

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE



PROJET DE FIN D'ETUDES

S U J E T

**Gestion de la Production
Assistee Par
Ordinateur**

Proposé par :
F. CHIGARA

Etudié par :
L. Y. LASSEL

Dirigé par :
F. CHIGARA

PROMOTION **JANVIER 88**

وزارة التعليم والبحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

S U J E T

Gestion de la Production Assistee Par Ordinateur

Proposé par :
F. CHIGARA

Etudié par :
L. Y. LASSEL

Dirigé par :
F. CHIGARA

PROMOTION : JANVIER 88

Ministère de l'enseignement supérieur

Ecole Nationale Polytechnique

Département : ELECTRONIQUE

Promoteur : F. CHIGARA

Elève Ingénieur : LASSEL Lidia Yasmina



Separator line of dots

الموضوع: تسيير الإنتاج اعتماداً على العقل الإلكتروني.
الغاية: الهدف من عملنا هو دراسة حوار زمنيين وهذا التحليل الأجهزة الكبيرة بطريقة
التحليل والتنسيق وتطبيقها لتسيير الإنتاج بالاعتماد على العقل الإلكتروني
الحوار زمنية الخوف، حوار زمنية دانتزيق- وولف. نحن أيجاد حل للمب للحارة
غير الخطية، أما الحوار زمنية الثانية- حوار زمنية أوزاوا- تستعمل طريقة البحث
عن نقطة التوازن.
وعند تحليل الأجهزة الجزئية من المسألة تؤدي إلى الحل الكلي الأعظم.

Separator line of dots

Sujet: Gestion de la Production Assistée par Ordinateur

Resume: L'objectif de ce travail est d'étudier deux algorithmes
de resolution de grands systemes par des methodes de de-
composition-coordination appliquee a la G.P.A.O.
Le premier algorithme, celui de DANTZIG-WOLFE linearise
un probleme convexe avant d'obtenir sa solution optimale.
Le second, l'algorithme d'UZAWA utilise la methode de
recherche de point-selle.
La resolution du probleme se fait en le decomposant en
sous-systemes. La coordination des solutions partielles
conduit a la solution globale optimale.

Separator line of dots

Subject: Management of Controlled Production through Computer

Synthesis: The objective of such a work is to study two al-
gorithms in order to understand to great systems
thanks to decomposition-coordination applied to
G.P.A.O.
The first algorithm, the one of DANTZIG-WOLFE studies
a convex problem before obtaining its optimal solu-
tion. The second, UZAWA 's algorithm uses a method
of research of the balance point.
The solution of the problem can be done through the
its decomposition in sub systems. the coordination
of partial solutions lead to the optimal global
solution.

Separator line of dots

Je tiens à offrir ce modeste travail

A mes parents, pour leur amour et leurs sacrifices
et pour m'avoir toujours aidé dans mes projets,

A mon fiancé, pour toute l'aide et les encouragements
qu'il n'a cessé de me donner,

A mes frères qui partagent aujourd'hui ma joie .

Et à toutes les personnes qui me sont chères en
particulier ma grand-mère.

Je remercie très vivement mon promoteur Monsieur F. CHIGARA, enseignant à l'Ecole Nationale Polytechnique, pour avoir proposé le sujet et m'avoir suivi et dirigé durant tout le semestre, et, pour avoir si généreusement enrichi mes connaissances.

Je tiens à remercier également:

M. AIT CHEIKH, enseignant au Département d'Electronique, pour l'intérêt qu'il a porté à mon travail.

M. SALHI, enseignant au Département de Génie Industriel, pour l'aide précieuse qu'il m'a apportée.

Ainsi que M. EL KOLLI, enseignant à l'Institut de Mathématique de BAB EL ZZOUAR et M. A. MIGNON.

Mes remerciements vont également à Mlle. HAMMOUTENE de l'INPG qui m'a fourni une riche documentation concernant la gestion de la production. Ainsi qu'à Mlle. NESSIRA G. pour avoir dactylographié mon travail.

Enfin, que tous les professeurs qui ont contribué à ma formation d'Ingénieur trouvent ici l'expression de ma profonde reconnaissance.

INTRODUCTION

CHAPITRE I: NOTION DE GRANDS SYSTEMES

I-1- Notion de système

I-2- Notion de grand système

CHAPITRE II: ANALYSE DES GRANDS SYSTEMES

II-1- Généralités

II-2- Agrégation

II-2-1- Réduction par agrégation

II-2-2- Structure du système agrégé

II-3- Méthodes de décomposition-coordination

II-3-1- Décomposition statique de grand système

II-3-1-1- Méthode de coordination du modèle

II-3-1-2- Méthode de coordination du critère

II-3-1-3- Condition d'application de la méthode du modèle

II-3-2- Décomposition dynamique de grand système

CHAPITRE III: PROBLEME DE COORDINATION

III-1- Généralités

III-2- Algorithmes courants

III-2-1- Méthode de programmation linéaire

III-2-2- Méthode de programmation non-linéaire

III-2-2-1- Méthode de coordination du but

III-2-2-2- Méthode de coordination du modèle

III-2-2-3- Méthode mixte

III-3- Algorithmes étudiés

III-3-1- Méthode de DANTZIG-WOLFE

III-3-1-1- Algorithme de DANTZIG-WOLFE

III-3-1-2- Résolution

III-3-1-3- Organigrammes

III-3-2- Méthode d'UZAWA

III-3-2-1- Généralités

III-3-2-2- Principe Général

III-3-2-3- Algorithme

III-3-2-4- Organigrammes

III-4- Synthèse

CHAPITRE IV: APPLICATION

IV-1- Généralités

IV-2- Exemple d'application

IV-3- Résultats

CHAPITRE V: CONCLUSION

BIBLIOGRAPHIE

ANNEXES :

- 1- Méthode du simplexe
- 2- Choix de la période d'échantillonnage
- 3- Démonstration du théorème (1)
- 4- Démonstration du théorème (2)
- 5- Démonstration du théorème (3)
- 6- Listing

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

- ° - INTRODUCTION - ° -

L'introduction de l'informatique dans les différentes fonctions de la production a commencé au début des années 60, et s'est développée beaucoup plus rapidement vers les années 80, période où l'environnement technique, économique, et social des entreprises industrielles est devenu plus complexe.

L'objectif de toute Entreprise de production confrontée à la concurrence est de maîtriser au mieux les trois contraintes principales (14).

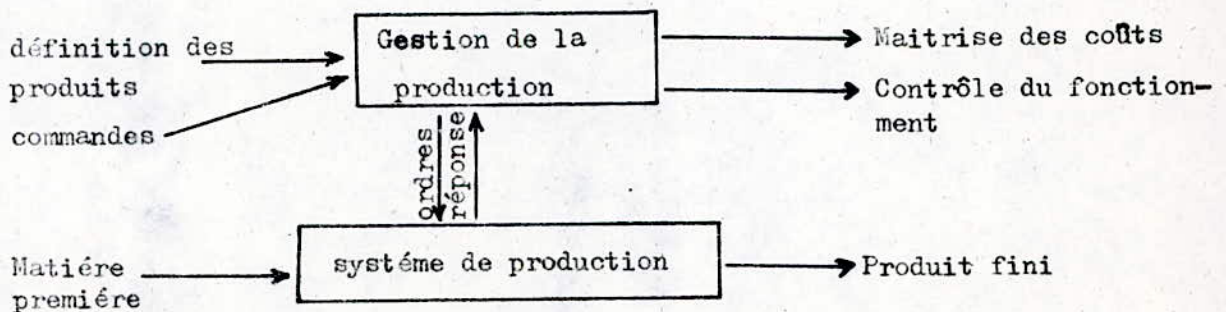
- Le moindre coût,
- le délais,
- la qualité

Pour mener à bien ces prestations, une bonne gestion de la production est nécessaire.

La G.P.A.O (Gestion de la production assistée par ordination) est l'outil moderne utilisé, ses avantages en particulier la commodité, la rapidité, l'efficacité, le minimum d'erreur, la standardisation et l'économie sont les plus performants.

Son but est :

- de donner les directives au système de production en tenant compte des commandes et de la définition des produits
- de contrôler le fonctionnement du système
- de maîtriser les coûts



Dans notre travail nous nous sommes intéressé à des systèmes industriels complexes dont la résolution relève de la théorie des grands systèmes.

Dans la première partie nous définissons ce qu'est un grand système et nous présentons les différentes méthodes d'analyse. Dans la deuxième partie, les algorithmes de décomposition - coordination courants sont étudiés et nous nous intéressons de manière particulière aux algorithmes de :

DANTZIG - WOLFE qui est un algorithme de programmation linéaire basé sur la méthode du simplexe et l'algorithme d'UZAWA basé sur la recherche de point selle. L'idée générale est de décomposer le problème global en sous systèmes et de coordonner la solution de ses problèmes locaux pour obtenir une solution globale optimale.

Comme illustration, nous choisissons un cas particulier de la G.P.A.O; la gestion des stocks pour une production optimale. Celle-ci est particulièrement intéressante vu le nombre important d'applications pratiques.

Dans cette application nous constatons que la résolution du problème nécessite un grand nombre de calculations. Pour palier à cette difficulté la solution de ce grand système est approchée par une décomposition horizontale il en résulte des sous systèmes et la coordination de leurs solutions permet d'avoir la solution optimale du système initial.

Nous résolvons ce système par les deux algorithmes étudiés, une synthèse permet de conclure quant aux possibilités d'utiliser l'un ou l'autre des deux algorithmes.

CHAPITRE I

-°- NOTION DE GRANDS SYSTEMES -°-

=====

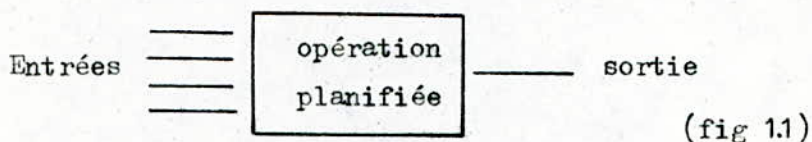
I - 1 NOTION DE SYSTEME

Un système est un ensemble d'éléments dont on a défini les inter - relations de type physique ou économique et surtout les finalités. Ces éléments sont situés par rapport à une frontière qui constitue la frontière du système. (12)

Le système a des objectifs, un environnement, il peut être concret ou abstrait, ouvert ou fermé, statique ou dynamique.

Tout système possède en général plusieurs entrées et plusieurs sorties. Les entrées peuvent être des commandes ou des perturbations. Donc pour pouvoir agir sur un système il est nécessaire de bien connaître son environnement et de bien le définir.

Le système le plus élémentaire avec ces entrées, son opération planifiée et sa sortie peut être représenté comme suit :



Avec le progrès des mathématiques surtout des probabilités et des statistiques la notion de système s'est imposée dans le domaine économique. Mais cette notion reste relative et donc difficile à définir avec exactitude et précision.

Les performances souhaitées pour un système économique sont définies par un critère (très souvent une relation quantité - qualité) . Il s'agit alors de trouver le meilleur réglage des paramètres constituant le système.

De ce fait l'étude du critère et sa mise en forme mathématique sont d'une grande importance car c'est de ce critère que dépend l'optimisation du système.

I - 2 NOTION DE GRAND - SYSTEME

Un grand système est constitué d'un ensemble de petits systèmes dont il diffère quantitativement et qualitativement par :

- une conception d'ensemble
- une grande dimension et un grand nombre de fonction à remplir
- complexité

On distingue généralement deux types de grands systèmes :

1 - Grands systèmes déterministes :

Ils sont caractérisés par des grandeurs qui varient de façon continue conformément à des lois parfaitement déterministes.

Pour l'analyse de ce type de systèmes on emploie les méthodes analytiques classiques.

2 - Grands systèmes stochastiques :

La variation des ~~des~~ grandeurs se fait d'une manière désordonnée et aléatoire.

La description de ces systèmes est faite à l'aide des méthodes probabilités.

Les grands - systèmes sont rencontrés non seulement dans l'industrie (système de production, etc ...) mais aussi dans le domaine socio-économique (transport et distribution, système d'énergie, etc ...).

Ces systèmes posent de grandes difficultés du point de vue de l'analyse (décomposition agrégation) et du contrôle.

Il n'existe pas une méthode générale d'identification et d'optimisation. Chaque système doit être considéré d'une manière spécifique.

CHAPITRE II

-°- ANALYSE DES GRANDS SYSTEMES -°-

II - 1 GENERALITES

Le comportement d'un système peut être représenté par un ensemble de relations mathématiques constituant le modèle mathématique de ce système.

Ce modèle doit refléter fidèlement les contraintes et l'objectif du système réel.

La formulation mathématique du modèle peut se faire de différentes façons selon le type du système :

- a) par des relations aux équations algébriques (processus statique)
- b) par des relations intégral-différentielles (système dynamique)
- c) par des équations aux dérivées partielles (système à paramètre distribués)
- d) par des équations aux différences (système à temps discret).

La relation entre la formulation du problème et les méthodes utilisées pour sa résolution (10) est représentée dans le tableau suivant :

REPRESENTATION DU PROBLEME	METHODE DE RESOLUTION
1 - <u>Processus statique</u>	
1.1 - Problèmes sans contraintes : max $J = J(x)$ (critère, x vecteur)	Méthode courante de recherche de l'optimum
1.2 - Problèmes dont la contrainte est sous forme d'équation : max $J = J(x)$ (critère) avec $g(x) = 0$ (contrainte)	Méthode du multiplicateur de Lagrange.
1.3 - Problèmes dont la contrainte est sous forme d'inéquation : $J = J(x)$ (critère) $h(x) \leq 0$ (contrainte)	Méthode du multiplicateur de KUHN - TUCKER
1.4 - Problèmes linéaires : $J = \sum_i C_i x_i \quad i = 1, 2, \dots, n$ avec $\sum_j a_{ij} x_i = b_j \quad j = 1, 2, \dots, m$ $x_i \geq 0 \quad i = 1, 2, \dots, n$	Méthode du simplexe
1.5 - Problèmes non linéaires $J = J(x)$ avec $g_j(x) (\leq = \geq) b_j \quad j = 1, 2, \dots, m$	programmation non linéaire en particulier la théorie de KUHN - TUCKER

REPRESENTATION DU PROBLEME	METHODE DE RESOLUTION
2 - <u>Systèmes Dynamiques</u> :	
2.1 - Problèmes linéaires	- Méthode variationnelle. - programmation dynamique - méthode du gradient.
J = J(x,u,t) avec $\dot{\phi} = \dot{x} - A(t)x - B(t).u = 0$	
2.2 - Problèmes non linéaires	Idem
J = J(x,u,t) avec $\dot{\phi} = \dot{x} - g(x,u,t) = 0$ J,g non linéaire	+ -quasilinearisation
3 - <u>Systèmes dynamique ou système statique complexe</u>	décomposition - coordination contrôle hierarchique et optimisation.

Mais lorsque le système à analyser est multidimensionnel il est nécessaire de le représenter par un modèle réduit pour les raisons suivantes :

- Difficultés d'ordre numérique qui apparaissent dès que l'ordre du système est suffisamment élevé.
- L'analyse ou la synthèse d'un système complexe à partir d'un modèle de dimension réduite est plus aisée, donc plus facile à réaliser.
- Le contrôle en temps réel est plus facile pour 1 modèle de petite dimension.

Dans ces conditions l'utilisations des modèles réduits permet l'extension du domaine d'application des techniques d'automatique (commande, filtrage, etc ...) qui fournissent des solutions sous optimales pour l'étude des systèmes complexes.

Dans ce qui suit nous exposerons quelques méthodes de résolution des grands systèmes multidimensionnels.

II - 2 AGREGATION

Soit un système décrit par un modèle linéaire M d'ordre N .

Il faut trouver un modèle \hat{M} d'ordre $m < n$ qui réalise une bonne approximation du modèle réel. Le modèle réduit doit être le plus simple possible et de faible dimensions et posséder les caractéristiques suivante :

- posséder les mêmes gains statiques entrée - sortie que le système réel.
- donner une bonne approximation de début du transitoire
- posséder les mêmes caractéristiques de stabilité que le système réel.
- fournir une bonne approximation des sorties mesurables du système et une bonne approche de la réponse en fréquence sur une gamme donnée.
- la technique de réduction doit être applicable à tous les systèmes linéaires.

II - 2 - 1 REDUCTION PAR AGREGATION

Puisque une relation explicite entre le modèle réel et son modèle réduit n'existe pas ce qui ne permet pas de prévoir rigoureusement les résultats du système réel à partir de son modèle, A. AOKI introduit en 1968 la méthode dite " Agrégation linéaire ". (10) Cette méthode s'applique aux systèmes dynamiques linéaire, elle consiste à imposer une relation linéaire entre le système réel et le modèle réduit de la forme :

$$Z(t) = L. X (t)$$

On obtient ainsi ce qu'on appelle " Un modèle agrégé ".

Soit le système suivant considéré complètement commandable et observable :

$$(2.1) \begin{cases} \dot{X}(t) = A X(t) + B U(t) \\ Y(t) = C X(t) + D U(t) \end{cases} \quad \begin{matrix} X \in \mathbb{R}^n \\ U \in \mathbb{R}^r, Y \in \mathbb{R}^p \end{matrix}$$

Le but est de déterminer le modèle réduit

$$(2.2) \begin{cases} \dot{Z}(t) = F Z(t) + G U(t) \\ \hat{Y}(t) = H Z(t) + K U(t) \end{cases} \quad \begin{matrix} Z \in \mathbb{R}^m \\ \hat{Y} \in \mathbb{R}^p \end{matrix} \quad m \ll n$$

si $Z(t)$ et $X(t)$ vérifient la relation linéaire d'agrégation

$$Z(t) = L X(t)$$

L : matrice de dimension $m \times n$

on aura

$$(2.3) \begin{cases} L \dot{X}(t) = F.L X(t) + G.U(t) \\ \hat{Y}(t) = H.L X(t) + K.U(t) \end{cases}$$

On en conclue qu'une relation d'agrégation est donnée si et seulement si

$$\begin{cases} FL = LA \\ G = LB \\ Z(0) = L X(0) \end{cases}$$

Ces trois relations constituent la condition nécessaire et suffisante d'agrégation. Si elles sont vérifiées on dira que le modèle réduit est un modèle agrégé du système (2.1) par la matrice L .

II - 2.2. STRUCTURE DU MODELE AGREGE

Pour établir la forme générale d'un modèle agrégé on procède de la façon suivante :

On fait le changement de base suivant :

$$X = T.V \quad T = \text{matrice de passage}$$

Le système (2.1) devient :

$$\begin{aligned} \dot{V}(t) &= T^{-1} A T V(t) + T^{-1} B U(t) \\ &= \beta V(t) + \Gamma U(t) \end{aligned}$$

Avec

$$\begin{aligned} \beta &= T^{-1} A T \quad \text{et} \quad \Gamma = T^{-1} B \\ \beta &= T^{-1} A T = \begin{bmatrix} \beta_1 & & & 0 \\ & \dots & & \\ 0 & & & \beta_2 \end{bmatrix} \end{aligned}$$

C'est la matrice diagonale de A sous forme de JORDAN car elle est constituée des vecteurs propres de A à 1 coefficient près.

$$\text{et} \quad \Gamma = T^{-1} B = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}$$

$$V = (V_1, V_2)^T$$

La forme diagonale du modèle agrégé retenant m modes du système réel est donnée par :

$$\dot{V}_1(t) = \beta_1 V_1(t) + \Gamma_1 U(t)$$

et la classe des modèles agrégés représentés par le système (2.2) est définie par une transformation régulière.

$$Z(t) = M V_1(t)$$

M : matrice quelconque de dimension (m x m) et de rang plein. La forme la plus générale d'un modèle agrégé est donnée par :

$$\dot{Z}(t) = M \beta_1 M^{-1} Z(t) + M \Gamma_1 U(t)$$

par analogie avec le système (2.2) on trouve :

$$F = M \beta_1 M^{-1} \quad \text{et} \quad G = M \Gamma_1$$

La relation d'agrégation correspondante est :

$$Z(t) = M V_1(t) = M \begin{pmatrix} I_m & \\ & 0 \end{pmatrix} V(t) = M \begin{pmatrix} I_m & \\ & 0 \end{pmatrix} T^{-1} X(t)$$

On pose :

$$L_0 = \begin{pmatrix} I_m & \\ & 0 \end{pmatrix} T^{-1}$$

On aura alors : $Z(t) = M L_0 X(t) = L X(t)$

* Remarque :

Il existe un grand nombre de méthodes d'agrégation.

exemple , les méthodes de DAVISON, CHIDAMBARA, FOSSAD, etc ... (10)

II - 3. METHODE DE DECOMPOSITION - COORDINATION

Les méthodes de décomposition consistent à diviser le grand système en plusieurs sous - systèmes invariants, de trouver la solution à ces sous systèmes et de coordonner toutes ces solutions afin de trouver la solution globale du système.

Une première méthode consiste à partir du modèle de processus invariant dans le temps et le subdiviser en plusieurs sous processus, c'est ce qu'on appelle décomposition statique du grand système.

Une autre méthode se fait en partant des observations du changement d'état d'un grand système, celle - ci se caractérise par la décomposition du système en sous-systèmes représentés par des vecteurs d'état convergent à la solution globale du système initiale , c'est ce qu'on nomme décomposition dynamique du grand système.

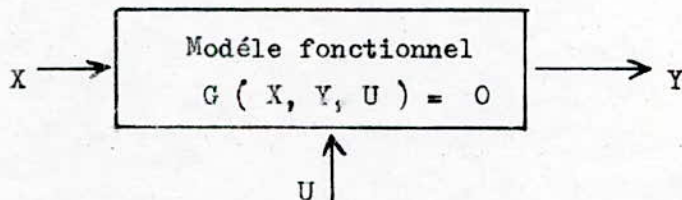
II - 3.1 . DECOMPOSITION STATIQUE DU GRAND SYSTEME

La décentralisation du problème d'optimisation statique est développée en deux étapes :

1°) Le problème dans son ensemble (fonction objective et contraintes) est transformé en une forme de deux ou plusieurs niveaux avec des objectifs séparés et distincts pour chaque niveau.

2°) Les parties de l'objectif du premier niveau ou problèmes n'ayant pas de relation avec d'autres parties sont mises à part formant ainsi une décomposition du problème du premier niveau.

II - 3.1.1. METHODE DE COORDINATION DU MODELE



X = Vecteur d'entrée

Y = Vecteur de sortie

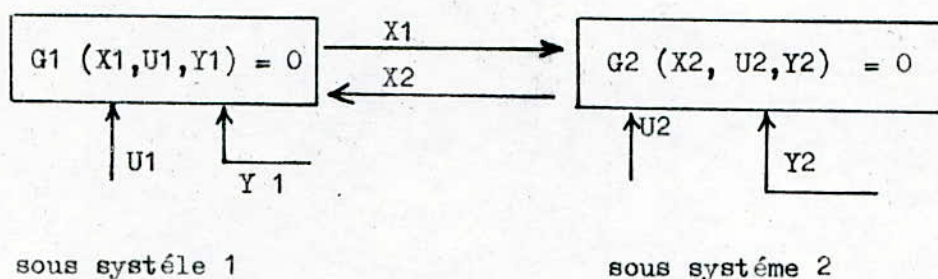
U = Vecteur de commande

Ce système est caractérisé par le modèle mathématique :

- processus $G(\vec{X}, \vec{U}, \vec{Y}) = 0$ (2.4)
 - critère $P(\vec{X}, \vec{U}, \vec{Y}) \rightarrow \min$ (2.5)

Dans la décomposition on suppose que le système est formé de plusieurs sous-systèmes à plusieurs niveaux de commande, qu'il y a une interaction entre les entrées et les sorties intermédiaires de ces sous systèmes et que l'objectif est assuré par la réunion des objectifs imposés par les sous-systèmes.

On a la représentation suivante d'une commande à deux niveaux :



- X 1 = vecteur d'interaction du sous système 1 avec le sous système 2
- X 2 = " " " " 2 " " " 1
- U 1 = vecteur de commande local du sous système 1
- U 2 = " " " " " 2
- Y 1 = vecteur de sortie du système globale et du sous système 1
- Y 2 = " " " " " " 2

Les équations (2.4) et(2.5) deviennent :

Processus : $G1(U1, Y1, X1, X2) = 0$ et $G2(U2, Y2, X1, X2) = 0$ (2.6)
 critère : $P(U, X, Y) = P1(\vec{U}1, \vec{X}1, \vec{Y}1) + P2(\vec{U}2, \vec{X}2, \vec{Y}2) \rightarrow \min$ (2.7)

l'équation (2.7) représente la façon dont on optimise le système globale. L'optimisation se fait en plusieurs niveau, dans notre cas (n=2) on convertit le problème intégral d'optimisation en un problème à deux niveaux. C'est à dire en fixant le vecteur d'interaction entre les deux systèmes tel que :

$$X = Z$$

Ainsi le problème intégral peut être séparé en deux problèmes :

a) - Problème du 1er niveau :

déterminer : $H(Z) = \min_{\vec{U}, \vec{Y}} P(U, Z, Y)$
 avec $G(U, Y, Z) = 0$

b) Problème du second niveau

Il consiste à chercher : $\min_{\vec{Z}} H(\vec{Z})$

Dans la première étape l'ensemble des solutions est :

$$S_1 = \{ (m, y) / G(m, y, z) = 0 \}$$

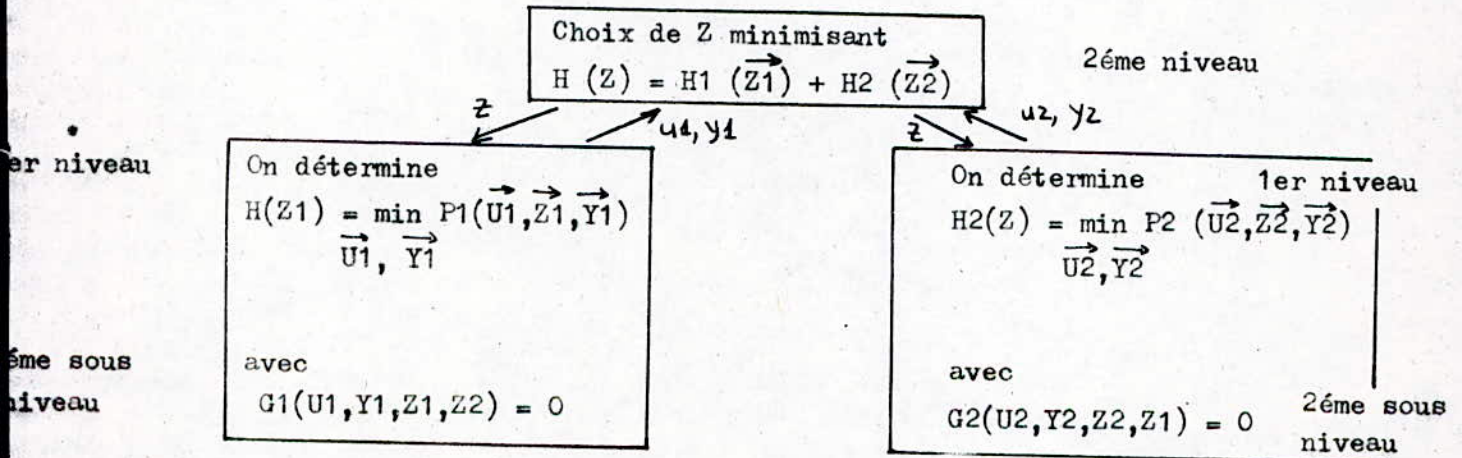
Dans la seconde étape l'ensemble est :

$$S_2 = \{ Z / H(Z) \text{ existe} \}$$

→ $Z = X_{opt}$ avec X_{opt} = valeur optimale de X

Les valeurs optimales de m et y correspondant à X_{opt} sont déterminées à partir de S_1 .

