exactement un sous-problème (on suppose ici pour simplifier que toutes les contraintes de couplage sont actives à l'optimum). Le choix de l'affectation des contraintes aux sous-problèmes n'est pas quelconque, il doit garantir en particulier l'unicité des solutions à l'optimum.

[11] "*Méthodes de décomposition et décentralisation en programmation linéaire* ", RAIRO- Recherche Opérationnelle, 20,4, 1986, pp. 287-306.

Ce travail reprend les résultats décrits précédemment sur la décentralisation des méthodes classiques de décomposition et présente

un nouvel algorithme de décomposition dont les caractéristiques sont les suivantes :

- il s'agit d'une méthode mixte qui affecte des prix et des ressources dans chaque sous-problème.
- chaque sous-problème est décentralisé au voisinnage de l'optimum.
- une méthode de sous-gradient est utilisée dans une première phase pour approcher la solution optimale duale.
- dans la dernière phase, on retrouve une procédure à un niveau, c'est-à-dire que les sous-problèmes échangent entre eux les paramètres de coordination sans l'aide d'un niveau supérieur de coordination.

On y montre tout d'abord l'existence d'une partition des contraintes de couplage telles que chaque sous-problème auquel ont été affectés les paramètres de coordination optimaux possède une unique paire de solutions primale- duale optimales, également optimales pour le problème de départ. Pour cela , chaque contrainte non affectée dans un sous-problème intervient dans sa fonction objectif par l'intermédiaire d'un prix. On met alors en évidence une certaine bifonction des paramètres de coordination primaux et duaux dont le point-selle caractérise la solution optimale du problème de départ. La forme très particulière du sous-différentiel de cette bifonction suggère un algorithme du type point-fixe. L'avantage de cette approche est qu'elle permet d'éviter la construction d'un programme-maître pour la mise à jour des paramètres. L'inconvénient principal est qu'elle nécessite, comme toute méthode de point fixe, de conditions qui garantissent sa convergence comme la dominance des blocs diagonaux mis à jour par les partitions des lignes et des colonnes. Si les conditions de convergence que nous avons déduites de la procédure sont difficilement exploitables, nous avons proposé une méthode heuristique qui permet d'effectuer l'affectation des contraintes aux sous-problèmes et qui a donné d'excellents résultats sur des modèles de programmation linéaire de taille moyenne.

L'algorithme décrit ci-dessus est en fait similaire à une méthode de décomposition à un niveau étudiée par Cohen (1984) dans le cas linéaire-quadratique. Celui-ci a d'ailleurs envisagé la substitution des itérations de point-fixe par une méthode de sous-gradient. Il fait partie également des approches par 'prédiction des interactions' analysées par Titli dans le cadre de la Commande Hiérarchisée (1975). Notre préoccupation principale fut tout d'abord de rechercher des modèles d'application où les conditions de convergence étaient naturellement satisfaites. En premier lieu, nous avons cherché à étendre cette approche au cas convexe.

[12] "*Mixed decomposition algorithms for decentralized planning* ", XI IFORS Triennial Conference, Buenos Aires, 1987.

Dans cet article, nous considérons un modèle général de

programmation convexe sur un espace produit avec des contraintes de couplage. Nous reprenons l'analyse de la convergence de la méthode mixte décrite ci-dessus en identifiant la matrice de transition de l'itération de point-fixe.

Les problèmes de multiflots constituent un cadre privilégié pour l'application des méthodes de décomposition . Gondran et Minoux (1979) décrivent plusieurs modèles où sont appliquées les méthodes de décomposition par les prix et par les ressources. L'intérêt de la décomposition réside dans le fait que chaque sous problème est un problème de flot simple. L'allocation de contraintes équivaut à fixer des capacités sur certains arcs. Ce dernier point est favorable à l'application de la méthode mixte, car les sous-problèmes conservent la même structure quel que soit le nombre de contraintes de couplage qui y sont affectées. La méthode a donc été appliquée à un problème de biflot issu d'un modèle original pour le problème du voyageur de commerce (Finke et al, 1984). Ce travail a fait l'objet d'une communication au Congrès Européen de Recherche Opérationnelle en 1989 et se poursuit dans le cadre d'un mémoire de DEA. L'objectif principal est l'amélioration des performances du modèle de biflot et la recherche de conditions simplifiées pour la convergence de la méthode de point-fixe décrite plus haut dans le cadre plus général des problèmes de multiflots.

[13] "Decomposition of optimal two-commodity networks : Application to the TSP", en préparation.

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 20, n° 4, novembre 1986, p. 287 à 306)

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

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/86/04 287 20/$ 4.00
© AFCET-Gauthier-Villars

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 :

$$\text{Minimiser} \sum_{i=1}^{n} c_i x_i$$

$$(\text{P0}) \quad \begin{cases} \text{sous les contraintes:} \quad \sum_{i=1}^{n} A_i x_i = b_0 & (1) \\ x_i \in S_i = \{ x_i \in R^{n_i} \mid B_i x_i = b_i,\ x_i \geqq 0 \},\ \forall i = 1, \ldots, n & (2) \end{cases}$$

avec

$$x_i \in R^{n_i}, \qquad i=1, \ldots, n$$

$$b_0 \in R^{m_0}$$

$$b_i \in R^{m_i}, \qquad i=1. \ldots, n.$$

Hypothèses initiales :

- $\forall i=1, \ldots, n$:

B_i est de rang plein.

S_i est borné.

- La solution optimale de (P0), $x^*=[x_1^*, \ldots, x_n^*]$ 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 :

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

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

$$(\mathrm{P1})_i \begin{cases} \text{Minimiser } (c_i+\bar{p}A_i)\, x_i \\ x_i \in S_i. \end{cases}$$

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

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

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 :

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

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 coordinateur calcule alors :

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

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 coordinateur utilise les hyperplans supports associés aux sous-gradients calculés aux itérations précédentes pour approcher la

fonction h. Il résoud alors le programme linéaire suivant :

$$(\mathrm{P3})\begin{cases}\text{Maximiser } \sigma \\ \sigma \leqq h(p^l)+g^l.(p-p^l), \qquad l=1,\ \ldots,\ t \\ \text{où} \\ \qquad\qquad g_l \in \partial h(p^l).\end{cases}$$

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, $T_1+1 \leqq l \leqq T_2$, 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 $p^{T_1+T_2}$.

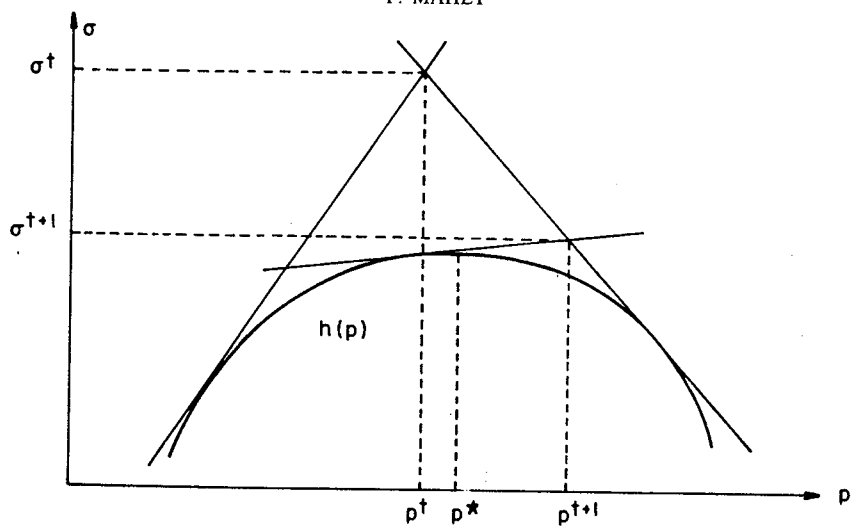


Figure 1. – Instabilité des solutions.
$\sigma^{t+1}<\sigma^t$ mais $\|p^{t+1}-p^*\|>\|p^t-p^*\|$.

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

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

3. DÉCOMPOSITION PAR LES RESSOURCES

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

$$\sum_{i=1}^{n} u_i=b_0. \tag{6}$$

Chaque sous-système doit alors résoudre :

$$(\mathrm{P}4)_i \begin{cases} \text{Minimiser } c_i x_i \\ A_i x_i=u_i \\ x_i \in S_i. \end{cases}$$

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

$$(P5) \quad \begin{cases} \text{Minimiser } v(u) = \sum_{i=1}^{n} v_i(u_i) \\ \sum_{i=1}^{n} u_i = b_o. \end{cases}$$

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

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

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

Comme dans le cas précédent, v n'est, en général, pas différentiable au point $u^* = (u_1^*, \ldots, u_n^*)$, 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 (A 4)

$$u^t \in \Omega_0 = \left\{ u = (u_1 \ldots u_n)^T \quad \sum_{i=1}^{n} u_i = b_0 \right\} \qquad \text{et} \qquad \pi^t \in \partial v(u^t).$$

On doit projeter orthogonalement π^t sur

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

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

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

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

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

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

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

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

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 coordinateur résoud un programme linéaire où on a rajouté la contrainte d'admissibilité :

$$(\mathrm{P6}) \quad \begin{array}{c} \mathrm{Min}\, w \\ w \geqq \sum_{i=1}^{n} [v_i(u_i^l) + \pi_i^l . (u_i - u_i^l)], \\ l = 1, \ldots, t, \qquad \pi^l \in \partial v(u^l) \\ \sum_{i=1}^{n} u_i = b_0. \end{array}$$

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 $\{\varepsilon_{1\,t}\}$ et $\{\varepsilon_{2\,t}\}$:

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

On choisira

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

et

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

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 $|\bar{h}-\bar{v}|\to 0$, 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 (A 2) deviendrait alors la méthode de Benders [2] qui a fait l'objet d'un grand nombre d'études et applications dont nous ne nous servirons 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 dégénéréescence 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 pertubations 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 :

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

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 :

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

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 $\det B_1^r . \det B_2^r$ (voir *fig.* 2). On a donc :

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

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,

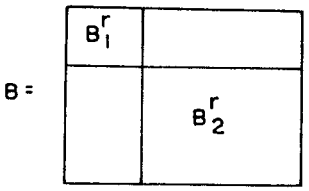


Figure 2

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 $u=(u_1, \ldots, u_n)$ telles que $u \in R^{m_0}$ et u_i a la même dimension que b_{0i} :

$$(P7)\begin{cases} \text{Minimiser } \sum_{i=1}^{n} c_i x_i \\ A_{ii} x_i - u_i = 0 \\ u_i + \sum_{k \neq i} A_{ik} x_k = b_{0i} \qquad i=1, \ldots, n \\ x_i \in S_i. \end{cases}$$

On effectue alors une relaxation lagrangienne des contraintes de couplage :

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

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 :

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

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 :

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

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 polyhèdre 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 résoud 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 :

$$\text{Sous-problème } \underline{i+1} \quad \left\{ \begin{array}{ll} \bar{p}_k = \pi_k^t, & k = 1, \ldots, i \\ \bar{p}_k = \pi_k^{t-1}, & k = i+2, \ldots, n \\ \bar{u}_{i+1} = b_{0,\, i+1} - \sum_{k=1}^{i} A_{i+1,\, k}\, x_k^t & \\ \quad - \sum_{k=i+2}^{n} A_{i+1,\, k}\, x_k^{t-1} & \end{array} \right. \tag{12}$$

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 coordinateur 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

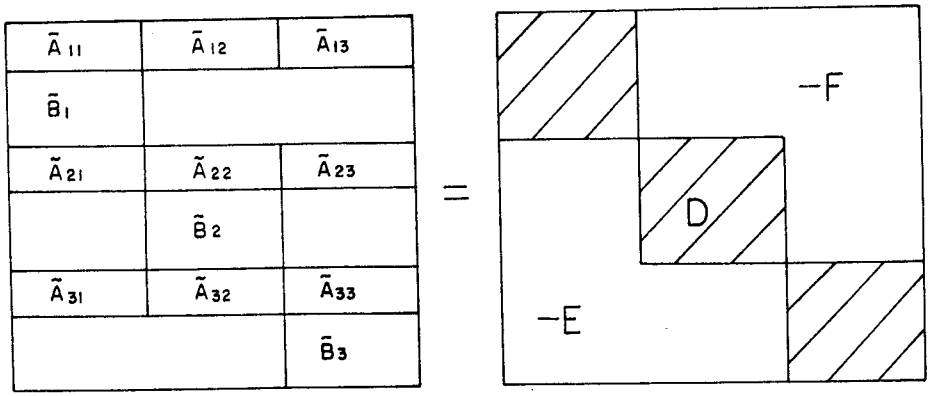


Figure 3. — Forme bloc-diagonale de la base optimale.

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 $r \in \{1, \ldots, m\}$ telle que :

$$g_r^1\, g_r^2 < 0 \tag{14}$$

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

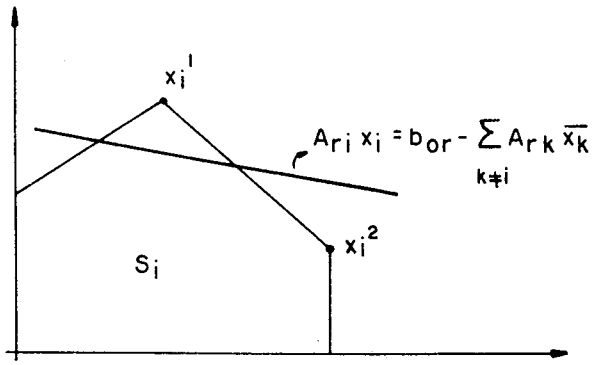


Figure 4

alors, l'hyperplan d'équation

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

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 successsifs. 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 dominance (*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 implanté 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'agorithme 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. LOUTE, *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.

MIXED DECOMPOSITION ALGORITHMS FOR DECENTRALIZED PLANNING

P. MAHEY
Dept Electrical Engineering
PUC/RJ - Rio de Janeiro
Brazil

Abstract:

In this paper, we investigate some mixed (prices + budgets) planning procedures for a coordinate decentralized organization using decomposition models. We show among other results that prices and budgets may be allocated to each subsystem in a very precise way in order to yield convergence towards an equilibrium point with totally decentralized optimal decisions. Indeed, no central agency is needed to coordinate the subsystems which exchange sequentially the primal and dual information between them.

1. Introduction

Given a typical multilevel organization, where firms have to share commodities or goods, a *decentralized planning procedure* is an iterative adjustment of the production and consumption levels of each firm which converges to an optimal plan without the need of centralizing all the local technological constraints. Generally, this is done with the help of an upper decision-making level, the central agency, which receives the proposals of each firm, then computes and sends back to the lower level some" prospective indices", to follow Malinvaud's terminology (Malinvaud, 1967) - It is well known for example that the Dantzig-Wolfe's decomposition method (Dantzig and Wolfe, 1960) can be interpreted as a decentralized planning procedure. A singular feature of this latter method is that the

central agency must recover a global feasible solution, henceforth the global optimal solution, by a convex combination of the local proposals. We call a *completely decentralized procedure*, one such that the sequence of the proposals of each firm converges to the globally optimal decisions.

A *mixed planning procedure* is one which includes prices (dual indices) and resources (primal indices) in the prospective indices. As we shall see below, these indices may or may not be updated by a central agency. We focus in this paper on the construction of completely decentralized procedures and consider their applications to the case of the decomposition of linear programs. In that sense, the mixed strategies proposed here have no common features with the methods proposed in Maier and Vanderweide (1976), Sengupta and Gruver (1976) or Obel (1978).

2. A simplified planning model

In a famous paper, Malinvaud (1967) concluded suggesting a mixed approach to "draw nearer to the methods used in pratice [...]".

We shall analyze below a class of mixed strategies at the light of a simple economic model in which the decentralization leads to local optimization subproblems depending on the price and/or resource allocations.

We consider a multidivision economy with n subdivisions represented by an index i varying from 1 to n. Each division must minimize its loss represented by a convex function $f_i(x_i)$ defined on $\Omega_i \subset X_i$, where X_i is the space of the decision variables of the ith division. Ω_i is a generally polyhedral convex set which represents the technological environment of the ith division. Some global constraints are now added to express either some limitations of the available amount of scarce resources, shared by the divisions or some global demand constraints, or some relations linking the outputs of various divisions, these constraints are represented in a separable form and associated to an index k varying from 1 to p:

$$\sum_{i=1}^{n} g_{ki}(x_i) \leq b_k, \quad k=1,\ldots,p$$

where g_{ki} are convex functions defined on Ω_i.

Let $g_i = [g_{1i}\ldots g_{pi}]^T$ and $b = [b_1..b_p]^T$

Then, the global problem of minimizing the total loss of the multidivision economy is:

$$\text{Minimize } f(x) = \sum_{i=1}^{n} f_i(x_i)$$

$$(P) \left| \begin{array}{l} g(x) = \sum_{i=1}^{n} g_i(x_i) \leq b \\ \\ x \in \Omega_1 \ldots x\ \Omega_n \end{array} \right.$$

We shall denote y_{ki} a primal allocation (a quantity) of the kth resource to the ith division, and u_{ik} a dual allocation (a price) of the kth resource for the ith division. These allocations are feasible when $\sum_{i=1}^{n} y_{ki}=b_k$, for each k=1,...,p,and when $u_{1k}=\ldots=u_{ik}=\ldots=\ldots=u_{nk}$, k=1,..,p. The duality between quantities and prices is structural in the separable case in the sense that a primal feasible direction satisfying $\sum_{i=1}^{n} y_{ki}=0$ is orthogonal to a dual feasible direction satisfying $u_{1k}=..=u_{nk}$, for each k.

