RÉPUBLIQUE DU SÉNÉGAL

ÉCOLE POLYTECHNIQUE DE THIÈS

PROJET DE FIN D'ETUDES

Titre Conception d'un logiciel interactif de ogrammation linéaire (simplexe et analyse nsibilité)

Auteur Pape Demba GNINGUE

Génie Mécanique

Date JUIN 1984

ECOLE POLYTECHNIQUE DE THIES

PROJET
DE
FIN D'ETUDES

TITRE : CONCEPTION D'UN LOGICIEL INTERACTIF DE PROGRAMMATION LINEAIRE

Auteur : Papa Damba GNINGUE
Génie : Mécanique
Date : Mai 1984

REMERCIEMENTS

Avant de commencer la rédaction de cette étude, je voudrais qu'il me soit permis de remercier très sincèrement mon directeur de projet, Mr. André LANGEVIN qui m'a été d'un grand secours par sa disponibilité, ses conseils précieux et pour la documentation fournie.

Mes remerciements iront aussi au technicien du centre de calcul Mr Belanger pour sa présence très utile mais aussi et surtout à tous mes camarade de promotion dont la présence et la compréhension ont été mon plus grand soutien moral.

Enfin je ne terminerai pas sans exprimer ma profonde gratitude à tous les professeurs qui ont bien voulu assurer la surveillance du centre de calcul à des heures non ouvrables.

SOMMAIRE

Il s'agit dans cette étude de mettre au point une méthode de résolution de programme linéaire par ordinateur utilisant l'algorithme du simplexe dans sa forme la plus adaptable.

Le programme établi devra pouvoir faire l'analyse de la sensibilité des résultats et avoir la possibilité de tirer les données du problème d'un fichier.

La procédure employée est la suivante.

- Faire l'analyse du problème après le choix de l'algorithme de résolution.
- Tracer l'organigramme puis rédiger le programme sur la base de cet organigramme.

L'exécution du programme fournit à l'utilisateur la solution du problème proposé et s'il le désire la variation de la solution suite à :

- un paramétrage de coefficient
- un changement de coefficient
- une addition d'une variable
- une addition d'une contrainte.

TABLE DES MATIERES

INTRODUCTION

La programmation linéaire est une méthode de calcul qui permet à l'utilisateur de résoudre des problèmes de gestion d'une manière optimum.

C[illegible] élément très important de la recherche op[illegible] en ce sens qu'il résoud des problèmes dé[illegible] comme par exemple la mise sur pied d'un p[illegible] de production optimum lorsque l'on dispose de ressources limitées en quantité et dans le temps.

Une lacune de la programmation linéaire est qu'elle résoud en principe seulement des problèmes déterministes aux données connues d'une manière certaine. Rares sont les problèmes de ce type. Mais la programmation linéaire peut être appliquée à des problèmes moins précis. Dans ces cas, l'analyse de sensibilité effectuée après l'optimisation permet de déterminer les effets de la variation des différents paramètres sur la solution

L'étude qui suit cette introduction a pour but la conception d'un logiciel interactif de programmation linéaire Dans ce logiciel, la résolution du programme linéaire sera suivie d'une analyse de la sensibilité des résultats. Cette analyse de sensibilité sera basée soit ~~sur~~ des variations ponctuelles des coefficients, soit sur le paramétrage de ces mêmes coefficients.

Ch.I ANALYSE DU PROBLEME

I La programme linéaire

Un programme linéaire se présentera en général sous la forme :

$$\text{Maximiser (minimiser)} \sum_j^m c_j x_j = z$$

$$\text{sous les contraintes} \quad b_{1i} \leq (\geq) \sum_j^m a_{ij} x_j \leq (\geq) b_{2i}$$

$$l_{1j} \leq x_j \leq l_{2j}$$

$$\text{avec} \quad i = 1, 2, \ldots, m$$

i = numéro d'une contrainte (ressources)

j = indice d'une variable (activités)

m = nombre de contraintes

n = nombre de variables

x_j = variable niveau d'une activité j

b_{1i}, b_{2i} = quantités minimum et maximum d'une ressource i

c_j = coefficient de la fonction économique (profit unitaire d'une activité j)

a_{ij} = coefficient des contraintes : quantité de ressource i pour une activité j

z = profit total

l_{1j}, l_{2j} = bornes d'une variable : limites d'une activité j

Pour simplifier l'analyse, on traitera uniquement du problème de maximisation, tout programme linéaire pouvant se mettre sous forme d'une maximisation.

Le but de l'étude étant la résolution par ordinateur d'un programme linéaire (P.L.), nous allons faire une brève étude de différentes méthodes pour choisir celle qui s'adaptera le mieux à la programmation. Les critères du choix seront entre autres la généralité de la méthode, la rapidité d'exécution et l'économie de l'espace mémoire. Les différentes méthodes étudiées seront toutes basées sur le simplexe.

Avant de passer à l'analyse des méthodes, nous ferons quelques transformations. Avant tout, chaque contraintes sera scindée en deux. Le problème devient :

$$\text{Max}: \quad z = \sum c_j x_j$$

$$\text{s.c.} \quad \sum a_{ij} x_j \leq (\geq) b_i$$

$$0 \leq x_j^+ \leq s(j)$$

$$\text{avec} \quad s_j = l_{2j} - l_{1j}$$

On suppose pour la facilité de l'analyse que $s_j = +\infty$ et on réajustera après la résolution du problème

II Méthode du simplexe (méthode primale)

Comme nous l'avons déjà dit, nous étudions un problème sous la forme :

$$\text{Max} \sum_j^n c_j x_j = z$$

$$\text{s.c} \quad \sum_j^n a_{ij} x_j \le (\text{ou} \ge)\ b_i$$

$$x_j \ge 0 \qquad i = 1, 2, \dots, m$$

Nous avons dans ce problème n variables (non négatives) et m contraintes.

Une façon pratique de résoudre le problème est de le mettre sous forme standard, c'est à dire remplacer les inéquations par des équations. Ceci permet d'avoir une solution de base.

Pour ce faire, on ajoute à chaque contrainte des variables dites d'écart et des variables artificielles si besoin est.

1. Contrainte ayant un signe $\le$

On ajoute une variable d'écart affecté d'un coefficient 1, x^e

2. Contrainte ayant un signe $\ge$

On ajoute une variable d'écart affectée d'un coefficient -1 et une variable artificielle affectée d'un coefficient $+1$, x^a

Dans le premier cas nous aurons une contrainte de la forme :

$$\sum a_{ij} x_j + x^e_{n+i} = b_i$$

Dans le deuxième cas, la contrainte sera de la forme :

$$\sum a_{ij} x_j - x^e_{n+i} + x^a = b_i$$

Lorsque le programme linéaire est écrit sous cette forme, les variables d'écart constituent la solution de base et sont affectées d'un profit unitaire nul. Les variables artificielles sont affectées d'un profit unitaire égal à l'infini négatif ($-\infty$) ceci pour éviter qu'elles entrent dans la base. On suppose que $-\infty$ est une valeur finie très faible $-M$ (perte)

Avant de commencer le processus d'optimisation, il faut éliminer les variables artificielles du système d'équations. Pour cela, on soustrait M fois de la fonction objectif chaque contrainte contenant une variable artificielle.

Il faut signaler que toutes ces opérations se font sur le tableau du simplexe

Lorsque toutes ces opérations sont faites, on peut alors passer à la phase d'optimisation du tableau qui est sous la forme suivante:

	x_1	x_2	$\cdots$	x_n	x_{n+1}	$\cdots$	x_{n+m}	b
x_{n+1}	a_{11}	a_{12}	$\cdots$	a_{1n}	1	$\cdots 0 \cdots$	0	b_1
$\vdots$					0		0	$\vdots$
x_{n+m}	a_{m1}			a_{mn}	0	$\cdots 0 \cdots$	1	b_m
x_0	$-c_1$	$-c_2$	$\cdots$	$-c_n$	0	$\cdots$	0	0
						base		

Nous avons obtenu une solution de base initiale composée dans notre tableau des variables x_{n+1} à x_{n+m} (dites de base).