Pour un modèle de deux sous systèmes l'organigramme de décision est représenté comme suit :

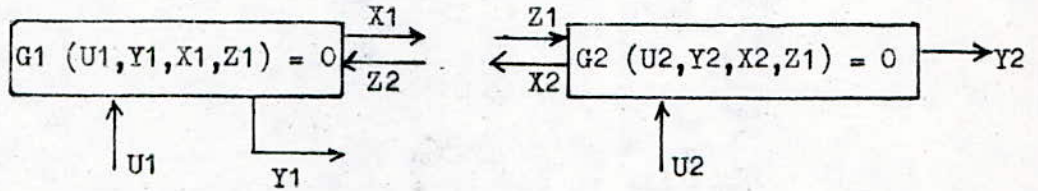


SIMULATION MULTINIVEAU AVEC LA
METHODE DE COORDINATION DU MODELE

II - 3 . 1 . 2 . METHODE DE COORDINATION DU CRITERE (11)

Considérons de nouveau les deux sous systèmes couplés et le problème d'optimisation défini au paragraphe précédent.

La méthode de coordination du critère annonce qu'un couple ne peut être séparé en deux parties et résolu séparément si l'on veut obtenir l'optimisation globale. En conséquence il est nécessaire de rompre les relations entre les sous systèmes



Systeme decouplé

- X1 : vecteur des sorties intermédiaires du sous système 1
- X2 : " " " " " " 2
- Z1 : vecteur des entrées intermédiaires du sous système 2 dues au sous système 1
- Z2 : " " " " " " 1 " " " 2
- Y1 : vecteur de sortie observable du système globale et du sous système 1
- Y2 : " " " " " " " " 2
- U1 : Vecteur de commande locale du sous système 1
- U2 : " " " " " " 2

L'optimisation consiste à déterminer un vecteur de coefficient de pénalisation λ (ou paramètre de coordination) de façon qu'il y ait une égalité entre X et Z

Soit $X = Z$

On exprime cette pénalité par le critère suivant :

$$P(\vec{U}, \vec{X}, \vec{Y}, \vec{Z},) = P1(\vec{U1}, \vec{X1}, \vec{Y1}) + P2(\vec{U2}, \vec{X2}, \vec{Y2}) + \lambda'(X - Z)$$

avec $G1(\vec{U1}, \vec{X1}, \vec{Y1}, \vec{Z1}) = 0$
 $G2(\vec{U2}, \vec{X2}, \vec{Y2}, \vec{Z2}) = 0$

Il faut définir l'ensemble des solutions S1 tel que

$$S1 = \left\{ (U, X, Z, Y,) / G1 = G2 = 0 \right\}$$

et déterminer $H(\lambda) = \min P(U, X, Y, Z, \lambda)$

Fonction objective (avec coefficient de pénalisation) à minimiser sur l'ensemble des variables admises.

Puis on définit $S_2 = \{ \lambda / H(\lambda) \text{ existe} \}$

Pour tout calcul on développe le terme de pénalisation en :

$$\lambda'(X - Z) = \lambda_1 (X_1 - Z_1) + \lambda_2 (X_2 - Z_2)$$

Alors le problème du premier niveau se sépare en :

a) Sous système 1 :

$$\begin{aligned} \min \quad & P_1 (U_1, Y_1, X_1, Z_2) + \lambda'_1 X_1 - \lambda_2 Z_2 \\ & U_1, X_1, Y_1, Z_1, \end{aligned}$$

avec $G_1 (U_1, X_1, Y_1, Z_2) = 0$

b) sous système 2 :

$$\begin{aligned} \min \quad & P_2 (U_2, Y_2, X_2, Z_1) - \lambda_1 Z_1 + \lambda'_2 X_2 \\ & U_2, Y_2, X_2, Z_1 \end{aligned}$$

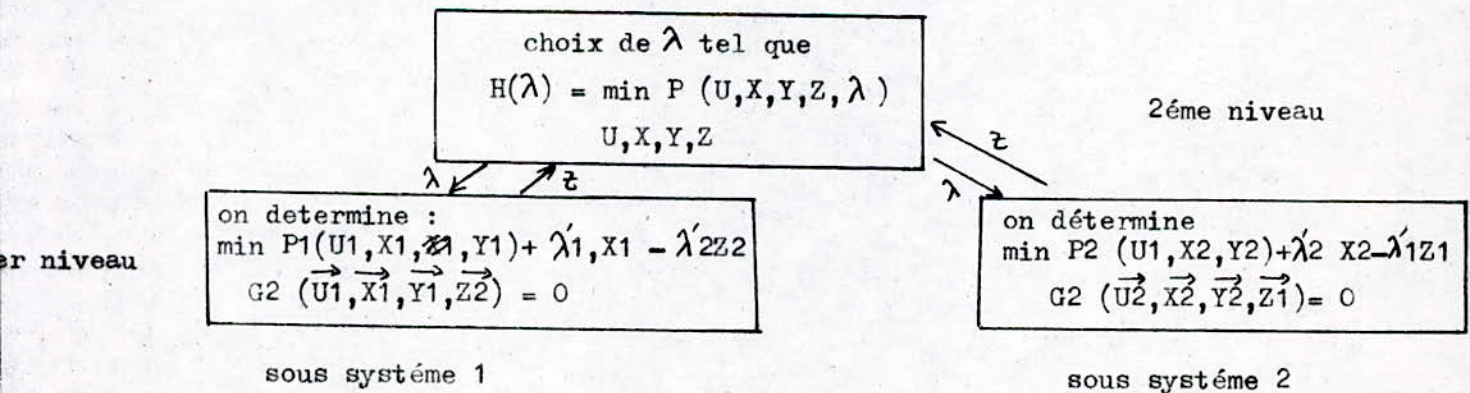
avec $G_2 (U_2, X_2, Y_2, Z_1) = 0$

* Remarques :

L'objectif de chaque sous système a été modifié du fait que la variable de coordination X et Z entre dans l'objectif de chaque sous système.

$\lambda \in \mathbb{R}$ s'appelle " multiplicateur de lagange "

L'organigramme des décisions de cette méthode est le suivant : (10)



III 3.1.3. CONDITION D'APPLICATION DE LA METHODE DU MODELE

On note :

m_{ui} = nombre de composants de la commande

m_{x2i} = nombre de composants de la sortie intermédiaire

On compare ces deux grandeurs : (13)

- si $m_{ui} < m_{x2i}$

\Rightarrow l'équation $\frac{dl}{d\mu_i} = T_i - Z_i = 0$ est un système à m_{xi} équation non linéaire et m_{ui} inconnues.

Dans ce cas là le système d'équation n'a pas de solution et la méthode n'est pas valable car ce n'est pas un problème d'optimisation.

- si $m_{ui} = m_{xi}$

L'équation $\frac{dl}{d\mu_i} = 0$ est un système à équation linéaire qui admet :

- 1) zéro solution
- 2) une solution
- 3) plusieurs solutions

Pour que le système soit résolu, il faut que ces équations aient au moins une solution dans U_i pour Z_i et X_i données.

Si cette condition est réalisée, c'est à dire que l'on a un problème d'optimisation non global (le i ème sous système n'est pas optimisable) la méthode du modèle ne peut pas être appliquée.

- si $m_{x2i} < m_{ui}$, on est dans le cas d'un problème d'optimisation car on a $(m_{x2i} + m_{xi})$ contraintes d'égalité et $(m_{ui} + m_{x2i})$ variables indépendantes

C'est dans ce cas là seulement, on peut utiliser la méthode du modèle.

Par contre la méthode du critère peut être utilisée quelque soit le nombres des variables.

Après avoir vu les différentes méthodes de décomposition statique. On peut dire que la méthode de coordination du critère est utilisée pour traiter les problèmes non linéaires à couplage par variable d'état, ceci pour sa grand simplicité (aucune condition d'utilisation) Par contre la méthode du modèle n'est applicable que si le nombre de composant de la commande est supérieur au nombre de composant des sorties intermédiaire. On en conclue que la méthode du critère à un plus large domaine d'application par rapport à la méthode du modèle.

Interprétation économique : la méthode du modèle est utilisée en planification économique et la méthode du critère en économie de l'Entreprise.

II . 3 . 2 DECOMPOSITION DYNAMIQUE DU GRAND SYSTEME

En dynamique on ne peut plus considérer le système comme invariant, car son état change au cours de l'observation dans le temps. L'analyse dynamique consiste à décomposer le système en plusieurs sous - systèmes couplés entre eux et ayant chacun des performances fonctionnelles et des contraintes . La solution s'obtient en deux étapes :

- 1) - Solution du sous -système posé à chaque sous système
- 2) - Coordination des solutions pour trouver la solution globale.

HYPOTHESE

Soit l'instant d'observation $t \in (t_0, t_i)$ et une hypersurface quelconque définie par :

$$h(\lambda_i, t) = 0 \quad \text{pour } t \in (t_0, t_i)$$

$$i = 1, \dots, n$$

$X_i(t)$: variable d'état continue par morceau pour que la commande soit admissible.

On définit alors les variables suivantes :

- $U_i(t)$: variable de commande
- $Z_i(t)$: variable d'entrée
- $Y_i(t)$: variable de sortie observable

Chaque sous système est caractérisé par :

- 1 - les équations d'état : $X_i(t) = F_i(X_i, U_i, Z_i, t)$
 $t \in (t_0, t_i)$ et $i = 1, \dots, n$
- 2 - Les conditions initiales : $X_{i0}(t_0) = X_{i0}$
- 3 - Les contraintes d'inégalités : $R_i(X_i, U_i, Z_i, t) \geq 0$
- 4 - couplage entre les sous systèmes: $Z_i(t) = \sum_{j=1}^n L_{ij} Y_j$

Ou

$[L_{ij}]$ est la matrice de couplage ou de permutation entre les sous systèmes et $Y_j = G_j(X_j, U_j, t)$ sortie observable ou pas.

.../...

En plus des contraintes énoncées dans 1, 2, 3, 4, on peut ajouter le critère principal, du système qui est une fonction objective telle que :

$$J_i (U_i, Z_i) = G_i (X_i, Z_i) + \int_{t_0}^{t_i} f_i (X_i, U_i, Z_i, t) dt$$

L'optimisation de chaque critère $J_i (U_i, Z_i)$ relatif au sous système i ($i = 1, 2, \dots, n$) doit aboutir à un compromis qui permet d'optimiser le grand système global.

L'admissibilité de la commande de chaque sous système nous permet d'écrire pour le système entier :

$$\begin{aligned} (X(t) = F (X, U, Z, t) & \quad \text{avec} \quad F = (F_1, F_2, \dots, F_n) \\ (R(U, Z, X, t) \geq 0 & \quad \text{avec} \quad R = (R_1, R_2, \dots, R_n) \\ (Z = LY & \quad \text{avec} \quad Y = G (X, U, Z) \\ & \quad L = L_{ij} \\ & \quad G = (G_1, G_2, \dots, G_n) \end{aligned}$$

S serait définie par $h(x, t) = 0$ pour $t \in (t_0, t_i)$
 $h = (h_1, h_2, \dots, h_n)$

La fonction objective est toujours la même :

$$J_i (U, Z) = g(x, t) + \int_{t_0}^{t_i} f(x, u, z, t) dt$$

$$\text{avec } g(x, t) = \sum_{i=1}^n g_i(x_i, t)$$

$$\text{et } f(x, u, z, t) = \sum_{i=1}^n f_i(x_i, u_i, z_i, t) dt$$

Les méthodes d'optimisation du critère sont nombreuses.

Parmi celles-ci on peut citer :

- La méthode variationnelle, suivant que les équations intégrales différentielles soient de Lagrange, d'Hamilton ou d'Euler.
- la méthode duale
- la méthode des sauts et des points singuliers
- la méthode de transition d'état qui est une méthode moderne.

C H A P I T R E I I I

- ° - P R O B L E M E D E C O O R D I N A T I O N - ° -



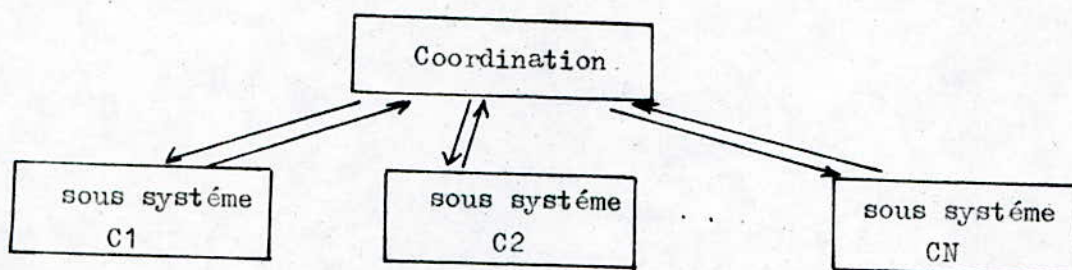
III . 1 . GENERALITE SUR LA COORDINATION

La coordination correspond à une motivation différente de celle de la décomposition, considérons n centre de décision (éventuellement $n + 1$ pour tenir compte du centre $n^{\circ} = 0$) et supposons que l'on ait à définir un problème global (soit un problème d'optimisation avec fonction d'utilité globale, soit un problème d'équilibre non coopératif, etc ...) (1).

La coordination consiste en l'éventuelle résolution de ce problème global en laissant les centres de décisions résoudre des problèmes locaux (ne faisant intervenir que les décisions individuelles, et en particulier sans contraintes globales) que l'on combinera pour donner une solution globale.

Il s'agit là d'un problème classique de l'économie, qui se pose en plus dans un contexte Management (étude de grandes Entreprises en particuliers).

Nous pouvons représenter la notion de décomposition - coordination par la figure suivante :



Ainsi un gestionnaire confronté à un système complexe ou apparaissent plusieurs sous systèmes, doit faire face à un problème de coordination entre les sous systèmes.

Les premiers résultats de la théorie mathématique de la coordinabilité ne concernent encore que des systèmes où l'objectif est un objectif d'optimisation (1). En générale le problème est formulé en terme de " Satisfaction " de " niveau de service " , notion plus réaliste mais dont il est plus difficile de quantifier. D'où l'intérêt des algorithmes de coordination que nous traiterons dans le paragraphe suivant.

III . 2 . ALGORITHMES COURANTS

Pour résoudre les différents problèmes de décomposition coordination différents algorithmes sont mis à notre disposition. Si le problème à résoudre présente une fonction objective et des contraintes linéaires on utilise la méthode de programmation linéaire, par contre si la fonction objective et /ou les contraintes ne sont pas linéaire on utilise la méthode de programmation non linéaire.

III . 2 . 1 . METHODE DE PROGRAMMATION LINEAIRE

La programmation linéaire est une des méthodes d'optimisation linéaire les plus connues et probablement les plus utilisées dans les sciences de la gestion (4). Elle est aussi incontestablement la méthode mathématique la plus proche de l'économie puisque ses hypothèses se fondent aussi sur la rareté de l'information. En traduisant un problème économique en un problème mathématique, l'optimisation linéaire cherche, en effet, à optimiser les résultats, pour un ensemble de moyen donnés ou à minimiser les moyens engagés pour un ensemble de résultats souhaités.

La méthode fondamentale pour trouver l'optimum de la fonction économique d'un programme linéaire est la méthode du simplexe ou la méthode de DANTZIG (4).

III 2. 2. METHODE DE PROGRAMMATION NON LINEAIRE

Nous présentons trois méthodes de décomposition - coordination en programmation non linéaire pour la résolution de problèmes d'optimisation statique :

Ces méthodes utilisent 2 approches : (2)

1 - Agir sur la fonction objective des sous systèmes (coordination du but)

2 - Prédiction des interactions. Dans ce cas on distingue.

a) prédiction de l'interaction des sorties Z_i au niveau du sous système (coordination du modèle)

b) Prédiction de l'interaction des entrée X_i du sous système. En réalité ce mode est rarement appliqué seul.

III 2 . 2 . 1 . METHODE DE COORDINATION DU BUT

Cette méthode est caractérisée par le paramètre ρ_j ($j = 1, N$) fixé préalablement au niveau 2 et qui constitue une donnée importante du niveau 1.

ρ_i étant fixé, on peut associer le terme $\rho_i^T X_i$ au langrangien du sous système "i" et distribuer le terme.

$$\rho_i^T \sum_j a_{ij} Z_j$$

entre tous les autres sous-système ce qui revient à faire :

$$\sum_i \rho_i^T \sum_j c_{ij} z_j = \sum_i \sum_j \rho_i^T c_{ij} z_i$$

et

$$L = \sum_{i=1}^N L_i = \sum \left[f_i(M_i, X_i) + \mu_i^T (T_i - Z_i) + \rho_i^T X_i - \sum_{j=1}^N \rho_i^T c_{ij} z_j \right] = \sum_{i=1}^N L_i(\beta_i, d)$$

$$(\beta_i^T = (X_i^T, M_i^T, z_i^T, \mu_i^T), d = \rho)$$

La fonction L peut se mettre sous forme "séparable" et chaque sous système du premier niveau est donc défini par son lagrangien associé L_i .

sous système i

$$\begin{cases} \max (f_i(M_i, X_i) + \rho_i^T X_i - \sum_{j=1}^N \rho_j^T c_{ji} Z_j) \\ Z_i = T_i(M_i, X_i) \text{ avec } \rho_i \quad i = 1, N \text{ fixé} \end{cases}$$

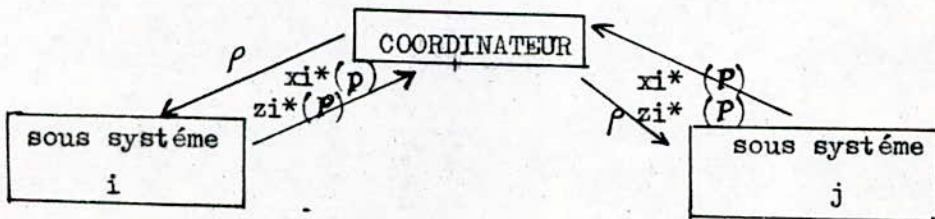
On voit alors que la fonction critère est considérablement modifiée. C'est pour cela que la méthode est appelée méthode de coordination du but. Cette modification s'exprime comme suit :

$$\sum_i \Delta f_i = \sum_i \rho_i^T (X_i - \sum_j c_{ij} z_j) = 0$$

si $X_i^*(P)$, $Z_i^*(P)$, $M_i^*(P)$ sont les solutions du sous système, l'ensemble de ces solutions de $i = 1$ à N doit satisfaire la condition :

$$X_i^*(P) - \sum_{j=1}^N c_{ij} Z_j^*(P) = \xi_i$$

Si $\xi_i = 0 \forall i$ nous avons la solution globale, si non la contrainte d'interconnection n'a pas été satisfaite et après avoir tenu compte des résultats du niveau 1, une action sur le niveau 2 est nécessaire pour satisfaire cette contrainte. Les informations nécessaires transmises dans ce cas sont schématisés dans la figure suivante :



D'un point de vue analytique, la tâche du niveau 1 est de résoudre le système suivant :

$$* \left\{ \begin{array}{l} \frac{\partial L}{\partial x_i} = 0 \\ \frac{\partial L}{\partial M_i} = 0 \\ \frac{\partial L}{\partial Z_i} = 0 \end{array} \right. \begin{array}{l} \text{conditions qui doivent \u00eatre} \\ \text{satisfaites par la solution} \\ \text{optimale} \end{array} \quad i = 1, \dots, N$$

Le niveau 2 permet de fixer ρ , cependant ρ apparait explicitement dans les \u00e9quations pr\u00e9c\u00e9dentes, il est donc n\u00e9cessaire d'introduire un algorithme de coordination

De l'expression du langrangien nous d\u00e9duisons :

$$dL = \sum_i \left[L \rho_i^T d\rho_i + L Z_i^T dZ_i + L x_i^T dx_i + L M_i^T dM_i + L \mu_i^T d\mu_i \right]$$

apr\u00e8s le passage du niveau 1 nous avons :

$$LZ_i = 0 \quad ; \quad LM_i = 0 \quad ; \quad Lx_i = 0 \quad ; \quad L\mu_i = 0$$

dL est donc r\u00e9duit \u00e0 :

$$dL = \sum_i L \rho_i^T d\rho_i$$

si l'on choisi :

$$d\rho_i = -K L \rho_i \quad K > 0$$

On pose $dL > 0$. Ceci est n\u00e9cessaire puisque nous recherchons une solution conforme aux suppositions que nous avons faites, nous aurons alors :

$$\rho_i(t+1) = \rho_i(t) - K L \rho_i(t)$$

Il est aussi possible d'utiliser directement la m\u00e9thode de Newton pour r\u00e9soudre l'\u00e9quation $L\rho = 0$ en se basant sur le coordinateur algorithmique :

$$* * \quad \rho(t+1) = \rho(t) - c \left[\frac{dL\rho(t)}{d\rho} \right]^{-1} \quad L\rho = L\rho(X,Z)$$

et

$$\frac{dL\rho}{d\rho} = L\rho_x X_p + L\rho_z Z_p$$

Cet algorithme de coordination n\u00e9cessite une inversion de matrice mais son avantage est qu'il converge facilement contrairement \u00e0 la m\u00e9thode du gradient

Nous pouvons conclure en disant qu'il est pr\u00e9f\u00e9rable d'utiliser la m\u00e9thode du gradient loin de l'optimum et la m\u00e9thode de Newton pr\u00e8s de l'optimum.

* * Nous avons vu que :

$$L_i = X_i - C_{ij} Z_j \quad 1 = 1 \quad (X,Z)$$

III 2. 2. 2. METHODE DE COORDINATION DU MODELE

Dans cette méthode, le niveau haut fixe Z_j et stipule le niveau bas.
 Dans ce dernier, le lagrangien peut être écrit de la manière "séparable" suivante :

$$L = \sum_i L_i(\beta_i, d_i) = \sum_i [f_i(M_i, X_i) + \mu_i^T (T_i - Z_i) + \rho_i^T (X_i - \sum_j C_{ij} Z_j)]$$

$$(\beta_i^T = (X_i^T, M_i^T, \mu_i^T, \rho_i^T), d = Z)$$

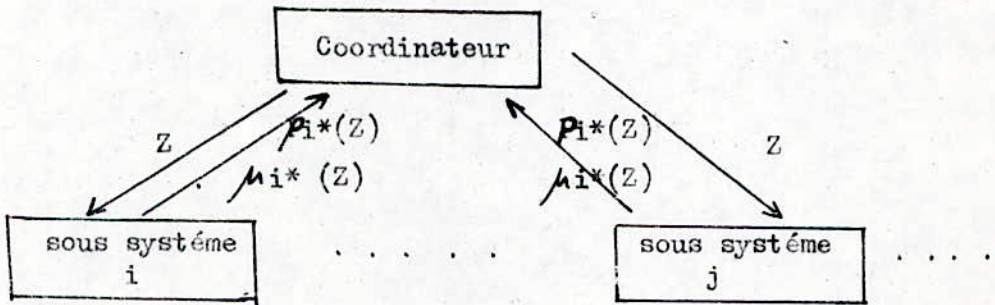
La forme de L_i montre que le i ème sous système est :

$$\begin{cases} \max & f_i(M_i, X_i) \\ & \begin{cases} Z_i = T_i(M_i, X_i) \\ X_i = \sum_j C_{ij} Z_j \end{cases} \end{cases} \quad Z_j, j = 1, N \text{ fixé}$$

Dans ce cas, quoique le critère reste inchangé, les variables Z_i et X_i sont indirectement fixées dans le modèle.

Les contraintes du modèle et les contraintes d'interconnection sont toujours satisfaites c'est pourquoi on appelle aussi cette méthode, méthode "faisable"

Les informations de transfert entre les deux niveaux sont schématisées comme suit :



μ et ρ sont nécessairement au niveau 2, puisque la seule équation que l'on peut traiter à ce niveau est :

$$L_Z(\mu, \rho) = 0$$

Les équations $L_X = 0$, $L_M = 0$, $L_\rho = 0$, $L_\mu = 0$ sont satisfaites au niveau local dans l'ordre de résolution.

Ainsi au niveau 2, $dL = L_Z^T dZ$ puisque nous voulons maximiser L en satisfaisant $dL > 0$, on doit choisir :

$$dZ = K L_Z^T$$

ce qui devient après descriptisation

$$Z(t+1) = Z(t) + K_L \dot{Z}(t)$$

lequel est l'algorithme de coordination du type gradient.

Si nous prenons l'algorithme de Newton, on peut écrire

$$Z(t+1) = Z(t) - C \left(\frac{dLZ^{-1}}{dZ} \right) LZ(t)$$

III 2. 2. 3. METHODE MIXTE

Dans cette méthode, ρ et Z sont déterminés par le niveau 2 et utilisés au niveau 1. Le lagrangien est alors :

$$L = \sum_{i=1}^N L_i(\beta_i, d) = \sum_{i=1}^N \left[f_i(X_i, M_i) + \rho_i^T (X_i - \sum_{j=1}^N C_{ij} Z_j) + \mu_i^T (T_i - Z_i) \right]$$

$$(\beta_i^T = (X_i^T, M_i^T, \mu_i^T), d^T = (\rho^T, Z^T))$$

Ce qui nous permet de décomposer le problème en sous système :

avec ρ_i et Z_i fixés $\left\{ \begin{array}{l} \max [f_i(X_i, M_i) + \rho_i^T X_i + C_t] \quad (- \rho_i^T C_{ij} Z_j = \text{constante}) \\ Z_i = T_i(X_i, M_i) \end{array} \right.$

Dans ce cas nous voyons que la fonction critère et le modèle sont tous deux utilisés dans la coordination.

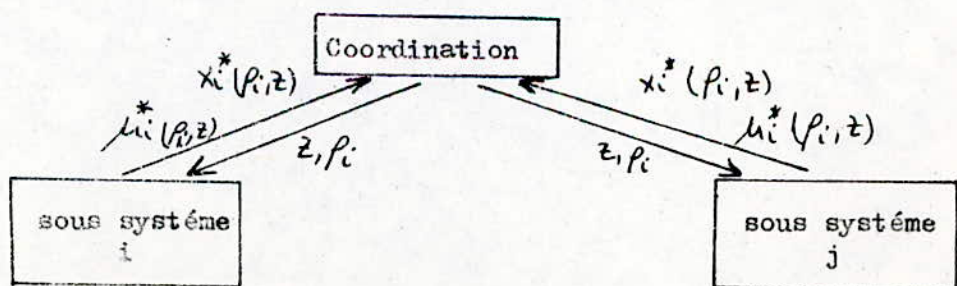
La solution de chaque sous système correspond au traitement des équations * ce qui donne :

$$L_p(X, Z) = 0$$

$$L_Z(\mu, \rho) = 0$$

$m_{xi} + m_{\mu i} \geq m_{zi}$ est la plus petite condition restrictive que l'on recherche pour la méthode de coordination du modèle.

Les informations de transfert, entre les niveaux sont schématiser comme suit.



En faisant l'analogie avec les méthodes précédentes, nous pouvons utiliser l'algorithme de coordination du gradient.

$$\left. \begin{aligned} \frac{dP}{dt} &= -L_P \\ \frac{dZ}{dt} &= L_Z \end{aligned} \right\} \Rightarrow \begin{cases} P(t+1) = P(t) - K L_P \\ Z(t+1) = Z(t) + K L_Z \end{cases} \quad K > 0$$

Autrement, nous pouvons utiliser l'algorithme de Newton pour résoudre les deux équations.

$$L_P = 0 \qquad L_Z = 0$$

En écrivant :

$$W(t+1) = W(t) - c \left[\frac{dLW}{dW} \right]^{-1} L_W$$

$$W = \begin{bmatrix} Z \\ P \end{bmatrix} \quad 0 < c < 2$$

Ou de nouveau, si certaines conditions sont faites ($\sum m_{xi} = \sum m_{zi}$), nous pouvons calculer Z en utilisant $L_P = 0$ et P en utilisant $L_Z = 0$.

Ainsi nous mettons en pratique une méthode d'itération directe avec une bonne convergence.

III . 3 . ALGORITHMES ETUDIÉS

Dans ce qui suit nous étudierons de manière plus approfondie deux algorithmes de décomposition - coordination :

L'algorithme de DANTZIG - WOLFE basé sur la méthode du simplexe et l'algorithme d'UZAWA basé sur la recherche du point selle.

III . 3 . 1 . METHODE DE DANTZIG - WOLFE

L'algorithme de DANTZIG - WOLFE est une généralisation de la décomposition des programmes linéaires aux cas des programmes convexes. Il est basé sur la méthode du simplexe que nous verrons en annexe.