The classical decentralized procedures realize feasible allocations and of the same kind for each division. The dual approach, which corresponds to a lagrangian relaxation of the problem,is illustrated by the famous methods of Arrow and Hurwicz (1960) and Dantzig and Wolfe (1960), and the primal approach is illustrated in the methods of Kornai and Liptak(1965) and Ten Kate (1972). In both cases, the decentralization is not complete.

3. A general mixed subproblem

For each division i, we denote $H_i \subset \{1,\ldots,p\}$ the subset of indices of the global constraints allocated to division i and if $\bar{H}_i = \{1,\ldots p\} \setminus H_i$, we suppose that $\bar{H}_i$ is the subset of indices of the price vector allocated to i. In other words, we determine that any division must receive a primal - or a dual-allocation for each resource but not both. The subproblem which must be solved by the ith division is then:

$$\text{Min} \quad f_i(x_i) + \sum_{k \in \bar{H}_i} u_{ik}\, g_{ki}(x_i) = v_i(u,y)$$

$$(SP_i) \left| \begin{array}{l} g_{ki}(x_i) \leq y_{ki}, \quad k \in H_i \\ \\ x_i \in \Omega_i \end{array} \right.$$

where $(u,y) \in R^{nm}$ is the vector of dual and primal allocations. Of course, the pure price-directive allocation corresponds to $H_i = \emptyset$, i,and the pure resource - directive allocation to $H_i = \{1,..p\}$.We first verify the coherence of such a decomposition. Suppose that (x^*,u^*) is a primal-dual pair of Kuhn-Tucker optimal solution for (P) and let $y_{ki}^* = g_{ki}(x_i^*)$.

Theorem 1: x_i^* solves the subproblem (SP_i) for $u_{ik} = u_k^*, k \in \bar{H}_i$, and $y_{ki} = y_{ki}^*, \forall\, k \in H_i$. Moreover u_k^*, $k \in H_i$, are dual optimal for the allocated constraints of SP_i.

Proof: We observe first that x_i^* is feasible for (SP_i) and that, if some constraint qualification holds at x^* for (P),then its specialization to (SP_i) holds at x_i^*. The Kuhn-Tucker first order condition at $x^* = [x_1^* \ldots x_n^*]$ can be written, with $f = [f_1 \ldots f_n]$, $g_k = [g_{k1},\ldots,g_{kn}], \forall\, k$, and $\Omega = \Omega_1 \times \ldots \times \Omega_n$:

$$\nabla f(x^*) + \sum_{k=1}^{p} u_k^*\, \nabla g_k(x^*) \in -\Omega^{*p}$$

where $\Omega^{*p} = \Omega_1^{*p} \times \ldots \times \Omega_n^{*p}$ is the polar cone of the feasible direction cone Ω^* at $x = x^*$, $\Omega^* = \Omega - x^*$. The global condition splits

into n local conditions:

$$\nabla f_i(x_i^*) + \sum_{k=1}^{p} u_k^* \nabla g_{ki}(x_i^*) \; \varepsilon\text{-}\Omega_i^* P \quad , \quad i=1,\ldots,n$$

Then it is clear that x_i^* and $\{u_k^*\}_{k\in H_i}$ satisfy the first-order optimality condition for (SP_i). Observe that the complementarity conditions are satisfied because $g_{ki}(x_i^*)=y_{ki}^*, k \in H_i$.

The decentralized procedure which leads to the splitting of (P) into the n subproblems (SP_i) stems from the following transformation of problem (P). In the same way as we have defined H_i and $\bar{H}_i$, we define:

$$I_k = \{i\in\{1,\ldots,n\} \mid k\in H_i\} \text{ and}$$

$$\bar{I}_k = \{1,\ldots,n\}\setminus I_k.$$

Then (P) is equivalent to:

$$\text{Minimize} \sum_{i=1}^{n} f_i(x_i)$$

$$\left.\begin{array}{l} g_{ki}(x_i) - y_{ki} \le 0, \; i\in I_k \\ \sum_{i\in I_k} y_{ki} + \sum_{i\in \bar{I}_k} g_{ki}(x_i) \le b_k \end{array}\right\} \quad k=1,\ldots,p$$

$$x_i \in \Omega_i, \; i=1,\ldots,n$$

Then, if we dualize only the p constraints with right-hand side b_k in the above program, we obtain a "lagrangian" function in x, y and u($u \ge 0$):

$$L(x,y,u) = \sum_{i=1}^{n} f_i(x_i) + \sum_{k=1}^{p} u_k \Big(\sum_{i\in I_k} y_{ki} + \sum_{i\in \bar{I}_k} g_{ki}(x_i) - b_k \Big)$$

$$= \sum_{i=1}^{n} \Big(f_i(x_i) + \sum_{k\in \bar{H}_i} u_k \, g_{ki}(x_i) + \sum_{k\in H_i} u_k \, y_{ki} \Big) - \sum_{k=1}^{p} u_k b_k$$

For fixed u and y, we obtain the decomposition in the n subproblems (SP_i) with $u_{ik}=u_k, \forall i$ and $k\varepsilon\bar{H}_i$. Supposed that u and y are such that $u\geq 0$ and each subproblem (SP_i) has a finite optimal solution with value $v_i(y,u)$, we define the associated bifunction $w(y,u)$:

$$w(y,u)=\sum_{i=1}^{n} v_i(y,u)+\sum_{k=1}^{p}\Big(\sum_{i\varepsilon I_k} u_k\, y_{ki} - u_k\, b_k\Big)$$

We show now that (y^*,u^*) is a saddle - point for w:

Theorem 2: with the same hypotheses as before on (x^*,u^*) and supposed that $H_i\neq\emptyset$ for at least one i; then (y^*,u^*), where $y^*_{ki} = g_{ki}(x_i^*)$ for $k\varepsilon H_i$ is a saddle - point of the function w on: y such that exist $x_i\varepsilon\Omega_i$ with $g_{ki}(x_i)\leq y_{ki}$ for each $k\varepsilon H_i$ and $u\varepsilon R^p$, $u\geq 0$.

Proof: From the construction of SP_i, we observe that, if $H_i\neq\emptyset$ and y_{ki} is such that (SP_i) has a solution, then v_i is convex w.r.t. each y_{ki} and concave w.r.t. u. We have too, if $X_i(y,u)$ is the optimal solution set of (SP_i), and $\Pi_i(y,u)$ is the set of optimal dual multipliers:

$$\frac{\partial v_i}{\partial y_{ki}} = \{-\pi_{ki},\ \pi_{ki}\ \varepsilon\ \Pi_i(y,u)\}$$

$$\frac{\partial v_i}{\partial u_k} = \{g_{ki}(x_i),\ x_i\varepsilon\ X_i(y,u)\}$$

The subdifferential of the convex-concave function w follows immediately:

$$\frac{\partial w}{\partial y_{ki}}(y,u) = \{u_k-\pi_{ki},\pi_{ki}\varepsilon\ \Pi_i(y,u)\}$$

$$\frac{\partial w}{\partial u_k}(y,u) = \Big\{\sum_{i\varepsilon I_k} y_{ki}+\sum_{i\varepsilon\bar{I}_k} g_{ki}(x_i)-b_k, x_i\varepsilon X_i(y,u)\Big\}$$

From theorem 1, (x^*,u^*), solve $(SP)_i$, $i=1,..,n$. This means that $(0,0)\ \varepsilon\ \partial w(y^*,u^*)$ (reducing the dual space to the binding constraints). Hence, (y^*,u^*) is a saddle - point for w.

In order to build completely decentralized planning procedures, we are interested in the differentiability of the convex-concave function w. This is the subject of the next section.

4. A completely decentralized procedure

It is well - known that the pure primal or dual decomposition methods lead both to partially decentralized procedure in the linear case. This is because the convex-concave function w is not differentiable at the saddle - point, or,in other words, that the dual or primal solutions of the respective subproblems are not all unique. We show here that there exist a mixed strategy which leads to complete decentralization without requiring the strict convexity of the functions.

We suppose then that (x^*,u^*) is a unique optimal primal-dual pair of solutions for (P). The optimality conditions for (x^*,u^*) are:

$$\nabla f(x^*) + \sum_{k\in I} u_k^* \nabla g_k(x^*) \varepsilon - \Omega^{*p}$$

$$u_k^*=0,\ k \notin I,\ u_k^* \geq 0, \forall k$$

$$g_k^*(x^*) \leq b_k,\ k \in I$$

$$x^* \in \Omega \qquad \text{where } I=\{k \mid g_k(x^*) = b_k\}$$

A sufficient condition to obtain a unique primal-dual point of optimal solutions in each sub-problem is to decompose these above conditions in n regular disjoint subsystems of equations, where disjoint means that each variable x_i appears in exactly one subsystem. This is done in two steps:

i) Define a partition of the set I of the binding constraints in n disjoint subsets $H_i, i=1,\ldots,n$.
Thus, $H_i \cap H_j = \emptyset\ \forall i \neq j$ and $\bigcup_{i=1}^{n} H_i = I$

ii) Decompose the global conditions in the following way:

For i=1,..,n, $(x_i^*, u_k^*, k\epsilon H_i)$ are unique solution of:

$$\nabla f_i(x_i) + \sum_{k\epsilon H_i} u_k^* \nabla g_{ki}(x_i) + \sum_{k\epsilon H_i} u_k \nabla g_{ki}(x_i) \epsilon - \Omega_i^{*p}$$

$$u_k \geq 0, \quad k\epsilon H_i$$

$$g_{ki}(x) = y_{ki}^*, \quad k\epsilon H_i \ , \ x_i \ \epsilon \ \Omega_i$$

We have shown (Mahey, 1986) that in the linear case, there always exists a partition leading to regular subsystems of equations which guarantees the unicity of the solution of each subproblem. Resuming the proof, it is shown that we must allocate the exact number of global binding constraints to the divisions to reconstruct the global optimal non-degenerate basis from the optimal basis of the subproblems. In other words, if B is the jacobian of the binding constraints, B is a non singular block-angular square matrix which we decompose in the following way:

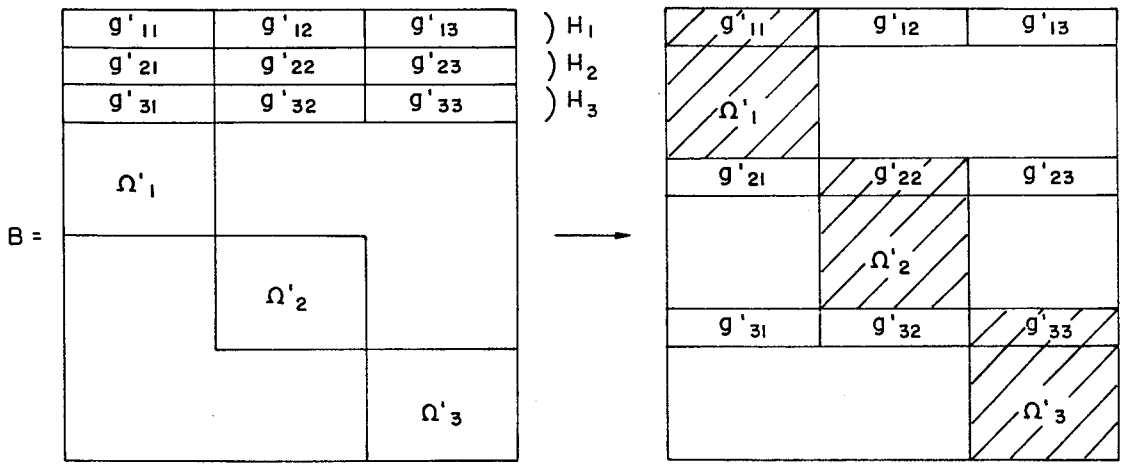


FIG.1- DECOMPOSITION WITH NON SINGULAR DIAGONAL BLOCKS

g'_{ki} are the blocks of the jacobian relative to the binding resource constraints g_k and Ω'_i is the block relative to the local constraints in Ω_i. The blocks on the diagonal of the matrix on the right are then nonsingular for some partition and represent the gradients of the constraints in the ith sub-problem.

These ideas characterize a completely decentralized planning procedure where each resource is allocated to exactly one division and each division receives price allocations for the resources that have not been allocated to it. The primal and dual allocations may be updated in various ways. The more attractive seems to be to exploit the special form of the subdifferential of the convex-concave function w, applying a fixed point algorithm to find u* and y*. This has been done in Mahey (1986) and the result is a one - level procedure where each subproblem receives at iteration t the following allocations:

$$\text{for} \quad k \notin H_i : u_k = \pi_{kj}^{(t-1)} \quad \text{where } I_k=\{j\}$$

$$\text{for} \quad k \in H_i : y_{ki} = b_k - \sum_{j \neq i} g_{ki} \, (x_j^{(t-1)})$$

The solution of (SP_i) is then $x_i^{(t)}$ and $\pi_{ki}^{(t)}$, $k \in H_i$. This procedure, when it converges, may work as well in a sequential manner as in the Gauss-Seidel iterative method.

5. Conclusion

We have shown how a mixed strategy based on a partition of the common resources could lead to a complete decentralization of the firm. The kind of one-level decomposition suggested by this approach has been analyzed by Cohen in the linear-quadratic case (Cohen, 1980). It seems that the allocation of constraints to the subproblems compensates the lack of strict convexity which would be sufficient to yield the complete decentralization. Strict convexity may be forced in the subproblems as in the methods proposed by Jennergren (1979) and Spingarn (1985) and it should be interesting to approximate both approaches.

6. Appendix:

An illustrative example

We illustrate the completely decentralized strategy on a simple academic example found in Lasdon (1970) which we solve by a pure dual decomposition scheme (DUAL), a pure primal decomposition scheme (PRIMAL) and the mixed strategy we have just described (MIXED):

$$
\begin{array}{rll}
\text{Maximize} & \xi_1 + \xi_2 + 2\xi_3 + \xi_4 & \\
(P) & \left|\begin{array}{ll}
\xi_1 + 2\xi_2 + 2\xi_3 + \xi_4 & \leq 40 \\
\xi_1 + 3\xi_2 & \leq 30 \\
2\xi_1 + \xi_2 & \leq 20 \\
\xi_3 & \leq 10 \\
\xi_4 & \leq 10 \\
\xi_3 + \xi_4 & \leq 15
\end{array}\right. & \\
& \xi_1, \xi_2, \xi_3, \xi_4 \geq 0 &
\end{array}
$$

Decomposition: $x_i = (\xi_1 \ \xi_2)^T \quad x_2 = (\xi_3 \ \xi_4)^T$

$n = 2$, $p=1$, $x_1^* = (25/3 \ 10/3)^T$ $x_2^* = (10 \ 5)^T$, $u^* = 1/3$

1) DUAL

The problem decomposes in two subproblems for a given price allocation $u \geq 0$:

$$
\begin{array}{rll}
\text{Max} & \xi_1 + \xi_2 + u(\xi_1+2\xi_2) & \\
(SP_1) & \left|\begin{array}{ll}
\xi_1 + 3\xi_2 & \leq 30 \\
2\xi_1 + \xi_2 & \leq 20
\end{array}\right. & \\
& \xi_1 \geq 0, \xi_2 \geq 0 &
\end{array}
\qquad
\begin{array}{rll}
\text{Max} & 2\xi_3 + \xi_4 + u(2\xi_3+\xi_4) & \\
(SP_2) & \left|\begin{array}{ll}
\xi_3 & \leq 10 \\
\xi_4 & \leq 10 \\
\xi_3 + \xi_4 & \leq 15
\end{array}\right. & \\
& \xi_1 \geq 0, \xi_2 \geq 0 &
\end{array}
$$

For the optimal dual solution $u^* = 1/3$, these subproblems give the following answers:

$$(SP_1) : X_1 = \text{conv} \left\{ \begin{pmatrix} 6 \\ 8 \end{pmatrix} , \begin{pmatrix} 10 \\ 0 \end{pmatrix} \right\}$$

$$(SP_2) : X_2 = \left\{ \begin{pmatrix} 10 \\ 5 \end{pmatrix} \right\}$$

Then $x_1^* \in X_1$, but the division 1 is unable to compute x_1^*. The second division computes x_2^*.