Si parmi les autres variables, il en existe qui ont des coefficients négatifs à la dernière rangée, il faut choisir la variable (hors base) ayant le plus petit coefficient pour la faire entrer parmi les variables de base: c'est le premier critère ou critère d'entrée.

Si tous les coefficients sont positifs ou nuls, la solution est optimale.

Le deuxième critère ou critère de sortie consiste au calcul du rapport entre les coefficients du second membre et ceux correspondant, de la variable entrante x_q (b_i/a_{iq}). Le plus petit rapport b_p/a_{pq} indique la variable qui doit sortir de la base, x_p

On procède alors à l'opération de pivotage qui consiste à intervertir x_p et x_q en faisant des opérations sur tous les coefficients.

Ainsi un a_{ij} quelconque devient :

$$a^*_{ij} = a_{ij} - (a_{iq}/a_{pq}) \times a_{pj}$$

et les a_{pj} deviennent :

$$a^*_{pj} = a_{pj}/a_{pq}$$

Il faut noter que ces transformations concernent tous les coefficients du tableau. (ref. Techniques et applications de la Recherche Opérationnelle – A. MARTEL)

Il s'agit maintenant, après l'opération de pivotage, de recommencer la procédure jusqu'à l'obtention d'une solution optimale.

III Méthode du dual-simplexe

Nous allons maintenant étudier une méthode qui utilise une propriété du programme linéaire, la dualité (ref. A Martel. p: 166).

Il faut seulement savoir que la dualité permet de transformer un problème de maximisation en un problème de minimisation et vice-versa.

Ainsi, le dual d'un problème (primal) a pour dual le problème primal lui-même.

Lorsque l'on pose un problème sous sa forme duale, les c_j du primal deviennent les coefficients du second membre et les b_i deviennent les coefficients de la fo[illegible]n objectif. Et dans un tableau dual optimisé p[illegible] critère du simplexe, on peut déduire les solutions [illegible]ème primal.

La méthode du dual-simplexe est utilisée lorsque pour un tableau primal donné :

- tous les c_j sont positifs ou nuls (solution optimale)
- et il existe des b_i négatifs (solution non réalisable)

L'algorithme du dual simplexe s'applique alors directement sur le tableau primal.

La procédure est l'inverse de celle du simplexe.

Le premier critère est le critère de sortie et consiste au choix de la variable sortante.

Il s'agit de choisir le plus petit coefficient du second membre b_p : x_p sera la variable sortante. Mais lorsque tous les b_i sont non négatifs, la solution est optimale et réalisable.

La deuxième critère ou critère ou critère d'entrée consiste au calcul des rapports $c_j/-a_{pj}$ pour les a_{pj} négatifs. Le plus petit rapport $c_q/-a_{pq}$ indiquera la variable entrante x_q.

L'opération de pivotage ici a le même rôle et se fait de la même façon que dans la méthode du simplexe.

La procédure continue jusqu'à ce que tous les coefficient du second membre soient positifs ou nuls. La solution est alors optimale.

IV LA forme révisée du simplexe

Cette méthode de résolution passe par la forme matricielle.

Soit le problème

$$\begin{aligned} \text{Max} : \; & z = \sum c_j x_j \\ \text{s.c} \; & \sum a_{ij} x_j = b_i \\ & x_j \geq 0 \end{aligned}$$

On peut écrire :

$$\begin{aligned} \text{Max} : \; & [z] = [C][x] \\ \text{s.c} \; & [A][x] = [b] \\ & [x] \geq [0] \end{aligned}$$

et on parlera de forme matricielle.

On a alors la matrice $[A]$, le vecteur ligne $[C]$ et le vecteur colonne $[b]$.

La matrix $[A]$ est décomposée et on obtient

$[B, D]$

B = matrix des variables de base

D = matrix des variables hors base

De même pour le vecteur $[C]$, on obtient $[C_B, C_D]$.

Le programme linéaire se met alors sous la forme

$$\begin{cases} [B, D]\begin{bmatrix} x_B \\ x_D \end{bmatrix} = [b] \\ -z + [c_B, c_D]\begin{bmatrix} x_B \\ x_D \end{bmatrix} = 0 \end{cases}$$

La forme matricielle sara

$$\left.\begin{array}{c|c:c|c} 0 & B & D & b \\ \hline -1 & C_B & C_D & 0 \end{array}\right\} \Rightarrow \begin{array}{l} 0 + Bx_B + Dx_D = b \\ -1 + C_B x_B + C_D x_D = 0 \end{array}$$

Si l'on multiplie les contraintes par l'inverse de la base B^{-1}, on obtient

$$B^{-1}Bx_B + B^{-1}Dx_D = B^{-1}b$$

$$x_B + B^{-1}Dx_D = B^{-1}b \Rightarrow x_B = B^{-1}b - B^{-1}Dx_D$$

Ceci donne pour la fonction objectif.

$$-z + c_B(B^{-1}b - B^{-1}Dx_D) + c_D x_D = 0$$

$$-z + (c_D - c_B B^{-1}D)x_D = -c_B B^{-1}b$$

et le tableau initial est :

0	I	$B^{-1}D$	$B^{-1}b$	=	0	I	$\hat{D}$	$\hat{b}$
-1	0	$c_D - c_B B^{-1}D$	$-c_B B^{-1}b$		-1	0	$\hat{c}_D$	$\hat{z}$

I est alors le vecteur unité et $c_B B^{-1}$ est la valeur actuelle de la fonction à optimiser.

Lorsque tous les éléments de $\hat{c}_D$ sont positifs ou nuls, la solution est optimale.

Dans le cas contraire, la variable entrante sera le vecteur $\hat{a}_j$ ayant le plus petit $\hat{c}_j$ et sera appelée $\hat{a}_q$.

Il faut alors calculer $[\hat{a}_j^*] = [B^{-1}][\hat{a}_j]$ puis les rapports $\hat{b}_i/\hat{a}_{iq}$ pour tout $\hat{a}_{iq}$ positif. Le plus petit rapport définira la variable sortante x_p

Pour continuer la procédure de calcul, on détermine l'inverse $[B^{-1}]$ de la nouvelle base puis $[\hat{b}^*] = [B^{-1}][\hat{b}]$.

Ce calcul correspond au pivôtage autour du coefficient $\hat{a}_{pq}$.

Pour cette méthode comme pour les autres, la solution sera optimale lorsque tous les éléments de c_D seront positifs ou nuls.

V La forme produit de l'inverse

Cette méthode de calcul n'est rien d'autre qu'une variante de la forme révisée.

Dans ce cas ci, l'inverse de la base s'obtient par le produit de deux matrice.

Le principal avantage de cette méthode est qu'elle nécessite moins de calculs que les autres.

Lors d'une opération de pivotage, l'élément pivot est a_{pq} et donc la colonne pivot est a_q et on obtient une matrice ayant la forme [E][D]

[E] est une matrice carrée telle que :

$E_{ij} = 1$ pour $i = j \neq p$

$E_{ij} = 0$ pour $i \neq j$ et $j \neq p$

$E_{ij} = -a_{ij}/a_{pj}$ pour $i \neq p$ et $j = p$

$E_{ij} = 1/a_{pj}$ pour $i = p$ et $j = p$

On calcule $[B^{-1}]$ à partir de [E]

Pour la k-ième opération de pivotage le calcul de $[B^{-1}]$ se fait comme suit :

$$[B^{-1}] = [E]_k \times [[E]_{k-1} \times [E]_{k-2} \times \cdots \times [E]_2 \times [E]_1]$$

Pour le reste des opérations, la procédure est la même que pour la forme révisée du simplexe.

Si la solution n'est pas optimale, on définit $[E]_{k+1}$ et on continue les itérations.

(Réf Programmation linéaire . M. SIMONNARD)

Ch.II COMPARAISON DES DIFFERENTES METHODES

I La méthode du simplexe

C'est la méthode la plus simple, du moins en ce qui concerne le calcul manuel.