III . 3 . 1 . 1 . ALGORITHME DE DANTZIG - WOLFE

- Problème :

Le programme convexe P1 que l'on désire résoudre peut se présenter de la forme :

$$(1.1) \quad \begin{array}{l} \min \sum_{i=1}^N f_i(x_i) \\ \sum_{i=1}^N A_i(x_i) \leq 0 \\ x_i \in S_i \end{array}$$

$f_i(x_i)$ et $A_i(x_i)$ sont des fonctions convexes c'est à dire convergentes.
On fait l'hypothèse que le domaine de P1 n'est pas vide.

Les générateurs X_i représentent les variables d'entrée qui génèrent le problème.
La fonction $f_i(x_i)$ n'est autre que la fonction objective ou fonction économique.

- X_i : colonne indicée par J
- $A_i(x_i)$: colonne indicée par L
- A domaine défini par (1.1)

Pour résoudre le programme P1 on le linéarise c'est à dire on passe de P1 convexe au programme P2 linéaire, P2 sera appelé maître programme.

- Maître programme

Pour comprendre comment se fait le passage du programme P1 convexe au programme P2 linéaire, on passe au cas général de problème linéaire.

Ce dernier peut se mettre sous la forme. (2)

$$\begin{aligned} & \min \quad [C_1 X_1 + \dots + C_p X_p] \\ \text{contraintes : } & \left. \begin{aligned} & [A_1 X_1 + \dots + A_p X_p = a_1 \\ & [B_1 X_1 \dots \dots \dots = B_1 \\ & [B_2 X_2 \dots \dots \dots = B_2 \\ & \vdots \end{aligned} \right\} \quad (1,3) \end{aligned}$$

On pose L égale au lagrangien associé au problème (1,3)

$$\begin{aligned} L = & C_1^T X_1 + \dots + C_p^T X_p + \pi_1^T [a_1 - A_1 X_1 - A_2 X_2 \dots - A_p X_p] \\ & + \alpha_1^T [b_1 - B_1 X_1] + \dots + \alpha_p^T [b_p - B_p X_p] \quad (1,4) \end{aligned}$$

$\alpha_i, i = 1, P$

π_1 : multiplicateur du lagrangien

Si l'on fixe π_1 , le lagrangien peut être additivement séparable :

$$L = \sum_i L_i = \sum_i \left\{ C_i^T X_i - \pi_i^T A_i X_i + \alpha_i^T [b_i - B_i X_i] \right\} \quad (1,5)$$

pour π_1 donné, $\pi_1^T \alpha_1$ est 1 constante.

Chaque lagrangien donne le sous - problème suivant :

$$\begin{aligned} & \min (C_i^T - \pi_i^T A_i) X_i \\ & \left. \begin{aligned} & [B_i X_i = b_i \\ & [X_i \geq 0 \end{aligned} \right\} \quad \pi_1 \text{ donné} \quad (1,6) \end{aligned}$$

Il reste à trouver comment déterminer \bar{x}_1 en utilisant une solution locale de sorte à pouvoir résoudre le problème (1.3)

Pour ce faire, nous commencerons par simplifier la structure du problème ($p = 1$), on a alors la représentation suivante.

$$\begin{aligned} \min C^T x & & (1,7) \\ \left[\begin{array}{ll} A1 x = b1 & m1 \text{ contraintes} \\ A2 x = b2 & m2 \text{ contraintes} \\ x \geq 0 & \end{array} \right. & \begin{array}{l} (1,8) \\ (1,9) \end{array} \end{aligned}$$

On peut maintenant utiliser le théorème suivant

" Soit $X = \{ x / Ax = b, x \geq 0 \}$, un ensemble non vide borné et x^i un point extrême de cet ensemble

alors tous les éléments $x \in X$ peuvent s'écrire de la forme :

$$x = \sum_{i=1}^r \lambda_i x^i \quad \lambda_i \geq 0, \quad \sum_{i=1}^r \lambda_i = 1 \quad (1.10)$$

$i = 1, \dots, r$, où r est le nombre limite des points extrêmes

$$\text{soit } S_2 = \{ x / A2 x = b2, x \geq 0 \}$$

alors pour $x \in S_2$

$$x = \sum_j \lambda_j x^j \quad \sum_j \lambda_j = 1 \quad \lambda_j \geq 0 \quad (1.11)$$

Où x^j est une solution basique réalisable

Ainsi le problème de (1.7) à (1.9) peut être réécrit :

En choisissant parmi les solutions de (1.9) celles qui satisfont (1.9) et minimisent (1.7)

Pour satisfaire la condition (1.8), en tenant compte de (1.11), on doit avoir

$$A1 \sum_j \lambda_j x^j = b1 \Rightarrow \sum_j (A1 x^j) \cdot \lambda_j = b1$$

$$\text{et } \min C^T x \Rightarrow C^T \sum_j \lambda_j x^j = \sum_j (C^T x^j) \lambda_j$$

$$\text{En posant } A1 x^j = p_j \Rightarrow \sum_j p_j \lambda_j = b1$$

$$C^T x^j = f_j \Rightarrow C^T \cdot x = \sum_j f_j \lambda_j$$

Le problème donné par les équations de (1.7) à (1.9) est équivalent à :

$$\left. \begin{array}{l} \min \quad \sum_j f_j \lambda_j \\ \sum_j p_j \lambda_j = b_1 \\ \sum_j \lambda_j = 1 \\ \lambda_j \geq 0 \end{array} \right\} \quad (1.12)$$

C'est ce qu'on appelle le "Maître programme "

Ainsi, on peut remplacer le programme d'origine P1 par un programme linéaire approché P2 (K), que l'on pourra écrire de la forme suivante :

$$\boxed{\begin{array}{l} \text{P2 (K)} \quad \min \quad \sum_{i=1}^N f_i^{K_i} \lambda_{K_i} \\ \sum_{i=1}^N A^{K_i} \lambda_{K_i} \leq 0 \\ \left. \begin{array}{l} e^{K_i} \lambda_{K_i} = 1 \\ \lambda_{K_i} \geq 0 \end{array} \right\} \quad i = 1, \dots, N \end{array}}$$

avec : . K_i , un ensemble fini d'indices

. x_i^j , $j \in K_i$, un point de l'ensemble S_i , et

. $x^{K_i} = \left\{ \begin{array}{l} x^j / j \in K_i \text{ une matrice indicée par } (J \times K_i) \\ f_i^j = f_i(x_i^j) \text{ un scalaire et, } f_i^{K_i} = \{ f_i^j / j \in K_i \} \end{array} \right\}$
une ligne indicée par $(J \times K_i)$

. $A_i^j = A_i(x_i^j)$ une colonne, et $A^{K_i} = \left\{ A_i^j / j \in K_i \right\}$

une matrice indicée par $(L \times K_i)$

. $e^j = 1$ un scalaire, et $e^{K_i} = \{ e^j / j \in K_i \}$ une ligne indicée par K_i

Ce programme linéaire n'est qu'une approximation de P1 par défaut. En effet, à toute solution réalisable $\lambda_{K_i} \quad i = 1, N$ de P2 (K) correspond par (1.1) , une solution réalisable de P1 , $x_i(\lambda_{K_i})$. Et l'ensemble des points $x_i(\lambda_{K_i})$ correspondants à l'ensemble des solutions réalisable λ_{K_i} de P2 (K) est contenu dans le domaine des solutions réalisables de P1.

D'après la méthode simpliciale sur laquelle est basé l'algorithme DZW pour améliorer la solution optimum de P2 (K),

On passe à une autre base, la nouvelle solution est réalisable par construction par rapport à la solution précédente.

.../...

En d'autres termes, la solution de $P2(K)$ est obtenue dans une certaine base I , pour s'approcher au mieux de la solution, on change de base (I'). On renouvelle le procédé d'amélioration sur I' jusqu'à obtenir la solution de base optimale. Le nombre de bases possibles étant fini, on est assuré de converger en un nombre fini d'itérations.

A chaque solution optimal, on associe le multiplicateur simplexe W qui permet de connaître la sensibilité de la solution.

A l'optimum le multiplicateur du simplexe est égale aux variables duales :

$$W(K) = [U1(K), U2(K)]$$

Ces variables représentent les contraintes du problème, $U^1(K)$ est indicé par L , et $U2(K)$ indicé par N .

si on écrit les conditions de KHUN et TUCKER

$$(1.13) \begin{cases} U^1(K) \leq 0 \\ d_i(K) = f_i^{K_i} - U^1(K) A_{i,K_i} - U_i^2(K) e^{K_i} \geq 0 \\ U^1(K) A_{i,K_i} \lambda_{K_i} = 0 \\ \lambda_i^j d_i^j = 0 \end{cases}$$

On remarque que les coûts relatifs $d_i(K_i)$ sont tous positifs, pour améliorer la solution de $P2(K)$ il faut trouver une colonne dont le coût soit négatif et aussi petit que possible et l'insérer dans le domaine de recherche c'est ce qu'on appelle changement de base.

La minimisation du coût relatif conduit aux problèmes locaux $P3_i(K)$

- Programmes locaux

Le coût relatif associé à un générateur quelconque est d'après la méthode simpliciale:

$$d_i^j = f_i^j - U^1(K) A_{i,j} (N_i^j) - U_i^2(K)$$

.../...

puisque une colonne P2 (K) s'écrit :

$$i \dots \dots \dots \begin{bmatrix} \frac{f_i^j}{A_i(x_i^j)} \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ L \\ N \end{matrix}$$

pour trouver le générateur dont le coût relatif est minimum négatif, il faut résoudre :

$$(1.14) \quad \min d_i^j \\ d_i^j < 0$$

Ce qui donne :

$$\begin{aligned} \min d_i^j &= \min \left\{ f_i^j - U^1(K), A_i^j(x_i^j) - U_i^2(K) \right\} \\ &= \min \left\{ \min \left[f_i^j - U^1(K) A_i^j(x_i^j) \right] - U_i^2(K) \right\} \end{aligned}$$

avec la condition $x_i^j \in S_i$

$$\Rightarrow P3i \quad \boxed{\begin{matrix} \min f_i(x_i) - U^1(K) A_i(x_i) \\ x_i \in S_i \end{matrix}}$$

La solution de ce programme doit être x_i^S dont d_i^S est minimum négatif.

Ce générateur x_i^S sera alors insérer dans P2 (K).

III . 3 . 1 . 2 . RESOLUTION

On peut résumer la méthode DZW dans ce qui suit :

Pour une valeur de $U^1(K)$ et $U^2(K)$, on recherche le générateur x_i^S qui minimise P3i (K) et vérifie la condition $d_i^S < -\epsilon_1$

Si ce générateur existe pour cette valeur de $U^1(K)$ et $U^2(K)$ il sera insérer dans P2 (K).

Si au moins un générateur est ajouté à P2(K), c'est à dire :

Si on fait un changement de base, on vérifie si l'itération est valable en d'autre termes on vérifie qu'on n'a pas dépasser le nombre d'itération nécessaire pour s'approcher au mieux de la solution, et ceci par les critères de terminaison (a) et (b), si ces deux critères ne sont pas encore atteind , on réscud à nouveau P2(K) en tenant compte du nouveau domaine de solution réalisable.

On aura alors des nouvelles solutions λ_{Ki} de $P_2(K)$ donc des nouvelles valeurs de $U_1(K)$ et $U_2(K)$ et le cycle reprend jusqu'à ce que les critères de terminaison soient atteints.

Notre objectif est de trouver des solutions pour chacun de ces K sous programmes et de déterminer parmi toutes ces solutions celle qui optimisera la fonction économique du programme global tout en satisfaisant à l'ensemble des contraintes générales.

- 1ère étape

Résolution du programme satellite $P_3 i (K)$

- 2ème étape

Résolution de $P_2 (K)$

On se basera essentiellement sur la méthode simpliciale.

Pour pouvoir appliquer cette méthode, on transforme les inéquations du problème en équations, en introduisant des nouvelles variables non négatives appelées variables d'écart. Les contraintes du problème peuvent donc s'écrire sous la forme suivante :

$$\sum_{i=1}^N A^{Ki} \lambda_{Ki} + I \lambda'_{Ki} = 0$$

I : matrice unité

λ'_{Ki} : variables d'écart

Solution de base - Solution de base réalisables (3)

En introduisant les variables d'écart on aura alors plus d'inconnues que d'équations en d'autre termes, soit un problème linéaire avec m contraintes (non compris les contraintes de non négativité) et n variables (y compris les variables d'écart), $n > m$ nous avons les définitions suivantes :

1 - Solution de base : rendons $(n - m)$ variable égale à zéro

Si le système de m équation à m inconnues restant admet une solution unique, on appelle cette solution une solution de base. Il existe au plus

$$\frac{n!}{m!(n-m)!} = C_m^n \text{ solution de base.}$$

.../...

2 - Solution de base réalisable : on appelle solution de base réalisable toute solution de base qui satisfait l'ensemble de contraintes du problème (y compris les contraintes de non - négativité).

3 - Solution de base réalisable optimale : on appelle solution de base réalisable optimale une solution de base réalisable qui optimise la fonction économique.

Pour P2 (k) les contraintes sont :

$$\sum_{i=1}^N A^{ki} \lambda_{ki} \leq 0 \quad \text{et} \quad e^{ki} \lambda_{ki} = 1$$

$A^{ki} = \left\{ A_i^j = A_i(x_i^j) / j \in K_i \right\}$ une matrice indexée par $L \times K_i$.

pour $i=1$ $A^{k1} \lambda_{k1}$

$$L \begin{pmatrix} A_{11} & A_{12} & \dots & \dots & A_{1k1} \\ A_{21} & & & & \vdots \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ A_{L1} & \dots & \dots & \dots & A_{Lk1} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_{k1} \end{pmatrix}$$

$\underbrace{\hspace{15em}}_{K1}$

$$\sum_{i=1}^N A^{ki} \lambda_{ki} \leq 0$$

$$\begin{pmatrix} A_{11} & \dots & \dots & A_{1k1} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ A_{L1} & \dots & \dots & A_{Lk1} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_{k1} \end{pmatrix} + \begin{pmatrix} A_{11} & \dots & \dots & A_{1k2} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ A_{L1} & \dots & \dots & A_{Lk2} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_{k2} \end{pmatrix} + \dots$$

$$\dots + \begin{pmatrix} A_{11} & \dots & \dots & A_{1kN} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ A_{L1} & \dots & \dots & A_{LkN} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_{kN} \end{pmatrix} \leq 0$$

$$\begin{pmatrix} A_{11} \lambda_1 + \dots + \lambda_{k_1} A_{1k_1} \\ \vdots \\ A_{L1} \lambda_1 + \dots + A_{Lk_1} \lambda_{k_1} \end{pmatrix} + \dots + \begin{pmatrix} A_{11} \lambda_1 + \dots + A_{1k_n} \lambda_{k_n} \\ \vdots \\ A_{L1} \lambda_1 + \dots + A_{Lk_n} \lambda_{k_n} \end{pmatrix} \leq 0$$

$$\begin{pmatrix} \sum_{i=1}^{k_1} A_{1i} \lambda_i + \sum_{i=1}^{k_2} A_{1i} \lambda_i + \dots + \sum_{i=1}^{k_n} A_{1i} \lambda_i \\ \vdots \\ \sum_{i=1}^{k_1} A_{Li} \lambda_i + \sum_{i=1}^{k_2} A_{Li} \lambda_i + \dots + \sum_{i=1}^{k_n} A_{Li} \lambda_i \end{pmatrix} \leq 0$$

On a : L équations et $\sum_{i=1}^N k_i$ inconnues, on doit donc introduire L variables d'écart.

$$\left. \begin{aligned} \sum_{i=1}^{k_1} A_{1i} \lambda_i + \sum_{i=1}^{k_2} A_{1i} \lambda_i + \dots + \sum_{i=1}^{k_n} A_{1i} \lambda_i + \lambda_{k_n+1} &= 0 \\ \sum_{i=1}^{k_1} A_{2i} \lambda_i + \dots + \sum_{i=1}^{k_n} A_{2i} \lambda_i + \lambda_{k_n+2} &= 0 \\ \vdots \\ \sum_{i=1}^{k_1} A_{Li} \lambda_i + \dots + \sum_{i=1}^{k_n} A_{Li} \lambda_i + \lambda_{k_n+L} &= 0 \end{aligned} \right\} L$$

et les contraintes d'égalité sont :

$$e^{k_i} \lambda_{k_i} = 1 \quad i = 1, \dots, N$$

$$(e^1, e^2, \dots, e^{k_1}) \lambda_{k_1} = 1$$

$$(e^1, e^2, \dots, e^{k_2}) \lambda_{k_2} = 1$$

⋮

$$(e^1, e^2, \dots, e^{k_N}) \lambda_{k_N} = 1$$

Les contraintes de non négativité sont : $\lambda_{k_i} \geq 0$

On a alors :

L : variables d'écart

et N : variables artificielles.

$$\begin{array}{l}
 L \left[\begin{array}{l}
 \sum_{i=1}^{k_1} A_{1i} \lambda_i + \sum_{i=1}^{k_2} A_{1i} \lambda_i + \dots + \sum_{i=1}^{k_N} A_{1i} \lambda_i + \lambda_{k_N+1} = 0 \\
 \sum_{i=1}^{k_1} A_{2i} \lambda_i + \sum_{i=1}^{k_2} A_{2i} \lambda_i + \dots + \sum_{i=1}^{k_N} A_{2i} \lambda_i + \lambda_{k_N+2} = 0 \\
 \vdots \\
 \sum_{i=1}^{k_1} A_{Li} \lambda_i + \sum_{i=1}^{k_2} A_{Li} \lambda_i + \dots + \sum_{i=1}^{k_N} A_{Li} \lambda_i + \lambda_{k_N+L} = 0
 \end{array} \right. \\
 \\
 N \left[\begin{array}{l}
 \lambda_1 e^1 + \lambda_2 e^2 + \dots + \lambda_{k_1} e^{k_1} + \lambda_{k_N+L+1} = 0 \\
 \lambda_1 e^1 + \lambda_2 e^2 + \dots + \lambda_{k_2} e^{k_2} + \lambda_{k_N+L+2} = 0 \\
 \vdots \\
 \lambda_1 e^1 + \lambda_2 e^2 + \dots + \lambda_{k_N} e^{k_N} + \lambda_{k_N+L+N} = 0
 \end{array} \right.
 \end{array}$$

Le programme utilisé se distingue des autres programmes du simplexe usuels par sa simplicité dans l'introduction des données.

Description du programme (4)

L'algorithme du simplexe fonde ces itérations sur la matrice de base appelée "Tableau du simplexe". Ce tableau n'est rien d'autre qu'une transformation d'un modèle de programmation linéaire écrit sous forme canonique en forme standard (introduction des variables d'écart et artificielles).

Les lignes 110 - 850 permettent de construire ce tableau de la façon la plus simple.

Les lignes 210 - 550 identifient et impriment les variables de décision, variables d'écart et variable artificielles.

Les lignes 870 - 1270 préparent et impriment le tableau du simplexe. L'examen d'optimalité est analysé dans les lignes 1290 - 1570. Lorsque la solution optimale est identifiée, les lignes 1590 - 1700 impriment les résultats du problème.

Lorsque la solution optimale n'est pas encore atteinte, il est nécessaire de transformer à nouveau le tableau. Pour cela, il faut d'abord déterminer une nouvelle variable de base (lignes 1300 - 1340), une variable sortant de la base (lignes 1740 - 1800) et ensuite, transformer les données du tableau (lignes 1840 - 1990).

3ème étape

Calcul des variables duales :

1 - Nécessité des variables duales :

Dans de nombreuses situations d'intérêt scientifique et pratique, il est nécessaire de commander des procédés (ou des systèmes) faisant appel à des incertitudes. En générale cette incertitude est remplacée par des effets stochastiques.

En effet, la capacité future de production, la demande future des produits, les prix et les quantités nécessaires des matières premières, les coûts et profits futurs entre autres informations, proviennent en général de projections à court terme des tendances observées par le passé ; cela met donc en doute l'exactitude des données. Pour cela, un système de commande optimale doit être de nature adoptive, c'est à dire que le système devra acquérir des informations sur les paramètres inconnus au fur et à mesure des prises de décision.

Une propriété importante de ces systèmes est que les décisions de commande affectent le niveau d'incertitude des paramètres du système, et réciproquement. De tels types de commandes sont appelées " duales ".

2 - Dualité :

Tout programme linéaire a un programme équivalent qu'on appelle DUAL, le premier est appelé PRIMAL et le second DUAL. Par définition, un programme primal est un programme linéaire consistant à "maximiser" une fonction économique dans un domaine défini par des contraintes. Par contre, le programme dual qui est le transformé du primal consiste à "minimiser" la fonction économique. La correspondance primal-dual est bijnivoque et l'un ou l'autre des problèmes peut s'identifier au primal ou au dual.

La structure des 2 programmes est semblables :

- Le problème primal a m contraintes ; le problème dual a m variables. Le primal a n variables principales tandis que le dual à n contraintes. Le vecteur de la fonction économique du primal devient le vecteur des seconds membres des contraintes du problème dual ; et inversement, le vecteur des seconds membres des contraintes du primal devient le vecteur de la fonction économique du dual.

3 - Rôle pratique des variables duales

La connaissance des valeurs des variables duales permet d'évaluer la rentabilité des ressources utilisées ainsi que d'ordonner ces ressources selon leur rentabilité. Elle permet de plus de fractionner selon les ressources utilisées la contribution marginale au profit des diverses activités de production.

4 - Coûts marginaux :

L'interprétation économique des valeurs optimales des variables duales révèle les similarités qui existent entre les solutions primales et duales optimales : Les variables duales représentent la valorisation marginale des quantités exprimées dans les contraintes du primal.

D'autre part on appelle cout marginaux, la valeur optimale des variables dual. On peut donc interpréter les coûts marginaux comme les coûts d'opportunité des ressources (contraintes) avec lesquelle ils sont associés. En d'autre termes, le coût marginal mesure l'effet sur la fonction économique de la décision de relâcher la contrainte correspondante en ajoutant une unité supplémentaire de la "ressource" associée à cette contrainte.

5 - Calcul des variables duales

Le tableau optimal du simplexe donne à la fois la solution du problème primal et du problème dual.

Il suffit donc de résoudre par la méthode du simplexe l'un ou l'autre des problèmes primal ou dual pour obtenir simultanément les solutions optimales des deux programmes.

Critères de terminaison

Nous avons dit plus haut que $P_2(K)$ n'est qu'une approximation par défaut de P_1 , et qu'une fois la résolution de $P_2(K)$ obtenue, on devait améliorer la solution par des changements de base successifs et ceci en ajoutant au programme $P_2(K)$ les générateurs X_i solutions de $P_3^i(K)$.

Donc, pour terminer l'algorithme on choisira entre une méthode utilisant la borne inférieure de l'optimum donnée par la relation :

$$\sum_{i=1}^N f_i [X_i(\lambda_{K_i})] \geq \sum_{i=1}^N f_i(\hat{X}_i) \geq \sum_{i=1}^N (f_i^{K_i} \lambda_{K_i} + d_i^s)$$

ou une évolution de la décroissance des solutions de $P_2(K)$.

a) Méthode de la borne inférieure :

d'après la relation précédente on a :

$$\sum_{i=1}^N f_i^{K_i} \lambda_{K_i} \geq \sum_{i=1}^N f_i(\hat{X}_i) \geq \sum_{i=1}^N (f_i^{K_i} \lambda_{K_i} + d_i^s)$$

En effet, du fait de la convexité de $f_i(X_i)$ et puisque $f_i^{K_i} \lambda_{K_i}$ est une combinaison convexe des $f_i(X_i)$ on a :

$$f_i [X_i(\lambda_{K_i})] \leq f_i^{K_i} \lambda_{K_i} \quad i = 1, \dots, N$$

si $\hat{X}_i, i = 1, \dots, N$ est la solution optimale de P_1 et si on pose $X_i(K_i) = X_i^{K_i} \lambda_{K_i}$, il vient

$$\begin{aligned} \sum_{i=1}^N f_i [X_i(K_i)] &\geq \sum_{i=1}^N f_i(\hat{X}_i) \\ \Rightarrow \sum_{i=1}^N f_i^{K_i} \lambda_{K_i} &\geq \sum_{i=1}^N f_i X_i(\lambda_{K_i}) \geq \sum_{i=1}^N f_i(\hat{X}_i) \end{aligned}$$

Par ailleurs, puisque

$$\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n \in A \cap \left[\bigcap_{i=1}^N S_i \right] \subset \left[\bigcap_{i=1}^N S_i \right]$$

on a pour $i = 1, \dots, N$

$$f_i(\hat{X}_i) - U_1(K) A_i(\hat{X}_i) \geq \min \left\{ f_i(X_i) - U_1(K) A_i(X_i) / X_i \in S_i \right\}$$

On en déduit

$$f_i(\hat{X}_i) - U_1(K) A_i(\hat{X}_i) - U_2 i(K) \geq d_i^s$$

et en utilisant la 2^{ème} condition de KUHN et TUCKER il vient alors :

$$f_i(\hat{X}_i) \geq f_i^{K_i} \lambda_{K_i} + d_i^s + U_1(K) A_i(\hat{X}_i)$$

avec $U_1(K) \leq 0$

$$\text{on a } f_i(\hat{X}_i) \geq f_i^{K_i} \lambda_{K_i} + d_i^s$$

On sommant sur i on obtient :

$$\sum_{i=1}^N f_i(\hat{X}_i) \geq \sum_{i=1}^N (f_i^{K_i} \lambda_{K_i} + d_i^s)$$

$|\sum d_i^s|$ représente donc une borne supérieure de l'écart entre l'optimum et la solution du programme approché $P_2(K)$ étant donné $\{\epsilon_2 > 0\}$, on en déduit le critère de terminaison (a) :

$$\frac{\sum_{i=1}^N d_i^s}{\sum_{i=1}^N f_i^{K_i} \lambda_{K_i}} < \epsilon_2$$

b) Méthode de la variation de l'optimum :

Cette méthode consiste à mesurer la "vitesse" de décroissance de l'optimum. Lorsque cette vitesse devient inférieure à une limite fixée, on arrête les itérations.