2) PRIMAL

$$(SP_1) \left| \begin{array}{rl} \text{Max} & \xi_1 + \xi_2 \\ & \xi_1 + 2\xi_2 \leq y_1 \\ & \xi_1 + 2\xi_2 \leq 30 \\ & 2\xi_1 + \xi_2 \leq 20 \\ & \xi_1 \geq, \xi_2 \geq 0 \end{array} \right. \qquad (SP_2) \left| \begin{array}{rl} \text{Max} & 2\xi_3 + \xi_4 \\ & 2\xi_3 + \xi_4 \leq y_2 \\ & \xi_3 \leq 10 \\ & \xi_4 \leq 10 \\ & \xi_3 + \xi_4 \leq 15 \\ & \xi_3 \geq 0, \xi_4 \geq 0 \end{array} \right.$$

For the optimal resource allocation $y^* = (15\ 25)^T$, each subproblem has a unique primal solution:

$x_1 = x_1^* = (25/3\ 10/3)^T$ and $x_2 = x_2^* = (10\ 5)^T$ but the dual solution of (SP_2) is not unique:

$$\Pi_1 = \{1/3\} \qquad \Pi_2 = \text{conv}\ \{0,1\}$$

Then the subproblems cannot decide upon the optimality which is $\pi_1 \in \Pi_2$ and $\pi_2 \in \Pi_2$ such that $\pi_1 = \pi_2$.

3) MIXED

We have seen before that the resource constraint must be allocated to the first subproblem to avoid the dual degeneracy observed with DUAL and must not be allocated to the second subproblem to avoid the primal degeneracy observed with PRIMAL.

Hence, the mixed allocation must be:

$$(SP_1)\begin{cases}\text{Max } \xi_1 + \xi_2 \\ \xi_1 + \xi_2 \le y_1 \\ \xi_1 + 3\xi_2 \le 30 \\ 2\xi_2 + \xi_2 \le 20 \\ \xi_1 \ge 0, \xi_2 \ge 0\end{cases} \qquad (SP_2)\begin{cases}\text{Max } 2\xi_3 + \xi_4 + u(2\xi_3 + \xi_4) \\ \xi_3 \le 10 \\ \xi_4 \le 10 \\ \xi_3 + \xi_4 \le 15 \\ \xi_3 \ge 0, \xi_4 \ge 0\end{cases}$$

for $y_1 = (y_1^*) = 15$ and $u = u^* = 1/3$, each subproblem has unique primal and dual optimal solutions, (x_1^*, π_1^*) and (x_2^*).

Finally the completely decentralized procedure to solve (P) is:

0. Initialize y_1
1. Solve SP_1. The solution is $\bar{x}_1$ and $\bar{\pi}_1$
2. Solve SP_2. for $u = \bar{\pi}_1$. The solution is $\bar{x}_2 = (\bar{\xi}_3\ \bar{\xi}_4)$
3. Solve SP_1 for $y_1 = 40 - 2\bar{\xi}_3 - \bar{\xi}_4$. Return to 2.

The fixed point (y_1^*, u^*) is obtained in one cycle of iterations in this case. Observe that no master program or central agency is needed and that the subproblems exchange their indices sequentially (Gauss-Seidel type algorithm).

References:

Arrow K.J. and L. Hurwicz, "Decentralization and computation in resource allocation", Essays in Economics and Econometrics, R.W. Pfouts,eds.,University of North Carolina Press (1960)

Cohen G., "Auxiliary problem principle and decomposition of optimization problems", JOTA, 32,3 (1980) pp. 277-305.

Dantzig G.B. and P. Wolfe, "Decomposition algorithm for linear programs", Econometrica, 29,4 (1960) pp. 767-778.

Jennergren P., "A price-schedules decomposition algorithm for linear programming problems", Econometrica, 41 (1973) pp. 965-980.

Kornai J. and J. Liptâk, "Two-level planning", Econometrica,33 (1965) pp. 141-169.

Lasdon L.S., Optimization Theory for Large Systems, Mac Millan, 1970.

Mahey P., "Méthodes de décomposition et décentralisation en programmation linéaire",RAIRO - Recherche Opérationnelle, 20,4 (1986) pp. 287-306.

Maier S.F. and J.H. Vanderweide, "Capital budgeting in the decentralized firm", Man. Sci., 23,4 (1976) pp. 433-443.

Malinvaud E.,"Decentralized procedures for planning", Activity Analysis in the Theory of Economic Growth, E. Malinvaud and M. Bacharach, eds., Mac Millan, 1967.

Obel B., "A note on mixed procedures for decomposing linear programming problems", Math. Oper. Forsch. Statist. Ser. Optimization, 9,4 (1978) pp. 537-544.

Sengupta J.K. and G.W. Gruver, "On the two-level planning procedure under a Dantzig-Wolfe decomposition", Int. J.Sys. Sci. 5, 9 (1974) pp. 857-875.

Spingarn J.E., "Applications of the method of partial inverse to convex programming: Decomposition", Math. Prog., 32,2 (1985) pp. 199-223.

Tenkate A., "Decomposition of linear programs by direct distribution", Econometrica, 40,5 (1972) pp. 883-898.

CHAPITRE 5

DECOMPOSITION ET METHODES PROXIMALES

On aborde ici la troisième ligne de recherche motivée par la construction de méthodes de décomposition décentralisées. Il s'agit maintenant de garantir l'unicité des solutions des sous-problèmes par des techniques de régularisation. La découverte des travaux de J. Spingarn en 1984 et le rapprochement avec les méthodes de Lagrangien Augmenté qui ouvrent la voie au domaine du non convexe nous ont poussé vers l'étude des techniques de régularisation proximales. La méthode du Point Proximal étudiée en détail par Rockafellar (1976) consiste à remplacer une fonction f convexe par sa régularisée f_c définie comme suit :

$f_c(x) = \text{Inf} \{ f(z) + (2c)^{-1}||z-x||^2 \}$, où c est un paramètre positif

L'infimum est toujours atteint de manière unique en un point appelé par Moreau (1965) $\text{prox}_{cf}(x)$. La régularisée f_c est différentiable, elle sous-estime f en tout point sauf au minimum, quand il existe, où les deux fonctions coincident. Spingarn a été le premier à adapter cet algorithme au cas sous contraintes. Il a proposé la méthode de l'Inverse Partielle dont l'attrait principal réside dans l'exploitation des informations primales et duales à chaque pas de l'algorithme (1983).

[14] "*Decomposition techniques in convex programming*", en collaboration avec Nguyen V. Hien, soumis à publication, 1990.

Ce travail fournit un cadre d'étude des méthodes de décomposition pour la programmation convexe. On y montre que la plupart des approches par décomposition reviennent à représenter le couplage entre les différents sous-systèmes par un sous-espace construit dans un espace produit approprié. Par exemple, le problème de rechercher un point dans l'intersection de p ensembles convexes fermés peut s'écrire :

$$\text{Minimiser} \sum_{i=1}^{p} f_i(x)$$

où chaque f_i est la fonction indicatrice d'un des ensembles convexes.

Ce problème est lui-même équivalent au problème suivant :

$$\text{Minimiser} \sum_{i=1}^{p} f_i(x_i)$$

$$x=(x_1,\ldots,x_p) \in A = \{ x_1=\ldots=x_p \}$$

A est bien un sous-espace de l'espace produit formé par p copies de l'espace d'origine.

Le modèle obtenu est favorable à l'application de la méthode de l'inverse partielle ou d'une méthode plus simple qui consiste à coupler une itération proximale sur la fonction avec une projection sur le sous-espace A. Cette méthode conduit à une forme séparable de la méthode du Lagrangien Augmenté à rapprocher de plusieurs algorithmes proposés dans la littérature, parmi lesquels ceux de Pierra (1984), Golstein (1985) et Bertsekas et Tsitsiklis (1989).

La convergence de la méthode de projection proximale a été démontrée dans le cadre de la régularisation de la somme de deux opérateurs maximaux monotones :

[15] "*Partial regularization of the sum of two maximal monotone operators* ", en collaboration avec Pham Dinh T., soumis à publication, 1990.

Cet article analyse la convergence d'un algorithme composé de deux étapes qui recherche un zéro de la somme de deux opérateurs maximaux monotones A et B : on détermine en premier lieu un zéro de A_c+B, où A_c est la régularisation de A (i.e. $A_c=c^{-1}(I-(I-cA)^{-1})$). Puis, la solution du problème est obtenue en faisant tendre c vers 0. La démonstration généralise des résultats obtenus par Brézis (1973) dans le cas fortement monotone.

Une autre voie de recherche liée aux méthodes proximales est l'exploration du cas non convexe. Spingarn (1981) a posé la première pierre en étendant la méthode proximale aux opérateurs hypomonotones, c.a.d. à la minimisation de fonctions non convexes mais convexes à une forme quadratique près. L'objectif actuel est d'étendre ces idées au cas de la minimisation de fonctions dc (différences de fonctions convexes) qui couvrent presque toutes les fonctions continues. L'idée de base est de chercher à exploiter la double structure convexe de ces fonctions. C'est ce qui a constitué le thème du travail de Doctorat de R. B. de Sampaio qui doit soutenir sa thèse sous notre direction en août 1990. Ce travail a permis d'établir d'importants résultats sur la régularisation des fonctions dc. Celle-ci consiste à remplacer la fonction par la différence des régularisées proximales, différentiables et de gradients lipschitz continus. Une deuxième partie de ce travail a été effectuée lors de deux séjours de Sampaio à Grenoble où un nouvel algorithme de type proximal pour les fonctions dc a été mis au point :

[16] "*Une méthode proximale pour la minimisation des fonctions dc* ", en collaboration avec R.B. de Sampaio, soumis à publication, 1990.

Une méthode pour la recherche d'un point critique (minimum local) d'une fonction dc est proposée dans cet article. La méthode est basée sur les propriétés de convexité des deux fonctions en présence. C'est une

méthode de descente dont les performances peuvent être modulées par un paramètre comme dans l'algorithme du point proximal.

Decomposition Techniques in Convex Programming

P. Mahey

Laboratoire ARTEMIS

IMAG, BP 53X, F-38041 Grenoble, France

Nguyen Van Hien

Département de Mathématique

Facultés Universitaires N.D. de la Paix, B–5000 Namur, Belgique

Abstract

We propose a general algorithm for the minimization of a convex function on a subspace. It combines a proximal step with a projection on the subspace and the proximal parameter must tend to zero to get the convergence. This algorithm is then applied to large-scale convex programs yielding a family of decomposition methods. We distinguish the cases of variable splitting, constraint splitting and block splitting. The common feature of the subproblems generated by these algorithms is the presence of a strictly convex quadratic proximal term in their objective function. In the particular case where the local functions are pieces of a dual functional associated to coupling constraints, this approach can lead to a separable version of the Augmented Lagrangian Method.

1 Introduction

The main characteristic of decomposition algorithms in mathematical programming is the splitting of a large-scale problem into a set of reduced size subproblems which may be solved either in parallel or in sequence. Very often, the structure of the system induces a splitting of the variable set in disjoint subsystems. The efficiency of the decomposition methods depends then not only on the specific properties of the functions as convexity, differentiability or separability but mostly on the degree of the coupling between these subsystems. Indeed, when the number of coupling constraints is large, we may need to split the constraint set too.

In this paper, we analyze both situations separately and present then a mixed decomposition method for general large-scale convex programs where both splitting are present.

Besides the motivation of reducing the size of the problems appears the possibility of decentralizing the optimal decisions among the local subproblems as in an ideal hierarchical organization. It is well-known ([1]) that most classical approaches perform only a partial decentralization and need a heavy coordination upper level to build a solution from the local proposals. This is due to the lack of unicity in the subproblems solutions which in turn is a direct consequence of the nonsmoothness of the coordination function. It is then natural to introduce regularizing terms in the decentralized process to cope with both nonuniqueness and nonsmoothness as in the Proximal Point Algorithm [21].

Our approach consists in solving regularized subproblems and projecting their solutions on an appropriate subspace which represents the coupling between the subsystems in a product space. The projection step can be viewed as a proximal iteration too (indeed, it is the proximal operation associated to the indicator function of the corresponding set). However, the composition of two proximal operators is not always a proximal operator itself and we need here to reconsider the whole problem of convergence.

The convergence of this Projected Proximal Algorithm has been extensively studied in a paper by Mahey and Pham Dinh [14] who have considered the general case of the sum of two maximal monotone operators. It appears that the type of convergence we get is similar to the one commonly used in penalty and barrier methods for constrained optimization. In section 2, the Projected Proximal Algorithm is presented in three distinct forms : the original one to solve the minimization problem on a subspace, the dual form where the algorithm is applied to the Fenchel dual of the former problem and the saddle-point form where both primal and dual mutually orthogonal subspaces are included.

This general method is then applied to three different situations of variables splitting, constraints splitting and block splitting. In the first case, a separable Augmented Lagrangian Method is obtained and we compare it to the multiplier approach given in Bertsekas and Tsitsikis [2] (see also the references cited therein) and the Partial Inverse Method of Spingarn [24]. In the second case, we obtain some algorithms which are close to the ones proposed by Pierra [19], Han and Lou [11], Mouallif et al. [18] and Spingarn [23]. Finally, in the third case, a completely symmetric setting of a block decomposition is given, which is very close to the one proposed by Gol'stein [10]. In this latter model, we build a product space from the copies (one for each block) or four kinds of variables : the original primal and dual variables of the problem and their perturbational counterparts.

2 A Projected Proximal Algorithm

In this section we are interested in solving the following constrained problem :

(P) $$\text{Minimize } F(x) \text{ subject to } x \in A$$

where F is a proper sci convex function from $\mathbb{R}^n$ into $\mathbb{R} \cup \{+\infty\}$ and A is a linear subspace of

$\mathbb{R}^n$.

We propose the following algorithm in view of solving (P) :

Algorithm A 1 *At iteration k , let x^k be a point in A and let $x^{k+1/2}$ be the unique minimizer of :*

$$\text{Minimize } F(x) + \frac{1}{2\lambda} \|x - x^k\|^2$$

where λ is some positive number.
Then, let x^{k+1} be the projection of $x^{k+1/2}$ on A .

We show first that the sequence $\{x^k\}$ generated by A1 converges to a point x_λ . Then, the point x_λ converges to a solution of (P) when $\lambda \to 0$. We shall give here only some elements of the proof which can be found in Mahey and Pham Dinh [14].

Before stating the general convergence result, we observe that x^{k+1} may be written as :

$$x^{k+1} = \text{Proj}_A \circ \text{Prox}_{\lambda F}(x^k) \tag{1}$$

where Proj_A is the projection operator on A and $\text{Prox}_{\lambda F}$ is the 'prox' operator defined by Moreau [17], i.e. $\text{Prox}_{\lambda F} = (I + \lambda \partial F)^{-1}$) where I is the identity and ∂F is the subdifferential of F (cf [20]) . It is well-known that both operators are non expansive.

Let T be the composed operator $T = \text{Proj}_A \circ \text{Prox}_{\lambda F}$. Then T is a nonexpansive and, assuming that it has a fixed point, the sequence $\{x^k\}$ defined by (1) and $x^0 \in A$ converges to that fixed point (see Browder and Petryshin [4] for the proof).

Now let F_λ be the Moreau-Yosida approximate of F , i.e. :

$$F_\lambda(x) = \inf\{F(z) + \frac{1}{2\lambda} \|x - z\|^2\} .$$

It is well-known that F_λ is convex and differentiable with gradient $\frac{1}{\lambda}(I - \text{Prox}_{\lambda F})$.

Theorem 1 *Any fixed point of $T = Proj_A \circ Prox_{\lambda F}$ is a solution of the regularized problem*

$$(\text{P}_\lambda) \qquad \text{Minimize } F_\lambda(x) \text{ on } x \in A$$

and conversely.

Proof : Let us write first the necessary and sufficient optimality conditions for (P_λ) :

x_λ solves (P_λ) , if and only if :
$\frac{1}{\lambda}(x_\lambda - \text{Prox}_{\lambda F}(x_\lambda)) \in A^\perp$
where $A^\perp$ is the orthogonal subspace to A .

To say that some vector is in $A^{\perp}$ is equivalent to say that its projection on A is zero. But, as $x_\lambda \in A$, we obtain :

$$x_\lambda = \mathrm{Proj}_A \circ \mathrm{Prox}_{\lambda F}(x_\lambda) .$$

which means that x_λ is a fixed point of T . ∎

When $\lambda \to 0$, we analyze the behaviour of the two sequences : $\{x_\lambda\}$ and $\{y_\lambda\}$, where y_λ is defined by $y_\lambda = \frac{1}{\lambda}(x_\lambda - \mathrm{Prox}_{\lambda F}(x_\lambda))$. We assume that these sequences are well-defined for any λ . It is shown in [14] that, if (P) has a solution, then the sequence $\{y_\lambda\}$ is bounded.

We can now assert the main result :

Theorem 2 *Assuming that (P) has a solution and that the sequence $\{x_\lambda\}$ has a limit point x^* when $\lambda \to 0$, then x^* solves (P) .*
Moreover, if (P) has a unique solution x^* and the sequence $\{x_\lambda\}$ is bounded, then the whole sequence converges to x^* and $\{y_\lambda\}$ converges to y^* , which is the minimum norm element in $\partial F(x^*) \cap A^{\perp}$.

Proof : See [14], Th.2, with $(I + \lambda A)^{-1} = \mathrm{Proj}_A$ and $B = \partial F$ which implies that $(I + \lambda B)^{-1} = \mathrm{Prox}_{\lambda F}$. We observe that the proof remains valid if we substitute the subspace A by any closed convex set . ∎

Remarks :

1. The necessity to reduce the parameter λ to zero can be a serious drawback to the numerical performance of the method. On the other hand, the procedure is quite simple to implement and robust. In relation to this observation, note that the fact that the parameter remains constant as in Spingarn's Partial Inverse method ([24]) is also a drawback from the computational point of view. The problem of the parameter setting to which the Proximal Point algorithm is very sensitive remains an open question in the constrained case .