Un inconvénient majeur est le nombre élevé d'opérations à effectuer (réf. A.Martel)

Pour un P.L. à n variables, il faut en général m à $\frac{3n}{2}$ opérations de pivotage pour obtenir l'optimum. A chaque pivotage, on recalcule m colonnes au moins, sans compter le calcul des ratios.

En fait le calcul de certains coefficients n'interviennent pas directement dans l'obtention de l'optimum.

Et l'algorithme du simplexe ne suffit pas toujours pour resoudre un P.L.

En effet, si tous les c_j deviennent non négatifs alors qu'il existe des b_i négatifs, la solution obtenue n'est pas réalisable. Il faut alors continuer les itérations avec la méthode du dual simplexe dont nous allons parler.

II Le dual simplexe et la dualité.

Comme on l'a dit plus haut, on peut obtenir la solution d'un P.L. en résolvant son dual.

Pour un programme donné, la résolution par le dual peut être plus rapide que la résolution par le primal. En fait la résolution dépend pour la vitesse du nombre de variable n et du nombre de contraintes m. Lorsque n < m, la résolution du dual est plus rapide et lorsque m < n, c'est la résolution du primal qui est plus rapide.

Malgré cela les différences entre les temps de résolution ne sont pas considérables.

Avec la dualité, on a l'algorithme du dual-simplexe qui permet de résoudre un P.L. entamé avec le simplexe et qui a abouti à une solution non réalisable pour les raisons vues plus haut.

En substance, on peut dire que le simplexe et le dual simplexe se valent et se complètent.

III La forme revisée du simplexe

Avec l'algorithme du simplexe, nous avons vu la résolution du programme linéaire avec des opérations de pivotage qui doivent à chaque itération aboutir à la modification de tous les coefficients a_{ij} de chaque variable x_j.

Comme on l'a déjà dit, le calcul de tous les coefficients des contraintes n'est pas nécessaire à la résolution d'un P.L.

L'algorithme révisé permet d'obtenir une solution en réduisant le nombre de calculs.

IV La forme produit de l'inverse.

Cette méthode est une variante de la forme révisée. Sa présentation montre qu'elle a un double avantage sur la forme révisée :

- le gain en temps-machine dans le calcul de l'inverse de la base.

- le gain en espace mémoire grâce à la réduction du nombre d'informations à stocker (en effet on se contente de stocker la colonne non nulle de la matrice unitaire [E] au lieu de conserver l'inverse de la base en entier).

Après cette brève analyse de différentes méthodes, on peut dire que la forme révisée du simplexe sous la forme produit de l'inverse est l'algorithme le plus adaptable à la programmation sur ordinateur.

Ceci est totalement vrai lorsque la contrainte sur les variables est seulement.

$$x_j \geq 0$$

Mais, lorsque la contrainte est de la forme :

$$l_{1j} \leq x_j \leq l_{2j} ,$$

ça fait changer de profil et on doit tenir compte de ces variations.

Pour ce faire, il est nécessaire d'intégrer dans la forme produit de l'inverse des étapes de calcul étrangère à cet algorithme, comme nous allons le voir dans ce qui suit.

Ch.III ANALYSE DETAILLEE ET METHODE DE CALCUL

Le choix de la méthode étant réalisé, il s'agit maintenant pour nous d'adapter cette méthode aux différents calcul que nous voulons faire.

Tout d'abord, rappelons que le programme pour lequel cette étude est faite doit être à même de résoudre le programme linéaire puis de paramétrer certains de ces coefficients pour mettre en évidence leurs effets sur les résultats. Il doit aussi pouvoir faire l'analyse de sensibilité des résultats.

Ces différents aspects doivent être pris en considération dès le début de l'analyse.

Ensuite comme nous l'avons dit à la fin du chapitre précédent, les variables du P.L. peuvent être soumises à des contraintes du type :

$$l_{1j} \leq x_j \leq l_{2j}$$

Dans ce cas, on pose la contrainte sous la forme :

$$0 \leq x_j^* \leq s_j \quad \text{avec } x_j^* = x_j - l_{1j} \text{ et } s_j = l_{2j} - l_{1j}$$

En conséquence, les x_j dans le programme linéaire seront remplacés par $x_j^* + l_{ij}$, aussi bien dans les contraintes que dans la fonction objectif.

Reste maintenant le problème de la borne supérieure s_j.

La façon d'en tenir compte est la suivante :

Chaque fois qu'une variable x_j^* atteint sa borne supérieure s_j pendant la résolution du P.L., cette variable est remplacée par $\bar{x}_j^* = s_j - x_j^*$ qui est le complément de x_j^* dans l'intervalle $[0, s_j]$ (réf : Techniques et applications de la R.O., A. MARTEL)

Nous verrons plus en détail dans l'organnigramme les transformations impliquées par ce remplacement.

I Résolution du programme linéaire

Il s'agit d'un problème de maximisation, présenté sous forme suivante :

$$\text{Max } \sum c_j x_j$$

$$\text{s.c } b_{i1} \leq \sum a_{ij} x_j \leq b_{i2}$$

$$l_{1j} \leq x_j \leq l_{2j}$$

Avant de passer à la résolution du P.L. il faut lui faire subir certaines transformations.

1. Aux bornes des variables

On pose : $0 \leq x_j^* \leq s_j$

Le problème devient : $\text{Max} \sum c_j x_j^*$

$$\text{s.c} \quad b_{i1}^* \leq \sum a_{ij} x_j^* \leq b_{i2}^*$$

$$0 \leq x_j^* \leq s_j$$

$$b_{i1}^* = b_{i1} - \sum a_{ij} l_{1j}$$

$$b_{i2}^* = b_{i2} - \sum a_{ij} l_{1j}$$

On se base sur le fait que :

$$\sum a_{ij} x_j^* = \sum a_{ij}(x_j - l_{1j}) = \sum (a_{ij} x_j - a_{ij} l_{1j})$$

2. Aux niveaux des contraintes

Plusieurs cas sont possibles.

— $b_{i1} = -\infty$

si $b_{i2} < 0$, on pose $\sum -a_{ij} x_i \geq -b_{i2}$

si $b_{i2} \geq 0$, on pose $\sum a_{ij} x_j \leq b_{i2}$

— b_{i1} = constante < 0

si $b_{i2} < 0$, on pose $\sum -a_{ij} x_j \geq -b_{i2}$

$\sum -a_{ij} x_j \leq -b_{i1}$

si $b_{i2} > 0$, on pose $\sum + a_{ij} x_j \leq b_{i2}$

$\sum -a_{ij} x_j \leq -b_{i1}$

si $b_{i2} = +\infty$, on pose $\sum -a_{ij} x_j \leq -b_{i1}$

— $b_{i1} > 0$

si $b_{i2} > 0$, on pose $\sum a_{ij} x_j \leq b_{i2}$

$$\sum a_{ij} x_j \geq b_{i1}$$

si $b_{i2} = +\infty$, on pose $\sum a_{ij} x_j \geq b_{i1}$

Ces différentes transformations sont faites dans le P.L. et d'éviter au maximum l'apparition de coefficients du second membre négatifs, qui a pour effet de rendre une solution irréalisable.

Nous nous trouvons maintenant devant un système d'inéquations et il nous faut une solution de base initiale pour pouvoir commencer la résolution.

Cette solution de base initiale est obtenue automatiquement en mettant le P.L. sous la forme canonique, c'est à dire en transformant les inéquations en équations.

L'opération qui permet cette transformation est expliquée dans la description de la méthode du simplexe (Chap. I). Après l'avoir effectuée, nous pouvons passer à la résolution proprement dite

3. Calcul du programme linéaire

Avant de commencer quelque calcul que ce soit, il faut mettre le P.L. sous la forme matricielle.

Il y aura une matrice $A(M,N)$,

un vecteur $C(N)$,

un vecteur $B(M)$,

une matrice $L(2,N)$,

un vecteur $S(N)$,

[illegible] matrices ont une correspondance directe avec les éléments déjà présentés au Chap I.