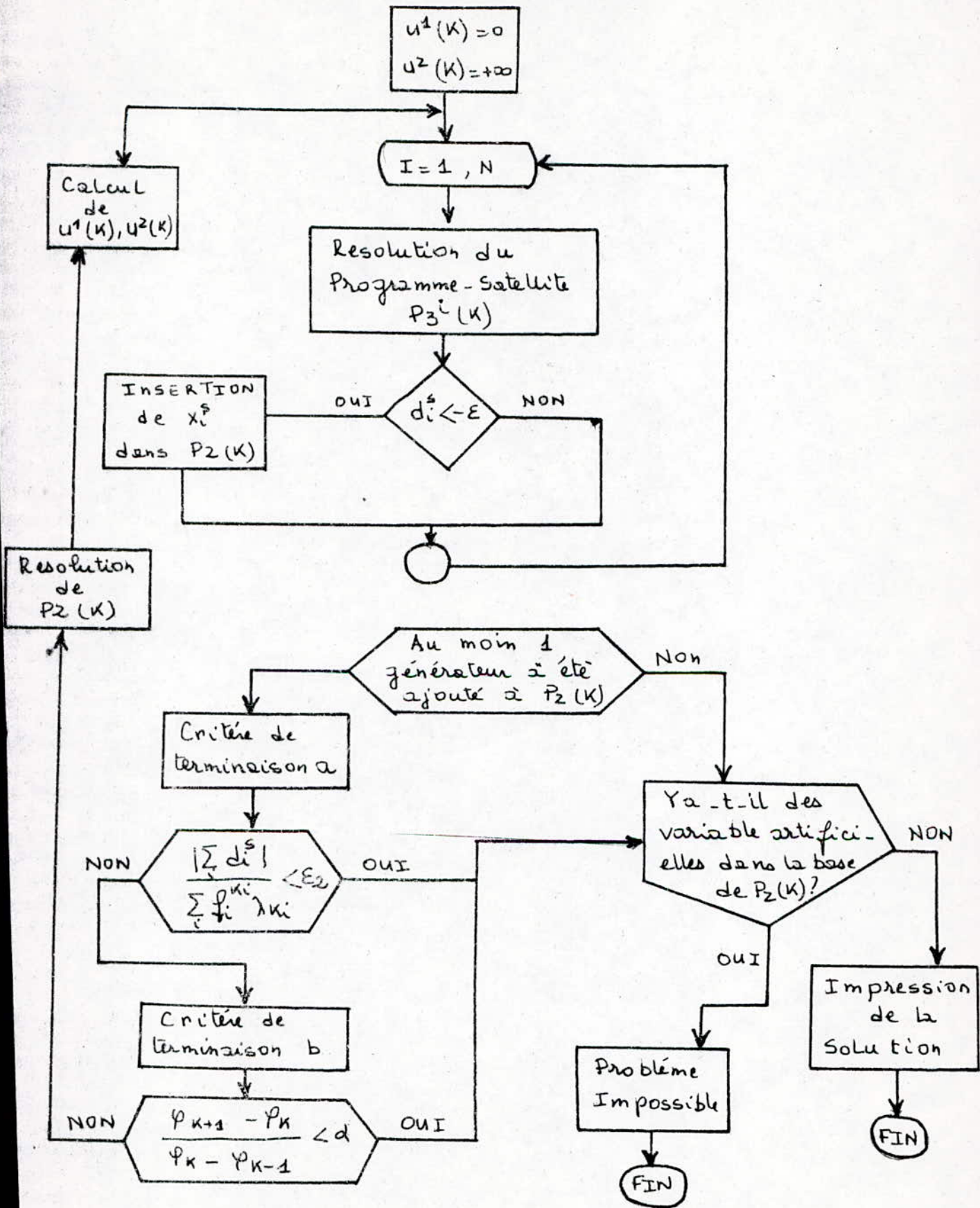
Soit ϕ_K la valeur de la solution de $P_2(K)$ à l'itération numéro K_i on arrête les itérations dès que :

$$\frac{\phi_{K+1} - \phi_K}{\phi_K - \phi_{K-1}} < d \quad 0 < d < 1$$

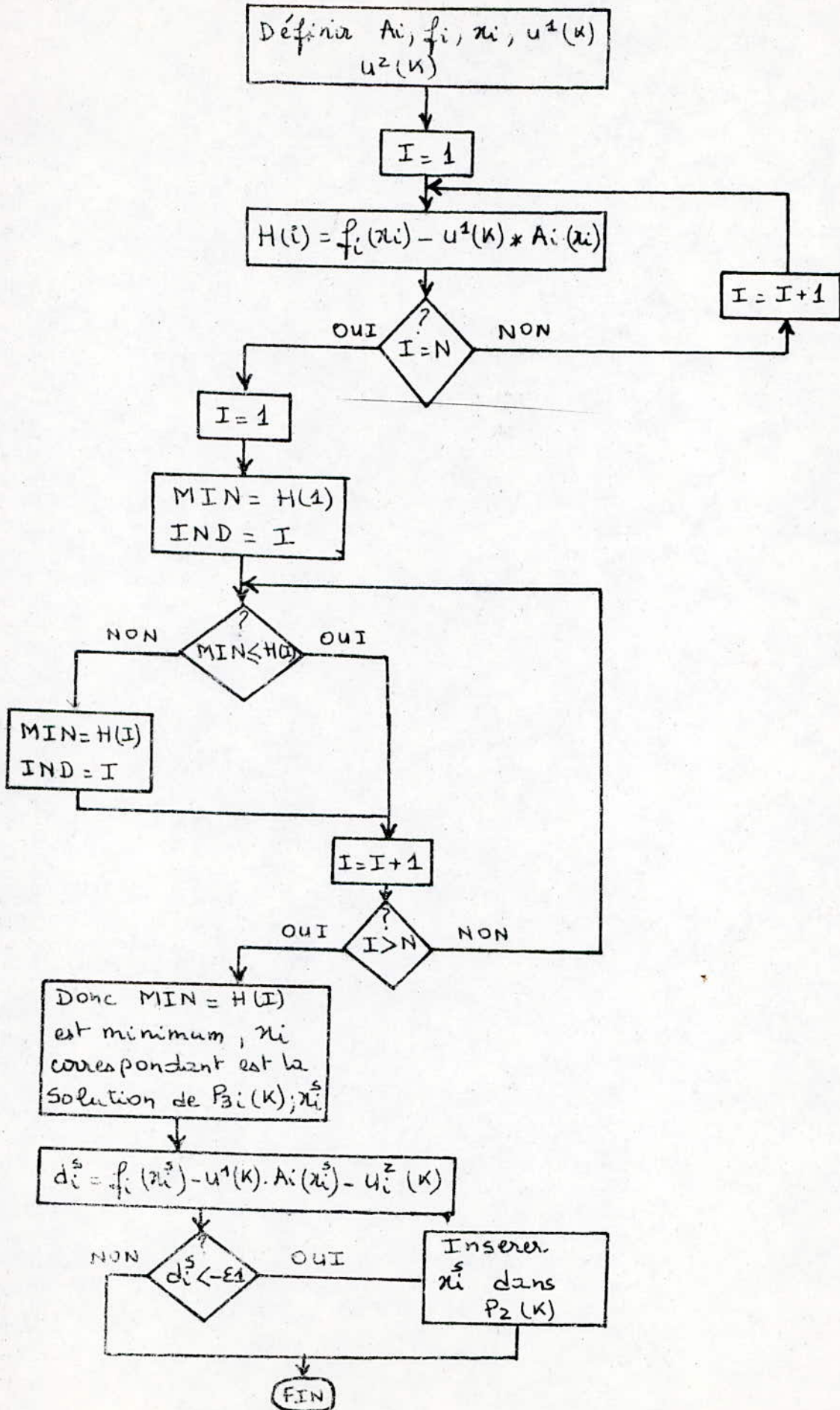
$$\phi_{K+1} - \phi_K \neq 0$$

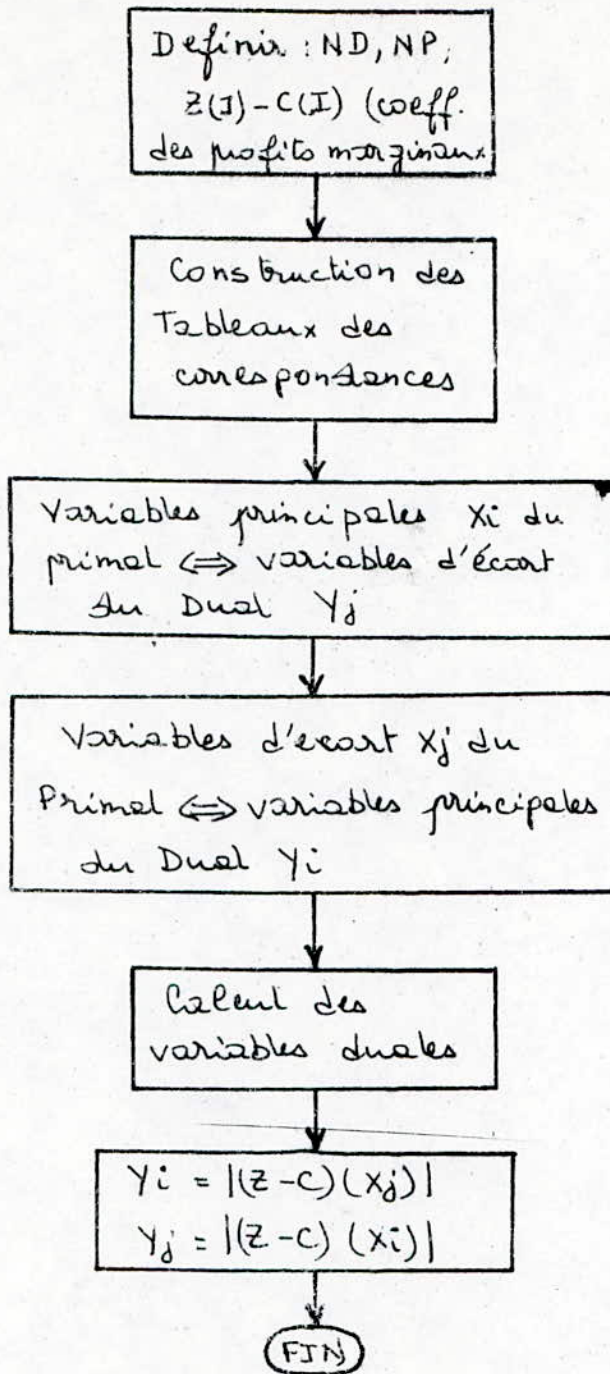
pour ne pas obtenir une solution très éloignée de la solution optimum, d doit être choisi suffisamment petit.

L'inconvénient de la méthode est que la variation de ϕ_K au cours des itérations progressent par sauts plutôt que régulièrement, on risque de terminer l'algorithme trop tôt.

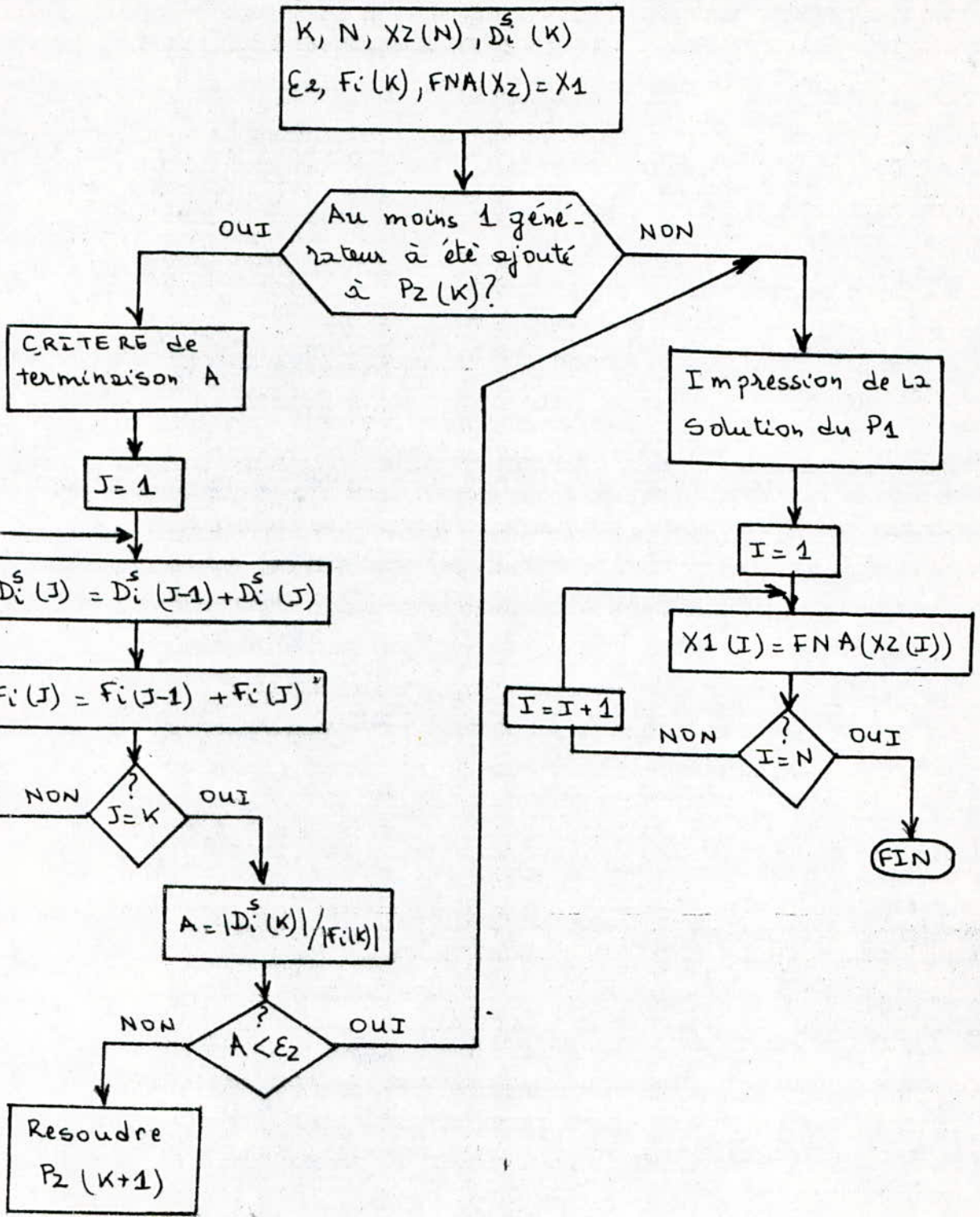


I- ORGANIGRAMME GENERAL DE LA METHODE DE DANTZIG-WOLFE (DZ)





3- ORGANIGRAMME DU TABLEAU DES
CORRESPONDANCES PRIMAL-DUAL ET
CALCUL DES VARIABLES DUALES.



4- ORGANIGRAMME DE LA FIN DE L'ALGORITHME DE DANTZIG-WOLFE .

III . 3 . 2 . ALGORITHME D'UZAWA (U Z W)

III . 3 . 2 . 1 . GENERALITES

Tout comme l'algorithme de DANTZIG - WOLFE, l'algorithme d'UZAWA est un algorithme de coordination basé sur la décomposition du problème en problèmes locaux qui peuvent être résolus indépendamment les uns des autres.

Mais si l'algorithme de DANTZIG - WOLFE est fondé sur la méthode simpliciale qui est une méthode de programmation linéaire, l'algorithme d'UZAWA (UZW) se base sur la recherche du point selle.

En effet, une condition nécessaire et suffisante de coordonnabilité est l'existence du point - selle.

Qu'est ce qu'un point-selle ?

La notion de point selle est surtout utilisée dans la théorie des jeux de stratégie développée à partir des travaux de BOREL et de VON NEUMANN (7). L'intérêt de cette théorie tient davantage dans la manière d'aborder les problèmes de combat ou de compétition que dans leur traitement systématique. A partir de jeux sur des tableaux rectangulaires, on peut définir un comportement rationnel face aux décisions adverses et faire apparaître un état d'équilibre dont l'importance est primordiale. Un point selle n'est autre qu'un point d'équilibre d'un jeu rectangulaire.

Définition :

On dit qu'un couple $\{\bar{u}, \bar{v}\} \in C \times D$ est un point selle de L sur $C \times D$ si

$$(2.1) \quad L(\bar{u}, \bar{v}) \leq L(\bar{u}, v) \leq L(u, \bar{v}) \quad \begin{matrix} \forall u \in C \\ \forall v \in D \end{matrix}$$

Avec C et D deux ensembles dont on précisera les propriétés

et L une fonction de $C \times D$ dans $\overline{\mathbb{R}} = [-\infty, +\infty]$

Condition nécessaire et suffisante d'existence de points selles pour les fonctionnelles :

- Proposition n° 1 :

Si L est une fonction définie sur $C \times D$ à valeur dans $\overline{\mathbb{R}}$

$$(2.2) \quad \sup_{v \in D} \inf_{u \in C} L(u, v) \leq \inf_{u \in C} \sup_{v \in D} L(u, v)$$

Démonstration :

On a en effet d'après (2.1)

$$L(U, V) \leq L(\xi, V) \quad \forall \xi \in C, \quad V \in D$$

$$\inf_{u \in C} L(U, V) \leq L(\xi, V)$$

$$\text{d'ou } \sup_{V \in D} \inf_{U \in C} L(U, V) \leq \sup_{V \in D} L(\xi, V)$$

$$\text{et } \sup_{V \in D} \inf_{U \in C} L(U, V) \leq \inf_{\xi \in C} \sup_{V \in D} L(\xi, V)$$

- proposition n°2 :

La fonction L définie sur C X D à valeur dans ... possède un point selle sur C X D si et seulement si :

$$(2.3) \quad \max_{V \in D} \inf_{U \in C} L(U, V) = \min_{U \in C} \sup_{V \in D} L(U, V)$$

et ce nombre est égale à L(\bar{U}, \bar{V})

Démonstration :

On suppose qu'il existe 1 point selle $\{\bar{U}, \bar{V}\}$

Alors on a d'après (2.1)

$$(2.4) \quad \sup_{V \in D} L(\bar{U}, V) = L(\bar{U}, \bar{V}) = \inf_{U \in C} L(U, \bar{V})$$

Mais

$$(2.5) \quad \inf_{U \in C} \sup_{V \in D} L(U, V) \leq \sup_{V \in D} L(\bar{U}, V)$$

$$(2.6) \quad \inf_{U \in C} L(U, \bar{V}) \leq \sup_{V \in D} \inf_{U \in C} L(U, V)$$

de sorte que

$$\inf_{U \in C} \sup_{V \in D} L(U, V) \leq \sup_{V \in D} \inf_{U \in C} L(U, V)$$

d'une part et (2.2) d'autre part, on voit qu'il y a égalité dans (2.5) et (2.6) ce qui donne.

$$\begin{aligned} L(\bar{U}, \bar{V}) &= \sup_{V \in D} L(\bar{U}, V) = \min_{U \in C} \sup_{V \in D} L(U, V) \\ &= \inf_{U \in C} \sup_{V \in D} L(U, V) = \max_{V \in D} \inf_{U \in C} L(U, V) \end{aligned}$$

Supposons réciproquement que (2.3) ait lieu, le minimum étant atteint en U et le maximum en V. Il est clair que :

$$(2.7) \quad \inf_{U \in C} L(U, \bar{V}) \leq L(\bar{U}, \bar{V}) \leq \sup_{V \in D} L(\bar{U}, V)$$

et d'après (2.3) les inégalité de (2.7) sont en fait des égalités ce qui montre que U, V est un point selle de L.

Remarque :

Si la fonction L possède un point selle, on a alors en particulier (2.8)

$$\sup_{V \in D} \inf_{U \in C} L(U, V) = \inf_{U \in C} \sup_{V \in D} L(U, V)$$

Existence de point selle :

Théorème :

On suppose que les conditions suivantes sont satisfaites :

(2.9) - $C \subset U$ est concave fermé, non vide

(2.10) - $\forall v \in D, U \rightarrow L(U, V)$ est quasi - convexe

(la quasi-convexité signifie que $\forall v \in D, \forall \lambda \in \mathbb{R}$

l'ensemble $\{ U \in C / L(U, V) \leq \lambda \}$ est convexe).

Et en outre C est compact dans U, D est compact dans V

Alors la fonction L possède au moins un point selle $\{\bar{U}, \bar{V}\}$ sur $C \times D$ et

$$(2.11) \quad L(\bar{U}, \bar{V}) = \min_{U \in C} \max_{V \in D} L(U, V) = \max_{V \in D} \min_{U \in C} L(U, V)$$

Corollaire :

Les conclusions du théorème restent valables sous les hypothèses (2.9) - (2.10) et (2.12) C est borné dans U, D est borné dans V

Remarque :

Le théorème s'étend aisément au cas où U et V sont des espaces vectoriels topologiques séparés. Il n'en est pas de même pour le corollaire.

III . 3 . 2 . 2 . PRINCIPE GENERAL

Définition du problème :

Soit le problème PA suivant :

$$\begin{aligned} \min \quad & \sum_{i=1}^N J_i (V_i) \\ & V_i \in U_i^{\infty} \\ & \sum_{i=1}^N B_i (V_i) \leq 0 \end{aligned}$$

et son lagrangien

$$L(v, q) = \sum_{i=1}^N J_i (v_i) + \langle q, \sum_{i=1}^N B_i (V_i) \rangle$$

$$q \geq 0$$

Le principe de l'algorithme d'UZAWA est de décomposer le problème global PA en problèmes locaux PA_i et de s'assurer de la coordonnabilité de ces problèmes, de sorte que les solutions des problèmes locaux PA_i soient solutions du problème global PA.

Pour ce faire on fixe q à 1 valeur quelconque et les problèmes locaux PA_i se mettront de la forme.

$$PA_i \quad \begin{aligned} \min \quad & J_i (v_i) + \langle q, B_i(V_i) \rangle \\ & V_i \in U_i^{\infty} \end{aligned}$$

Notation :

$J_i, i = 1.. n,$ sont les critères du centre de décision.

$V_i, i = 1, \dots, n,$ sont les variables de décision appartenants à U qui est un espace vectoriel normé, donc U est un vecteur de décision.

En générale $V \in U^{\infty}$ qui est un sous ensemble de contraintes inclu dans U.

B_i est une application de U_i dans \mathcal{Y} espace vectoriel normé.

$\sum_{i=1}^N B_i (V_i) \leq 0$ représente les contraintes du problème.

- q est un artifice mathématique qui permet de décomposer le problème PA.

Le problème PA étant décomposé en problèmes locaux PA_i, comment prouver que la solution U_i de PA est bien la solution des PA_i ?

Ceci peut se faire en recherchant la coordonnabilité des PA_i. Or l'étude de la coordonnabilité des PA_i est ramenée à la recherche du point selle, d'après le théorème suivant :

Théorème (1)

La condition nécessaire et suffisante pour que les PA_i soient coordonnables sans pertes est que la fonctionnelles :

$$\sum_{i=1}^N J_i(v_i) + \langle q, \sum_{i=1}^N B_i(v_i) \rangle$$
 possède un point - selle sur l'ensemble

$$\{ q \geq 0 ; v_i \in U_i^\infty \}$$

démonstration voir annexe 3.

Or la fonctionnelle

$$\sum_{i=1}^N J_i(v_i) + \langle q, \sum_{i=1}^N B_i(v_i) \rangle$$

représente le lagrangien associé au problème PA

Alors le problème de recherche du point selle se rapporte à :

$$\begin{aligned} & \text{Max} \min L(v, q) \\ & q \geq 0 \quad v \end{aligned}$$

qui est équivalent au problème

$$\text{Max } M(q) = \sum_{i=1}^N J_i(u_i(q)) + \langle q, \sum_{i=1}^N B_i(u_i(q)) \rangle$$

avec $u_i = u_i(q)$ les solutions des PA_i et $M(q)$ une fonctionnelle dépendant de q . (1)

La fonctionnelle $M(q)$ est associée au problème PA posé et dépend de ce dernier.

Théorème :

Si les problèmes locaux PA_i ont une solution, $\forall q \geq 0$
alors la fonctionnelle $M(q)$ est concave en q .

Démonstration

Considérons $M_i(q) = J_i[v_i(q)] + \langle q, B_i[v_i(q)] \rangle$ on a par définition de $u_i(q)$.

$$M_i(q) \leq J_i(v_i) + \langle q, B_i(v_i) \rangle \quad \forall v_i \in U_i^\infty$$

Soit pour q_1 et $q_2 \geq 0$

$$\begin{aligned} M_i(q_1) & \leq J_i(v_i) + \langle q_1, B_i(v_i) \rangle \quad \forall v_i \in U_i^\infty \\ M_i(q_2) & \leq J_i(v_i) + \langle q_2, B_i(v_i) \rangle \quad \forall v_i \in U_i^\infty \end{aligned}$$

Soit la combinaison convexe de $M_i(q_1)$, $M_i(q_2)$ suivante

$$\begin{aligned} \alpha M_i(q_1) + (1 - \alpha) M_i(q_2) & \leq J_i(v_i) + \langle \alpha q_1 + (1 - \alpha) q_2, B_i(v_i) \rangle \\ \forall v_i \in U_i \end{aligned}$$

soit $U_i (\alpha q_1 + (1-\alpha) q_2)$ minimisant le second membre, on en déduit

$$\alpha M_i(q_1) + (1-\alpha) M_i(q_2) \leq J_i [u_i(\alpha q_1 + (1-\alpha) q_2)] + \langle \alpha q_1 + (1-\alpha) q_2, B_i(u_i(-)) \rangle \leq M_i(\alpha q_1 + (1-\alpha) q_2)$$

On est donc ramené au problème de la maximisation d'une fonctionnelle concave sur l'orthant positif (1).

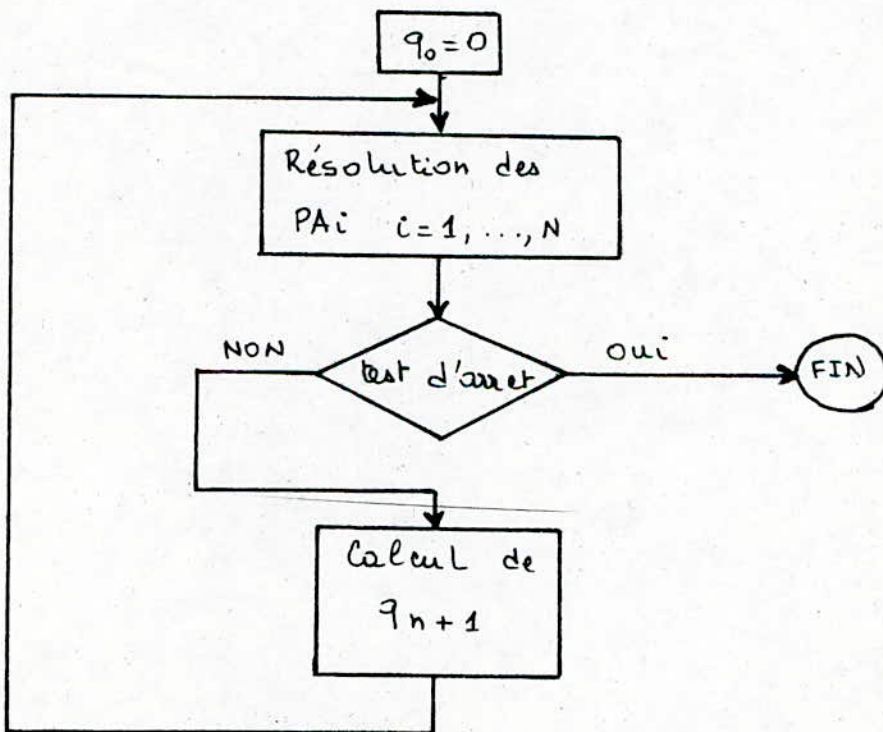
D'après ce théorème, nous savons que $M(q)$ est concave.

Nous pouvons alors montrer facilement que si, dans le cas particulier choisi, la fonction critère est convexe linéaire par morceaux alors $M(q)$ est concave - linéaire par morceaux.

ET ceci facilite beaucoup l'obtention du résultat, par le fait qu'il n'y a pratiquement aucun calcul pour déterminer le gradient de $M(q)$.

D'après ce qui précède on peut dès lors donner un organigramme de principe de l'algorithme UZW.

Organigramme de principe



Convergence numérique et critère d'arrêt:

L'idéale lors de la résolution de l'algorithme d'UZAWA serait d'évaluer, au cours des itérations, la valeur des solutions obtenues. C'est à dire qu'à chaque itération faire un test comparant la solution obtenue avec une certaine valeur qui serait l'écart minimum entre la solution obtenue et la solution qu'on devrait obtenir.

Ceci, surtout que les solutions successives n'appartiennent pas au domaine des contraintes.

Les deux théorèmes suivants (dûs à EVERETT) nous permettent de construire une borne supérieure de l'erreur commise. (1)

Théorème (2)

Etant donné $q \geq 0$ et $U_i(q)$ les solutions des PA_i $i = 1, n$

$$PA_i \quad \begin{cases} \min J_i(V_i) + \langle q, B_i(v_i) \rangle \\ V_i \in U_i \end{cases}$$

Alors les $U_i(q)$ sont solutions du problème global PA'

$$PA' \quad \begin{cases} \min \sum_{i=1}^N J_i(V_i) \\ V_i \in U_i \\ B_i(V_i) \leq \sum_{i=1}^N B_i(U_i(q)) \end{cases}$$

Démonstration voir annexe 4

Interprétation

Ce théorème dit que, pour tout $q \geq 0$, les solutions $U_i(q)$ des PA_i , si elles existent, sont aussi les solutions d'un problème global PA' dont la contrainte s'écrit :

$$\sum_{i=1}^N B_i(V_i) \leq \sum_{i=1}^N B_i(U_i(q))$$

Donc, pour déterminer le problème global qui est résolu exactement par résolution des problèmes locaux il suffit de bien choisir q .(1).

Malheureusement ce problème n'est pas connu à l'avance, mais défini par les solutions mêmes des problèmes locaux.

On peut toutefois dire que, au cours de la résolution de l'algorithme d'UZAWA on résoud à chaque itération un problème global approché du problème global posé, jusqu'à en obtenir une approximation aussi satisfaisante que possible.