2. We could be tempted to substitute λ by λ_k in (1) with $\lambda_k \to 0$ to avoid the two-stages convergence (we mean here that we must solve (P_λ) completely before reducing λ and solving it again). Unfortunately, it does not converge to a solution anymore except in the ergodic sense, i.e. very slowly (see [3] for instance).

Let us now consider the Fenchel dual of (P) :

$$\text{(D)} \qquad \inf_{y \in B} F^*(-y)$$

where F^* is the conjugate function of F and $B = A^{\perp}$. (D) has then the same form as (P) and we can apply Algorithm A1 :

Let $y^k \in B$ and $y^{k+1/2}$ be the unique minimizer of :

$$\text{Minimize } F^*(-y) + \frac{1}{2\lambda} \| - y + y^k\|^2$$
$$\text{and } y^{k+1} = \text{Proj}_B y^{k+1/2} .$$

The proximal minimization above can be expressed directly with the primal function noting that $y^{k+1/2}$ must solve :

$$0 \in \lambda \partial F^*(-y^{k+1/2}) - y^{k+1/2} + y^k$$

or equivalently (see [20]) :

$$-y^{k+1/2} \in \partial F(\lambda^{-1}(y^{k+1/2} - y^k)) .$$

Then $y^{k+1/2}$ satisfies the following optimality condition :

$$0 \in y^{k+1/2} + \partial F(\lambda^{-1}(y^{k+1/2} - y^k)) .$$

This leads to the following dual version of Algorithm A1 which we denote by A2 :

Algorithm A 2 *At iteration k , let y^k belong to B and $y^{k+1/2}$ be the unique minimizer of the following problem :*

$$\textit{Minimize } F(\lambda^{-1}(y - y^k)) + \frac{1}{2\lambda} \|y\|^2 ,$$
$$y^{k+1} = \textit{Proj}_B y^{k+1/2} .$$

Further on, a primal-dual version of the basic algorithm can be designed as well. Indeed, (P) is equivalent to solve the following saddle-point problem :

$$\text{(S)} \qquad \sup_{y \in B} \inf_{x \in A} F(x) - \langle x, y \rangle .$$

Applying the Projected Proximal Algorithm to (S) in the primal-dual product space leads to the following iteration :

Let (x^k, y^k) belong to $A \times B$ and $(x^{k+1/2}, y^{k+1/2})$ be the unique saddle-point of :

$$\max_y \min_x F(x) - \langle x, y \rangle + \frac{1}{2\lambda} \|x - x^k\|^2 - \frac{1}{2\lambda} \|y - y^k\|^2 .$$
$$\text{Then, } (x^{k+1}, y^{k+1}) = \text{Proj}_{A \times B}(x^{k+1/2}, y^{k+1/2}) .$$

By solving the outer problem w.r.t. y in the above saddle-point problem, we get :

$$0 = -x^{k+1/2} - \frac{1}{\lambda}(y^{k+1/2} - y^k)$$

and we can now eliminate y by substituting it in the inner minimization :

$$\min_x F(x) - \langle x, y^k - \lambda x \rangle + \frac{1}{2\lambda} \|x - x^k\|^2 - \frac{1}{2\lambda} \|\lambda x\|^2$$

Combining the two norms with the cross term and eliminating the constants, we obtain the following algorithm :

Algorithm A 3 *At iteration k , let (x^k, y^k) belong to $A \times B$ and $x^{k+1/2}$ be the unique minimizer of :*

$$\begin{aligned} &\textit{Minimize } F(x) + \frac{\gamma}{2\lambda} \|x - \frac{1}{\gamma}(x^k + \lambda y^k\|^2 \\ &\quad \textit{where } \gamma = 1 + \lambda^2 , \\ &\textit{Let } y^{k+1/2} = y^k - \lambda x^{k+1/2} \\ &\textit{Then, } (x^{k+1}, y^{k+1}) = Proj_{A\times B}(x^{k+1/2}, y^{k+1/2}) . \end{aligned}$$

In the next section, we consider optimization problems with a large number of variables where an underlying structure induces a decomposition in subproblems with a small number of variables. Algorithms A1 and A3 are then applied to the dual problem to yield some new version of the multiplier method. In section 4, algorithm A1 and A2 are applied to problems with many constraints (or nonseparable objective functions which are the sum of many simple functions). Finally, we consider the application of algorithm A3 to the more general situation where both the variables set and the constraints set are splitted. The other variants not mentioned in this paper are being implemented with the present ones to yield a complete numerical experience with this family of decomposition methods which should be published in a forthcoming paper.

3 Decomposition by splitting the variable set

3.1 No coupling constraints

Let us consider first a convex program on a finite product space $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p} = \mathbb{R}^n$. Let $x = (x_1, \cdots, x_p)$ with $x_j \in \mathbb{R}^{n_j}$. Then, let f_0 be a proper convex function on this product space with values in $\mathbb{R} \cup \{+\infty\}$ and let S_j , $j = 1, \cdots, p$, be closed convex subsets of $\mathbb{R}^{n_j}$ respectively.

$$\text{(P1)} \qquad \begin{aligned} &\text{Minimize } f_0(x) \\ &x_j \in S_j , \quad j = 1, \cdots, p . \end{aligned}$$

It is clear that, if f_0 is separable w.r.t. the product space, i.e., if there exist p convex functions f_{0j} defined on $\mathbb{R}^{n_j}$ such that :

$$f_0(x) = \sum_{j=1}^{p} f_{0j}(x_j) ,$$

then (P1) splits into the p independent subproblems :

$$\text{(SP1)} \qquad \begin{aligned} &\text{Minimize } f_{0j}(x_j) \\ &x_j \in S_j . \end{aligned}$$

If f_0 is not separable, then we can decompose it directly by a relaxation strategy, i.e., at iteration k , we solve sequentially for $j = 1, \cdots, p$:

$$\text{(SP2)} \qquad \begin{aligned} &\text{Minimize } f_0(x_j) + \tfrac{1}{2\lambda} \|x_j - x_j^k\|^2 \\ &x_j \in S_j \ , \end{aligned}$$

where

$$f_{0j}(x_j) = f_0(x_1^{k+1}, \cdots, x_{j-1}^{k+1}, x_j, x_{j+1}^k, \cdots, x_p^k) \text{ and } c > 0 \ .$$

The convergence is proved in Martinet [15] without any further assumptions on f_0 (see, too, Martinet and Auslender [16] for acceleration schemes using a relaxation parameter).

When f_0 is differentiable, a similar way to decompose is to substitute it by its first-order approximation around x^k , which is obviously separable. We obtain then a block-projected gradient method and the subproblem is the same as before with :

$$f_{0j}(x_j) = \langle \nabla f_0(x^k), x_j - x_j^k \rangle \ .$$

This idea has been extended by Cohen [5] who replaced the proximal term in (SP2) by any separable convex kernel, yielding what he called the *Auxiliary Problem Principle.*

3.2 Coupling contraints

Let us now introduce a 'small' number of coupling constraints and assume that all the functions are separable on the product space :

$$\text{(P2)} \qquad \begin{aligned} &\text{Minimize } \sum_{j=1}^{p} f_{0j}(x_j) \\ &\sum_j f_j(x_j) \leq 0 \ , \quad x_j \in S_j \ , \quad j = 1, \cdots, p \ . \end{aligned}$$

Here, each function f_j is a proper closed convex function from $\mathbb{R}^{n_j}$ into $\mathbb{R}^m$ and each S_j is again a closed convex set in $\mathbb{R}^{n_j}$.

As all the functions are convex, we can substitute (P2) by its Lagrangian dual formulation. Let $u \in \mathbb{R}^m$, $u \geq 0$, the dual variables associated to the coupling constraints and consider the classical dual problem :

$$\sup_{u \geq 0} \inf_{x_j \in S_j} \sum_{j=1}^{p} (f_{0j}(x_j) + \langle u, f_j(x_j) \rangle) \ .$$

By letting u fixed in the subproblems and updated on a second level, we could decompose this problem into p independent subproblems. This is the so-called Price-Coordination Principle which leads to classical two-level algorithms. To avoid the difficulties which generally appear with this approach, like non-smoothness, partial decentralization or typical slowness of the convergence,

we introduce new local variables $u^1, \cdots, u^p$, such that $u^j \geq 0$, $\forall\, j$. Then, the problem above is equivalent to solve :

$$\text{(P3)} \qquad \begin{aligned} &\max_{u^j \geq 0} \sum_j h_j(u^j) \\ &(u^1, \cdots, u^p) \in A = \{(u^1, \cdots, u^p) \in (I\!R^m)^p \mid u^1 = \cdots = u^p\} \ , \end{aligned}$$

where

$$h_j(u^j) = \inf_{x_j \in S_j} f_{0j}(x_j) + \langle u^j, f_j(x_j) \rangle \ .$$

We have in fact created p copies of the dual variables u and obtained the desired formulation in the product space $V = (I\!R^m)^p$.

If we apply now Algorithm A1 to (P3), we obtain the following iteration :

At iteration k, let $((u^1)^k, \cdots, (u^p)^k) \in A$, i.e. $(u^1)^k = \cdots = (u^p)^k$ and let $(u^j)^{k+1/2}$ be the unique minimizer of :
Maximize $h_j(u^j) - \frac{1}{2\lambda} \|u^j - (u^j)^k\|^2$
$u^j \geq 0$.
Then $(u^j)^{k+1} = \dfrac{1}{p} \sum_{j=1}^{p} (u_j)^{k+1/2}$.

A straightforward calculation as above (see Algorithm A3) shows that everything can be written in the primal space :

Algorithm A 11 *At iteration k, let $((u^1)^k, \cdots, (u^p)^k) \in A$ and let $(x_j)^k$ be the unique minimizer of the following subproblem :*

$$\text{(SP3)} \qquad \begin{aligned} &\textit{Minimize } f_{0j}(x_j) + \tfrac{1}{2\lambda} \phi_\lambda(f_j(x_j), (u^j)^k) \\ &x_j \in S_j \ , \end{aligned}$$

where $\phi_\lambda(a, b) = \sum_i [\max(0, \lambda a_i + b_i)]^2$
for any $a = (a_1, \cdots, a_m)$ *and* $b = (b_1, \cdots, b_m)$, *both in* $I\!R^m$.
Compute $(u_j)^{k+1/2} = \max\{0, (u^j)^k + \lambda f_j((x_j)^k)\}$.
Then

$$(u^j)^{k+1} = \frac{1}{p} \sum_{j=1}^{p} (u_j)^{k+1/2} \ .$$

Thus the algorithm can be interpreted as a separable version of the Augmented Lagrangian Method and could be compared to similar approaches which appear in Glowinski and Marocco [9], Cohen and Zhu [6] or in Tseng [25] (see also [2] and [8]). On the other hand, we can also establish the relationship with Spingarn's Partial Inverse method ([24]) by applying Algorithm

A3 to (P3). This is done by introducing the ressource-allocation vector $(v_1, \cdots, v_p)$. Indeed, (P3) is equivalent to :

$$\text{(P4)} \qquad \begin{array}{l} \max\limits_{\substack{(u^1,\cdots,u^p)\\ u^j \geq 0}} \min\limits_{(v_1,\cdots,v_p)} \sum\limits_j (h_j(u^j) - \langle u^j, v_j \rangle) \\ u^1 = \cdots = u^p \ , \\ v_1 + \cdots + v_1 \leq 0 \ . \end{array}$$

Observe that here the constraints in duality are polyhedral cones and no more subspaces. This is due to the presence of the nonnegativity constraints on the dual variables u which implies the following mutually polar cones :

$$\begin{aligned} A &= \{(u^1, \cdots, u^p) \mid u^j \geq 0;, \ u^1 = \cdots = u^p\} \ , \\ B &= \{(v_1, \cdots, v_p) \mid v_1 + \cdots + v_p \leq 0\} \ . \end{aligned}$$

As noted above, this substitution does not affect the convergence of the method nor its applicability as the projection steps on these cones are computationally very easy. The application of the projected-proximal algorithm leads now to the following algorithm :

Algorithm A 31 *At iteration k, let $((u^1)^k, \cdots, (u^p)^k) \in A$ and $((v_1)^k, \cdots, (v_p)^k) \in B$ and let $(x_j)^k$ be the unique minimizer of the following subproblem :*

$$\text{(SP4)} \qquad \begin{array}{l} \textit{Minimize } f_{0j}(x_j) + \frac{1}{2\lambda\gamma}\phi_\lambda(f_j(x_j) - v_j^k, (u^j)^k) \\ x_j \in S_j \ . \end{array}$$

where ϕ_λ is the same penalty function as before (see (SP3))
Compute $(u^j)^{k+1/2} = \frac{1}{\gamma}\max\{0, (u^j)^k + \lambda f_j((x_j)^k) - \lambda(v_j)^k\}$
and $(v_j)^{k+1/2} = (v_j)^k + \lambda(u^j)^{k+1/2}$.
Then, $((u^1)^{k+1}, \cdots, (u^p)^{k+1}) = Proj_A((u^1)^{k+1/2}, \cdots, (u^p)^{k+1/2})$
and $((v_1)^{k+1}, \cdots, (v_p)^{k+1}) = Proj_B((v_1)^{k+1/2}, \cdots, (v_p)^{k+1/2})$

We observe that the suproblems (SP4) correspond to applying the method of multipliers to the so-called primal resource-directive subproblems :

$$\begin{array}{l} \text{Minimize } f_{0j}(x_j) \\ f_j(x_j) \leq v_j \\ x_j \in S_j \end{array}$$

where v_j is a resource allocation vector for the substysem j such that $\sum v_j \leq 0$ (i.e. $(v_1, \cdots, v_p) \in B$) (see Lasdon [12] for example). This algorithm is quite close to the one given by Spingarn in [24]. Note also that Eckstein et al [7] have proved that Spingarn's Partial Inverse method is

a particular case of Douglas-Rachford splitting algorithm. In this latter algorithm, the central iteration (1) is substituted by

$$x^{k+1} = \mathrm{Proj}_A \circ (2\mathrm{Prox}_{\lambda F}(x^k) - x^k) + (x^k - \mathrm{Prox}_{\lambda F}(x^k)) .$$

4 Splitting of the constraint set

Lions and Teman [13] seem to have been the first to propose the splitting of the constraint set for the decomposition of large-scale variational problems.

Let $f_1, \cdots, f_q$ be proper closed convex functions from $\mathbb{R}^n$ into $\mathbb{R} \cup \{+\infty\}$. Let consider the following convex program :

$$\text{(P4)} \qquad \text{Minimize } \sum_{i=1}^{q} f_i(x) .$$

In most applications, f_i is the indicator function of a closed convex set C_i of $\mathbb{R}^n$. In some models, a smooth strongly convex functional f_0 is added to take into account an objective function on the intersection of the convex sets. Although it may lead to slightly different algorithms, we shall not at first distinguish this case from the general one.

Following Pierra [19], we substitute (P4) by an equivalent problem in the product space $(\mathbb{R}^n)^q$, i.e., we create q copies of the original variables x :

$$\text{(P5)} \qquad \begin{aligned} &\text{Minimize } \sum_{i=1}^{q} f_i(x^i) \\ &(x^1, \cdots, x^q) \in A = \{(x^1, \cdots, x^q) \in (\mathbb{R}^n)^q \mid x^1 = \cdots = x^q\} . \end{aligned}$$

Then we can apply Algorithm A1 to (P5) with the function $F(x^1, \cdots, x^q)$ given by

$$F(x^1, \cdots, x^q) = \sum_{i=1}^{q} f_i(x^i) .$$

Algorithm A 12 *At iteration k, let $((x^1)^k, \cdots, ((x^q)^k) \in A$ and $((x^i)^{k+1/2})$ be the unique minimizer of :*

$$\text{(SP5)} \qquad \textit{Minimize } f_i(x^i) + \frac{1}{2\lambda} \|x^i - (x^i)^k\|^2 .$$

Then,

$$(x^i)^{k+1} = \frac{1}{q} \sum_{l=1}^{q} (x^l)^{k+1/2} , \quad i = 1, \cdots, q .$$

The algorithm has been studied by Pierra in [19] and is intimately related to the classical projection method in the case where f_i is the indicator function of a closed convex set.