Toutefois nous préciserons certaines informations

M : nombre d'équations

N : nombre de variables (hors base et de base)

$S(J) = L(2,J) - L(1,J)$

Nous pouvons maintenant passer à la procédure de calcul.

Comme pour la méthode primale du simplexe, il faut déterminer la variable la plus apte à entrer dans la base, c'est à dire la variable ayant le $C(J)$ le plus négatif, $C(Q)$. c'est le premier critère

A partir de là, il faut déterminer la variable qui doit quitter la base. Ce sera alors la variable de de la ligne ayant le rapport $R(I) = B(I)/A(I,Q)$ le plus faible, à condition que cette variable ne dépasse pas sa borne supérieure et que $A(I,Q)$ soit positif. Ceci est le cas général.

Deux autres cas peuvent se présenter :

a) Cas où la variable $X(Q)$ atteint sa borne supérieure, [illegible] lorsque :

$$R(I) = B(I)/A(I,Q) > S(Q)$$

Si $R(I)$ est le plus petit ratio, la variable $X^*(Q)$ ne peut pas sortir de la base et est remplacée par sa variable complémentaire $\overline{X}^*(Q) = S(Q) - X^*(Q)$.

Cela se fait en changeant les signes des éléments de la colonne Q et en remplaçant la colonne des $B(I)$ par :

$$B(I) - A(I,Q) \times S(Q)$$

Dans ce cas, on ne définit pas de variable sortante et on retourne au premier critère.

b) Cas où le coefficient $A(I,Q)$ est négatif.

Si la variable est bornée, le rapport se calcule comme suit :

$$R(I) = (S(K) - B(I))/(-A(I,Q))$$

Pour une variable non bornée, le rapport ne s'évalue pas.

Lorsque $R(I)$ se calcule de cette façon, cela signifie que la variable $X^*(K)$ a été remplacée par sa variable complémentaire $\overline{X}^*(K)$ [ce qui se fait en changeant les signes des coefficients de la ligne I et en remplaçant $B(I)$ par $S(K)-B(I)$] avant de sortir de la base (réf. Alain MARTEL).

[illegible] ratio le plus petit $R(P)$ a été trouvé, le [illegible] $A(P,[illegible])$ est pris comme pivot.

Il s'agit maintenant de calculer le nouveau tableau matriciel en définissant la matrice $E(M+1,M+1)$ telle que :

si $I \neq J$ et $J \neq P$, $E(I,J) = 0$

si $I = J$ et $J \neq P$, $E(I,J) = 1$

si $I \neq J$ et $J = P$ $E(I,P) = -A(I,P)/A(P,P)$

si $I = J = P$ $E(P,P) = 1/A(P,P)$

Cette matrice multipliée par le tableau du P.L sous forme matricielle donne le nouveau tableau matriciel. On retourne alors au premier critère jusqu'à ce que la solution soit optimale.

Lorsque le tableau optimum est obtenu, il ne faut perdre de vue qu'on a travaillé avec $X^*(J)$ et que la vrai variable est $X(J)$.

4. Lecture des résultats dans le tableau final.

Lorsque $X^*(J)$ est une variable de base, sa valeur est égale au $B(I)$ qui se trouve sur la même ligne.

Dans le cas contraire sa valeur est nulle.

Dans [illegible] :

$$X(J) = L(1,J) + X^*(J)$$

Si la variable $X^*(J)$ a atteint sa borne supérieure pendant la résolution, la valeur qu'on obtient du tableau est $\overline{X}^*(J)$.

Dans ce cas on a :

$$X(J) = L(1,J) + X^*(J)$$

$$\Rightarrow \quad X(J) = L(1,J) + S(J) - \overline{X}^*(J)$$

$$\Rightarrow \quad X(J) = L(1,J) + L(2,J) - L(1,J) - \overline{X}^*(J)$$

$$\Rightarrow \quad X(J) = L(2,J) - \overline{X}^*(J)$$

II Analyse de sensibilité des résultats

L'analyse de la sensibilité des résultats a pour objectif de déterminer le comportements des résultats à la suite de la variation de certains paramètres du problème.

Nous considérerons dans cette analyse la variation des coefficients c_j, a_{ij} et b_i et aussi l'addition d'une variable et l'addition d'une contrainte.

L'analyse de sensibilité s'effectue toujours à partir du tableau final du P.L.

La méthode consiste à insérer la variation dans le tableau final, puis à effectuer si nécessaire certaines itérations dans le but d'obtenir une solution de base.

Après l'obtention de la solution de base, il s'agit maintenant de vérifier que la solution est optimum et réalisable. Dans le cas contraire, il faut utiliser une procédure d'optimisation qui sera selon le cas la méthode vue précédemment ou le dual simplexe.

Comme nous l'avons vu au début de l'étude, le dual simplexe ne diffère du simplexe qu'au niveau des choix des variables sortantes et entrante. En effet, pour le dual-simplexe on choisit d'abord la variable sortante alors que pour le simplexe on choisit d'abord la variable entrante.

Pour l'adaptation de cette méthode aux variables bornées, il suffit de remplacer chaque variable qui atteint sa borne par son complémentaire aussitôt que cela arrive. Cela se fait en changeant les signes de tous les coefficients de la ligne de la variable, puis de remplacer le coefficient du côté droit b_i par $s_k - b_i$ (s_k étant la borne de la variable x_k.

Après [illegible] exposés de la méthode du dual simplex[illegible] nous maintenant passer à l'analyse de sensibilité proprement dite.

1. Changement dans la fonction économique (c_j)

Lorsqu'on fait varier un coefficient de la fonction économique, soit $C(Q)$ d'une valeur constante T, il suffit d'ajouter cette valeur T au coefficient $C(Q)$ du tableau final

Si $X(Q)$ est une variable de base, il faut réécrire la ligne des coefficient $C(I)$ de façon à retrouver $C(Q) = 0$. Pour arriver à cela, il faut faire la transformation suivante pour chaque $C(J)$:

$$C(J) = C(J) - C(Q) \times A(K,J)$$

K étant l'indice de la ligne de la variable de base $X(Q)$.

Après cela, il faut maintenant vérifier si la tableau est optimum avec les critères déja exposés, c'est à dire vérifier si tous les C(J) sont positifs ou nuls.

Si tel n'est pas le cas, la solution n'est pas optimum et il faut recommencer la procédure d'optimisation avec le même algorithme que pour le P.L. initial.

2. Changement au second membre (b_i)

Le problème est maintenant la variation d'un coefficient du second membre soit B(K) d'une valeur de T.

Dans ce cas, il faut faire subir à tous les coefficients du second membre la transformation suivante

B(I) = B(I) + A(I, K+N) x T (réf. A. MARTEL)

Le tableau obtenu est optimum si tous les B(I) sont positifs ou nuls.

S'il existe un seul nouveau B(I) qui soit négatif, il faut réoptimiser le tableau en utilisant la méthode du dual-simplexe

3. Changement aux coefficients des contraintes (a_{ij})

Ce changement est un peu plus compliqué que les deux précédents qui sont relativement simples à effectuer

Il s'agit de faire varier le coefficient A(K,L) d'une valeur T. Pour insérer cette variation dans le tableau final, il faut d'abord transformer la colonne L de la façon suivante :

$$A(I,L) = A(I,L) + A(I,K+N) \times T$$

$$\text{et} \quad C(L) = C(L) + C(K+N) \times T$$

Mais ce changement du coefficient A(K,L) entraîne un changement du coefficient B(K).

Au début du chapitre, on a vu que du fait de l'existence d'une borne inférieure pour les variables, les coefficients b_i se transformaient comme suit :

$$b_i^* = b_i - \sum a_{ij}\, l_{ij}$$

l_{ij} étant la borne inférieure de x_j

De ce fait toute variation d'un coefficient A(I,J) influence le coefficient B(I) correspondant.

Ainsi, faire varier le coefficient A(K,L) d'une valeur T implique la variation de B(K) d'une valeur T×L(1,L).