Théorème (3)

Si, pour $q \geq 0$ donné, $U_i(q)$ minimise

$$L(v, q) = \sum_{i=1}^N J_i(v_i) + \langle q, \sum_{i=1}^N B_i(v_i) \rangle$$

avec une erreur inférieure à ϵ , alors $U_i(q)$ est solution du problème PA' suivant à ϵ près.

$$\begin{aligned} & \min \sum_{i=1}^N J_i(v_i) \\ & v \in U_i \quad \infty \\ & \sum_{i=1}^N B_i(v_i) \leq \sum_{i=1}^N B_i(U_i(q)) \end{aligned}$$

PA'

Démonstration voir annexe 5

Interprétation :

Ce théorème permet de connaître la valeur de l'approximation obtenue. Supposons qu'au cours des itérations, on ait trouvé un $q \geq 0$ tel que les solutions $U_i(q)$ des PA_i ($i = 1, \dots, n$) correspondants définissent une contrainte :

$\sum_{i=1}^N B_i(v_i) \leq \sum_{i=1}^N B_i(U_i(q))$ " Bonne ", mais encore trop éloignée de la contrainte réelle :

$$\sum_{i=1}^N B_i(v_i) \leq 0$$

Alors pour s'approcher aux mieux de la contrainte réelle, on conçoit une procédure de modification, qui à partir de la solution $U_i(q)$ ($i=1, \dots, n$) la modifie de sorte que les contraintes du problème global initial soient respectées. On obtient la solution modifiée $U_i(q)$ ($i = 1, \dots, n$)

La procédure de modification sera expliquée de façon détaillée dans l'exposé de l'algorithme.

.../...

III 3 . 2 . 3 . ALGORITHME

Nous nous proposons d'étudier un problème de nature économique qui comporte :

1 - n vecteurs de décision

$$V_i \quad i = 1, \dots, n$$

2 - n fonctions critéres

$$J_i \quad i = 1, \dots, n$$

3 - et n fonctions contraintes

$$B_i \quad i = 1, \dots, n$$

Pour chaque fonction critère où contrainte on applique les n vecteurs de décision, tel que :

J _{1,1}	J _{1,2}	J _{1,n}	
J _{2,1}	J _{2,2}	J _{2,n}	n fonctions critéres
.			.	
.			.	
.			.	
.			.	
J _{n,1}	J _{n,2}	J _{n,n}	

n. vecteur de décision

idem pour B_i

Ce problème est décomposé en problèmes locaux, et pour résoudre chaque problème local on procède suivant une méthode de monté itérative.

Pour un q fixé (à zéro), on résoud les P_{Ai}, c'est à dire qu'on fixe les fonctions J et B à une valeur et on fait varier les vecteurs de décision de i = 1 à i = n, et on recherche la valeur du vecteur qui va minimiser le P_{Ai}, ce qui correspondra à la solution du problème local, pour cette valeur de J et de B. On incrémente ensuite J et B et on refait le même travail, à la fin nous aurons donc n solutions pour n problèmes locaux et chaque solution sera unique.

Pour vérifier que les n solutions que nous avons trouvé sont bien solution du problème global, on procède à une série d'opérations qui constitueront un test d'arrêt.

Si ce test est positif, c'est la FIN du problème si non on calcule q_n + 1 qui permettra de passer à l'itération suivante et on reprend tout l'algorithme.

La série d'opération consiste en ce qui suit :

1) On calcule la norme suivante :

$$\left\| \sum_{i=1}^N B_i (U_i(q)) \right\|$$

* Rappel sur les normes :

1) Norme :

On désigne par K le corps \mathbb{R} des nombres réels ou le corps \mathbb{C} des nombres complexes.

Définition :

Soit E un K -espace (K est \mathbb{R} ou \mathbb{C}). On appelle norme sur E une application, notée $X \rightarrow \|x\|$ de E dans \mathbb{R}^+ ayant les propriétés suivantes : (8)

(1) $\|x\| = 0 \Leftrightarrow x = 0$

(2) $\|\lambda \cdot x\| = |\lambda| \cdot \|x\| \quad \forall \lambda \in K, \forall x \in E$

(3) $(\forall x, y \in E) \quad \|x+y\| \leq \|x\| + \|y\|$

E est alors nommé espace vectoriel normé

2) Distance associé à une norme

Soit E un K -espace normé. On appelle distance associée à la norme $\| \cdot \|$, l'application d :

$$E \rightarrow \mathbb{R}^+, \text{ définit par } d(x, y) = \|x - y\|$$

Tout espace vectoriel normé est donc un espace métrique avec la distance associée, et, les propriétés des espaces métriques s'appliquent intégralement aux espaces normés.

Alors :

L'application $X \rightarrow \|X\|$ définie par $\|X\| = \sqrt{X \cdot X}$ est une norme sur E , nommé norme euclidienne.

Si, E est de dimension finie et $B = \{e_1, \dots, e_n\}$ une base orthonormale de E , pour tout $X = X_1 e_1 + \dots + X_n e_n$, la norme euclidienne est :

$$\|X\| = \sqrt{X_1^2 + X_2^2 + \dots + X_n^2}$$

Dans notre cas nous allons donc utilisé la norme euclidienne de

$$\left\| \sum_{i=1}^N B_i (U_i)(q) \right\| = \sqrt{\sum_{i=1}^N B_i (U_1(q))^2 + \dots + \left(\sum_{i=1}^N B_i (U_n(q))^2 \right)}$$

Par ce calcul on veut évaluer la distance (d'où notion de norme) qui existe entre la solution optimum et la solution trouvée, en d'autre termes, si la solution trouvée s'approche au mieux de la solution optimum.

Pour le calcul, on prend la première valeur de $U_i \equiv U_1(q)$ et on fait varier B_i de $i = 1$ à $i = n$. On additionne toutes les fonctions $B_i (U_1(q))$ $i = 1, n$, et on élève au carré, on passe à $U_2(q)$ et on procède de la même manière.

Une fois qu'on a fait cela n fois, on calcul alors la racine carré de la somme de tous les $\sum_{i=1}^N B_i (U_j(q))^2$ j fixé.

C'est ce qu'on appelle la norme euclidienne.

2 - On compare cette norme à une certaine valeur positive η qui doit être aussi petite que possible car elle est l'erreur permise entre la solution trouvée et la solution optimum. et à chaque itération la valeur de η doit être diminuer pour affiner au fur et à mesure des itérations le domaine d'approche de la solution optimum.

3 - Si la norme calculée est inférieure a η ce qui veut dire que les U_i $i = 1, N$ sont dans le domaine proche de la solution optimum, mais, qu'ils ne sont pas encore égales à cette solution, on procède à une modification de U_i $i = 1, N$ qui va donner $\tilde{U}_i(q)$ $i = 1, N$.

* Pourquoi cette modification ?

Pour pouvoir apprécier la solution fournie par l'algorithme, on construit une borne supérieure de l'erreur commise. D'après le théorème (2) (Vu précédemment) les solutions des PAi sont aussi solutions d'un problème global ayant comme contrainte :

$$\sum_{i=1}^N B_i (V_i) \leq \sum_{i=1}^N B_i (U_i(q))$$

Imaginons que l'on trouve les solutions des PAi vérifiant cette condition mais que cette contrainte est encore loin de la contrainte réelle

$$\sum B_i (V_i) \leq 0$$

C'est pour cela qu'on effectue la procédure de modification de la solution trouvée.

* Comment faire cette modification

Cette procédure de modification n'est pas indépendante du problème résolu, mais doit être réécrite en fonction de chaque nouveau problème. C'est là l'un des inconvénients de l'algorithme d'UZAWA comme nous le verrons plus loin

Une chose est toutefois certaine ; si les U_i ($i = 1, \dots, n$) vérifient la contrainte

$$\sum_{i=1}^N B_i (V_i) \leq \sum_{i=1}^N B_i (U_i(q))$$

et non $\sum_{i=1}^N B_i (V_i) \leq 0$

C'est qu'ils sont supérieurs à la solution optimum donc lors de la modification, on fait de sorte que

$$\sum_{i=1}^N B_i(U_i(q)) \text{ soit aussi petit que possible}$$

pour s'approcher au mieux de la contrainte réelle.

C'est à dire qu'on modifie $U_i(q)$ $i = 1, \dots, n$ de façon que $\sum_{i=1}^N B_i(U_i(q))$ tende vers zéro (0).

Les nouvelles solutions obtenues seront $\tilde{U}_i(q)$ $i = 1, \dots, n$.

4 - Une fois cette modification effectuée on veut vérifier que les nouvelles solutions U_i ($i = 1, n$) sont bien les solutions optimums

Alors, on calcule l'écart :

$$\gamma = L(\tilde{U}(q), q) - L(U(q), q)$$

Cette écart γ est en quelque sorte l'écart maximum que l'on peut se permettre entre la valeur optimal des fonctions critères et la valeur de ces même fonctions appliquées aux solutions modifiées c'est à dire $\sum_{i=1}^n J_i(U_i(q))$ pour que nous puissions dire que les \tilde{U}_i ($i = 1, n$) sont bien les solutions optimums du problème globale PA.

D'après ce qui précède il nous paraît évident que γ doit être aussi petit que possible. On le compare donc à ϵ (que nous nous fixerons antérieurement). Si γ est inférieure à ϵ l'algorithme d'UZAWA est terminé et $\tilde{U}_i(q)$ $i = 1, n$ sont les solutions optimums de PA à ϵ près.

Par contre si γ est supérieur ou égal à ϵ on doit faire une nouvelle itération comme dans le cas ou la norme $\| \sum_{i=1}^n B_i(U_i(q)) \|$ est supérieur ou égal à η .

Pour le calcul de γ :

$$\gamma = L [\tilde{U}(q), q] - L [U(q), q]$$

$$\text{or } L(V, q) = \sum_{i=1}^N J_i(V_i) + \langle q, \sum_{i=1}^N B_i(V_i) \rangle$$

on a alors :

$$L[\tilde{U}(q), q] = \sum_{i=1}^N J_i(\tilde{U}_i(q)) + \langle q, \sum_{i=1}^N B_i(\tilde{U}_i(q)) \rangle$$

$$\text{et } L[U(q), q] = \sum_{i=1}^N J_i(U_i(q)) + \langle q, \sum_{i=1}^N B_i(U_i(q)) \rangle$$

On fixe $U(q)$ et $\tilde{U}(q)$ à leur première valeur et on fait la somme des J_i et de B_i ($i = 1, \dots, n$) et on calcule

$L[\tilde{U}(q), q] - L[U(q), q]$, n fois, donc γ sera de dimension n et chaque valeur sera comparée à ϵ et il faut que chaque γ soit inférieur à ϵ pour que les $\tilde{U}_i(q)$ $i = 1, \dots, n$ soient considérés comme les solutions optimales de PA ; le problème global.

Dans le cas contraire, pour pouvoir calculer l'itération suivante on doit trouver la valeur de q_{n+1} qui permet le passage de l'itération n à l'itération $n+1$.

Or dans l'algorithme d'UZAWA ou algorithme du gradient projeté :

$$q_{n+1} = \text{Pr} \left[q_n + \rho_n \nabla M(q) \right]$$

$\nabla M(q)$ représente le gradient de $M(q)$

nous savons que :

$$M(q) = \sum_{i=1}^N J_i(U_i(q)) + \langle q, \sum_{i=1}^N B_i(U_i(q)) \rangle$$

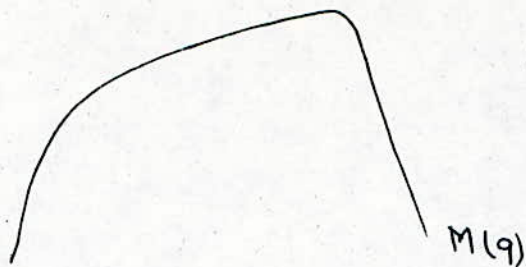
$$\Rightarrow \nabla M(q) = \sum B_i[U_i(q)]$$

ce qui donne bien :

$$q_{n+1} = \text{Pr} \left[q_n + \rho_n \cdot \sum_{i=1}^N B_i(U_i(q)) \right]$$

Pr = opérateur de projection sur le cône positif

Si $M(q)$ est concave en q cela veut dire que $M(q)$ possède un ou plusieurs maximums en q .

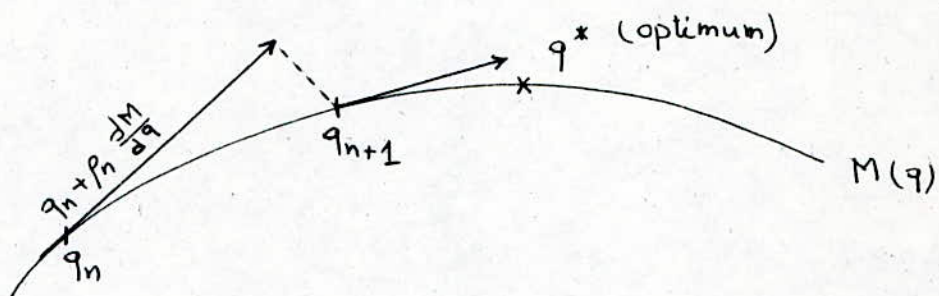


$M(q)$ concave \Rightarrow présente

1 palier \Rightarrow plusieurs maximums

La projection sur le cône positif veut dire que l'on ne prend que la partie positif de $\sum B_i (U_i(q))$ pour $q \geq 0$

On peut représenter graphiquement l'expression de q_{n+1} comme suit



dépend

On choisi q_n , ce choix du problème économique à résoudre, et pour calculer q_{n+1} on prend la direction de la tangente à $M(q)$ en q_n et on fait la projection, on trouve alors q_{n+1} . On procède de la même manière avec q_{n+1} jusqu'à atteindre l'optimum.

L'opérateur de projection donne :

$$\begin{cases} \text{Pr } q_i = \max(q_i, 0) \\ i = 1, \dots, n \end{cases}$$

en d'autres termes :

$$\text{si } F(X) > 0 \Rightarrow \text{Pr}(X) = F(X)$$

$$\text{si } F(X) \leq 0 \Rightarrow \text{Pr}(X) = 0$$

dans notre cas :

On calcule l'expression

$$P = q_n + \beta_n \cdot \sum_{i=1}^N B_i [U_i(q_n)]$$

$$\text{si } P > 0 \Rightarrow q_{n+1} = P$$

$$\text{si } P \leq 0 \Rightarrow q_{n+1} = 0$$

Le choix du q_0 initiale est très important, de sa valeur peut dépendre la convergence de l'algorithme.

Calcul de ρ_n

ρ_n est choisie de sorte à assurer le maximum de

$$M(q_n + 1) - M(q_n)$$

comme il est impossible mathématiquement d'avoir $M(q_n + 1)$ avant le calcul de $q_n + 1$. ρ_n doit être de sorte que :

$$M(q_n + \rho_n \cdot \frac{dM(q)}{dq}) - M(q_n) \text{ soit le plus grand possible.}$$

Dans notre cas :

$$M(q_n + \rho_n \cdot \sum_c B_i(U_i(q))) - M(q_n) \quad (\text{Max})$$

$$M(q_n) = \sum_c J_i(U_i(q)) + \langle q_n, \sum_c B_i(U_i(q)) \rangle$$

$$M(q_n + \rho_n \cdot \sum_c B_i(U_i(q))) = \sum_c J_i(U_i(q)) + \langle q_n + \rho_n \cdot \sum_c B_i(U_i(q)), \sum_c B_i(U_i(q)) \rangle$$

$$M(q_n + \rho_n \cdot \sum_c B_i(U_i(q))) - M(q_n) =$$

$$\langle q_n + \rho_n \cdot \sum_c B_i(U_i(q)), \sum_c B_i(U_i(q)) \rangle - \langle q_n, \sum_c B_i(U_i(q)) \rangle$$

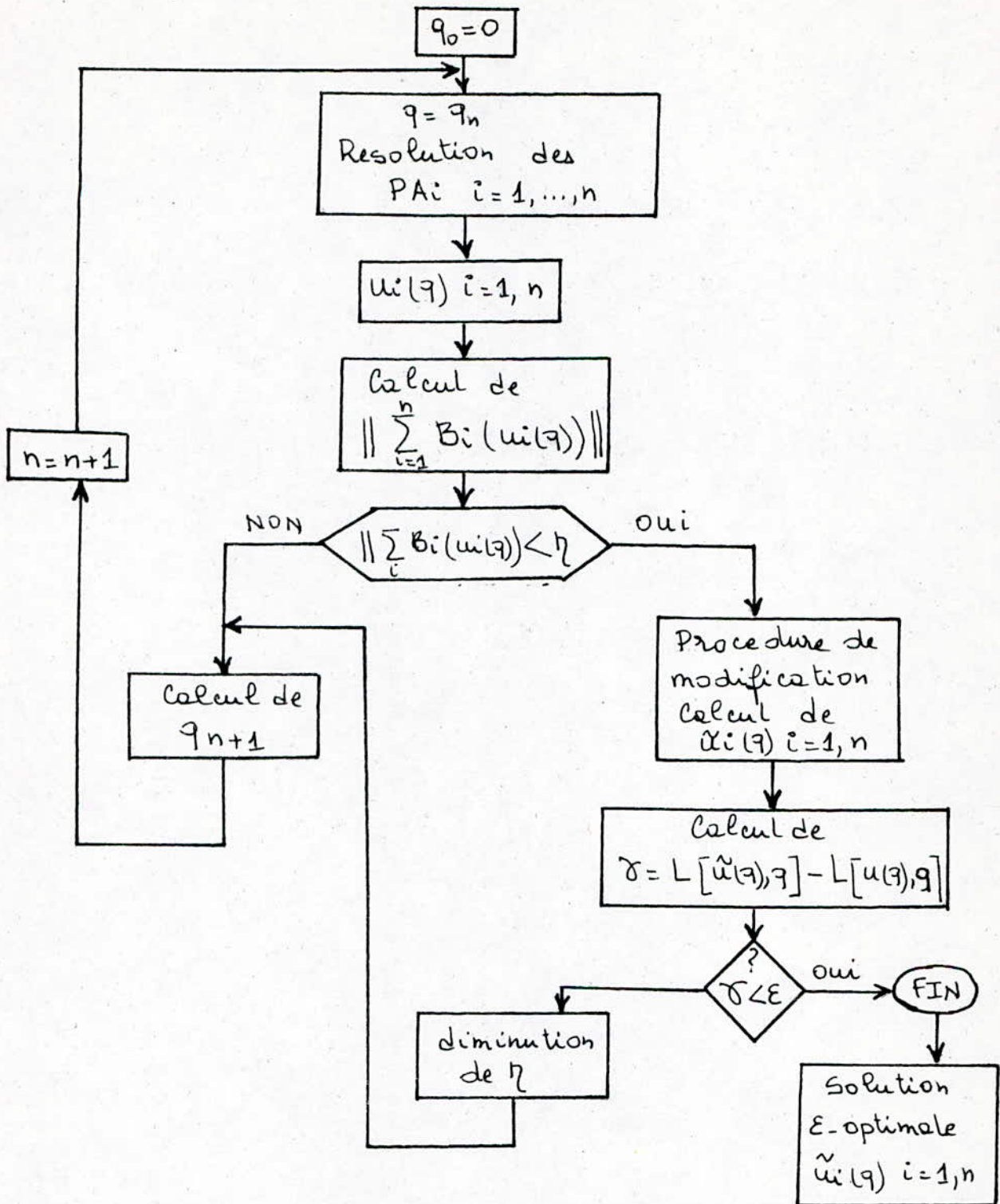
$$= (q_n + \rho_n \cdot \sum_c B_i(U_i(q))) \cdot \sum_c B_i(U_i(q)) \cdot T - (q_n \cdot \sum_c B_i(U_i(q))) \cdot T$$

On doit trouver ρ_n de façon à maximiser cette expression.

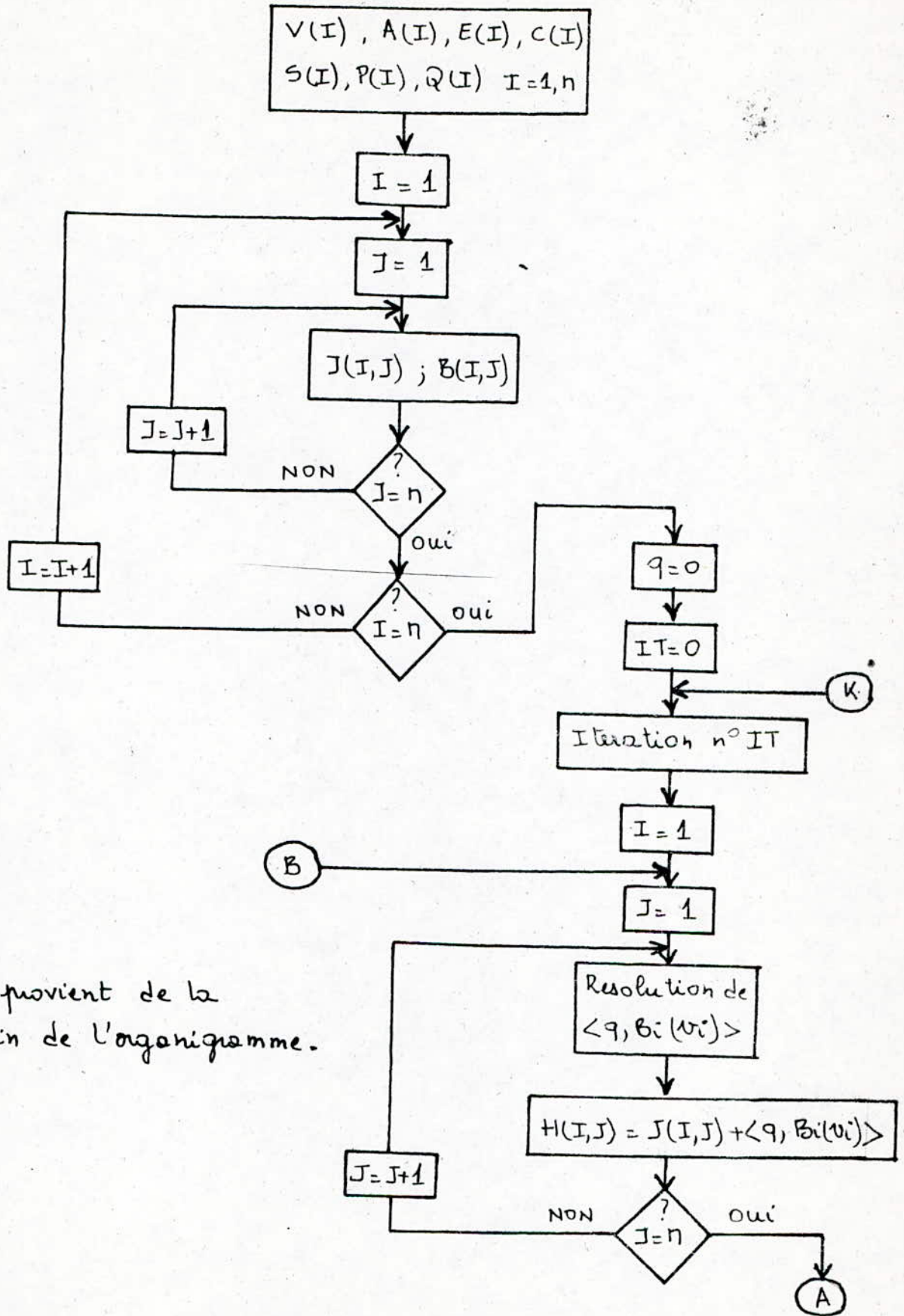
En effet, on fixe ρ_n à une valeur que l'on devra trouver à partir du problème à résoudre lui-même, et on augmente (ou l'on diminue) cette valeur jusqu'à ce que la valeur de $M(q_n + \rho_n \cdot \frac{dM}{dq}) - M(q_n)$ soit maximum.

Une fois $q_n + 1$ calculé on peut alors passer à l'itération $n + 1$ et reprendre l'algorithme en faisant attention de diminuer à chaque itération la valeur de η pour sélectionner le plus rapidement possible la solution optimum.

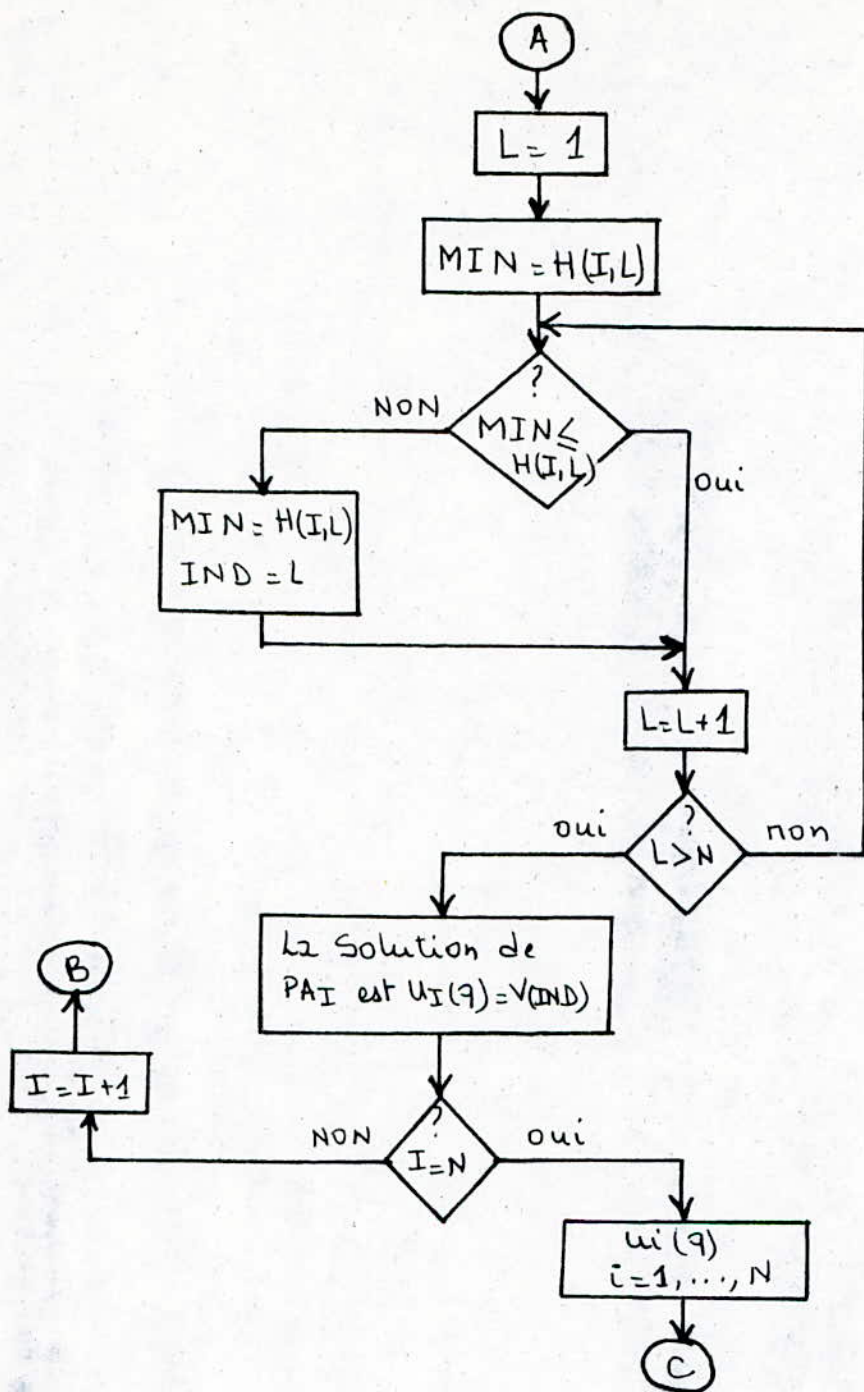
III 3. 2 . 4 . ORGANIGRAMMES



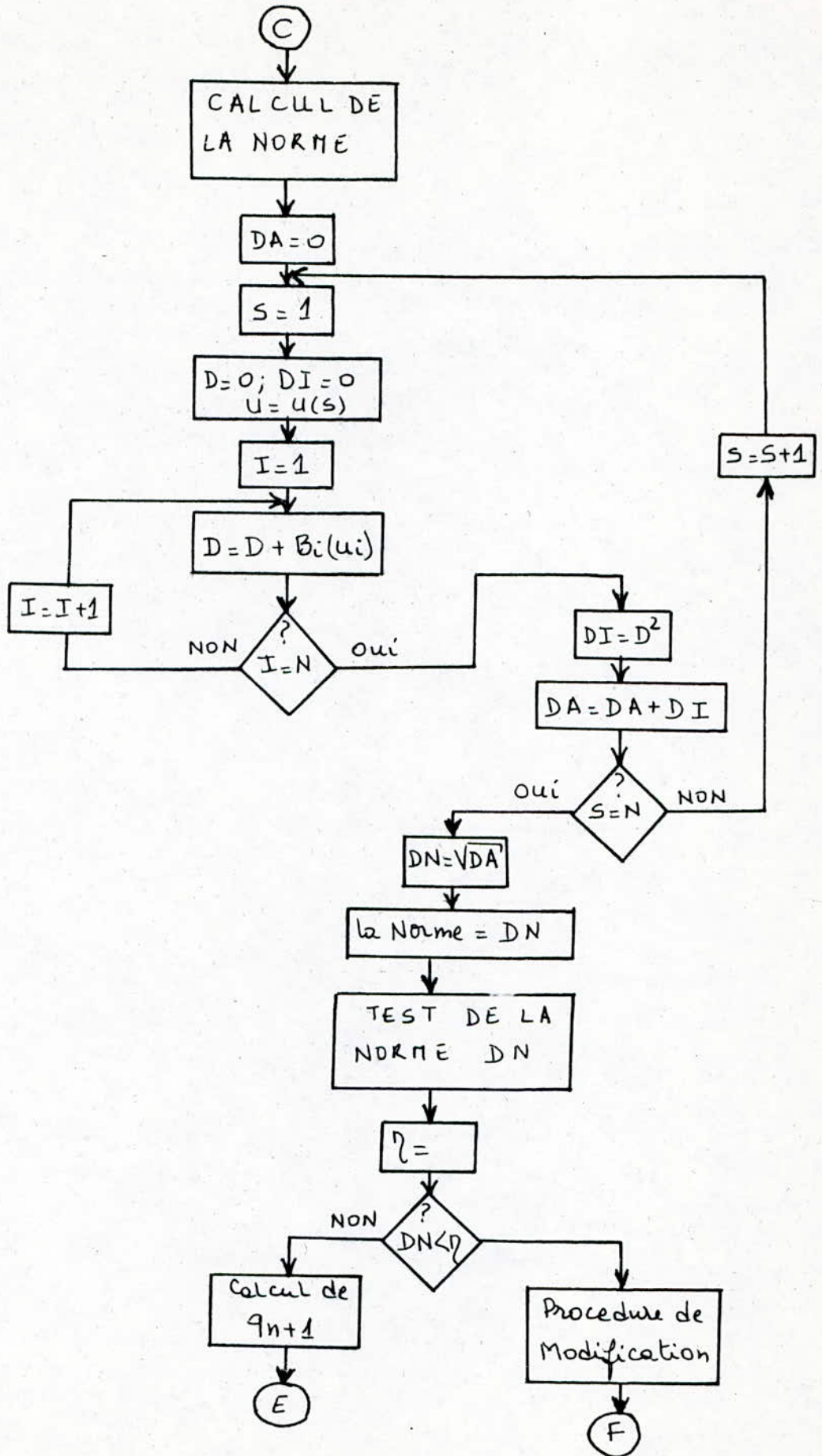
I- ORGANIGRAMME DE L'ALGORITHME MODIFIE D'UZAWA.