If we apply now Algorithm A2 to the Fenchel dual of (P5), we obtain the following algorithm :

Algorithm A 22 *At iteration k, let* $((y^1)^k, \cdots, ((y^q)^k) \in B$. *For each* i , *let* $((y^i)^{k+1/2})$ *be the unique minimizer of :*

(SP6) $$\text{Minimize } f_i(\lambda^{-1}(y^i - (y^i)^k) + \frac{1}{2\lambda} \|y^i\|^2 .$$

Then,

$$(y^i)^{k+1} = (y^i)^{k+1/2} - \frac{1}{q}\sum_{l=1}^{q}(y^l)^{k+1/2} , \quad i = 1, \cdots, q .$$

If we add a smooth strongly convex function f_0 to F , we don't need the projection step on the subspace any more and this yields a decomposition method proposed in Han and Lou [11] and in Mouallif et al. [18]. Indeed, the Fenchel dual of (P5) can be written :

(D5) $$\text{Minimize } f_0^*(y^1 + \cdots + y^q) + \sum f_i^*(-y^i) .$$

As f_0 is smooth and strongly convex, so is f_0^* and we can treat it by Cohen's Auxiliary Problem Principle avoiding the explicit use of subspace B . It results in the following subproblems closely related to (SP6) :

For each $i = 1, \cdots, q$, let $(y^i)^{k+1}$ be the unique minimizer of :

(SP7) $$\text{Minimize } f_i(\lambda^{-1}(y^i - (y^i)^k) + \nabla f_0^*((y^1)^k + \cdots + (y^q)^k)) + \frac{1}{2\lambda} \|y^i\|^2 .$$

Observe that in (SP7) each subproblem involves f_0. This is the main difference with the application of A22 which would have led to $q + 1$ subproblems and the projections of their respective solutions on the subspace $\{y^0 + \sum_{i=1}^q y^i = 0\}$. Moreover, in Han and Lou's algorithm, the parameter λ need not decrease to 0 .

5 Block splitting of large constrained convex programs

We consider now the general case of splitting of both sets, the variables set and the constraints set. We shall apply the same principles as before and obtain subproblems which are similar to those proposed by Gol'stein [10]. Following the latter and without any loss of generality, we assume that the objective function is linear. This representation will lead to nice symmetric dual formulations. Indeed, we can use the fact that any linear objective function $\langle c, x\rangle$ on X may be written in a separable form on the product space $X \times \cdots \times X$:

$$\langle c, x\rangle = \sum_i \langle y_i, x_i\rangle$$

where $\sum_i y_i = c$ and $(x_1, \cdots, x_p) \in X \times \cdots \times X$ such that $x_1 = \cdots = x_p$.

Let us first partition the variables space $\mathbb{R}^n$ into the product space $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p}$ and the constraint space $\mathbb{R}^m$ into the product space $\mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_q}$.

The primal variables are denoted by x_j , $j = 1, \cdots, p$ and the dual variables by u_i , $i = 1, \cdots, q$. As we are going to create copies of the original primal and dual variables for the purpose of

decomposition, we denote these copies by x_j^i , $i = 1,\cdots,q$ and u_i^j , $j = 1,\cdots,p$, respectively. Let us consider the following problem :

$$\text{(P6)} \qquad \begin{aligned} &\text{Minimize } \sum_{j=1}^{p}\langle c_j, x_j\rangle \\ &\sum_{j=1}^{p} f_{ij}(x_j) \le b_i\,, \quad i = 1,\cdots,q \\ &x_j \in S_j\,, \quad j = 1,\cdots,p \end{aligned}$$

where $c_j \in I\!R^{n_j}$, $j = 1,\cdots,p$ and $b_i \in I\!R^{m_i}$, $i = 1,\cdots,q$. Each function f_{ij} is supposed to be convex proper from $I\!R^{n_j}$ into $I\!R^{m_i} \cup \{+\infty\}$.

To decompose (P6) into $p \cdot q$ independent subproblems, we proceed as before, introducing q (resp. p) copies of the original primal (resp. dual) variables and the auxiliary variables v_i^j , y_j^i , for each i and j . Then we obtain the following equivalent problem :

$$\max_{\substack{\{u_i^j\}\ \{x_j^i\}\\ \{y_j^i\}\ \{v_i^j\}}} \min \sum_i \sum_j \langle y_j^i, x_j^i\rangle - \langle v_i^j, u_i^j\rangle + \langle f_{ij}(x_j^i), u_i^j\rangle$$

$$x_j^i \in S_j\,, u_i^j \ge 0 \quad i = 1,\cdots,q\,,\ j = 1,\cdots,p\,,$$

and subject to :

$$\text{(P7)} \qquad \begin{aligned} &x_j^1 = \cdots = x_j^q\,, \quad j = 1,\cdots,p\,, \\ &u_i^1 = \cdots = u_i^p\,, \quad i = 1,\cdots,q\,, \\ &\sum_{j=1}^{p} v_i^j = b_i\,, \quad i = 1,\cdots,q\,, \\ &\sum_{i=1}^{q} y_j^i = c_j\,, \quad j = 1,\cdots,p\,. \end{aligned}$$

We observe that the auxiliary variables $\{v_i^j\}$ and $\{y_j^i\}$ must lie in some affine subspaces which are orthogonal to the feasibility subspaces of the primal and the dual variables respectively. We may treat again these subspaces by adding proximal terms in the objective funciton and by successive projections onto these subspaces. The resulting algorithm is quite similar to one given by Gol'stein [10].

Algorithm A 33 *At iteration k, let $(\cdots,(x_j^i)^k,\cdots) \in A = \{x_j^1 = \cdots = x_j^q\ ,\ j = 1,\cdots,p\}$, $(\cdots,(u_i^j)^k,\cdots) \in B = \{u_i^1 = \cdots = u_i^p\ ,\ i = 1,\cdots,q\}$, $(\cdots,(v_i^j)^k,\cdots) \in C = \{\sum_{j=1}^{p} v_i^j = b_i\ ,\ i = 1,\cdots,q\}$ and $(\cdots,(y_j^i)^k,\cdots) \in D = \{\sum_{i=1}^{q} y_j^i = c_j\ ,\ j = 1,\cdots,p\}$. For each block (i,j), let $(x_j^i)^{k+1/2}$ be the unique minimizer of the following subproblem :*

$$\text{(SP8)} \qquad \begin{aligned} &\textit{Minimize } \langle (y_j^i)^k, x_j^i\rangle + \tfrac{1}{2\lambda\gamma}\phi_\lambda(f_{ij}(x_j^i) - (v_i^j)^k, (u_i^j)^k) + \tfrac{1}{2\lambda}\,\|x_j^i - (x_j^i)^k\|^2 + \tfrac{\lambda}{2}\|x_j^i\|^2 \\ &x_j^i \in S_j\,, \end{aligned}$$

where ϕ_λ is the same penalty function as before. We may observe that subproblem (SP8) is identical to the one proposed by Gol'stein unless the quadratic term $\frac{\lambda}{2}\|x_j^i\|^2$ which is a consequence here of the derivation w.r.t. y_i^j.

The projections on the respective subspaces are as follows :

- x^{k+1} *is obtained by projecting* $((x_1^1)^{k+1/2},\cdots,(x_p^q)^{k+1/2})$ *on the subspace* A .[1]
- u^{k+1} *is obtained by projecting* $((u_1^1)^{k+1/2},\cdots,(u_p^q)^{k+1/2})$ *on the subspace* B , *where* $(u_i^j)^{k+1/2} = \frac{1}{\lambda\gamma}\max\{0, \lambda(f_j((x_j^i)^{k+1/2}) - (v_i^j)^k) + (u_i^j)^k\}$.
- y^{k+1} *is obtained by projecting* $((y_1^1)^{k+1/2},\cdots,(y_p^q)^{k+1/2})$ *on the affine subspace* C , *where* $(y_j^i)^{k+1/2} = (y_j^i)^k + \lambda(x_j^i)^{k+1/2}$.
- v^{k+1} *is obtained by projecting* $((v_1^1)^{k+1/2},\cdots,(v_p^q)^{k+1/2})$ *on the affine subspace* D , *where* $(v_i^j)^{k+1/2} = (v_i^j)^k + \lambda(u_i^j)^{k+1/2}$.

As a final remark, we may introduce in every updating formula a relaxation parameter but the proof of convergence of the resulting generalized algorithm is omitted here and we refer to [7] for a similar approach.

References

[1] K. Arrow and L. Hurwicz, "Decentralization and computation in resource allocation", in *Essays in Economics and Econometrics*, R. Phouts ed., Chapell Hill, 1960.

[2] D.P. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation*, Prentice-Hall, 1989.

[3] H. Brézis and P.L. Lions, "Produits infinis de résolvantes", *Israel J. of Math.* 29, 4 (1978), pp. 329–345.

[4] F. Browder and W. Petryshin, "Construction of fixed point of nonlinear mappings in Hilbert spaces", *J. Math. Anal. and Appl.* 20 (1967), pp. 197–228.

[5] G. Cohen, "Auxiliary problem and decomposition of optimization problems", *J. Optimization Theory and Appl.* 32 (1980), pp. 277–305.

[6] G. Cohen and D.L. Zhu, "Decomposition coordination methods in large-scale optimization problems. The nondifferentiable case and the use of augmented lagrangians", in *Advances in Large-Scale Systems*, J.B. Cruz ed., Vol. I, JAI Press, 1983.

[7] J. Eckstein and D.P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators", Working Paper 90–033, Harvard Business School, to appear in *Mathematical Programming*, 1990.

[1] Observe that $(x_j)^{k+1}$ belongs to S_j, $\forall j$.

[8] M. Fortin and R. Glowinski, *Résolution Numérique de Problèmes aux Limites par des Méthodes de Lagrangien Augmenté*, Dunod, 1982.

[9] R. Glowinksi and A. Marocco, "Sur l'approximation par éléments finis d'ordre un et la résolution par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires", *RAIRO*, R-2, 9 (1975), pp. 41–76.

[10] E.G. Gol'stein, "The block method of convex programming", *Soviet Math. Dokl.* 33 (1986), pp. 584–587.

[11] S.P. Han and G. Lou, "A parallel algorithm for a class of convex programs", *SIAM J. Control and Optimization* 26 (1988), pp. 345–355.

[12] L.S. Lasdon, *Optimization Theory for Large Systems*, MacMillan, 1970.

[13] J.L. Lions and R. Temam, "Eclatement et décentralisation en calcul des variations", *Lecture Notes in Mathematics* 32 (1970), pp. 196–217.

[14] P. Mahey and Pham Dinh T., "Partial regularization of the sum of two maximal monotone operators", Working paper, ARTEMIS/IMAG, 1990.

[15] B. Martinet, "Minimisation d'une fonctionnelle dans un espace produit par une méthode de relaxation", *RIRO*, R-3 (1971), pp. 121–126.

[16] B. Martinet and A. Auslender, "Méthodes de décomposition pour la minimisation d'une fonction sur un espace produit", *SIAM J. Control* 12 (1974), pp. 635–642.

[17] J.J. Moreau, "Proximité et dualité dans un espace Hilbertien", *Bull. Soc. Math. de France* 93 (1975), pp. 273–299.

[18] K. Mouallif, Nguyen V.H. and J.J. Strodiot, "Decomposition based on nonsmooth optimization methods", Tech. Rep. Math., FUNDP, Namur, 1988.

[19] G. Pierra, "Decomposition through formalization in a product space", *Math. Progr.* 28 (1984), pp. 96–115.

[20] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[21] R.T. Rockafellar, "Monotone operators and the proximal point algorithm", *SIAM J. Control and Optim.* 14 (1976), pp. 877–898.

[22] R.T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm to convex programming", *Math. of Oper. Res.* 1 (1976), pp. 97–116.

[23] J.E. Spingarn, "Partial inverse of a montone operator", *Appl. Math. and Optim.* 10 (1983), pp. 247–265.

[24] J.E. Spingarn, "Applications of the method of partial inverse to convex programming : Decomposition", *Math. Progr.* 32 (1985), pp. 199–223.

[25] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities", Working Paper LIDS-P-1836, Laboratory for Information and Decision Sciences, MIT, 1988.

PARTIAL REGULARIZATION OF THE SUM OF TWO MAXIMAL MONOTONE OPERATORS

P. Mahey[1] and Pham Dinh Tao[2]
IMAG, BP53X, 38041 Grenoble

Abstract. To find a zero of the sum of two maximal monotone operators, we analyze a two-steps algorithm where the problem is first approximated by a regularized one and the regularization parameter is then reduced to converge to a solution of the original problem. We give a formal proof of the convergence which , in that case, is not ergodic. The main result is a generalization of one given by Brezis [2] who has considered operators of the form I+A+B. Additional insight on the underlying existence problems and on the kind of convergence we aim at are given with the hypothesis that one of the two operators is strongly monotone. A general scheme for the decomposition of large scale convex programs is then induced.

1 INTRODUCTION

We consider the following inclusion problem in a finite dimensional space X ($X=R^n$) :

Find $x \in X$ such that $0 \in (A+B)x$ **(P)**

where A and B are two maximal monotone operators.

We analyze here the convergence of some specific 'splitting' algorithms, i.e. such that separate steps on A and B are made to avoid the difficulties derived from the coupling between the two operators like in decomposition methods (other splitting algorithms are described in [8]).

Maximal monotone operators in Hilbert spaces have been extensively studied, mainly in the context of evolution equations, by Brezis [3] who, in particular, has given some conditions for the sum of two maximal monotone operators to be maximal monotone. Indeed, this fact happens when one of the operators is Lipschitzian. This result induces a strategy to solve (**P**) which consists in substituting one of the operators, say A , by a regularized approximation, for instance its Moreau- Yosida approximation A_λ, i.e. , for some $\lambda>0$:

[1] Laboratoire ARTEMIS
[2] Equipe Modélisation et Optimisation

$$A_\lambda = {}^1/_\lambda (I - (I + \lambda A)^{-1})$$

Then, the regularized problem is :

$$\text{Find } x_\lambda \in X \text{ such that } 0 \in (A_\lambda + B)\, x_\lambda \qquad (P_\lambda)$$

Properties of A_λ are well-known (see Moreau [12] and Brezis[3]). In particular, it tends in some way to the original operator A when $\lambda \downarrow 0$. We analyze here the convergence of the following iteration :

$$x_k^{t+1} = (I+\lambda_k B)^{-1} (I+\lambda_k A)^{-1} x_k^t \qquad (1)$$

where $\{\lambda_k\}$ is an a priori defined sequence of positive numbers.

It is shown in Lions[7] and Passty[13] that, if both subscripts k and t are incremented together, i. e. , if the iteration takes the form :

$$x_{k+1} = (I+\lambda_k B)^{-1} (I+\lambda_k A)^{-1} x_k \qquad (2)$$

with $\lambda_k \downarrow 0$ and $\Sigma \lambda_k = +\infty$, then the sequence of weighted averages

$$z_k = \sum_{i=1}^{k} \lambda_i x_i / \sum_{i=1}^{k} \lambda_i$$

converges to a solution of (**P**) if one exists. Our purpose is to show the convergence of a two-steps version of iteration process (1) where we iterate first on subscript t for a fixed k , then increment k and perform another cycle of iterations. We show in particular that the sequence generated by (1) for a fixed k converges to a solution of (P_λ) when it exists. The important thing is that the solution x_λ converges to a solution of (1) when $\lambda \downarrow 0$, avoiding ergodic convergence. In fact, the parameter λ acts like a penalty parameter in a penalty method for constrained programming, which means that we cannot set it to a too small value at first to avoid ill-conditioning but we may reduce it if some convergence criterion is not met. In fact, the iterative process (1) can be seen as a decomposed version of the Proximal Point Algorithm but the need of reducing the parameter λ to zero forbids to apply the classical results of convergence given by Rockafellar [15] in the case of a single operator.

As in all iterative methods where different subproblems are to be solved at each step, two important questions arise beside the necessity of convergence : the problem of existence of solutions of the successive subproblems which is not necessarily guaranteed by the existence of a solution of (**P**) and the computational simplicity of these subproblems

without which there is no interest to manipulate the original problem in that way. We shall see that these questions are linked to the choice of the operator to be regularized. Sufficient conditions for existence of solutions in the subproblems and for the convergence of the whole sequence are obtained when one of the operators is strongly monotone. In this case, it is natural to regularize the other one.

The interest for Proximal methods has increased recently motivated by the theoretical works of Rockafellar[15] and Spingarn[17]. The numerical behaviour in the applications, initially limited to variational inequalities as in the pioneering work of Martinet [10], have been analyzed in the context of Mathematical Programming (see [2], [6], [11], [19] for example). The positive aspects that come out of these experience are :

- The numerical stability.
- The ability of dealing with nonsmoothness.
- The nice effect of the regularization on the decentralization of subproblems in the decomposition of large-scale programs (see [1], [9] and [14]).

The negative aspects are mostly :

- The complexity of the proximal computation which limits the domain of applications.
- The slow rate of convergence.

This latter drawback is minimized by the constatation that, in some specific applications as the Fermat-Weber location problem ([11]) or the numerical solution of evolution equations for low level vision ([6]), the proximal algorithms seem to exhibit the best stability and efficiency.

A particular motivation we have in mind beside the general problem (**P**) is the case where $A=\partial f$, the subdifferential mapping of a convex lsc function f and B is the subdifferential mapping of the indicator function of a closed convex subset C of X, i.e. $B=N_C$, the normal cone to C. Then, under mild conditions, (**P**) is the optimality condition for the following convex program :

$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \end{aligned} \tag{3}$$

When C is an appropriate subspace of a product space, this formulation is attractive to decompose large-scale problems. The subspace represents the coupling between the subsystems. The iteration (1) decomposes in two steps, one proximal iteration on f which maintains separability and a projection on C which satisfies the coupling. This idea has been used by Pierra[14] in this context, but the convergence proof we propose here, beside the fact that it applies to the splitting of general operators, is more straightforward and less restrictive than his proof.