On résoud alors le problème comme dans le cas d'un changement au second membre :

Pour tous les B(I), $\quad B(I) = B(I) - T \times L(1,L) \times A(I,K+N)$

Ces deux transformations effectuées, si X(L) est une variable de base, il faut transformer le tableau par une itération en prenant comme pivot le coefficient A(P,L), P étant l'indice de la ligne de la variable X(L).

Après cela, si la solution n'est pas réalisable, il faut optimiser avec le dual-simplexe comme pour le changement au second membre.

4. Addition d'une variable x_{n+1}

L'introduction d'une variable additionnelle dans le système de contrainte implique l'apparition de nouveaux coefficients A(I, N+1), d'un coefficient C(N+1) et des bornes de la variable, L(1,N+1) et L(2,N+1)

Ce problème se résoud en insérant une nouvelle colonne dans le tableau final. Cette colonne aura les coefficients suivants (ref. A. MARTEL)

$$A(K,J) = \sum A(I,N+1) * A(K,N+I)$$

$$C(J) = \left(\sum A(I,N+1) * C(N+I)\right) - C(N+1)$$

Dans ce cas ci aussi, la borne inférieure L(1,N+1) et les coefficients A(I,J) introduisent une variation de chaque B(I)

Cela se traduit par les changements suivants à effectuer sur chaque B(K)

$$B(K) = B(K) - \sum A(I,N+1) \times L(1,N+1) * A(K,N+I)$$

Dans ce cas, on a deux niveaux d'opérations:

On optimise d'abord le tableau par le simplexe s'il n'est pas optimum.

Ensuite si la solution n'est pas réalisable, on utilise le dual simplexe.

5. Addition d'une contrainte (m+1)

L'introduction d'une contrainte additionnelle implique aussi l'apparition de coefficients A(M+1,J) et de coefficients du second membre B(M+1,1) et B(M+1,2), la contrainte étant de la forme :

$$B(M+1,1) \leq \sum A(M+1,J) * X(J) \leq B(M+1,2)$$

Pour ne pas revenir sur ce qui a déjà été dit, il faut simplement signaler que la contrainte ci dessus doit être ramenées à deux contraintes (voir la page 20).

Ces deux contraintes seront insérées dans le tableau final si elles ne sont pas satisfaites par la solution déja obtenue.

Si les deux contraintes sont satisfaites, le problème ne se pose plus et la solution est toujours optimum.

Dans le cas contraire, il faut transformer le tableau pour réobtenir une solution de base.

La transformation est pratiquement la même que pour le changement aux $C(I)$, mais cette fois, il se fait pour toutes les variables de base.

Cela veut dire que pour chaque variable de base $X(L)$, on fait les opérations suivantes:

$$A(M+1,J) = A(M+1,J) - A(M+1,L) * A(K,J) \quad ,$$

K étant l'indice de la ligne de la variable de base $X(L)$.

Au niveau des $B(I)$, il faut recalculer le coefficient $B(M+1)$:

$$B(M+1) = B(M+1) - \sum A(M+1,J) * L(1,J)$$

Si après toutes ces opérations le tableau n'est pas optimum, il faut l'optimiser avec le dual-simplexe.

III Programmation paramétrique

La programmation paramétrique est une forme d'analyse de sensibilité. La différence est qu'au lieu d'apporter une variation constante à un coefficient, on fait varier ce dernier à un taux fixé jusqu'à une valeur donnée.

Dans cette partie, nous étudierons le paramétrage d'un coefficient de la fonction économique et celui d'un coefficient du second membre.

1 Paramétrage de la fonction économique.

Pour paramétrer un coefficient, il faut savoir avant de commencer à résoudre le P.L qu'on fait de la programmation paramétrique.

Ainsi le tableau initial devra tenir compte de la possibilité de paramétrer un coefficient.

Pour la fonction économique, chaque coefficient C(J) sera affecté d'un vecteur ligne aux éléments nuls sauf à la colonne J où l'élément est égal à -1.

Il faut alors, pendant le processus d'optimisation traiter ces vecteurs-ligne supplémentaires (au nombre de N) comme les coefficients de la fonction économique.

Après l'obtention de la solution optimale, nous avons donc N vecteurs lignes qui ont subi toutes les transformations qui ont été faites au niveau des coefficients de la fonction économique.

Et si l'on veut faire varier le coefficient C(L) d'une valeur T=1, il faut tout simplement ajouter la ligne supplémentaire L au vecteur ligne des coefficients C(J), et optimiser si besoin est.

Sur la base de cela, chaque fois qu'on veut faire varier le coefficient C(J) d'une constante T, on multiplie le vecteur-ligne L par T, puis le résultat est ajouté au vecteur des C(J).

Quand il s'agit de programmation paramétrique, on utilise cette propriété pour déterminer le comportement des résultats dans un domaine de variation défini d'un coefficient. Elle permet aussi de contrôler la vitesse de variation.

2. Paramétrage du second membre.

Le problème est pratiquement résolu par la même méthode que pour le paramétrage de la fonction économique.

Mais à chaque coefficient B(I) sera affecté un vecteur-colonne aux éléments nuls sauf à la ligne i [illegible] l'élément est égal à +1.

[illegible] vecteurs-colonne subiront les même trans[illegible] que les coefficients du second membre.

A ce niveau, le paramétrage d'un B(I) se fait de la même façon que celui d'un C(J) sauf en ce qui concerne l'optimisation qui dans ce cas, se fait à l'aide de l'algorithme du dual simplexe.

IV Conclusion

Nous venons d'étudier l'algorithme du simplexe dans sa généralité et les différentes parties qui doivent être incluses dans le programme que nous nous proposons de mettre au point.

Comme nous pouvons le voir dans l'analyse, l'analyse de sensibilité des résutats fait sans cesse appel à l'algorithme du simplexe et à celui du dual simplexe.

Aussi le programme comprendra-t-il deux sous-programmes pour les critères de choix des variables entrantes et sortantes (un pour le simplexe un pour le dual-simplexe), et un sous-programme commun pour les itérations.

De même un sous-programme sera con-[illegible] chaque type d'analyse de sensibilité, et ceci sera permis par un questionnaire d'uti-[illegible]tôt après la résolution du problème