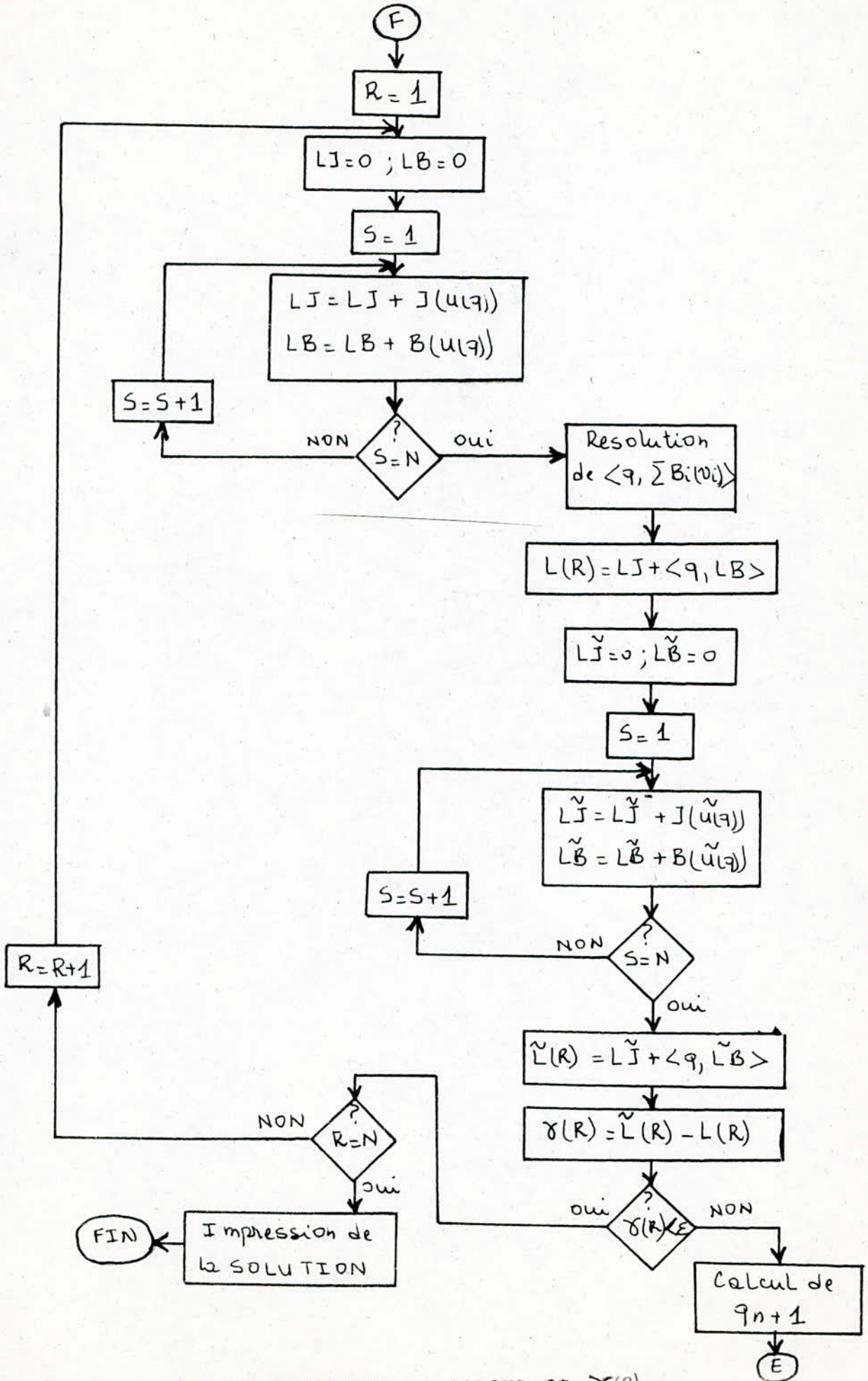
(K) provient de la fin de l'organigramme.



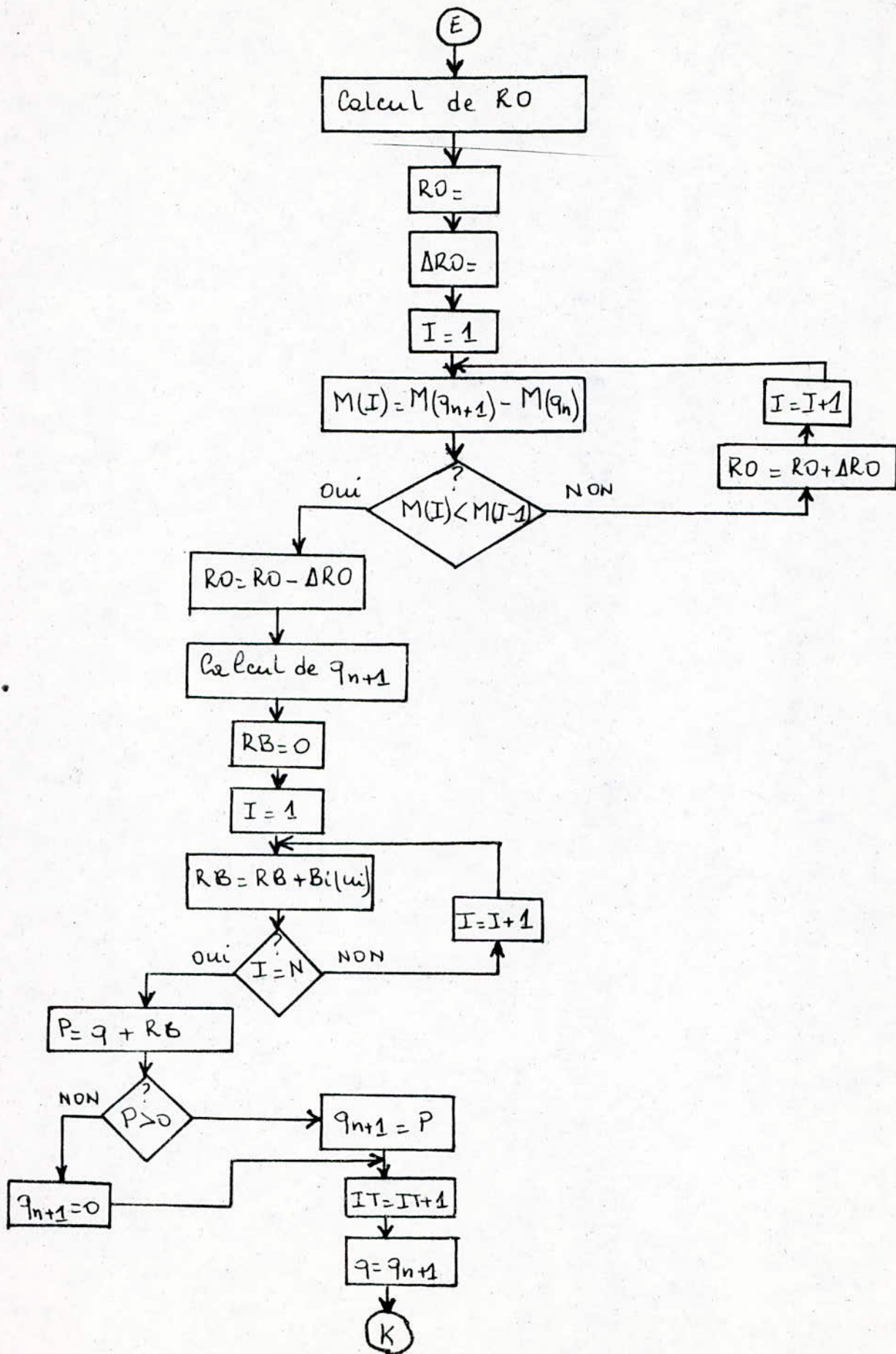
2- ORGANIGRAMME DE LA RESOLUTION
DES PAI.



3- ORGANIGRAMME DU CALCUL ET DU TEST DE LA NORME.



ORGANIGRAMME DU CALCUL DE $\gamma(R)$



5- ORGANIGRAMME DE CALCUL DE Q_{n+1} .

Pour pouvoir faire une application numérique, il faut résoudre le problème de la modification que doivent subir les solutions $U_i(q)$ ($i = 1, \dots, n$). Pour cela nous avons développé l'idée suivante.

Les nouvelles valeurs des solutions doivent vérifier

$$\sum_{i=1}^n B_i(u_i(q)) \text{ tend vers } 0$$

Donc:

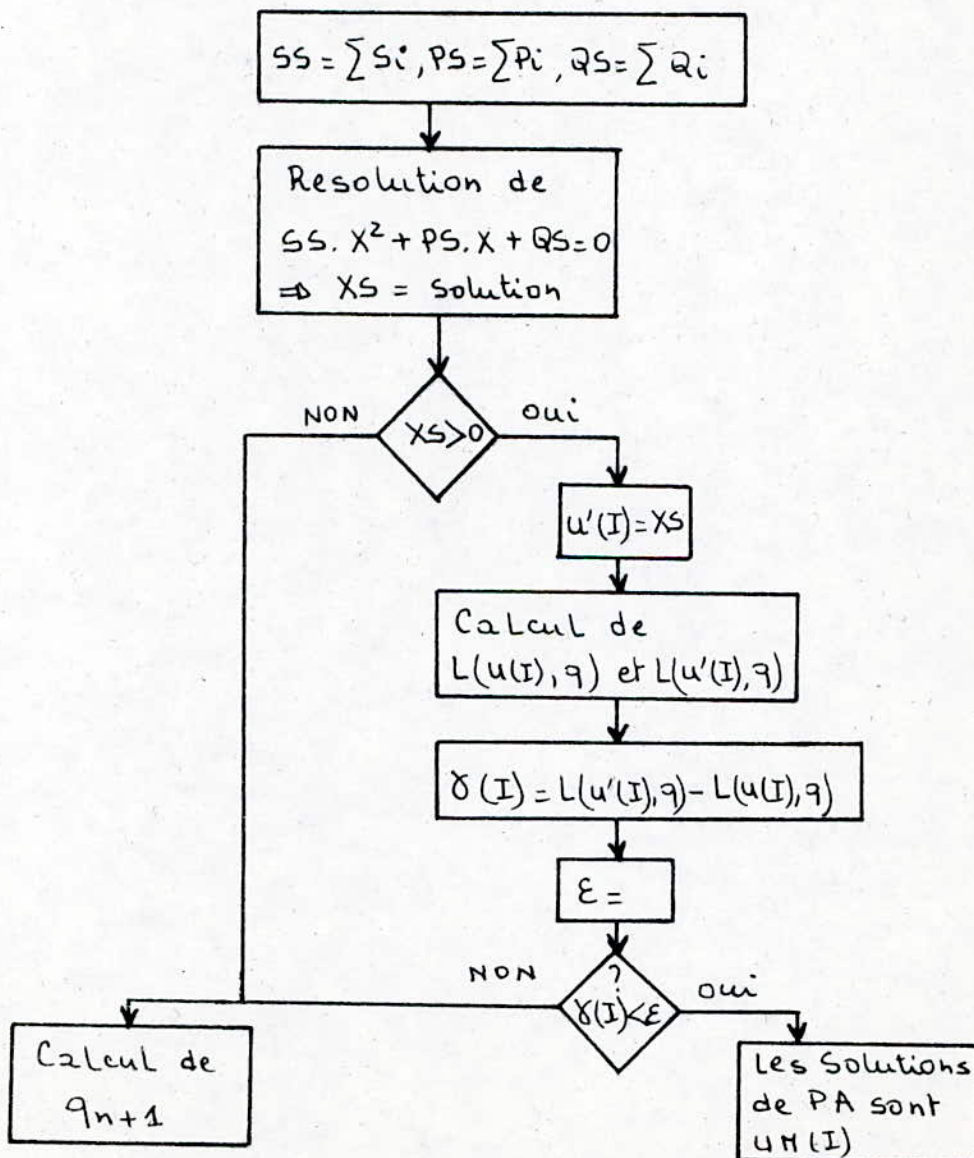
On résout $\sum B_i(X) = 0$, la solution obtenue doit être tel que

$$\delta(I) < \epsilon \text{ avec}$$

$$\delta(I) = L(X, q) - L(U_i(q), \epsilon)$$

$$\text{et } L(u, q) = \sum_{i=1}^n J_i(U_i) + \langle q, \sum_{i=1}^n B_i(U_i) \rangle$$

Organigramme:



III . 4 . SYNTHESE

Après avoir étudié séparément les deux algorithmes de décomposition - coordination, nous pouvons déduire que :

L'algorithme d'UZAMA présente une exécution rapide car la partie coordination du programme est simple et de taille réduite. Donc la mémoire du micro-calculateur nécessaire au programme peut ne pas être importante.

Mais il présente un important inconvénient :

son manque de souplesse, en effet, il est nécessaire de réécrire la procédure de modification à chaque nouveau problème, il est aussi difficile de généraliser cette technique particulièrement si le problème à résoudre présente plusieurs contraintes de couplage (les contraintes de couplage sont les contraintes qui concernent l'ensemble des sous-systèmes). Il n'est plus possible de définir une procédure heuristique simple mais une solution, dans le cas de contraintes linéaires, serait alors d'obtenir une solution réalisable à l'aide d'un simplexe, on perd alors tout l'intérêt de la coordination, puisque finalement on est conduit à résoudre directement le problème global.

Une autre caractéristique de cet algorithme est le choix des constantes q_0 , ρ_n , η et ϵ , qui est très important et influe directement sur la convergence. Un mauvais choix de q_0 et/ou ρ_n peut entraîner la divergence de l'algorithme.

L'algorithme de DANTZIG - WOLFE a la particularité d'être très facile à manipuler : quelque soit la forme de la fonction objective ou des contraintes il suffit d'initialiser le programme en introduisant la matrice du programme principal, opération facilement automatisable. De plus la convergence est assurée sans aucune difficulté.

Ceci est indispensable lorsque nous avons à résoudre de nombreux problèmes du même type mais de structure différentes. L'inconvénient de cet algorithme est sa longueur entraînant un temps de calcul très long et la nécessité d'une capacité de mémoire du micro-calculateur importante.

Nous pouvons alors conclure que :

* l'algorithme d'UZAMA modifié est utilisé lorsque :

- le problème est à structure de couplage fixe vu le manque de souplesse de l'algorithme.

- L'on/ressent la nécessité de construire une procédure de modification.

- la mémoire du micro-calculateur disponible n'est pas importante.

- L'on désire un calcul rapide.

* Par contre l'algorithme de DANIZIG - WOLFE est surtout sollicité lorsque :

- le problème est à structure de couplage variable

- la mémoire du micro-calculateur est importante

- l'on désire des solutions intermédiaires réalisables.

- ° - CHAPITRE IV - ° -

- ° - A P P L I C A T I O N - ° -

IV - 1 . GENERALITES

L'application que nous avons choisie traite d'un problème de production et de gestion des stocks, préoccupation principale de nombreuses entreprises.

Le rôle du système de gestion de production et de piloter le système physique de production (15), il présente différentes fonctions qui, indépendantes mais aussi coordonnées et synchronisées entre elles, permettent les prises de décision par mi ces fonctions nous citons :

- la gestion des données techniques et des ressources,
- la planification,
- le suivi de production,
- la gestion des approvisionnement et des stocks,
- la fonction achat,
- la fonction gestion ^{des} coûts

Chaque fonction donne naissance à un système informatique spécifique. De ce fait, il est important, avant d'automatiser, d'analyser le mode de gestion (15).

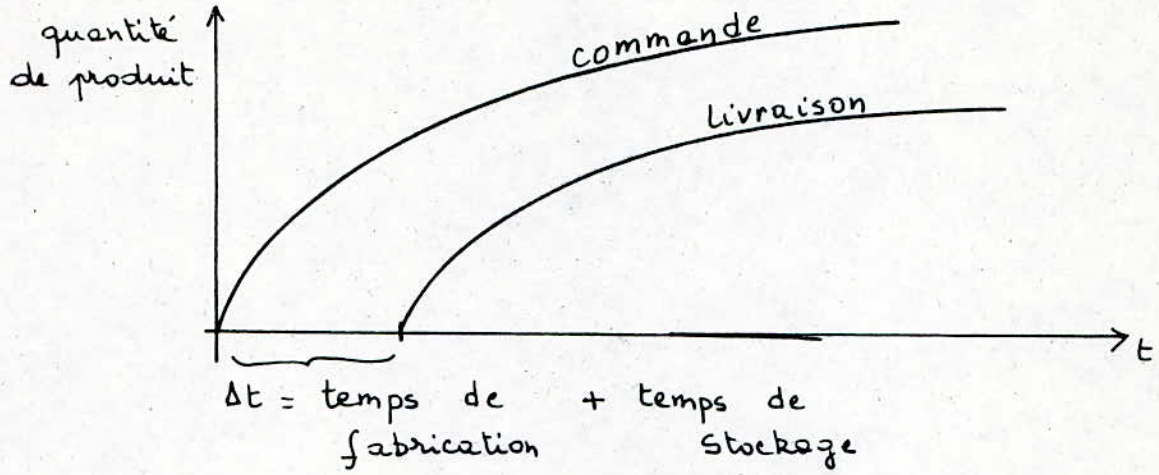
Dans le cas de notre application, la production est optimisée en utilisant la gestion des stocks.

Le rôle de cette fonction est d'assurer aux opérations de la production, la fourniture des éléments de base dont elle a besoin, quand elle en a besoin (15). Le stock mis en place pour faciliter cet objectif permet la régulation de la production. Ce rôle d'amortisseur peut intervenir à n'importe quel stade de la production.

La création de stock permet de :

- maintenir l'indépendance entre opérations et par conséquence permettre une certaine souplesse dans l'ordonnement.
- faire face à la variation des délais d'approvisionnement soumis à des aléas.
- assurer un taux de service élevé vis à vis des clients tout en répondant à des délais de livraison beaucoup plus courts que les délais de fabrication.

Un autre rôle de la ~~gestion des stocks~~ est de diminuer le délais entre la commande et la livraison, si l'on représente sur un graphique les courbes de la commande et de la livraison :



On remarque que : si l'on diminue le temps de stockage le délais diminue, ce qui représente une donnée importante pour la production. Du point de vue économique la gestion de la production présente un impact décisif sur la rentabilité financière de l'Entreprise, ainsi que sur le financement à court terme.

IV . 2 . EXEMPLE D'APPLICATION

Le problème que nous avons à résoudre est du type " coordination des activités de N unités de production partageant une ressource R ".

Pour chaque unité, nous définissons :

U_i = niveau d'activité

X_i = niveau du stock du produit i

U_i = domaine des niveaux d'activités admissible caractérisé par les contraintes :

$$\begin{cases} 0 \leq U_i(t) \leq (t) \\ \alpha_i(t) \leq X_i(t) \leq \beta_i(t) \\ \dot{X}_i = k_i U_i(t) - d_i(t) \end{cases}$$

Nous avons à minimiser :

$$G_i(U_i) = \int_0^T \left\{ C_1 U_i(t) + C_2 |U_i(t)| + C_3 [X_i(t)]^+ + C_4 [X_i(t)]^- \right\} dt$$

qui est le coût de fonctionnement de l'unité sur $[0, T]$
 les unités sont liées entre elles par la relation :

$$\sum_{i=1}^N \delta_i u_i \leq R$$

Le problème est posé dans une formulation continue (variable temps non discrétisé). Par la suite nous travaillerons sur la formulation discrète que nous présentons ci-dessous :

$$C_i(U_i) = \sum_{t=1}^T C_1 U_i(t) + C_2 |U_i(t) - U_i(t-1)| + C_3 [X_i(t)]^+ + C_4 [X_i(t)]^-$$

L'intervalle $[0, T]$ est partagé en intervalles élémentaires égaux Δt . La quantité de produit i fabriqué par l'unité élémentaire de fabrication i ($U E F i$) pendant une période dépend linéairement du niveau d'activité $U_i(t)$ pendant la période t et ne dépend que de lui. D'où l'équation suivante :

$$\dot{X}_i(t) = k_i U_i(t) - d_i(t)$$

$$X_i(t) - X_i(t-1) = k_i U_i(t) - d_i(t)$$

$$X_i(t) = X_i(t-1) + k_i U_i(t) - d_i(t)$$

ce qui signifie :

Le stock à l'instant t = stock initial + production - la demande

k_i = constante positive de rendement

d_i = quantité produit i en 1 période

niveau d'activité

Par cette contrainte, on suppose que la décision $U_i(t)$ se concrétise immédiatement et directement par une production $k_i U_i(t)$ pendant la période T . On écarte donc tous délais supérieur à la période de temps retenue, ainsi que tout phénomènes annexes qui pourraient retarder ou empêcher l'application de la décision. Le fonctionnement des $U E F$ est donc défini essentiellement déterministe.

Les contraintes locales, c'est à dire les contraintes ne portant que sur 1 unités :

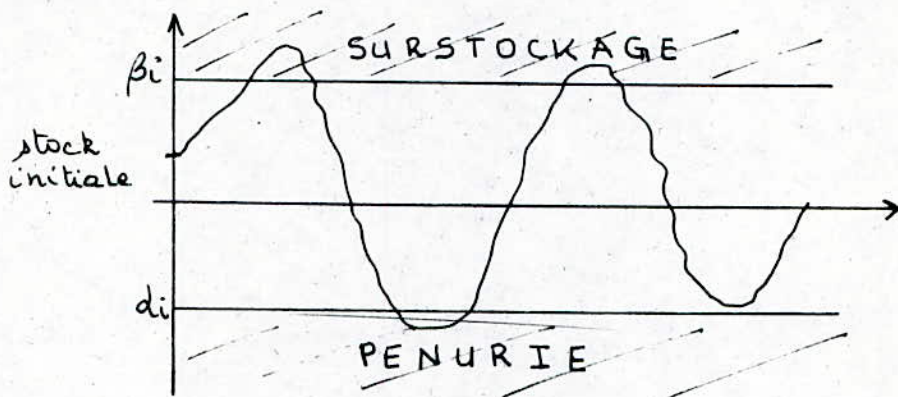
$$0 \leq U_i(t) \leq m_i(t)$$

$$\alpha_i(t) \leq X_i(t) \leq \beta_i(t)$$

peuvent être interprétées comme suit :

La première est évidente : le niveau d'activité $U_i(t)$ ne peut être négatif, ni arbitrairement grand .

La seconde relation exprime que pour des raisons techniques (volume physique) financière (valeur des stocks) et commerciales (stock de sécurité) le niveau du stock ne peut descendre sous $\alpha_i(t)$, ni dépasser $\beta_i(t)$ (1).



Le coût de fonctionnement de l'unité i pendant la période t ne dépend que des trois grandeurs suivantes :

- $U_i(t)$ niveau d'activité
- $X_i(t)$ niveau du stock
- $(U_i(t) - U_i(t-1))$ changement de niveau d'activité d'une période à l'autre.

$$C_i(U_i) = \int_0^T C_1 U_i(t) + C_2 |U_i(t) - U_i(t-1)| + C_3 [X_i(t)]^+ + C_4 [X_i(t)]^- dt$$

Les $C_i(U_i)$ sont les coûts de fonctionnement dont on précisera :

C_1 = Coût de production qui peut intervenir de deux façons : d'une part, dans le cas d'un coût de production non linéaire, d'autre part parce que la quantité totale produite peut n'être pas égale à la demande totale, celle-ci pouvant être en partie prélevée sur le stock initial (1).

C_2 = Coût de changement de cadence

C_3 = Coût de stockage

C_4 = Coût de rupture de stock

.../...

L'objectif que nous cherchons à atteindre est de minimiser le coût de fonctionnement du système.

La contrainte qui relie toutes les unités :

$$\sum_i \gamma_i \cdot U_i \leq R$$

peut être interprétée comme suit :

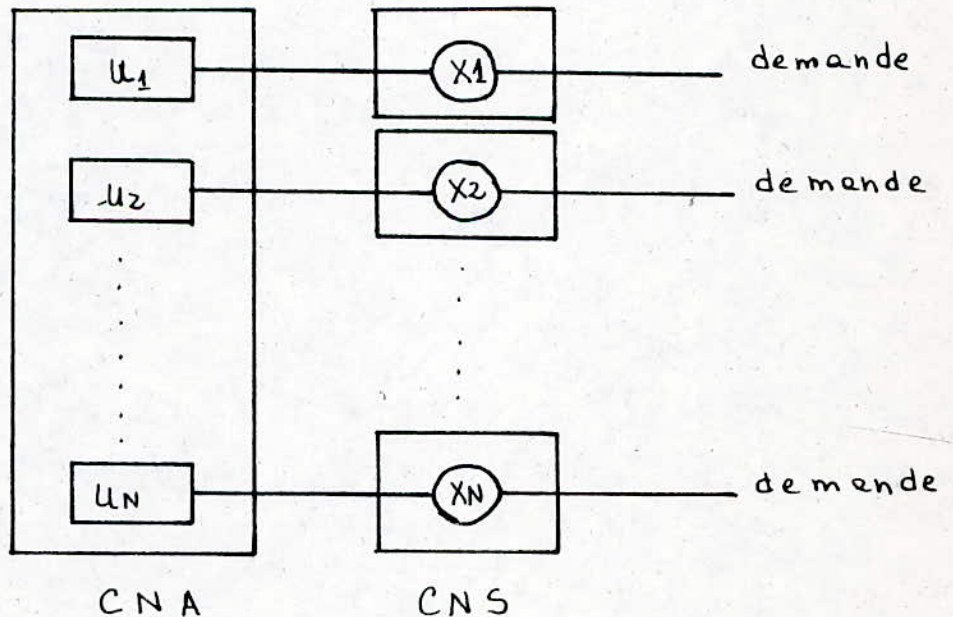
La ressource disponible R devant être partagée entre N unités il faut que la somme des dépenses des N unités soit inférieure ou au plus égale à R.