2 PRELIMINARY RESULTS

In this section, A and B are maximal monotone operators on X, finite dimensional, with respective domains D(A) and D(B), and we denote by A_λ (resp. B_λ) their Moreau-Yosida approximations,i.e. $A_\lambda = {}^1/_\lambda(I - (I+\lambda A)^{-1})$. The operator $(I+\lambda A)^{-1}$ is generally called the resolvent. We recall below some important properties of these operators which proofs can be found in Brézis[3] :

Properties :

1.0 *A is closed in the sense that its graph* $Gr(A)=\{(x,y)\in XxX/\ y\in A(x)\}$ *is closed in XxX. Furthermore, for any* $x\in D(A)$, *the set* $\{y\in X/\ y\in Ax\}$ *is closed, convex and nonempty.*

1.1 $(I+\lambda A)^{-1}$ *is a contraction defined on the whole space X for any* $\lambda>0$.

1.2 A_λ *is maximal monotone and lipschitzian with ratio* ${}^1/_\lambda$.

1.3 $\forall x\in X,\ A_\lambda x \in A((I+\lambda A)^{-1}x)$.

1.4 $\overline{D(A)}$ *is convex, the range of* $(I+\lambda A)^{-1}$ *is D(A) and*

$$\lim_{\lambda\downarrow 0} (I+\lambda A)^{-1}x = Proj_{\overline{D(A)}}\, x$$

1.5 *For any* $\lambda>\mu>0$, *for any* $x\in D(A)$, $||A_\lambda x||\le||A_\mu x||\le||A_0 x||$, *where* A_0x *is the minimum norm element of Ax.*

Moreover, when $\lambda\downarrow 0$, *we have the following limits :*

$\forall x\in D(A),\ \lim ||A_\lambda x|| = ||A_0 x||$

If $x\notin D(A),\ \lim ||A_\lambda x|| = +\infty$

The Proximal Point algorithm is based on property **1.1**. Rockafellar [15] has analyzed the convergence of the proximal iteration :

$$x_{k+1} = (I+\lambda_k A)^{-1}x_k \qquad (4)$$

The main result tells us that, when λ_k is bounded away from zero, problem (**P**) has a solution if and only if the sequence $\{x_k\}$ is bounded . Then it converges to a point x^∞ such that $0\in Ax^\infty$ and $\lim ||A_{\lambda_k}x_k|| = 0$.

Rockafellar concludes that the convergence becomes faster when λ_k increases. Observe that the Proximal Point method, as it searches a fixed point of $(I+\lambda A)^{-1}$, converges to a zero of the operator A_λ which is also a zero of A. This is no more true if we look for a zero of A+B and substitute A+B by A_λ+B. In fact, we need a sequence $\{\lambda_k\}$ which decreases to zero.

Brezis and Lions [4] have given the following conditions for the convergence of the sequence $\{x_k\}$ generated by (4) to a zero of A :

i) a sufficient condition for the convergence is : $\Sigma\lambda_k{}^2 \to \infty$.

ii) if $\Sigma\lambda_k{}^2$ is bounded and $\Sigma\lambda_k \to \infty$, then the sequence $\{x_k\}$ converges to a point which is generally not a solution, but on the other hand, the sequence of average solutions $\{z_k\}$, $z_k = \sum_{i=1}^{k} \lambda_i x_i / \sum_{i=1}^{k} \lambda_i$ does converge.

iii) if $A=\partial f$, the subdifferential of a convex lsc mapping, then it suffices to suppose that $\Sigma\lambda_k \to \infty$ to get the convergence of the sequence $\{x_k\}$.

3 REGULARIZATION TECHNIQUES

The regularization we are interested in is the substitution of a maximal monotone operator by its Moreau-Yosida approximation. It is well-known (see [3] or [16]) that the sum of two maximal monotone operators A and B is maximal monotone when one of the following conditions is fulfilled :

- $\mathrm{Int}(D(A)) \cap D(B) \neq \emptyset$
- one of the operators is single-valued and lipschitzian.

Then, a natural way to regularize the sum of two operators is to regularize one of them. We are faced with three distinct strategies : to regularize A only, to regularize B only or to regularize both.

In the first case, we approximate (**P**) by ($\mathbf{P}_\lambda$) which can be transformed in the following way :

$$0 \in x - (I+\lambda A)^{-1} x + \lambda B x$$

which, in turn, as B is maximal monotone, is equivalent to :

$$x = (I+\lambda B)^{-1} (I+\lambda A)^{-1} x \qquad (5)$$

Thus, x is a fixed point of the operator $T_\lambda = (I+\lambda B)^{-1}(I+\lambda A)^{-1}$, which is a contraction, inducing the fixed point iteration :

$$x^0 \in X$$
$$x^{t+1} = (I+\lambda B)^{-1}(I+\lambda A)^{-1} x^t \qquad (6)$$

Before discussing the convergence of iteration (6), we recall some useful results on the existence of solutions for problems (**P**) and ($\mathbf{P}_\lambda$).

3.1 Existence results

Existence of solutions is linked with the property of *surjectivity* of the operator : we know for example that, if T is a maximal monotone operator such that D(T) is *bounded*, then T is surjective (see Brézis [3]).

If T is the sum of two maximal monotone operators, it is clear that its domain is bounded if one of the domains is bounded. On the other hand, we have seen before that the sum of two maximal monotone operators is maximal when one of them is single-valued and Lipschitzian. This implies that $(\mathbf{P}_\lambda)$ has a solution if D(B) is bounded (this last result is similar to the conditions of existence for variational inequalities involving single-valued and hemicontinuous monotone operators on bounded convex sets given in Stampacchia [18]).

Another interesting situation which we develop hereafter is the case where one operator is *strongly monotone* :

If B is strongly monotone, i.e., if $\exists\alpha>0$ such that :

$< x-x', y-y'> \geq \alpha||x-x'||^2$, $\forall x$, x' and $\forall y\in Bx$, $\forall y'\in Bx'$

then $(\mathbf{P}_\lambda)$ has a unique solution for any $\lambda>0$. Indeed, the operator T_λ is now a strict contraction :

$$||T_\lambda(x)-T_\lambda(x')|| \leq \frac{1}{1+\alpha\lambda}||(I+\lambda A)^{-1}x - (I+\lambda A)^{-1}x'|| \leq \frac{1}{1+\alpha\lambda}||x- x'|| \qquad (7)$$

Note that, in this case, $(\mathbf{P})$ has at most one solution.

Furthermore, as B is strongly monotone, $C=B-\alpha I$ is maximal monotone and $(\mathbf{P}_\lambda)$ is equivalent to :

$$0 \in x_\lambda + \frac{1}{\alpha} Cx_\lambda + \frac{1}{\alpha} A_\lambda x_\lambda$$

We observe here that, when $\alpha=1$, the present situation fits in the model analyzed by Brezis[3,p.34] :

Note that a similar result to (7) exists if we regularize B in the place of A. We will show in Theorem 3 that the strongly monotone case allows some refinements in the final convergence theorem.

3.2 Convergence of the proximal iteration

Theorem 1 :

For some $\lambda>0$, assume that the sequence $\{x^t\}$ generated by (6) is bounded, then it converges to a point x_λ which is a solution of the regularized problem $(\mathbf{P}_\lambda)$.

Proof : Each resolvent is nonexpansive as was seen before (Prop. **1.1**), so that the composed operator T_λ is nonexpansive too. As the iterates are all in a bounded set, by a theorem of Browder et al(see [5]), the sequence (6)

converges to a fixed point of T_λ, say x_λ, which means that :

$$x_\lambda = (I+\lambda B)^{-1}(I+\lambda A)^{-1} x_\lambda$$

Remark : If we know that $(\mathbf{P}_\lambda)$ has a solution x_λ (this is the case when one operator is strongly monotone), then it is a fixed point of T_λ and for any t, we have :

$$||x^t - x_\lambda|| = ||T^t x^0 - T^t x_\lambda|| \leq ||x^0 - x_\lambda||$$

We have seen in section 3.1 that the choice of the operator to be regularized is not indifferent. More insight on that question can be given on the application of the regularization to the convex problem (3) (see too **Th.3**) :

a) *Regularizing A* :

Then $(\mathbf{P}_\lambda)$ is the optimality condition for the regularized convex program :

$$\begin{aligned} &\text{Minimize} \quad f_\lambda(x) \\ &x \in C \end{aligned} \qquad (8)$$

where $f_\lambda(x)=\text{Inf}\ \{ f(z)+ {}^1/_{2\lambda}||z-x||^2\}$. It is known [12] that for every x, there is a unique z_x such that $f_\lambda(x)=f(z_x)+ {}^1/_{2\lambda}||z_x-x||^2$, and that $z_x=(I+\lambda\partial f)^{-1}x$. The regularized function f_λ is smooth and its gradient is $A_\lambda x={}^1/_\lambda(x-z_x)$.

A sufficient condition for the existence of solutions of $(\mathbf{P})$ and $(\mathbf{P}_\lambda)$ is that C is a bounded set (or equivalently that f is coercive).

Now, iteration (6) takes the form :

$$\begin{aligned} &x^0 \in C \\ &z^{t+1}=\text{Argmin}\{f(z)+ {}^1/_{2\lambda}||z-x^t||^2\} \\ &x^{t+1}=\text{Proj}_C z^{t+1} \end{aligned} \qquad (9)$$

b) *Regularizing B* :

If we regularize B, we get the following equation :

$$0 \in Ax + B_\lambda x \qquad (10)$$

and, as $B_\lambda x= {}^1/_\lambda(x-\text{Proj}_C x)$, (10) is the optimality condition for the unconstrained problem :

$$\begin{aligned} &\text{Minimize} \quad f(x) + {}^1/_{2\lambda} d(x,C)^2 \\ &x \in X \end{aligned} \qquad (11)$$

where d(x,C) is the Euclidian distance between x and C. In fact, the constraint is treated here like in a penalty method. Again, (11) has a

solution if C is bounded.

Equation (10) leads to the following fixed point equation :

$$z_\lambda = (I+\lambda A)^{-1}\,(I+\lambda B)^{-1}\,z_\lambda \qquad (12)$$

and we obtain the same sequences $\{z^t\}$ as in (9) if we initialize by $z^1 = (I+\lambda A)^{-1}\,x^0$.

In conclusion, we are faced with two dependent sequences, $\{x^t\}$ and $\{z^t\}$, the first one in C and the second one outside C such that $x^t = Proj_C z^t$. Both converge from **Theorem 1** and if x_λ and z_λ are their respective limit points, we have $x_\lambda = Proj_C\, z_\lambda$.

Remarks : 1) If both operators were regularized, we should add the penalty term to (8) but no direct fixed point iteration could be induced.
2)Some additional insight on these models is given at the end of section 4.

4 MAIN CONVERGENCE RESULT

We analyze here the convergence of the sequence $\{x_k\}$ defined by : $x_k \in X$ and x_k solves problem (P_λ) for $\lambda=\lambda_k$, where $\{\lambda_k\}$ is a sequence of positive real numbers such that $\lambda_k \downarrow 0$, when $k\to\infty$. We assume in this section that (P_λ) has a solution for any $\lambda_k>0$. Then, for each k, x_k satifies :

$$x_k = (I+\lambda_k B)^{-1}(I+\lambda_k A)^{-1}\,x_k \qquad (13)$$

To simplify our notation, we denote by A_k the operator A_{λ_k}

We denote by $\{y_k\}$ the sequence associated to the sequence $\{x_k\}$ such that $y_k = A_k x_k$, $\forall k$. That sequence plays a central role in the convergence results we present hereafter.

Theorem 2 :

(2.1) *Let x be a solution of (P) and $y \in Ax \cap (-Bx)$. Then, $\|y_k\| \le \|y\|$ for any $\lambda_k>0$.*

(2.2) *Let x be a limit point of $\{x_k\}$ and assume that the sequence $\{y_k\}$ is bounded. Then, x solves (P).*

(2.3) *Assume that the sequence $\{x_k\}$ has a limit point. Then, (P) admits a solution if and only if the sequence $\{y_k\}$ is bounded.*

(2.4) *If (P) has a unique solution x^* and if the sequence $\{x_k\}$ is bounded, then $x_k \to x^*$ and $y_k \to y^*$ which is the element of minimum norm in $Ax^* \cap (-Bx^*)$.*

Proof :

(2.1) : As $y \in -Bx$ and $y_k \in -Bx_k$, the monotonicity of B implies that :

$$< y_k - y, x_k - x > \leq 0$$

But, as $y_k = A_k x_k = (\lambda_k)^{-1}(x_k - (I+\lambda_k A)^{-1}x_k)$, we can write :

$$x_k = \lambda_k y_k + (I+\lambda_k A)^{-1}x_k \qquad (14)$$

Then, $< y_k - y, \lambda_k y_k > \leq - < y_k - y, (I+\lambda_k A)^{-1}x_k - x >$

The right-hand side is non positive because $y_k = A_k x_k \in A(I+\lambda_k A)^{-1}x_k$ (**Prop.1.3**), $y \in Ax$ and A is monotone. Then :

$$\|y_k\|^2 \leq \|y\| \, \|y_k\|$$

(2.2) : We show first that any limit point x of the sequence $\{x_k\}$ is in $\overline{D(A)} \cap \overline{D(B)}$: from (13) and **Prop.1.4**, we see that $x_k \in D(B)$. Again, from (14) and **Prop.1.4**, we see that $x_k \in \lambda_k y_k + D(A)$. Then, as y_k is bounded, the first term of the sum tends to zero when $\lambda_k \downarrow 0$ and the limit point x must be in $\overline{D(A)}$. Then, $x \in \overline{D(A)} \cap \overline{D(B)}$.

As x_k solves $(\mathbf{P}_\lambda)$ for $\lambda = \lambda_k$, it satisfies :

$$-A_k x_k \in Bx_k \quad \text{or equivalently} : -y_k \in Bx_k$$

By assumption, the sequence $\{y_k\}$ is bounded and we can extract a convergent subsequence which we denote too $\{y_k\}$ to avoid overloaded notations . Let y be its limit.

On the other hand, if we put $z_k = (I+\lambda_k A)^{-1}x_k$, then, from **Prop. 1.3** , we know that $y_k \in Az_k$ for any k. To prove that $\lim z_k = x$, we compute $\|z_k - x\|$:

$$\|z_k - (I+\lambda_k A)^{-1}x + (I+\lambda_k A)^{-1}x - x\| \leq \|(I+\lambda_k A)^{-1}x_k - (I+\lambda_k A)^{-1}x\| + \|(I+\lambda_k A)^{-1}x - x\|$$

The first norm is bounded by $\|x_k - x\|$because $(I+\lambda_k A)^{-1}$ is a contraction. The second norm tends to zero when λ_k tends to zero (Prop. **1.4** with the fact that $x \in \overline{D(A)}$).

Finally, as $y_k \to y$ and $z_k \to x$, A being a closed map, we must have :

$y \in Ax$

But, $-y_k \in Bx_k$, and from the closedness of B, we conclude that :

$-y \in Bx$, which means that x solves (**P**).

(2.3) : Immediate consequence of **(2.1)** and **(2.2)**.

(2.4) : Suppose now that problem (**P**) possesses a unique solution x^*. As the sequence $\{x_k\}$ is bounded, we know from **(2.2)** that its limit point solves (**P**). Then the sequence $\{x_k\}$ has a unique limit point which is exactly x^* and $x_k \to x^*$. Let y^* be the element of minimum norm in

$Ax^* \cap (-Bx^*)$ (note that y^* exists and is unique according to **Prop.1.0**). According to **(2.1)** we must have for any k :

$$\|y_k\| \leq \|y^*\|$$

Let y be a limit point of the sequence $\{y_k\}$. Then, $\|y\| \leq \|y^*\|$. But, as in the proof of **(2.2)**, we have $y \in Ax^* \cap (-Bx^*)$. It follows that $y=y^*$ and the whole sequence $\{y_k\}$ converges to y^*.

Remarks :

We have used the fact in the proof of **(2.2)** that $x_k \in D(B)$ for any k. Then, the sequence $\{x_k\}$ is bounded if D(B) is bounded. This is the case for example if (**P**) is the optimality condition for problem (3) with C bounded.

On the other hand, if D(A) is bounded, the existence of a solution of (**P**) implies that $\{x_k\}$ is bounded. Indeed, it follows from **(2.1)** that the sequence $\{y_k\}$ is bounded and relation (14) with the same arguments as in the proof of **(2.2)** shows that x_k remains bounded.