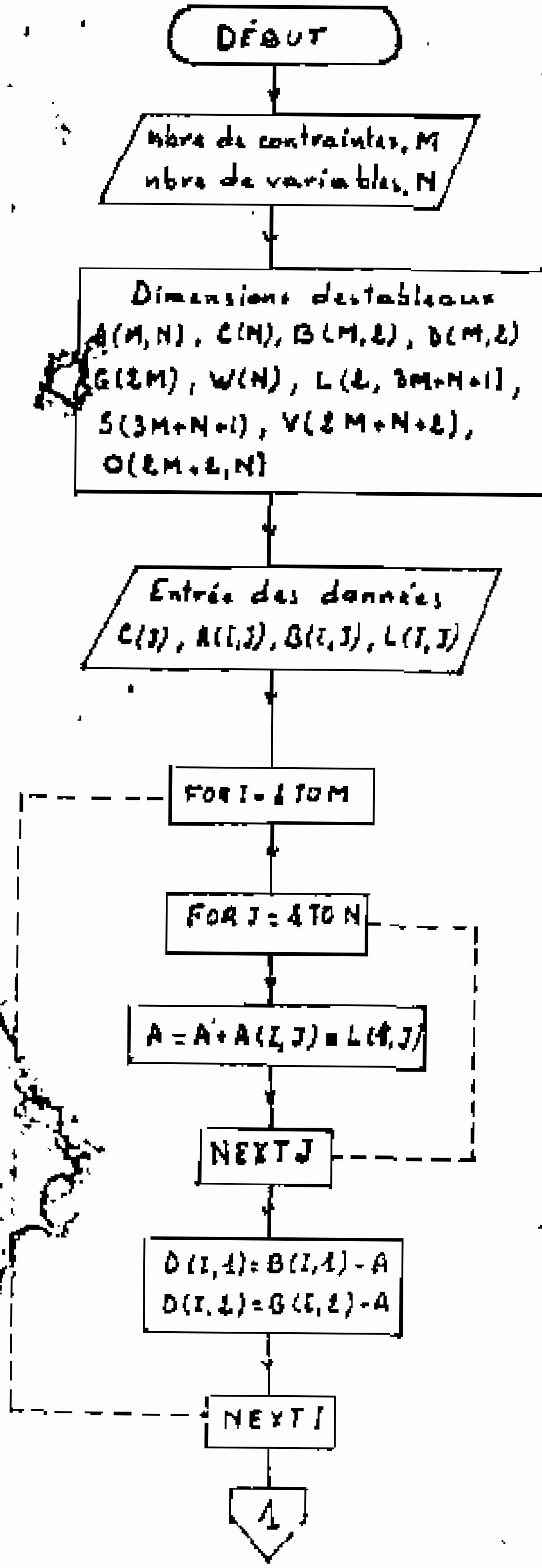
Nous all

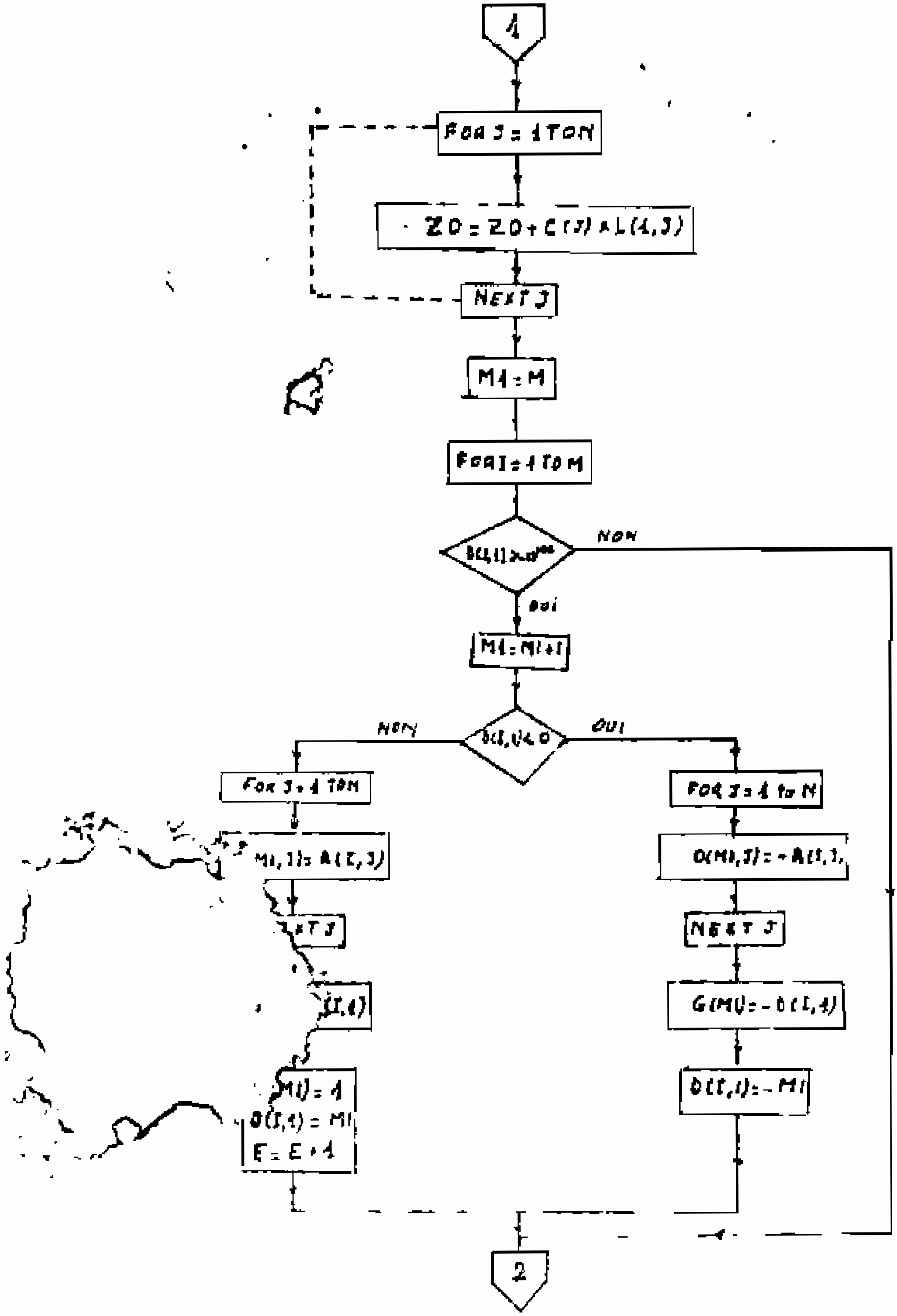
Nous allons maintenant passer à l'organigramme qui est un préalable important à la mise au point du programme.