Nous schématisons une unité de production :



- . U_i : niveau d'activité de l'unité i pendant la période t
- . X_i : niveau du stock de l'unité i pendant la période t
- . d_i : demande du produit i fabriqué par l'unité i pendant la période t.

Nous pouvons alors donner une représentation schématique du système complet :



Les rectangles entourant plusieurs unités représentent une contrainte CNA ou CNS liant ces unités.

CNA : contraintes des niveaux d'activités ; il s'agit alors de contraintes de ressources au sens large (énergie, matière première, main d'oeuvre).

CNS : contraintes de niveaux des stocks ; ces contraintes traduisent les limitations sur le volume physique total des stocks ou leur valeur financière globale.

Ce problème sera résolu par :

- La méthode d'UZAWA

On en déduit alors le problème global PA :

$$\begin{array}{l} \text{PA} \\ \min \sum_{i=1}^N C_i(u_i) \\ u_i \in U_i^{\infty} \\ \sum_{i=1}^N \gamma_i u_i \leq R \end{array}$$

et les problèmes locaux PA_i (i = 1, ..., N)

$$\begin{array}{l} \min C_i(u_i) + \langle q, \gamma_i u_i \rangle \\ u_i \in U_i^{\infty} \end{array}$$

La méthode de DANZIG - WOLFE :

On en déduit le Maître - programme ;

soit U_i^j un ensemble de points du domaine U_i , pour $i = 1, \dots, N$, le maître programme s'écrit :

$$\begin{array}{l} \text{P2.} \\ \sum_{i=1}^N \sum_j \lambda_i^j C_i(u_i^j) \\ \sum_{i=1}^N \gamma_i \sum_j \lambda_i^j u_i^j \leq R \\ \sum_{i=1}^N \lambda_i^j = 1 \quad j = 1, \dots, N \end{array}$$

Pour les programmes locaux :

soit $q = (q^1, \dots, q^t)$ le vecteur des multiplicateurs du maître programme associés aux contraintes de P_2 . Le programme local numéro i s'écrit :

PA_i

$$\begin{array}{l} \min C_i(U_i) + (q, \gamma_i U_i) \\ U_i \in U_i^{\text{ad}} \end{array}$$

IV .3. RESULTATS

voir la page suivante:

----- ALGORITHME DE COORDINATION -----

***** * * * * * METHODE D'UZAWA * * * * * *****

DEFINIR LES VALEURS DE C1(N),C2(N),C3(N),C4(N):

C1(1)= 3
C2(1)= 1
C3(1)= 0
C4(1)= 1

C1(2)= 11
C2(2)= 2
C3(2)= 1
C4(2)= 5

C1(3)= 44
C2(3)= 9
C3(3)= 4
C4(3)= 22

C1(4)= 10
C2(4)= 2
C3(4)= 5
C4(4)= 5

VALEURS INITIALES DES STOCKS :

X0(1)=-5
X0(2)=-7
X0(3)= 3
X0(4)= 25

DEFINIR K(N) ET D(N):

K(1)= 1
D(1)= 5

K(2)= 1
D(2)= 5

K(3)= 1
D(3)= 6

K(4)= 2
D(4)= 11

DEFINIR LES VALEURS DE S(N):

S(1)= 1
S(2)= 1
S(3)= 1
S(4)= 1

DEFINIR LA VALEUR DE LA RESSOURCE DISPONIBLE:

R= 7500

DEFINIR LES VECTEURS DE DECISION :

LE VECTEUR $V(1)$ EST DONNE PAR :
SA MOYENNE $A(1) = 11$
SA VARIANCE $E(1) = 7$

LE VECTEUR $V(2)$ EST DONNE PAR :
SA MOYENNE $A(2) = 9$
SA VARIANCE $E(2) = 5$

LE VECTEUR $V(3)$ EST DONNE PAR :
SA MOYENNE $A(3) = 10$
SA VARIANCE $E(3) = 6$

LE VECTEUR $V(4)$ EST DONNE PAR :
SA MOYENNE $A(4) = 12$
SA VARIANCE $E(4) = 5$

SOLUTION OPTIMUM	Nbr. D'ITERATIONS
1875	27

CONCLUSION :

A partir du résultat obtenu nous pouvons conclure que, pour diminuer le coût de fonctionnement de cette Entreprise constituée de quatre Unités de production, le niveau d'activité total ne doit pas excéder 1875.

- ° - C O N C L U S I O N - ° -



Ce travail a permis l'étude de la résolution d'un grand système multidimensionnel, par deux algorithmes de décomposition coordination.

L'algorithme de DANTZIG - WOLFE, qui est une méthode de programmation linéaire, ne présente pas de grande difficulté pour la résolution. Basé sur la méthode du simplexe, qui est une méthode assez répandue, il présente un avantage très important : sa simplicité.

Mais lors de l'exécution, il nécessite une grande capacité mémoire du micro-calculateur, ce qui représente un certain inconvénient si les dimensions du problème sont trop grandes.

L'algorithme d'UZAWA contrairement à DANTZIG-WOLFE nécessite une capacité mémoire beaucoup plus restreinte et présente une grande rapidité d'exécution. C'est une méthode très peu connue. Sa résolution est complexe et présente plusieurs difficultés dont le choix des constantes de tests, le calcul de q (constante permettant le passage aux itérations suivantes) qui est une projection sur l'orthant positif de la fonctionnelle $M(q)$, et la procédure de modification qui varie selon le problème.

Dans l'application que nous avons choisie, la résolution a été faite par la méthode d'UZAWA que nous avons jugée plus adéquate.

En effet, si la précision (ϵ) choisie aurait été plus petite, le nombre d'itération nécessaire pour trouver la solution optimale serait très grand et par conséquent la capacité mémoire du micro-calculateur, dans le cas de la résolution par l'algorithme de DANTZIG-WOLFE n'aurait pas suffi.

Enfin, un certain nombre d'améliorations pourraient être apportées à ce travail. Entre autre; améliorer la procédure de modification en tenant compte d'un problème réel, résoudre ce même problème avec l'algorithme de DANTZIG-WOLFE en utilisant un calculateur plus puissant que L'OLIVETTI (M24) sur lequel nous avons travaillé, et comparer les deux algorithmes.

ANNEXES

ANNEXE 1 : METHODE DU SIMPLEXE

ENONCE MATHEMATIQUE GENERAL DES PROGRAMMES LINEAIRES

Soient $n + m$ variables non négatives satisfaisant à m équations linéaires que nous appellerons contraintes :

$$a_{1,1} X_1 + a_{1,2} X_2 + \dots + a_{1,n} X_n + a_{1,n+1} X_{n+1} + \dots + a_{1,n+m} X_{n+m} = B_1$$

$$a_{2,1} X_1 + a_{2,2} X_2 + \dots + a_{2,n} X_n + a_{2,n+1} X_{n+1} + \dots + a_{2,n+m} X_{n+m} = B_2$$

(1) .

.

.

$$a_{m,1} X_1 + a_{m,2} X_2 + \dots + a_{m,n} X_n + a_{m,n+1} X_{n+1} + \dots + a_{m,n+m} X_{n+m} = b_m$$

où les coefficients a_{ij} ($i=1,m, j=1,n+m$) et b_i ($i=1,m$) sont des nombres réels sur lesquelles nous ne ferons aucune hypothèse particulière.

On se donne aussi une fonction linéaire qui constitue la fonction économique :

$$(2) Z = C_1 X_1 + C_2 X_2 + \dots + C_n X_n + C_{n+1} X_{n+1} \dots + C_{n+m} X_{n+m}$$

où les coefficients c_j ($j=1,n+m$) sont des nombres réels sur lesquels nous ne ferons également aucune hypothèse particulière.

Pour condenser l'écriture, les relations (1) et (2) seront représentées comme suit :

$$(3) \quad \sum_{j=1}^{n+m} a_{ij} X_j = b_i \quad i = 1, m$$

$$(4) \quad z = \sum_{j=1}^{n+m} C_j X_j$$

Résoudre un programme linéaire c'est trouver la valeur des variables non négatives X_j , soumises aux contraintes (3) et rendant optimum la fonction z donnée par (4).

EXPOSE GENERAL

Reprenons les m équations (3) :

$$\sum_{j=1}^{n+m} a_{ij} X_j = b_i \quad i = 1, 2, \dots, m$$

où la matrice a_{ij} est de rang m par hypothèse.

Chaque colonne de la matrice a_{ij} et la colonne second nombre formée avec les quantités b_i peuvent être considérées comme des vecteurs $P_j, j=1, n+m$ et P_0 d'un espace linéaire à m dimensions. On supposera qu'en prenant arbitrairement m vecteurs distincts P_i , parmi les $n+m$ vecteurs P_j , on réalise un système de vecteurs linéairement indépendants. On posera donc :

$$\sum_{j=1}^{n+m} P_j x_i = P_0 \quad \text{où } P_j = \begin{Bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{Bmatrix} \quad (6) \quad \text{et } P_0 = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{Bmatrix} \quad (7)$$

Ayant supposé que ce système de m équations linéaires de $n + m$ variables est de rang m , nous pouvons alors exprimer n vecteurs P_j numérotés de 1 à n et P_0 , en fonction des m vecteurs P_i numérotés de $n + 1$ à $n + m$ et linéairement indépendants par hypothèse.

Supposons qu'on ait trouvé une solution de base dont toutes les variables sont positifs, soient m quantités X_i positives, numérotées de $n + 1$ à $n+m$ tandis que toutes les autres numérotées de 1 à n seront supposées nulles. Pour cette solution de base, les équations (3) et (4) se réduisent à :

$$(8) \quad \sum_{i=n+1}^{n+m} P_i x_i = P_0 \quad \text{et} \quad Z_0 = \sum_{i=n+1}^{n+m} C_i x_i \quad (9)$$

$$x_i > 0 \quad i = n+1, n+2, \dots, n+m$$

les n autres vecteurs P_j , $j = 1, n$ peuvent, comme nous venons de l'écrire, être exprimés linéairement en fonction des m vecteurs

P_i qui constitueront, ainsi qu'il est usuel de les nommer dans la théorie des espaces linéaire une "base" ou "repère".

On écrira :

$$(10) \quad P_j = \sum_{i=n+1}^{n+m} x_{ij} P_i \quad j = 1, 2, \dots, n.$$

les coefficients X_{ij} peuvent être calculés par inversion de la matrice carrée formée par les m vecteurs P_i qui constituent la base.

Nous appellerons Z_j les quantités :

$$(11) \quad Z_j = \sum_{i=n+1}^{n+m} x_{ij} C_i \quad j = 1, 2, \dots, n$$

Z_j est l'écroissement de z correspondant à P_j .

Maintenant, soit un des vecteurs P_j qui ne sont pas dans la base; multiplions (10) par 1 quantité scalaire θ positive et écrivons en reprenant (8).

$$(12) \quad \sum_{i=n+1}^{n+m} P_i x_i = \sum_{i=n+1}^{n+m} (x_i - \theta x_{ij}) P_i + \theta P_j = P_0$$

(On n'a pas écrit $j = 1, 2, \dots, n$, le vecteur P_j ayant été choisi parmi les n vecteurs)

De la même manière, en ajoutant $\Theta(C_j - Z_j)$ aux deux membres de (9) on obtient

$$(13) \quad Z_0 + \Theta(C_j - Z_j) = \sum_{i=n+1}^{n+m} (X_i - \Theta X_{ij}) C_i + \Theta c_j$$

Nous avons ainsi réalisé une nouvelle solution de base, à la condition que tous les X_{ij} ne soient pas négatifs et que $\Theta \neq 0$ (Si $\Theta = 0$, la solution ne serait pas changée). Si certains X_{ij} sont positifs (au moins un) choisissons la quantité positive Θ telle que $\Theta = \Theta_0 = \min_i \frac{X_i}{X_{ij}}$

En ne considérant que les $X_{ij} > 0$;

C'est à dire Θ correspond à la plus petite valeur des rapports X_i/X_{ij} parmi ceux qui sont positifs : ($i = n + 1, \dots, n + m$)

Si Θ est choisi de cette façon, tous les termes

$$(15) \quad (X_i - \Theta X_{ij})$$

seront positifs sauf un, qui sera nul et correspondra à (14) Θ étant ainsi déterminé et appelé Θ_0 on fera sortir de la base le vecteur P_i tel que

$\Theta_0 = X_i/X_{ij}$ puisque son coefficient s'annulera dans (12) où il sera remplacé par le vecteur P_j de coefficient Θ_0 , la solution de base obtenue est alors distincte de la précédente.

Enfin, en ce qui concerne l'augmentation de Z_0 ; en consultant (13), on voit que ceci se produira si l'on choisit j de telle sorte que

$$(16) \quad C_j - Z_j > 0$$

A la valeur de $C_j - Z_j$ la plus élevée correspondra la plus grande augmentation possible de Z_0 par ce procédé ; c'est cette valeur que nous choisirons et qui nous désignera le vecteur P_j qui entrera dans la base. On dit encore par commodité, que l'on choisit la valeur de j correspondant à la quantité la plus négative des valeurs de $Z_j - C_j$.

Il en résulte donc qu'en choisissant P_j tel que $Z_j - C_j$ soit le plus négatif et en sélectionnant la ligne i pour laquelle $\Theta = X_i/X_{ij}$ est minimum mais positif (critères de DANTZIG)

On pourra déterminer une transformation de la solution de base initiale telle que l'on ait la plus grande augmentation possible de Z_0 .

Lorsqu'il ne sera plus possible de trouver une seule quantité $z_j - C_j$ négative, il ne sera plus possible d'augmenter Z_0 et l'on aura ainsi atteint le maximum de Z . Un raisonnement analogue en prenant $C_j - Z_j$ le plus négatif nous aurait conduit à diminuer Z_0 et à atteindre le minimum.

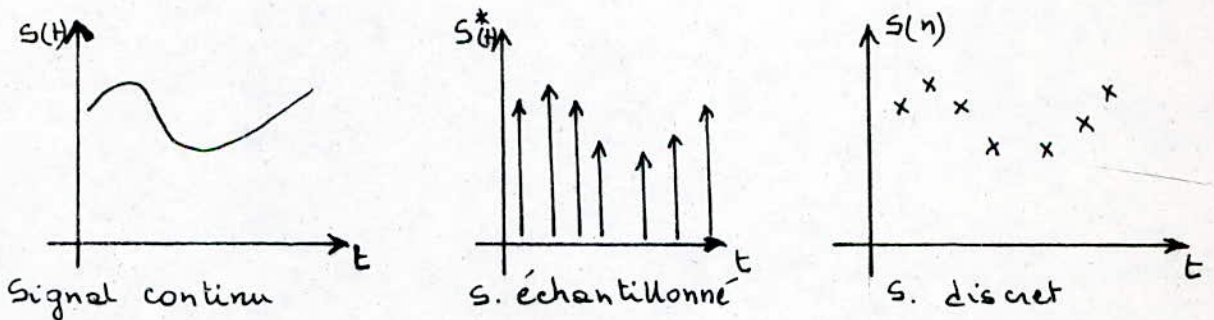
Le nombre de solution de base (m variables non - négatives et non toutes nulles, n variables nulles) est, au plus, $C_n^m + m$, ce nombre est très élevé lorsque m et n sont grands. La recherche de la solution optimum en calculant toutes les solutions de base exigerait des calculs énormes, on conçoit alors l'intérêt de la méthode analytique par itération donnée par DANTZIG.

ANNEXE 2

CHOIX DE LA PERIODE D'ECHANTILLONNAGE

Etant donné que l'on travaille sur ordinateur, et que la fonction objective est continue $C_i (U_i(t))$, il faut donc l'échantillonner. Le passage de la représentation continue à la représentation discrète est appelé "discrétion".

Pour faire le lien entre le signal discret et continu, on utilise un troisième signal échantillonné $S^*(t)$ obtenu par la multiplication du signal continu par un train d'impulsion de Dirac



Par définition on peut écrire :

$$S^*(t) = \sum_{-\infty}^{+\infty} C(n\Delta t) \delta(t - n\Delta t)$$

choix de la période d'échantillonnage ΔT de manière à éviter une dégradation d'information importante due à la discrétisation :

- Théorème de Shannon : Si la fréquence d'échantillonnage d'un signal est au moins supérieur au double de la fréquence la plus haute contenue dans le signal on obtient une information discrète équivalente à l'information continue.

- Pour être plus précis quant au choix de ΔT , on procède comme suit :

On a :

- Le signal continu $S(t)$
- Le signal échantillonné $S^*(t) = \sum_{-\infty}^{+\infty} S(t) \delta(t - n\Delta T)$
- Le signal discret $S_n = S(n\Delta T)$

Signal Continu	Signal Discret
Fonction d'autocorrelation	
$C_{xx}(\eta) = \frac{1}{T} \int_0^T S$	
Densité Spectrale	
$C_{xx1}(\omega) = \int_{-\infty}^{+\infty} C_{xx}(\eta) e^{-j\omega\eta} d\eta$	$C_{xx1}(\omega) = \sum_{-\infty}^{+\infty} C_{xx}(k) e^{-jk\omega\Delta T}$
Transformée de Fourier inversée	
$C_{xx}(\eta) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} C_{xx}(\omega) e^{j\omega\eta} d\omega$	$C_{xx}(k) = \frac{1}{2\pi} \sum_{-\infty}^{+\infty} C_{xx}(\omega) e^{j\omega k\Delta T}$

En calculant $C_{xx}(\eta)$ et $C_{xx}(k)$ et les transformées inverses on fait la comparaison:

$$C_{xx1}(\omega) = C_{2xx}(\omega) \quad C_{xx}(\eta) = C_{xx}(k)$$

Alors T est bien choisie, dans le cas contraire on prend $\Delta T' > \Delta T$ et on fait un nouveau échantillonnage.

* Remarque: $N = \frac{T}{\Delta T}$: qualité de l'information

on prend soin à ce que $\frac{1}{\Delta T} \geq \frac{2}{T}$.

ANNEXE 3 : Démonstration du théorème (1)

Condition nécessaire

Par hypothèse, on a :

$$(3.1) \quad J_i(U_i) + \langle p, B_i(U_i) \rangle \leq J_i(V_i) + \langle p, B_i(V_i) \rangle \\ \forall V_i \in U_i^\infty, \quad i = 1, \dots, N$$

et si les PA_i sont coordonnables :

$$(3.2) \quad \langle p, \sum_c B_i(U_i) \rangle = 0$$

De plus, puisque U_i est solution du problème global PA on a :

$$(3.3) \quad \sum_c B_i(U_i) \leq 0$$

par addition de (3.1), on déduit :

$$(3.4) \quad \sum_c J_i(u_i) + \langle p, \sum_c B_i(U_i) \rangle \leq \sum_c J_i(V_i) + \langle p, \sum_c B_i(V_i) \rangle \\ \forall V_i \in U_i^\infty$$

et d'après (3.3)

$$(3.5) \quad \sum_c J_i(U_i) + \langle q, \sum_c B_i(U_i) \rangle \leq \sum_c J_i(U_i) \quad \forall q \geq 0$$

il vient finalement, avec (3.2) :

$$(3.6) \quad \sum_c J_i(U_i) + \langle q, \sum_c B_i(U_i) \rangle \leq \sum_c J_i(U_i) + \langle p, \sum_c B_i(\omega) \rangle$$

$$(3.7) \quad \sum_c J_i(U_i) + \langle q, \sum_c B_i(U_i) \rangle \leq \sum_c J_i(V_i) + \langle p, \sum_c B_i(V_i) \rangle \\ \forall q \geq 0 \quad \forall V_i \in U_i^\infty$$

Ces deux relations expriment que la fonctionnelle

$$\sum_c J_i(V_i) + \langle q, \sum_c B_i(V_i) \rangle \text{ possède un point selle} \\ \text{sur } \left\{ q \geq 0 ; U_i \in U_i^\infty \right\}$$

Condition suffisante :

Si la fonctionnelle $\sum_c J_i(V_i) + \langle q, \sum_c B_i(V_i) \rangle$ possède un point selle sur $\left\{ q \geq 0 ; V_i \in U_i^\infty \right\}$, il existe U_i et p vérifiant

$$\sum_c J_i(U_i) + \langle q, \sum_c B_i(V_i) \rangle \leq \sum_c J_i(U_i) + \langle p, \sum_c B_i(U_i) \rangle \\ \leq \sum_c J_i(V_i) + \langle p, \sum_c B_i(V_i) \rangle \\ \forall q \geq 0 ; V_i \in U_i^\infty ; p \geq 0 ; U_i \in U_i^\infty$$

(3.6) entraîne

$$(3.8) \quad \langle q - p, \sum_c B_i(U_i) \rangle \leq 0, \quad \forall q \geq 0$$

On en déduit :

$$(3.9) \quad \sum_c B_i (U_i) \leq 0$$

$$(3.10) \quad \langle p, \sum_c B_i (U_i) \rangle = 0$$

De (3.7) on obtient également :

$$(3.11) \quad \sum_{i=1}^N \{ J_i (V_i) - J_i (U_i) \} + \langle p, \sum_c B_i (V_i) - \sum_c B_i (U_i) \rangle \geq 0 \\ \forall U_i \in U_i^\infty$$

En prenant tous les V_i égaux à U_i , sauf un, on déduit aisément

$$(3.12) \quad J_i (V_i) + \langle p, B_i (V_i) \rangle \geq J_i (U_i) + \langle p, B_i (U_i) \rangle \\ \forall V_i \in U_i^\infty$$

qui exprime que les U_i sont bien solution des problèmes locaux PA_i.

ANNEXE 4 : Démonstration du théorème(2)

Par hypothèse :

$$\sum_{i=1}^N \{ J_i [U_i(q)] + \langle q, B_i [U_i(q)] \rangle \} \leq \sum_{i=1}^N \{ J_i (V_i) + \langle q, B_i (V_i) \rangle \} \\ \forall u \in U^\infty \text{ en posant } \begin{cases} v = (v_1, v_2, \dots, v_n) \\ U^\infty = U_1^\infty \times U_2^\infty \times \dots \times U_n^\infty \end{cases}$$

$$\text{d'où } \sum_{i=1}^N J_i [U_i(q)] \leq \sum_{i=1}^N J_i (V_i) - \sum_{i=1}^N \langle q, B_i (U_i(q)) - B_i (V_i) \rangle$$

Cette relation étant vraie quelque soit $V \in U^\infty$, est vraie en particulier pour le sous espace $U^{*\infty} \subset U^\infty$ pour lequel on a :

$$\sum_{i=1}^N \{ B_i (V_i) - B_i (U_i(q)) \} \leq 0$$

On en déduit :

$$\sum_{i=1}^N \langle q, B_i (V_i) - B_i (U_i(q)) \rangle \leq 0 \\ \text{et } \sum_{i=1}^N J_i (U_i(q)) \leq \sum_{i=1}^N J_i (V_i) \quad \forall V \in U^{*\infty}$$

ANNEXE 5 : Démonstration du théorème (3)

Par hypothèse

$$\sum_{i=1}^N J_i (U_i(q)) \leq \sum_{i=1}^N J_i (V_i) - \sum_{i=1}^N \langle q, B_i (U_i(q)) - B_i (V_i) \rangle + \varepsilon \\ \forall V \in U^\infty$$

Cette relation est également vraie pour $V \in U^{*\infty}$, $U^{*\infty}$ étant défini de la même façon que précédemment.

$$\text{Alors : } \sum_{i=1}^N \langle q, B_i (U_i(q)) - B_i (V_i) \rangle \leq 0$$

$$\text{et } \sum_{i=1}^N J_i (U_i(q)) \leq \sum_{i=1}^N J_i (V_i) + \varepsilon \quad \forall V \in U^{*\infty}$$


```

5 CLS
10 REM *****PRESENTATION *****
15 LOCATE 5,22 : PRINT "*****"
20 LOCATE 7,22 : PRINT "RESOLUTION DU PROBLEME LINEAIRE P2(K) "
25 LOCATE 9,22 : PRINT "*****"
30 PRINT :PRINT :PRINT
35 PRINT TAB(20) "OPTIMISATION LINEAIRE (METHODE DU SIMPLEXE)":PRINT :PRINT
40 PRINT "PROGRAMMATION LINEAIRE"
45 PRINT "-----"
50 *****SAISI DE DONNEES*****
60 PRINT "FONCTION ECONOMIQUE:"
70 INPUT "MAXIMISATION /MINIMISATION (MAX/MIN) "; C$
80 IF LEFT$(C$,2)<>"MA" AND LEFT$(C$,2)<>"MI" THEN 70
90 IF LEFT$(C$,2)="MI" THEN PT=-1 : GOTO 110
100 PT=1
110 INPUT "NOMBRE DE VARIABLE DE DECISION. ";MN
115 **LES VARIABLES DE DECISION SONT LES SOLUTIONS DE P2(K)**
120 PRINT
130 PRINT "NOMBRE DE CONTRAINTES "
140 PRINT "(CONTRAINTES DE NON-NEGATIVITE EXCLUES):"
150 INPUT "plus petite(s) ou egale(s)";NS
160 INPUT "plus grande(s) ou egale(s)";NB
170 INPUT "d'egalite ";NE
180 M=NS+NB+NE : REM M=NOMBRE TOTAL DE CONTRAINTES
190 N=M+MN+NE : REM NOMBRE TOTAL DE VARIABLES
200 DIM B(M),CI(N),CJ(N),Z(N),ZC(N),XI(N),XJ(N),A(M,N)
210 **IDENTIFICATION DES VARIABLES DE DECISION, DES VARIABLES D'ECART
220 ET DES VARIABLES ARTIFICIELLES*****
230 PRINT :PRINT
240 PRINT "DEFINITION DES INDICES DES VARIABLES:" : PRINT
250 K=1 : FOR J=M+1 TO M+MN
260 PRINT "VARIABLE DE DECISION ";K;
270 XJ(J)=K
280 PRINT "=X("XJ(J);)"
290 K=K+1 : NEXT J :PRINT
300 IF NS <=0 THEN 390
310 PRINT "VARIABLE(S) D'ECART DES CONTRAINTES "
320 PRINT "PLUS-PETITES-OU-EGALES :"
330 K=MN+1 : FOR J=1 TO NS
340 PRINT "de la contrainte ";J;
350 XJ(J)=K
360 PRINT "=X(";XJ(J);)"
370 K=K+1 :NEXT J :PRINT
380 FOR I=1 TO N :CJ(I)=0 :NEXT I
390 IF NB=0 THEN 470
400 PRINT "VARIABLE(S) D'ECART DES CONTRAINTES"
410 PRINT "PLUS-GRANDES-OU-EGALES-A (VS. SURPLUS):"
420 K=M+MN+1 : FOR J=M+MN+1 TO N
430 PRINT "de la contrainte";J+NS-M-MN;
440 XJ(J)=K
450 PRINT "=X(";XJ(J);)"
460 K=K+1 : NEXT J : PRINT
470 IF NB=0 AND NE=0 THEN 560
480 PRINT "VARIABLE(S) ARTIFICIELLE(S) DES CONTRAINTES"
490 PRINT "PLUS-GRANDES-OU-EGALES-A ET D'EGALITE :"
500 K=MN+NS+1 : FOR J=NS+1 TO M