An important case where unicity of the solutions is observed is the strongly monotone case. Indeed, if B is strongly monotone and A is regularized, problem (**P**) has at most one solution and problem ($\mathbf{P}_\lambda$) has a unique solution for any $\lambda>0$ (see section 3.1). Then, **(2.4)** can be refined in the following theorem which is, in fact, inspired by a theorem given in Brézis [3,p.35] and which proof will therefore be omitted here :

Theorem 3 :

Suppose that B is strongly monotone with constant α and that A is regularized. Then, (P) admits a unique solution x^* if and only if the sequence $\{y_k\}$ is bounded. In this case, $x_k \to x^*$ and $y_k \to y^*$ which is the element of minimum norm in $Ax^* \cap (-Bx^*)$.
Moreover, we have the following estimation :

$$\|x_k - x^*\| \leq [{}^1/_\alpha \, \lambda_k \|y^*\| \, \|y_k - y^*\|]^{1/2} = o(\sqrt{\lambda})$$

Application : Strongly convex programming
Let consider the following problem :
(**P**) Minimize f(x) on $x \in C$
where f is a strongly convex function on C, a closed convex set of R^n
This means now that $B=\partial f$ is strongly monotone. In this particular case, we know that (**P**) has a unique solution. In addition, **Theorem 3** implies the following convergence results :
a) B is regularized. Then, ($\mathbf{P}_\lambda$) is :

Minimize $f_\lambda(x)$ on $x \in C$, which admits a unique solution x_λ for any $\lambda>0$

b) A is regularized. Then (P_λ) is :

Minimize $f(x) + {}^1/_{2\lambda} d(x,C)^2$ which admits a unique solution x'_λ for any $\lambda>0$.

When $\lambda\downarrow 0$, $x_\lambda \to x^*$ and $x'_\lambda \to x^*$. Furthermore, we get the estimations in a neighbourhood of x^* :

$||x_\lambda - x^*|| = o(\sqrt{\lambda})$

$||x'_\lambda - x^*|| = o(\sqrt{\lambda})$

Observe that these last results mean that the whole sequence generated by that specific penalty method converges at the speed of $\sqrt{\lambda}$. We have then given a elegant and clearcut proof of the convergence for both regularizations.

5 APPLICATION TO THE DECOMPOSITION OF CONVEX PROGRAMS

We consider first the following problem in R^n :

$$\text{Minimize} \quad \sum_{i=1}^{p} f_i(x)$$

$$x \in S = \bigcap_{i=1}^{p} S_i \qquad (15)$$

where each f_i is a proper convex and lsc function on S_i, which are closed and bounded convex sets of R^n. We assume too that S is not empty.

To decompose it, we create p copies of the variables denoted by $x_1,\ldots,x_p$, with $x_i \in R^n$ for all i. Then problem (15) is equivalent to solve in the product space $X=(R^n)^p$ the following problem :

$$\text{Minimize} \quad \sum_{i=1}^{p} f_i(x_i)$$

$$x_i \in S_i, \ i=1,\ldots,p$$

$$x=(x_1,\ldots,x_p) \in L = \{x \in X \mid x_1=\ldots=x_p\} \qquad (16)$$

To apply the precedent algorithm to (16), we put :

$$A = \sum_{i=1}^{p} \partial f_i(x_i) + \sum_{i=1}^{p} \partial g_i(x_i) = \partial\Big(\sum_{i=1}^{p} [f_i(x_i) + g_i(x_i)]\Big)$$

where g_i is the indicator function of the set S_i, and :

$B = L^{\perp}$, the orthogonal subspace to L.

Then, it can be easily verified that :

$(I+\lambda A)^{-1}x = (z_1,\ldots,z_p)$, with $z_i = \text{Argmin}\ \{ f_i(\xi_i) + {}^1/_{2\lambda}||\xi_i - x_i||^2 \mid \xi_i \in S_i\}$

$(I+\lambda B)^{-1}x = \text{Proj}_L x = {}^1/_p \sum x_i$

Indeed, each step of the algorithm splits in p convex subproblems, each one solved on an isolate set S_i :

Initialize $\lambda>0$, $x^0 \in R^n$;

a) Solve the independent subproblems for each i :

$$\text{Minimize}_{x_i \in S_i} \quad f_i(x_i) + {}^1/_{2\lambda}||x_i - x^t||^2 \qquad (17)$$

Let z_i^t be the optimal solution of (15).

$$\text{b)} \quad x^{t+1} = {}^1/_p \sum_{i=1}^{p} z_i^t \qquad (18)$$

This algorithm has been proposed in Pierra [14] . As (17) has a unique optimal solution, from **Theorem 1**, we conclude that the sequence $\{x^t\}$ converges to a limit point x_λ and, when $\lambda \downarrow 0$, from **Theorem 2**, x_λ tends to a solution of (15). In practice, we need a test to decide whether we must reduce λ and solve another cycle of iterations {(17)-(18)} or not.

Other situations where decomposition is induced on a certain product space are reviewed in Mahey and Nguyen [9]. In particular, when f is the dual function of a convex constrained separable problem, the proximal algorithm looks like a separable version of the Augmented Lagrangian method. In this sense, it can be compared to similar approaches given in Bertsekas [1] and in Spingarn [17]. Numerical experiments on these methods for the decomposition of large-scale convex programs will be published elsewhere.

6 CONCLUSION

Regularization techniques for inclusion problems depending on the sum of two maximal operators lead to an iterative process composed of two separate proximal steps. To avoid ergodic. convergence, we need to iterate with a fixed sufficiently small parameter.

If we come back to the doubly subscripted sequence defined by (1), we know from elementary analysis that there exists a subsequence $\{x_k^{t(k)}\}$ which converges to a solution x^* of (**P**). This means that iteration (2) can converge without the ergodic artifice. We conjecture that the sequence $\{\lambda_k\}$ must decrease to zero very slowly (no counterexample with $\Sigma\lambda_k{}^2 \to \infty$ has been found to our knowledge). Some of these numerical aspects will be presented in a forthcoming paper.

The splitting of the operators induces some classical iterative schemes for the decomposition of convex programs which, though theoretically very slow, are numerically stable and quite simple to implement in comparison with some direct approach like Spingarn's Partial Inverse method [16].

REFERENCES

[1] D.P. Bertsekas and J.N. Tsitsiklis , *Parallel and Distributed Computation*, Prentice-Hall Int. Ed., 1989.

[2] M.A. Boughazi, Contribution à l'étude des algorithmes d'optimisation en Analyse des Données, Thèse de Doctorat, Université de Grenoble, 1987.

[3] H. Brezis, *Opérateurs maximaux monotones*, Mathematics Studies n°5, North Holland, 1973.

[4] H. Brezis and P.L. Lions, "Produits infinis de résolvantes", *Israel J. of Math.* 29, 4 (1978), pp.329-345.

[5] F. Browder and W. Petryshin, "Construction of fixed points of nonlinear mappings in Hilbert spaces", *J. of Math. Anal. and Appl.* 20 (1967), pp. 197-228.

[6] J. Lemordant and Pham Dinh T., "Algorithme proximal pour la résolution numérique d'équations d'évolution en vision de bas niveau", preprint.

[7] P.L. Lions, "Une méthode itérative de résolution d'une inéquation variationnelle", *Israel J. of Math.* 31, 2 (1978), pp. 204-208.

[8] P.L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators", *SIAM J. Numer. Anal.* 16, 6 (1979), pp.964-979.

[9] P. Mahey and Nguyen van H., "Proximal methods and decomposition of large convex programs", Preprint, 1990.

[10] B.Martinet, Algorithmes pour la résolution de problèmes d' optimisation et de minimax, Thèse d'Etat, Univ. de Grenoble, 1972.

[11] C. Michelot, Problèmes de localisation : propriétés géométriques et résolution par des méthodes d'optimisation, Thèse de Doctorat, Université de Bourgogne, 1988.

[12] J.J. Moreau, "Proximité et dualité dans un espace Hilbertien", *Bull. Soc. Math. France* 93 (1965), pp. 273-299.

[13] G.B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in Hilbert spaces", *J. of Math. Anal. and Appl.* 72 (1979), pp. 383-390.

[14] G. Pierra, "Decomposition through formalization in a product space", *Math. Prog.* 28 (1984), pp. 96-115.

[15] R.T.Rockafellar, "Monotone operators and the proximal point algorithm in convex programming", *SIAM J. on Control and Optim.* 14 (1976), pp. 877-898.

[16] R.T. Rockafellar, "On the maximality of sums of nonlinear monotone operators", Trans. Amer. Math. Soc. 149, 1970, pp. 75-88.

[17] J.E. Spingarn, "Partial inverse of a monotone operator", *Appl. Math. Optim.* 10 (1983), pp. 247-265.

[18] G. Stampacchia, "Variational inequalities", in *Theory and Applications of Monotone Operators*, A. Ghizetti ed., 1969.

[19] A.Yassine, Etudes adaptatives et comparatives de certains algorithmes en optimisation. Implementations effectives et applications, Thèse de Doctorat, Université de Grenoble, 1989.

UNE METHODE PROXIMALE POUR LA MINIMISATION DES FONCTIONS DC

Raimundo J. B. de Sampaio[1] et Philippe Mahey[2]

Résumé - Nous présentons un nouvel algorithme pour la minimisation locale de fonctions dc (différences de fonctions convexes). Il s'agit d'un algorithme de descente de type proximal qui prend en compte séparément la convexité des deux fonctions.

A PROXIMAL METHOD FOR THE MINIMIZATION OF DC FUNCTIONS

Abstract - We present a new algorithm for the local minimization of dc functions (difference of two convex functions). It is a descent method of the proximal kind which takes in consideration the convex properties of the two functions separately.

1 INTRODUCTION

L'intérêt pour les fonctions dc (différences de deux fonctions convexes) est motivé par deux raisons principales :

- L'ensemble des fonctions dc sur C, convexe compact d' un espace vectoriel X muni de la topologie de la convergence uniforme, est dense dans l'ensemble des fonctions continues sur C.
- La convexité sous-jacente dans la décomposition peut être exploitée pour minimiser ces fonctions même dans le cas non différentiable.

Récemment, un certain nombre de travaux ont été consacrés aux fonctions dc (cf par exemple [4],[6],[7],[9],[11]). Assez peu d'articles par contre présentent des méthodes numériques qui exploitent la structure particulière de ces fonctions. On retiendra la méthode de régularisation introduite par Gabay [5], l'approche duale de Auchmuty [1] et les méthodes de sous-gradients de Pham Dinh et El Bernoussi [11]. En fait, seules ces dernières méthodes ont fait l'objet d'implantations numériques concrètes (cf également [2]) (on ne parle pas ici des diverses méthodes existantes en optimisation non convexe qui peuvent naturellement s'appliquer à la programmation dc, cf par exemple Tuy [15]).

Comme l'a signalé Hiriart-Urruty [8], l'avantage de disposer d'une représentation dc est de pouvoir utiliser deux fois tout l'arsenal de propriétés fourni par l'Analyse Convexe. Cette idée est concrétisée dans l'algorithme proposé ci-dessous. Il s'agit d'un algorithme de descente pour

[1]Departamento de Matematica, Universidade do Parana, Curitiba, Brésil
[2] Laboratoire ARTEMIS, IMAG, BP53X, 38041 Grenoble

la recherche de points critiques de la fonction, c'est-à-dire de points satisfaisant des conditions nécessaires de minimum local de f.

2 PRELIMINAIRES

Soit X un espace de dimension finie identifié avec R^n muni d'un produit interne <.,.>. On désigne par $\Gamma_0(X)$ l'ensemble des fonctions convexes, semi-continues inférieurement et propres sur X. Soit f une fonction dc sur X, i.e. telle qu'il existe g et h, éléments de $\Gamma_0(X)$ avec :

$$\forall x \in X,\ f(x) = g(x)-h(x)$$

Supposons de plus que $h(x)\neq+\infty$, $\forall x\in X$ et que $Dom(g)\cap Dom(h)\neq\emptyset$. Il est clair que g et h peuvent être choisies fortement convexes car on peut toujours ajouter à chacune d'elles le terme $k||x||^2$ avec $k>0$. Les fonctions conjuguées respectives de g et h seront désignées par g^* et h^*, leurs sous-différentiels respectifs par $\partial g(.)$ et $\partial h(.)$.

Propriétés (cf [7] et [14]) :

1) $\inf_x g(x)-h(x) = \inf_y h^*(y)-g^*(y)$ (1)

2) *Une condition nécessaire pour que $x\in Dom(f)$ soit un minimum local de f est :*

$$\partial h(x) \subset \partial g(x) \quad (2)$$

La propriété 2) étant souvent difficile à tester, on la relaxe par :

$$\partial g(x)\cap\partial h(x) \neq \emptyset \quad (3)$$

On appelle généralement *point critique* un point qui satisfait (3).

La méthode proposée dans cet article est à rapprocher de l'algorithme du *point proximal* (cf par exemple [13]). Cet algorithme permet de rechercher un zéro d'un opérateur maximal monotone A par l'itération de point fixe :

$$x_{k+1}=(I+cA)^{-1}x_k \quad (4)$$

où c est un paramètre positif. L'opérateur $(I+cA)^{-1}$ est la résolvante de A. C'est une contraction définie sur tout l'espace, donc en particulier, $(I+cA)^{-1}$ est Lipschitz continu de constante 1 (cf [3] pour plus de détails sur cet opérateur).

Quand A est le sous-différentiel d'une fonction g convexe, sci, propre, l'itération (4) s'écrit :

$$x_{k+1} = \text{Argmin}\{ g(x) + (2c)^{-1}||x- x_k||^2\} \quad (5)$$

L'intérêt de ce type de méthode est fonction de la simplicité relative des sous-problèmes (5). Rockafellar [13] a fourni une étude détaillée de la convergence de la méthode proximale. En particulier, l'algorithme converge linéairement avec un taux qui tend vers zéro avec

1/c (convergence superlinéaire) et la résolution des sous-problèmes (5) peut se faire avec une approximation ε_k si $\Sigma\varepsilon_k<+\infty$.

3 UNE METHODE DE DESCENTE POUR LA MINIMISATION DES FONCTIONS DC

La clef de la méthode que nous proposons ci-après se trouve dans la décomposition de chaque itération en deux pas distincts, chaque pas prenant en compte la convexité d'une des deux fonctions.

Grossièrement, la méthode consiste à augmenter h dans la direction d'un sous-gradient puis à réduire g par une itération proximale. En effet, si on ne peut garantir que la direction opposée d'un sous-gradient est toujours une direction de descente pour une fonction convexe, ce sous-gradient est par contre toujours une direction de montée :

Lemme 1 - *Soient $h\in\Gamma_0(X)$ et $x\in X$. Alors $\forall w\in\partial h(x)$ tel que $w\neq 0$ et $\forall c>0$, on a :*

$$h(x+cw) > h(x)$$

Démonstration : Conséquence immédiate de la convexité de h :

$$h(x+cw) \geq h(x) + \langle w, cw\rangle$$

Montrons maintenant qu'un point critique de f peut s'écrire sous la forme d'un point fixe d'un certain opérateur :

Lemme 2 : *Une condition nécessaire et suffisante pour que x soit un point critique de f est que x soit solution de :*

$$x = (I + c\partial g)^{-1}(x + cw) \text{ pour un } c>0 \text{ et } w\in\partial h(x)$$

Démonstration : Soit $w\in\partial g(x)\cap\partial h(x)$. Donc $w\in\partial g(x)$, ce qui équivaut à : $x+cw \in x+c\partial g(x)$. Comme ∂g est maximal monotone, on obtient le résultat cherché.

ALGORITHME LRK :

0. $x_0\in X$ *et* $c>0$, *arbitraires*
   $k=0$

1. *Déterminer* $w_k\in\partial h(x_k)$ *et* $y_k = x_k+cw_k$

2. *Calculer* $x_{k+1}=(I+c\partial g)^{-1}y_k$
   Si $x_{k+1}=x_k$, x_k *est un point critique de f*
   Sinon, incrémenter k et retourner en 1.

Remarques :

i) Dans le cas convexe ($h\equiv 0$), on retrouve la méthode du point proximal ([13]).

ii) Dans le cas où h est différentiable, cet algorithme s'apparente à une méthode proposée par Mine et Fukushima [10].

iii) Comme le montre la démonstration du Lemme 2, la méthode fait partie des approches par régularisation. Plus précisément, il est clair que minimiser $f(x)=g(x)-h(x)$ est équivalent à minimiser $f'(x)=g'(x)-h'(x)$ avec $g'=cg+{}^1/_2\|\ .\ \|^2$ et $h'= ch+{}^1/_2\|\ .\ \|^2$ pour un $c>0$. L'algorithme peut alors s'écrire :

$$y_k \in \partial h'(x_k) \text{ et } x_{k+1} \in \partial (g')^*(y_k) \qquad (6)$$

où $(g')^*$ est la conjuguée de g' (on peut vérifier que $\partial (g')^* = (I+c\partial g)^{-1}$)
Les pas alternés sur les sous-différentiels de h' et $(g')^*$ montrent le rapprochement avec les méthodes de sous-gradients introduites par Pham Dinh et El Bernoussi[11]. Il est important de noter toutefois que toutes les méthodes adaptées aux fonctions dc dépendent de la décomposition de ces fonctions en différence de fonctions convexes, décomposition qui n'est bien sûr pas unique.

iv) Comme toutes les méthodes de type proximal, leur efficacité dépend de la complexité du calcul du prox. Ici, l'étape 2. équivaut à résoudre :

Minimiser $g(x) + (2c)^{-1}\|x-y_k\|^2$

$x \in X$

4 CONVERGENCE DE L'ALGORITHME

Montrons tout d'abord que l'algorithme LRK est un algorithme de descente.

Théorème 1 : *La suite $\{x_k\}$ générée par l'algorithme LRK satisfait pour tout k :*

Soit, l'algorithme termine avec un point critique de f, soit :
$f(x_{k+1}) < f(x_k)$

Démonstration : Si $x_{k+1}=x_k$, cela implique d'après le lemme 2 que x_k est un point critique de f.