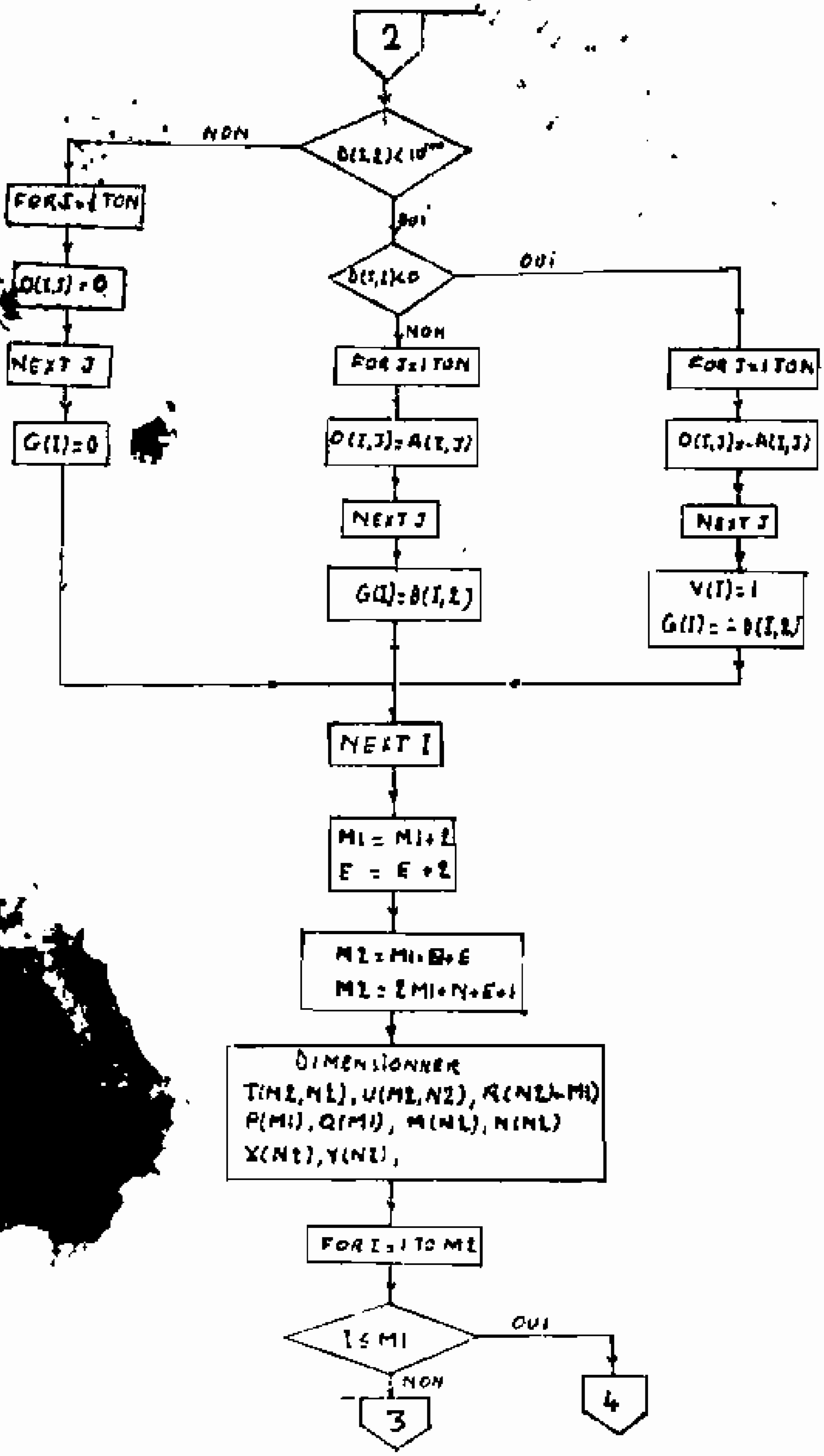
Ch. IV ORGANIGRAMME

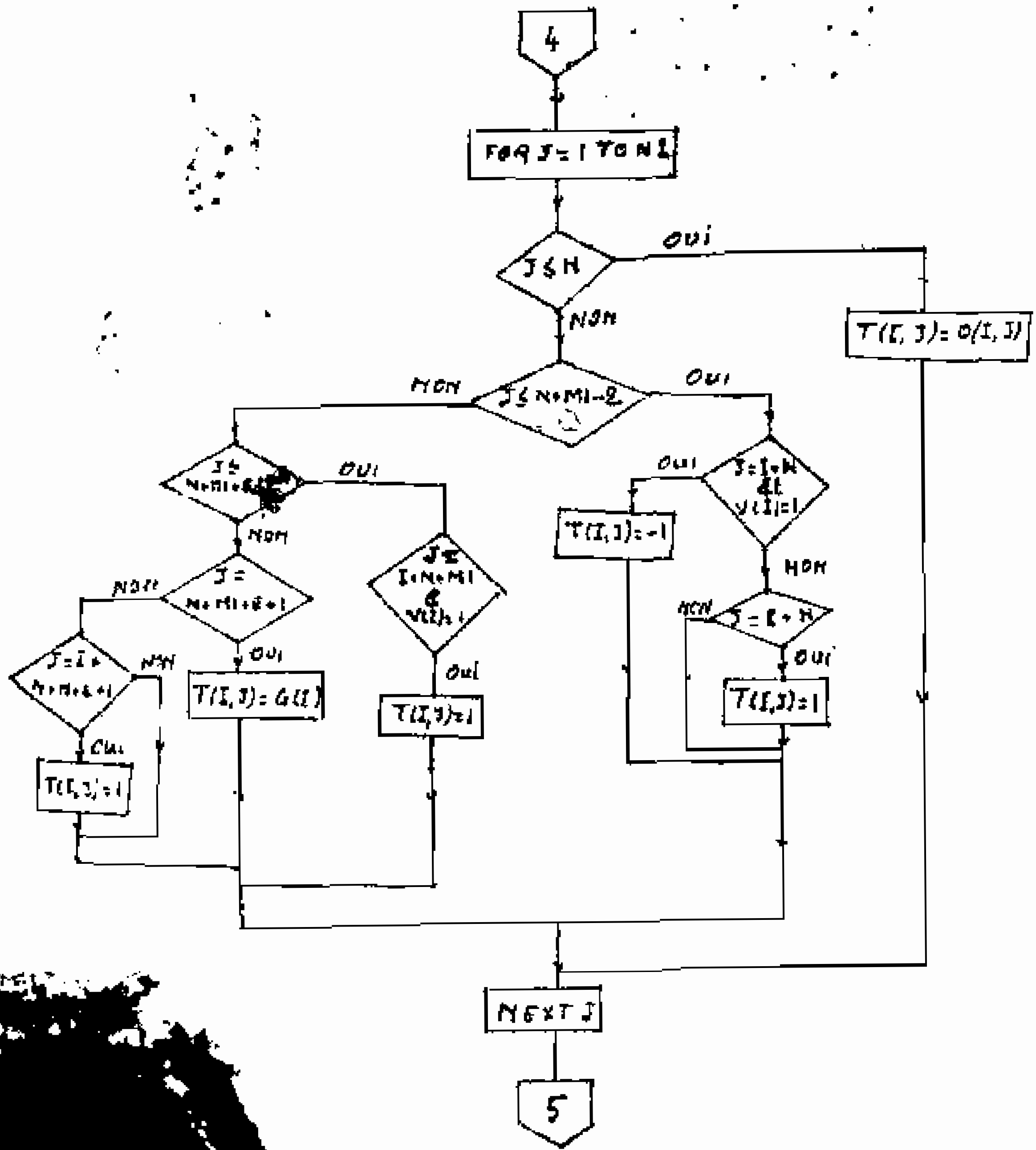
L'organigramme que nous allons maintenant présenter, ne sera en aucun cas le détail du programme.

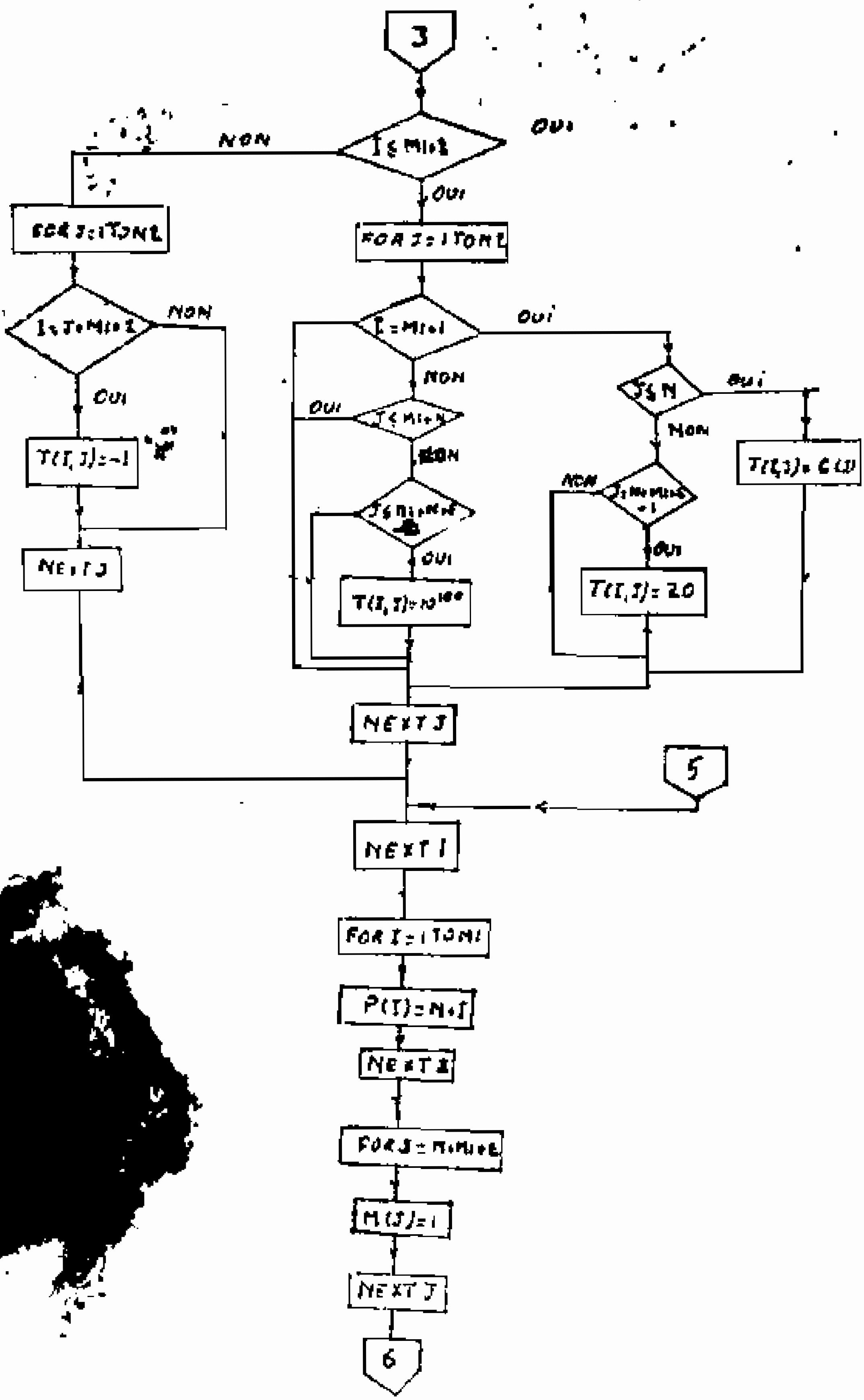
Il servira de guide à l'établissement du programme, les principaux calculs et opérations à faire ayant été décrits dans l'analyse du problème.