```



```

510 PRINT "de la contrainte ";J;
520 XJ(J)=K
530 PRINT "=X(";XJ(J);")"
540 CJ(J)=10000
550 K=K+1 : NEXT J :PRINT
560 FOR I=1 TO M : XI(I)=XJ(I) : NEXT I
570 PRINT "COEFFICIENTS TECHNIQUES DE LA FONCTION OBJECTIVE :"
580 FOR I=M+1 TO M+MN
590 PRINT "coefficient de la variable de decision ";I-M;
600 INPUT CJ(I)
610 CJ(I)=CJ(I)*PT*(-1)
620 NEXT I : PRINT
630 FOR I=1 TO M
640 PRINT "VALEUR DU COTE DROIT DE LA CONTRAINTE ";I;
650 INPUT B(I) : NEXT I
660 *****CONSTRUIRE LA MATRICE UNITAIRE AUGMENTEE*****
670 FOR I=1 TO M : FOR J=1 TO N
680 IF I<>J THEN 710
690 A(I,J)=1
700 GOTO 720
710 A(I,J)=0
720 NEXT J,I
730 PRINT
740 PRINT "COEFFICIENTS TECHNOLOGIQUES DANS LES CONTRAINTES:"
750 FOR I=1 TO M
760 PRINT "coeff. technologique de la contrainte";I
770 FOR J=M+1 TO M+MN
780 PRINT "-variable de decision";J-M;
790 INPUT A(I,J)
800 NEXT J,I
810 IF NB=0 THEN 870
820 *****INTRODUIRE LES COEFFICIENTS DES VS. SURPLUS*****
830 FOR I=1 TO NB
840 A(NS+I,M+MN+I)=-1
850 NEXT I
860 *****ALGORITHME DU SIMPLEXE*****
870 FOR I=1 TO M: FOR J=1 TO N
880 IF XI(I)<>XJ(J) THEN 900
890 CI(I)=CJ(J)
900 NEXT J,I
910 IT=0
920 FOR J=1 TO N
930 Z(J)=0
940 FOR I=1 TO M
950 Z(J)=Z(J)+CI(I)*A(I,J)
960 NEXT I
970 ZC(J)=Z(J)-CJ(J)
980 NEXT J
990 OB=0
1000 FOR I=1 TO M
1010 OB=OB+CI(I)*B(I)
1020 NEXT I
1030 PRINT :PRINT
1040 PRINT "ITERATION N°.";IT
1050 PRINT "
1060 PRINT "variables de base          valeur"

```



```

1070 FOR I=1 TO M
1080 PRINT TAB(6);"X(";XI(I);")"; TAB(22);B(I) : NEXT I
1090 PRINT : N1=1 : N2=8
1100 IF N2<=N THEN 1120
1110 N2=N
1120 PRINT "Var. du tableau simplexe:"
1130 K=0 : FOR I=N1 TO N2
1140 PRINT TAB(8*K); "X(";XJ(I);"),"; : K=K+1 : NEXT I
1150 PRINT :PRINT
1160 PRINT "Matrice des coefficients A(I,J):"
1170 FOR I=1 TO M: K=0: FOR J=N1 TO N2
1180 PRINT TAB(8*K); INT(100*A(I,J)+.5)/100;: K=K+1
1190 NEXT J: PRINT : NEXT I: PRINT
1200 PRINT "Coeff. des profits marginaux Z(J)-C(J):"
1210 K=0: FOR I=N1 TO N2
1220 PRINT TAB(8*K); INT(100*ZC(I)+.5)/100;: K=K+1
1230 NEXT I : PRINT
1240 IF N2>=N THEN 1270
1250 N1=N1+8 : N2=N2+8
1260 GOTO 1100
1270 PRINT :PRINT "Fonction economique Z="; OB :PRINT
1280 PRINT : INPUT "pour continuer taper <C> ";Cs
1290 IT=IT+1 : CM=ZC(1) : JM=1
1300 FOR J=2 TO N
1310 IF ZC(J)<=CM THEN 1330
1320 CM=ZC(J) : JM=J
1330 NEXT J
1340 IF CM>0 THEN 1740
1350 M3=M+MN : MO=M+1
1360 IF M=NS THEN 1420
1370 FOR I=1 TO M
1380 M4=NS+1
1390 FOR J=M4 TO M
1400 IF XI(I)=XJ(J) THEN 1720
1410 NEXT J,I
1420 FOR K=MO TO M3
1430 FOR I=1 TO M
1440 IF XJ(K)=XI(I) GOTO 1470
1450 NEXT I
1460 IF ZC(K)=0 THEN 1490
1470 NEXT K
1480 GOTO 1500
1490 PRINT "*****PLUSIEURS SOLUTIONS OPTIMALES POSSIBLES*****"
1500 PRINT :PRINT :PRINT
1510 PRINT "*****SOLUTION OPTIMALE TROUVEE*****"
1520 PRINT "***** APRES";IT;"ITERATIONS *****"
1530 FOR I=1 TO M
1540 IF B(I)<>0 THEN 1570
1550 PRINT :PRINT "*****SOLUTION DEGENEREE*****"
1560 GOTO 1580
1570 NEXT I
1580 PRINT
1590 PRINT "-----"
1600 PRINT "VARIABLE DE DECISION VALEUR"
1610 PRINT "-----"
1620 FOR I=1 TO M

```



```

1630 PRINT TAB(8);"X(";XI(I);")";TAB(16);"=";TAB(25);B(I)
1640 NEXT I
1650 PRINT "NOTE :";PRINT "Toute les variables non-indiquees"
1660 PRINT "dans ce tableau sont nulles."
1670 PRINT "-----"
1680 IF PT=1 THEN PRINT TAB(5);"MAXIMUM      Z=";ABS(OB)
1690 IF PT=-1 THEN PRINT TAB(5);"MINIMUM      Z=";ABS(OB)
1700 PRINT "-----"
1710 STOP
1720 PRINT :PRINT "*****SOLUTION IMPOSSIBLE*****"
1730 STOP
1740 XM=9.999999E+24 : IM=0
1750 FOR I=1 TO M
1760 IF A(I,JM)<=0 THEN 1800
1770 XX=B(I)/A(I,JM)
1780 IF XX>=XM THEN 1800
1790 XM=XX : IM=I
1800 NEXT I
1810 IF IM>0 THEN 1840
1820 PRINT "*****SOLUTION IMPOSSIBLE*****"
1830 STOP
1840 XX=A(IM,JM)
1850 B(IM)=B(IM)/XX
1860 FOR J=1 TO N
1870 A(IM,J)=A(IM,J)/XX
1880 NEXT J
1890 FOR I=1 TO M
1900 IF I=IM THEN 1960
1910 XX=A(I,JM)
1920 B(I)=B(I)-XX*B(IM)
1930 FOR J=1 TO N
1940 A(I,J)=A(I,J)-XX*A(IM,J)
1950 NEXT J
1960 NEXT I
1970 CI(IM)=CJ(JM)
1980 XI(IM)=XJ(JM)
1990 GOTO 920

```

5 CLS

```

10 *****PRESENTATION*****
20 LOCATE 5,22 : PRINT "*****"
30 LOCATE 7,22 : PRINT "CALCUL DES VARIABLES DUALES "
40 LOCATE 9,22 : PRINT "*****"
50 PRINT :PRINT :PRINT
60 PRINT "CE PROGRAMME DE CALCUL DES VARIABLES DUALES EST LA SUITE DU "
70 PRINT "PROGRAMME DE RESOLUTION DE P2(K) PAR LA METHODE DU SIMPLEXE"
75 PRINT "      *****":PRINT
80 *****SAISI DE DONNEES*****
90 INPUT "NOMBRE DE VARIABLE DE DECISION DU PRIMAL ";NP
100 INPUT "NOMBRE DE VARIABLE D'ECART DU PRIMAL ";ND
110 M=NP+ND : REM M=NOMBRE TOTAL DE VARIABLES
120 DIM X(M),Y(M),B(M)
125 PRINT :PRINT
130 PRINT "LES COEF. DES PROFITS MARGINAUX DES VARIABLES DU PRIMAL SONT:"
140 FOR I=1 TO M

```



```

150 PRINT "-pour la variable ";I;
160 INPUT B(I)
165 NEXT I :PRINT :PRINT
170 *****CONSTRUCTION DU TABLEAU DES CORRESPONDANCES *****
175 *****PRIMAL-DUAL*****
180 PRINT " TABLEAU DE CORRESPONDANCE PRIMAL-DUAL "
190 PRINT "-----"
200 PRINT
210 PRINT " VARIABLES PRINCIPALES !";
230 FOR I=1 TO NP
240 PRINT " X(";I;") !" ;
250 NEXT I
270 FOR J=1 TO ND
280 PRINT " Y(";J;") !";
290 NEXT J
300 PRINT "-----"
320 PRINT " VARIABLES D'ECART          !";
340 FOR S=ND+1 TO M
350 PRINT " Y(";S;") !";
360 NEXT S
380 FOR L=NP+1 TO M
390 PRINT " X(";L;") !";
400 NEXT L
410 PRINT "-----"
420 PRINT :PRINT
430 *****CALCUL DES VARIABLES DUALES*****
440 J=NP+1
450 FOR K=1 TO ND
460 Y(K)=ABS(B(J))
465 J=J+1
470 NEXT K
480 J=1
490 FOR K=ND+1 TO M
500 Y(K)=ABS(B(J))
505 J=J+1
510 NEXT K
520 *****IMPRESSION DU RESULTAT*****
530 PRINT "-----"
540 PRINT " VARIABLES DUALES          VALEURS"
550 PRINT "-----"
560 J=NP+1
570 FOR K=1 TO ND
580 PRINT TAB(8); "Y(";K;")" ; TAB(16); "="; TAB(25); ABS(B(J))
585 J=J+1
590 NEXT K
600 J=1
610 FOR K=ND+1 TO M
620 PRINT TAB(8); "Y(";K;")"; TAB(16); "="; TAB(25); ABS(B(J))
625 J=J+1
630 NEXT K
640 PRINT "-----"
650 END

```



```

5 CLS
6 *****PRESENTATION*****
10 PRINT TAB(25) "***ALGORITHME DE COORDINATION**"
20 PRINT
30 PRINT TAB(25) "METHODE DE DANTZIG-WOLFE (DZW)"
40 PRINT:PRINT:PRINT
50 PRINT " RESOLUTION DU PROGRAMME SATELLITE P31(K)"
60 PRINT
65 *****SAISI DE DONNEES*****
68 PRINT " LES COEFFICIENTS DE LA FONCTION OBJECTIVE SONT : "
69 PRINT "-----"
70 INPUT "S=";S : INPUT "P=";P : INPUT "Q=";Q
75 PRINT:PRINT " LES COEFFICIENTS DE LA FONCTION CONTRAINTE SONT : "
78 PRINT "-----"
80 INPUT "A=";A : INPUT "B=";B : INPUT "C=";C
90 DEF FNA(X)= S*X^2+P*X+Q
100 DEF FNB(X)= A*X^2+B*X+C
110 PRINT:INPUT " NOMBRE TOTAL DE VARIABLES R=";R
111 DIM A(R),F(R),X(R), D(R)
112 PRINT:INPUT "NOMBRE DE VARIABLES D'ECART L=";L
113 DIM U1(L),H(R,L)
114 FOR I=1 TO L
115 PRINT " LA VARIABLE DUALE U1(";I;")=";: INPUT U1(I)
116 NEXT I
118 PRINT:INPUT "NOMBRE DE VARIABLES ARTIFICIELLES N=";N
120 DIM U2(N)
122 FOR J=1 TO N
124 PRINT " LA VARIABLE DUALE U2(";J;")=";:INPUT U2(J)
126 NEXT J
128 *****CALCUL*****
130 FOR I=1 TO R
140 PRINT:PRINT "LE GENERATEUR X(";I;")=";: INPUT X(I)
150 LET F(I)=FNA(X(I))
160 LET A(I)=FNB(X(I))
170 PRINT:PRINT "LA FONCTION OBJECTIVE F(";I;")=";F(I)
180 PRINT "LA FONCTION CONTRAINTE A(";I;")=";A(I)
190 NEXT I
200 FOR J=1 TO R
205 FOR T=1 TO L
210 H(J,T)=F(J)-U1(T)*A(J)
215 NEXT T
220 NEXT J
230 *****RECHERCHE DU MINIMUM*****
240 MIN=H(1,1)
250 FOR M=1 TO R
260 FOR I=1 TO L
270 IF MIN<H(M,I) THEN 295
280 MIN= H(M,I)
290 INDICE=M
295 IF M<>1 THEN 300 ELSE INDICE=1
300 NEXT I : NEXT M
305 X1=X(INDICE)
310 PRINT:PRINT "LE MINIMUM EST H(1)=";MIN
315 PRINT "*****"
320 PRINT " * * * L'INDICE CORRESPONDANT EST * * * ";INDICE
330 PRINT "-----"

```



```

310 *****CALCUL DU CRITERE DE TERMINAISON (A)*****
315 PRINT
320 PRINT "CALCUL DU CRITERE DE TERMINAISON (A)"
330 PRINT "-----"
340 PRINT :PRINT "INTRODUIRE LA VALEUR DES COUTS RELATIFS"
350 PRINT "DE TOUTES LES ITERATIONS PRECEDANTES."
360 PRINT
370 FOR L=1 TO K
380 PRINT "POUR L'ITERATION Nl";L;"," ;
390 PRINT "Dl^S(";L;")=" ;:INPUT D(L)
400 NEXT L
410 FOR I=1 TO K
420 D(I)=D(I-1)+D(I)
430 NEXT I
440 PRINT " LA SOMME DES COUTS RELATIFS EST:";ABS(D(K))
450 PRINT :PRINT
460 PRINT "INTRODUIRE LA VALEUR DES FONCTIONS ECONOMIQUES DE TOUTES"
470 PRINT " LES ITERATIONS PRECEDANTES."
480 PRINT
490 FOR L=1 TO K
500 PRINT "POUR L'ITERATION Nl";L;"," ;
510 PRINT "Fl(";L;")=" ;:INPUT F(L)
520 NEXT L
530 FOR I=1 TO K
540 F(I)=F(I-1)+F(I)
550 NEXT I
560 PRINT "LA SOMME DES FONCTIONS ECONOMIQUES EST :";ABS(F(K))
570 PRINT :PRINT
580 A=ABS(D(K))/ABS(F(K))
590 IF A<EPS2 THEN 180
595 PRINT "****LE CRITERE (A) N'EST PAS VERIFIE ON PASSE A "
597 PRINT "L'ITERATION ";K+1;"ET ON RESOUD P2(";K+1;")****"
600 PRINT TAB(14) "*****"

```


5 CLS

```
10 *****PRESENTATION*****
20 PRINT :PRINT :PRINT
25 PRINT "-----"
30 PRINT "-----"
40 PRINT "----- ALGORITHME DE COORDINATION -----"
50 PRINT "-----"
70 PRINT "***** METHODE D'UZAWA *****"
80 PRINT "-----"
90 PRINT "-----"
100 PRINT :PRINT
110 PRINT "***** RESOLUTION DES PAi*****"
120 PRINT "-----"
130 *****SAISI DE DONNEES*****
140 INPUT "LE NOMBRE TOTAL DES VARIABLES DE DECISION N EST :";N
150 DIM V(N),A(N),E(N),C(N),S(N),P(N),Q(N),J(N,N),B(N,N),H(N,N),U(N),UM(N)
155 DIM DELTA(N),M(N),MP(N),BQ(N,N),IBQ(N,N),UA(N),GAM(N),L(N),LM(N),MIN(I
160 REM J= FONCTION CRITERE
165 REM B= FONCTION CONTRAINTE
170 REM V= VECTEUR DE DECISION
180 PRINT
190 PRINT "DEFINIR LES VECTEURS DE DECISION V(N):"
200 FOR I=1 TO N
210 PRINT "LE VECTEUR V(";I;")=";: INPUT V(I)
220 NEXT I
230 PRINT
240 PRINT "LES FONCTIONS CRITERES SONT DE LA FORME A.V^2+E.V+C "
245 PRINT "-----"
250 PRINT :PRINT "DEFINIR LES DIFFERENTES VALEURS DE A(N),E(N),C(N). "
260 FOR R=1 TO N
270 PRINT :PRINT "A(";R;")=";: INPUT A(R)
280 PRINT "E(";R;")=";: INPUT E(R)
290 PRINT "C(";R;")=";: INPUT C(R)
300 NEXT R
310 PRINT
320 PRINT "LES FONCTIONS CONTRAINTES SONT DE LA FORME S.V^2+P.V+Q"
325 PRINT "-----"
330 PRINT :PRINT "DEFINIR LES DIFFERENTES VALEURS DE S(N),P(N);Q(N). "
340 FOR T=1 TO N
350 PRINT :PRINT "S(";T;")=";: INPUT S(T)
360 PRINT "P(";T;")=";: INPUT P(T)
370 PRINT "Q(";T;")=";: INPUT Q(T)
380 NEXT T
390 PRINT :PRINT
400 *****CALCUL DE J(N) ET B(N)*****
405 -----
410 FOR I=1 TO N
420 FOR J=1 TO N
430 J(I,J)=A(I)*V(J)^2+E(I)*V(J)+C(I)
440 B(I,J)=S(I)*V(J)^2+P(I)*V(J)+Q(I)
450 NEXT J :NEXT I
460 PRINT :PRINT
470 *****RECHERCHE DU MINIMUM*****
475 -----
480 QI=0
490 IT=0
```



```

500 PRINT "-----"
510 PRINT " ITERATION N $\frac{1}{2}$ ";IT;" : "
520 PRINT "-----"
525 PRINT :PRINT "DEFINIR L'INTERVAL SUR LEQUEL ON VEUT TRAVAILLER:"
527 INPUT "      T=";T
530 FOR I=1 TO N
540 FOR J=1 TO N
550 BQ(I,J)=QI*B(I,J)
560 GOSUB 10000
570 H(I,J)=J(I,J)+IBQ(I,J)
580 NEXT J
590 MIN(I)=H(I,1)
600 FOR L=1 TO N
610 IF MIN(I)<=H(I,L) THEN 635
620 MIN(I)=H(I,L)
630 IND=L
635 IF L<>1 THEN 640 ELSE IND=1
640 NEXT L
650 U(I)=V(IND)
660 PRINT " LA SOLUTION DE PA";I;" EST : ";
670 PRINT "U(";I;")=";V(IND)
680 PRINT
690 PRINT V(IND) "minimise PA";I;" pour un QI fixe a :";QI
700 NEXT I
710 PRINT :PRINT
720 ***** CALCUL DE LA NORME *****
725 -----
730 DA=0
740 FOR S=1 TO N
750 D=0 : DI=0
760 U=U(S)
770 FOR I=1 TO N
780 D=D+(S(I)*U^2+P(I)*U+Q(I))
790 NEXT I
800 DI=D^2
805 DA=DA+DI
810 NEXT S
820 DN= SQR(DA)
830 PRINT " LA DISTANCE ENTRE LA SOLUTION TROUVEE ET LA SOLUTION "
840 PRINT "OPTIMUM EST DONNEE PAR LA NORME EUCLIDIENNE DE LA "
850 PRINT "SOMME DES Bi(Ui(q))."
860 PRINT
870 PRINT " POUR QI=";QI;" LA NORME EST EGALE A :";DN
880 PRINT :PRINT :PRINT
890 ***** TEST DE LA NORME *****
895 -----
900 INPUT " L'ERREUR PERMISE EST DE NU=";NU
910 IF DN>=NU THEN 1760
920 PRINT "ON EFFECTUE UNE PROCEDURE DE MODIFICATION"
930 ***** PROCEDURE DE MODIFICATION *****
940 -----
950 PRINT :PRINT
960 PRINT "PROCEDURE DE MODIFICATION "
965 PRINT "-----"
970 FOR I=1 TO N
980 SS=0 : PS=0 : QS=0

```



```

990 FOR T=1 TO N
1000 SS=SS+S(T)
1010 PS=PS+P(T)
1020 QS=QS+Q(T)
1030 NEXT T
1040 PRINT "RESOLUTION DE L'EQUATION ";SS;"*X^2+";PS;"*X+";QS;"=0 "
1050 IF SS=0 THEN 1240
1060 D=PS^2-4*SS*QS
1065 PRINT "VOICI LE DESCRIMINANT : D=" ; D
1070 IF D=0 THEN 1200
1080 IF D<0 THEN X5=-PS/(2*SS) : PRINT " LES DEUX SOLUTIONS SONT IMAGINAIRE
UR PARTIE RELLE EST EI=";X5 : GOTO 1355
1090 X1=(-PS+SQR(D))/(2*SS)
1100 X2=(-PS-SQR(D))/(2*SS)
1110 IF X1 <=0 THEN 1170
1120 IF X2 <=0 THEN UA(I)=X1 : PRINT " LA RACINE UA(";I;")=";UA(I); GOTO 1
1130 PRINT " L'EQUATION ADMET DEUX RACINES POSITIVES : " 1360
1140 PRINT " UA1(";I;")= "; X1
1150 PRINT " UA2(";I;")= "; X2
1160 IF X1>X2 THEN UA(I)=X1 ELSE UA(I)=X2 : GOTO 1360
1170 IF X2 >0 THEN UA(I)=X2 : PRINT "LA RACINE UA(";I;")= "; X2 : GOTO 13
1180 PRINT "LES DEUX SOLUTIONS DE L'EQUATION SONT NEGATIVES ." 1360
1190 GOTO 1760
1200 X3= -PS/(2*SS)
1210 UA(I)=X3
1220 PRINT "LA RACINE EST UA(";I;")= "; X3
1230 IF X3>0 THEN 1360 ELSE 1760
1240 IF PS=0 THEN 1290
1250 IF QS=0 THEN 1340
1260 X4=-QS/PS
1265 UA(I)=X4
1270 PRINT "LA RACINE EST UA(";I;")= "; X4
1280 IF X4>0 THEN 1360 ELSE 1760
1290 IF QS=0 THEN 1320
1300 PRINT " EQUATION IMPOSSIBLE "
1310 GOTO 1760
1320 PRINT " LA RACINE EST INDETERMINEE "
1330 GOTO 1760
1340 PRINT " UNE RACINE UNIQUE UA(";I;")= 0 "
1350 UA(I)=0 : GOTO 1360
1355 IF EI>0 THEN UA(I)=EI ELSE 1760
1360 ***** CALCUL DES GAM(N) *****
1365 -----
1370 PRINT " CALCUL DE GAM(";I;") : "
1375 PRINT "-----"
1380 LJ=0
1390 LB=0
1400 FOR S=1 TO N
1410 LJ=LJ+ (A(S)*U(I)^2+E(S)*U(I)+C(S))
1420 LB=LB+ (S(S)*U(I)^2+P(S)*U(I)+Q(S))
1430 NEXT S
1440 BQ=QI*LB
1450 GOSUB 10000
1460 L(I)=LJ+IBQ
1470 PRINT " LE LAPLACIEN DE LA ";I;"ieme SOLUTION OBTENUE EST "
1475 PRINT " L(";I;")= "; L(I)

```



```

1480 LJM=0
1490 LBM=0
1500 FOR S=1 TO N
1510 LJM=LJM+ (A(S)*UA(I)^2 + E(S)*UA(I) +C(S))
1520 LBM=LBM+ (S(S)*UA(I)^2 + P(S)*UA(I) +Q(S))
1530 NEXT S
1540 BQ=QI*LBM
1550 GOSUB 10000
1560 LM(I)=LJM+IBQ
1570 PRINT " LE LAPLACIEN DE LA ";I;"ieme SOLUTION TRANSFORMEE EST "
1575 PRINT " LM(";I;")= "; LM(I)
1580 GAM(I)=LM(I)-L(I)
1590 PRINT " L'ECART PERMIS ENTRE UM(";I;") ET U(";I;") EST :"
1595 PRINT " GAM(";I;")= "; GAM(I)
1600 INPUT " DEFINIR LA VALEUR DE EPS (du test final)"; EPS
1610 IF GAM(I)>EPS THEN 1740
1620 UM(I)=UA(I)
1630 NEXT I
1640 PRINT " ***** LA SOLUTION OPTIMUM DU PROBLEME GLOBAL ***** "
1645 PRINT " ***** (PA) EST ATTEINTE A EPS PRES ***** "
1650 PRINT " ***** * * * * * ***** "
1655 PRINT "-----"
1660 PRINT " ***** VALEUR DE LA SOLUTION OPTIMUM ***** "
1670 PRINT "-----"
1680 FOR J=1 TO N
1690 PRINT " * * * UM(";J;") = "; UM(J) ; " * * * "
1700 NEXT J
1710 PRINT "-----"
1720 PRINT :PRINT TAB(30)"*****"
1730 END
1740 PRINT "***** LA CONDITION SUR LES GAM(N) N'EST PAS *****"
1750 PRINT "***** VERIFIEE POUR TOUTES LES VALEURS DE GAM(N)*****"
1760 PRINT " * * * * * ON PASSE A L'ITERATION SUIVANTE * * * * * "
1770 '***** CALCUL DE QI(N+1) *****
1775 '-----
1780 PRINT :PRINT " ** CALCUL DE RO ** "
1790 PRINT
1800 PRINT " FIXER LA PREMIERE VALEUR DE RO (d'apres le prb. a resoudre) "
1810 INPUT " RO= ";RO
1820 PRINT
1830 INPUT "LA VARIATION A APPORTEE SUR RO EST ALP ="; ALP
1840 BU=0
1850 FOR I=1 TO N
1860 SS=0 : PS=0 : QS=0
1870 FOR T=1 TO N
1880 SS=SS+S(T)
1890 PS=PS+P(T)
1900 QS=QS+Q(T)
1910 NEXT T
1920 BU=BU + (SS*U(I)^2 + PS*U(I) + QS)
1930 NEXT I
1940 BQ=QI*BU
1950 GOSUB 10000
1960 SBR=IBQ
1965 FOR I=1 TO N
1970 QIR= QI+RO*BU

```



```

1980 BQ= QIR*BU
1990 GOSUB 10000
2000 SSR=IBQ
2010 M(I)= SSR-SBR
2020 PRINT " LA VALEUR APPROCHEE DE M(q(n+1))-M(q(n)) "
2030 PRINT " POUR RO =";RO;" EST : M = "; M(I)
2040 IF M(I)<M(I-1) THEN 2070
2050 RO=RO+ALP
2060 NEXT I
2070 RO=RO-ALP
2080 PRINT " LA VALEUR QUI MAXIMISE M(q(n+1))-M(q(n)) EST : "
2090 PRINT "      RO = "; RO
2095 PRINT "      ----- " : PRINT
2100 PRINT " *** CALCUL DE QI(n+1) ***"
2110 PRINT " -----"
2120 RB=0
2130 FOR T=1 TO N
2140 RB=RB+(SS*U(T)^2 + PS*U(T) + QS )
2150 NEXT T
2160 P=QI + RO*RB
2170 IF P>0 THEN QIN=P : GOTO 2190
2180 QIN=0
2190 PRINT " LA VALEUR DE QI(n+1) EST : "
2200 PRINT "      QI(";IT+1;")= "; QIN
2210 PRINT " -----"
2220 IT=IT+1
2230 QI=QIN
2240 GOTO 500
10000 PRINT
10010 '***** CALCUL DE L'INTEGRAL (1/2 SUBROUTINE)*****
10015 '-----
10020 IBQ=BQ*T
10030 RETURN

```


- (10) S. MOUSSAOUI et Y. FOUJILCHERIF " Modélisation , Simulation
Contrôle Optimal d'un grand système. "
Projet de fin d'étude Juin 1986
- (11) F. CHIGARA " Simulation sur ordinateur d'une unité de
montage du point de vue du contrôl optimal du stock
de composants "
Thèse de Majister Mars 1987
- (12) G. DOUMEINGTS , D. BREUIL et L.PUN " La gestion de la
production assistée par ordinateur "
HERMES Juin 1983
- (13) S. MALLAOUI et M. MAZIDI " Analyse de systèmes multidimen-
sionnelles. Agrégation-Décomposition-Coordination "
Projet de fin d'étude Janvier 1988
- (14) J.Y. GOURET " Fichiers Techniques de la production "
Techniques de l'Ingénieur
H6530 Mai 1985
- (15) G. DOUMEINGTS et M.C. MAISONNEUF " Gestion de la Production
Assistée par Ordinateur "
Techniques de l'Ingénieur
H6540 Mars 1987
- (16) J. MENEVEAU " Introduction-à la gestion informatisée de la
production "
Techniques de l'Ingénieur
H6510 Mars 1982