Supposons alors que $x_{k+1}\neq x_k$. On peut réécrire l'itération de la manière suivante :

$$x_k+cw_k \in x_{k+1}+c\partial g(x_{k+1})$$

$$\Leftrightarrow \quad c^{-1}(x_k-x_{k+1})+w_k \text{ est un sous-gradient de g en } x_{k+1}$$

$$\Leftrightarrow \quad g(x_k) \geq g(x_{k+1}) + \langle c^{-1}(x_k-x_{k+1})+w_k,\ x_k-x_{k+1}\rangle \qquad (7)$$

D'autre part, w_k étant un sous-gradient de h en x_k, on a :

$$h(x_{k+1}) \geq h(x_k) + \langle w_k, x_{k+1}-x_k \rangle \qquad (8)$$

Si on retranche membre à membre (8) de (7), on obtient :

$$f(x_{k+1}) \leq f(x_k) - c^{-1}\|x_k-x_{k+1}\|^2 \qquad (9)$$

D'où on conclut que $f(x_{k+1}) < f(x_k)$.

Pour démontrer la convergence de l'algorithme, il suffit de montrer que la multiapplication qui à x_k associe x_{k+1}, est fermée.

Théorème 2 : *Supposons que les suites $\{x_k\}$ et $\{y_k\}$ générées par l'algorithme LRK sont bornées. Alors il existe une sous-suite convergente $\{x_{k'}\}$qui converge vers un point critique de f.*

Démonstration : D'après Zangwill[16], la convergence globale de l'algorithme dépend de trois propriétés de la suite des itérés : descente, continuité et non divergence. Soit S l'ensemble des points critiques de f.
i) La fonction f est une fonction de descente pour l'ensemble des points critiques. En effet, le théorème 1 garantit que $f(x_{k+1}) < f(x_k)$, $\forall x_k$ tel que $x_k \neq x_{k+1}$, c'est-à-dire tel que $x_k \notin S$, d'après le lemme 2. Trivialement, si $x_k \in S$, $f(x_k)=f(x_{k+1})$.
ii) L'algorithme peut s'écrire sous la forme : $x_{k+1} \in B.Cx_k$, avec $B=(I+c\partial g)^{-1}$ et $C=I+c\partial h$. B est l'opérateur résolvant de ∂g, donc son graphe est fermé . De plus, comme h est une fonction propre et semi-continue inférieurement, le graphe de ∂h est également fermé (Rockafellar [12], Th.23.4). Donc, la suite $\{y_k\}$ avec $y_k \in Cx_k$ étant bornée par hypothèse, il existe une sous-suite $\{y_{k'}\}$ qui converge et la multiapplication B.C est fermée (voir Théorème sur la composition de multiapplications fermées dans Zangwill [16]).
iii) La suite $\{x_k\}$ est bornée par hypothèse.
Donc, d'après [16], il existe une sous-suite $\{x_{k''}\}$ qui converge vers un point de S.

Remarques : 1) Si f est bornée inférieurement, la suite entière converge vers un point critique. En effet, si on somme les inégalités (9) jusqu'à l'itération k, on obtient :

$$\frac{1}{c}\sum_{i=0}^{k} \|x_{i+1}- x_i\|^2 \leq f(x_0) - f(x_{k+1})$$

Soit f^- la borne inférieure de f. On a à la limite :

$$\sum_{i=0}^{\infty} \|x_{i+1}- x_i\|^2 \leq c(f(x_0) - f^-)$$

d'où on déduit que la suite $\{x_k\}$ est de Cauchy.

2) La suite $\{y_k\}$ est bornée si la suite $\{x_k\}$ est bornée et s'il existe un point d'adhérence de cette suite dans l'intérieur relatif du domaine de h.

5 APPLICATION A LA PROGRAMMATION CONCAVE

On va considérer ici l'application de l'algorithme LRK au problème suivant :

(P) Maximiser h(x)
$x \in C$

où h est une fonction convexe sci sur C, sous-ensemble convexe fermé de R^n.

Le problème (P) a fait l'objet d'un grand nombre d'études, certaines préconisant des algorithmes visant l'obtention de la solution optimale globale ou de points critiques suivant les cas ([11], [15], ...). Nous ne prétendons pas ici analyser les performances de l'algorithme LRK dans ce cas, mais seulement illustrer son comportement dans une situation classique.

(P) se met immédiatement sous la forme d'un programme dc en introduisant la fonction indicatrice χ_C de l'ensemble C :

(P') Minimiser $f(x) = \chi_C(x) - h(x)$

L'algorithme LRK prend alors la forme suivante :

- $x_0 \in C$, $c>0$
- $x_{k+1} = \mathrm{Proj}_C(x_k + cw_k)$ où $w_k \in \partial h(x_k)$

La progression de l'algorithme est illustrée sur la figure 1.

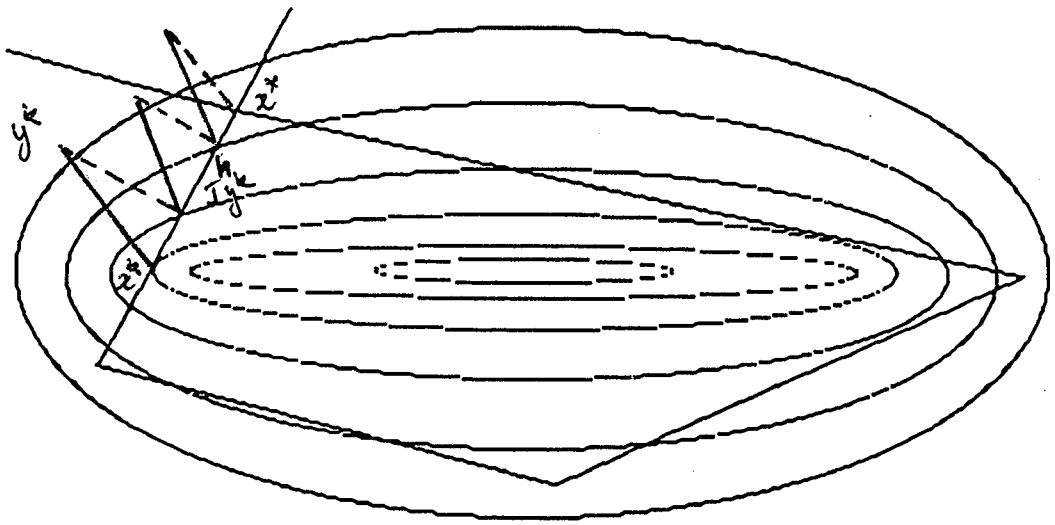


Illustration de la performance de l'Algorithme LRK

Par exemple, quand $h(x) = \langle x, Qx \rangle$ où Q est une matrice symétrique définie positive et C est la boule unité de R^n, le problème (P) détermine la plus grande valeur propre de Q. Dans ce cas, l'algorithme LRK consiste à calculer à chaque itération :

$$x^{k+1} = \frac{(I+cQ)x_k}{\|(I+cQ)x_k\|}$$

C'est-à-dire que LRK se comporte comme la méthode des puissances itérées pour le calcul de la plus grande valeur propre.

REFERENCES BIBLIOGRAPHIQUES

[1] G. Auchmuty, Duality algorithms for nonconvex variational principles, Research Report UH/MD-41, University of Houston, 1988.

[2] R. Benacer, Contribution à l'étude des algorithmes de l'optimisation non convexe et non différentiable, Thèse de Doctorat de l'Université de Grenoble, 1986.

[3] H. Brézis, *Opérateurs Maximaux Monotones, Mathematics Studies n°5*, North-Holland, 1973.

[4] R. Ellaia, Contribution à l'analyse et l'optimisation de différences de fonctions convexes, Thèse de 3ème cycle de l'Université Paul Sabatier, Toulouse, 1984.

[5] D. Gabay, Minimizing the difference of two convex functions. Part 1 : Algorithms based on exact regularization, Rapport de Recherche, INRIA, 1982.

[6] J.B. Hiriart-Urruty, From convex optimization to nonconvex optimization. Necessary and sufficient conditions for global optimization, *Nonsmooth Optimization and Related Topics*, Plenum Press, 1989, pp.219-240.

[7] J.B. Hiriart-Urruty, Generalized differentiability, duality and optimization for problems dealing with differences of convex functions, *Convexity and Duality in Optimization, Lectures Notes in Economics and Mathematical Systems 256*, 1986, pp.37-70.

[8] J.B. Hiriart-Urruty, Conditions nécessaires et suffisantes d'optimalité globale en optimisation de différences de deux fonctions convexes, *C. R. Acad. Sci. Paris 309, Série I*, 1989, pp. 459-462.

[9] J.B. Hiriart-Urruty et H. Tuy (Eds.), Essays on Nonconvex Optimization, special issue, *Math. Prog. Series A 41*, 1988.

[10] H. Mine et M. Fukushima, A minimization method for the sum of a convex function and a continuously differentiable function, *J. Optim. Theory and Appl. 33*, 1981, pp. 9-23.

[11] Pham Dinh T. et S. El Bernoussi, "Algorithms for solving a class of nonconvex optimization problems. Methods of subgradient", *Fermat Days 85 Mathematics for Optimization,* 1986.
[12] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
[13] R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control and Optimization*, 1976, pp. 877-898.
[14] J.F. Toland, "Duality in nonconvex optimization", *J. of Math. Anal. and Appl.* 66, 1978, pp.399-415.
[15] H. Tuy, A general deterministic approach to global optimization via dc programming, *Fermat Days 85 : Mathematics for Optimization*, North-Holland, 1986.
[16] W.I. Zangwill, *Nonlinear Programming : A Unified Approach*, Prentice-Hall, 1969.

6 EVOLUTION ET PERSPECTIVES

On a essayé de présenter un ensemble de travaux autour de quelques idées directrices, cela dans le but, non pas de justifier nos lignes de recherche au cours de ces dix dernières années, mais d'en simplifier la description. Certains aspects et même certains travaux ont été omis alors que d'autres ont pris plus de poids qu'ils n'en ont eu véritablement.

Les recherches sur les méthodes de décomposition ont toutefois dominé nos activités et il semble que le regain d'intérêt pour ces techniques, surtout ce qui concerne le calcul parallèle, nous maintiendra dans cette ligne où de nombreuses questions restent encore en suspens. En particulier, nous sommes conscients d'avoir négligé les aspects numériques et les applications de ces méthodes, ayant eu assez peu d'occasions de rencontrer des modèles réels de grande dimension. C'est le cas de la méthode de décomposition mixte présentée au chapitre 4 qui nous semble être notre contribution la plus originale dans ce domaine. Il apparait que les problèmes de multiflots sont un cadre très propice aux expériences numériques qui confirmeront, nous l'espérons, les espoirs placés dans cette méthode. D'autres modèles comme les problèmes en escalier issus de la discrétisation de processus dynamiques ou les modèles économiques de Leontieff doivent être également utilisés pour la validation définitive de cet algorithme. De plus, des critères de convergence facilement exploitables doivent être découverts pour éviter la phase de sous-gradients qui précède l'algorithme à un niveau et les heuristiques de choix des contraintes allouées dans les sous-problèmes.

Les recherches présentées dans la dernière partie sur les méthodes proximales tendent à dépasser le cadre des problèmes soulevés par la décomposition. L'extension au cas non convexe par le biais des fonctions dc devrait permettre de comprendre mieux la question suivante : comment agir sur le paramètre de la méthode proximale pour garantir la convergence dans le cas non monotone ? Cette question rejoint le problème du choix du paramètre dans les méthodes de Lagrangien Augmenté en optimisation non convexe.

REFERENCES

K.J. Arrow et L. Hurwicz, Decentralization and computation in ressource allocation, Essays in Economics and Econometrics, R. Phouts ed.,Chapel Hill, 1960.

W.J. Baumol et T. Fabian, Decomposition, pricing for decentralization and external economies, Man. Sci. 11, 1964, pp. 1-32.

D.P. Bertsekas et J. Tsitsiklis, Parallel and Distributed Computations, Prentice-Hall, 1989.

H. Brézis, Opérateurs Maximaux Monotones, Mathematics Studies n°5, North Holland, 1973.

G. Cohen, Décomposition et coordination en optimisation déterministe, différentiable et non différentiable, Thèse d'Etat, Paris, 1984.

G.B. Dantzig et P. Wolfe, The decomposition algorithm for linear programs, Econometrica, 29, 4, 1960, pp. 767-778.

Y. Dirickx et L.P. Jennergren, System Analysis by Multilevel Methods, John Wiley, 1979.

G. Finke, A. Claus et E. Gunn, A two-commodity network flow approach to the traveling-salesman problem, Congressus Numerantium 41, 1984, pp. 167-178.

E. G. Golstein, The block method of convex programming, Soviet Math. Dokl. 33, 1986, pp. 584-587.

M. Gondrand et M. Minoux, Graphes et Algorithmes, Eyrolles, 1979.

M. Held et R.M. Karp, The traveling-salesman problem and minimum spanning trees, Math. Prog. 1, 1971, pp. 6-25.

L.P. Jennergren, A price-schedules decomposition algorithm in linear programming problems, Econometrica, 41, 1973, pp. 965-980.

J. Kennington et M. Shalaby, An effective subgradient procedure for minimal cost muticommodity flow problems, Man. Sci. 23, 1977, pp. 994-1004.

F. Kydland, Hierarchical decomposition in linear economic models, Man. Sci. 21, 9, 1975, pp. 1020-1039.

C. Lemaréchal, An extension of Davidon's method to nondifferentiable problems, Math. Prog. Study 3, 1975, pp. 95-109.

D. Medhi, Decomposition of structured large-scale optimization problems and parallel optimization, PhD Thesis, University of Wisconsin-Madison, 1987.

M. Minoux, Programmation Mathématique, Dunod, 1983.

M. Minoux et J.Y. Serreault, Subgradient optimization and large-scale programming : an application of optimum multicommodity network synthesis with security constraints, RAIRO 15, 1981, pp. 185-203.

J.J. Moreau, Proximité et dualité dans un espace Hilbertien, Bull. Soc. Math. de France, 93, 1965, pp. 273-299.

G. Pierra, Decomposition through formalization in a product space, Math. Prog. 28, 1984, pp. 96-115.

D. Potier, Algorithmes de coordination - Application à la gestion d'unités de production interdépendantes, Méthodes Numériques d'Analyse des Systèmes, tome 2, Cahiers de l'IRIA n°11, 1972.

R.T. Rockafellar, Monotone operators and the proximal point algorithm in convex programming, SIAM J. Control and Optim. 14, 1976, pp. 877-898.

R.J.B. de Sampaio, Contribuiçao ao estudo da Programaçao DC (Diferença de duas funçoes convexas) , Thèse de Doctorat, Université Catholique de Rio de Janeiro, 1990.

N.Z. Shor, Minimization Methods for Non-Differentiable Functions, Springer-Verlag, 1985.

J.E. Spingarn, Submonotone mappings and the proximal point algorithm, Numer. Funct. Anal. and Optim. 4, 2, 1981, pp.123-150.

J.E. Spingarn, Partial inverse of a monotone operator, Appl. Math. and Optim. 10, 1983, pp. 247-265.

A. Titli, Commande Hiérarchisée et Optimisation des Processus Complexes, Dunod, 1975.

Les travaux présentés pour cette habilitation ont été réalisés au Laboratoire d'Automatique et d'Analyse des Systèmes de Toulouse (1975-1978), au Département de Génie Electrique de l'Université Catholique de Rio de Janeiro (1978-1988) et au Laboratoire ARTEMIS de l'IMAG à Grenoble (1988-1990). J'exprime toute ma reconnaissance aux directeurs respectifs de ces institutions ainsi que globalement à tous ceux qui ont de près ou de loin contribué à mes activités académiques et de recherches au cours de ces années et, plus particulièrement, à MM. G. Giralt et F. Roubellat du LAAS, MM. C. Kubrusly et C. Ribeiro de la PUC et Mr. J. Fonlupt d'ARTEMIS.

Je suis très honoré par la présence à la présidence de ce jury du Professeur P.J. Laurent de l'INPG et del'IMAG et je le remercie pour m'avoir consacré un peu de son temps.

Que Messieurs J. Fonlupt, Professeur à l'INPG et Directeur du Laboratoire ARTEMIS, A. Titli, Professeur à l'INSA de Toulouse et P. Tolla, Professeur à l'Université Paris-Dauphine, soient également remerciés d'avoir bien voulu être les rapporteurs de cette habilitation.

Je remercie chaleureusement MM. B. Lemaire, Professeur à l'Université de Montpellier, M. Minoux, Directeur de Recherches au MASI, et Nguyen Van Hien, Professeur et Directeur du Département de Mathématique des Facultés Universitaires de Namur, pour leur participation à ce jury mais surtout pour leur amitié et leurs contributions actives à mes travaux de recherche qui, je l'espère, se poursuivront dans l'avenir.

Finalement, j'aimerais associer dans ma reconnaissance les participations décisives de MM. G. Cohen, J.B. Lasserre, Pham Dinh Tao et R.J.B. de Sampaio.