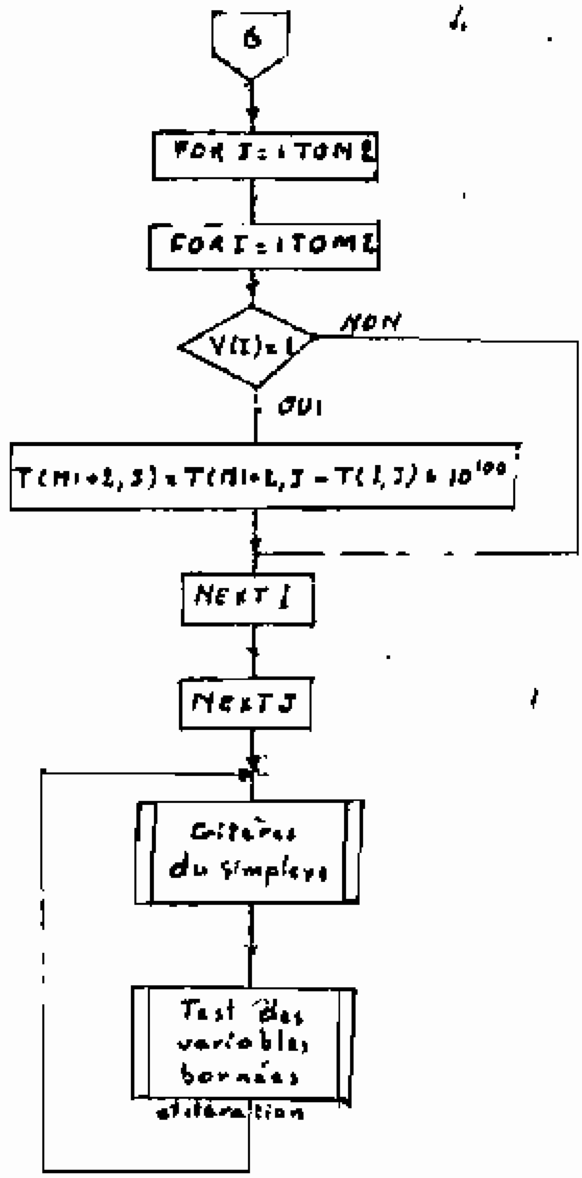

DÉBUT
nbre de contraintes, M
nbre de variables, N
Dimensions des tableaux
A(M,N), C(N), B(M,2), D(M,2)
G(2M), W(N), L(2, 3M+N+1),
S(3M+N+1), V(2M+N+2),
O(2M+2,N)
Entrée des données
C(J), A(I,J), B(I,J), L(I,J)
FOR I = 1 TO M
FOR J = 1 TO N
A = A + A(I,J) * L(1,J)
NEXT J
D(I,1) = B(I,1) - A
D(I,2) = B(I,2) - A
NEXT I
1


1
FOR J = 1 TO N
Z0 = Z0 + C(J) * L(1,J)
NEXT J
M1 = M
FOR I = 1 TO M
NON
OUI
M1 = M1 + 1
NON
OUI
FOR J = 1 TO N
FOR J = 1 to N
NEXT J
G(M1) = - B(I,1)
E = E + 1
2

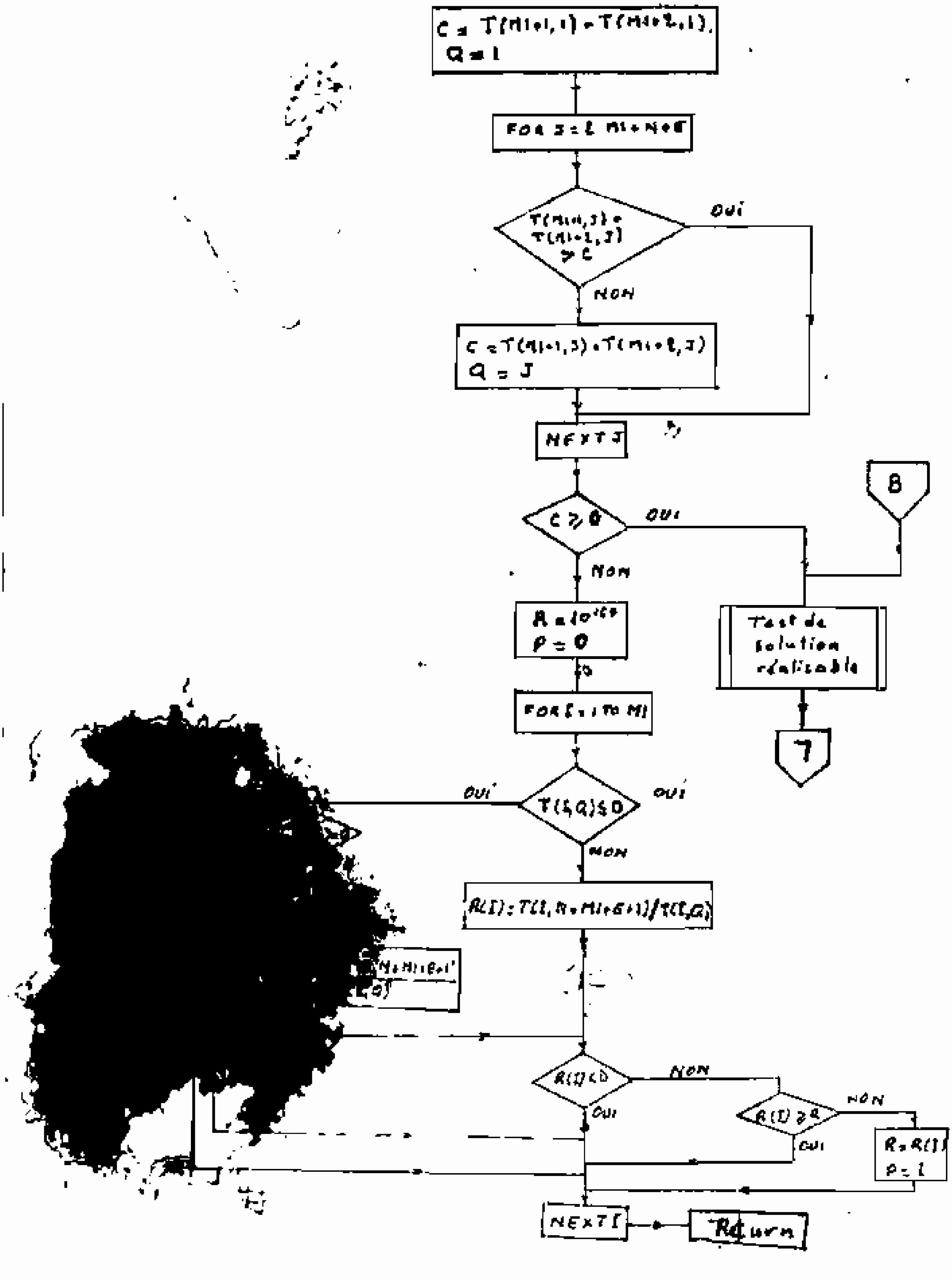

2
NON
B(I,I)<10
OUI
FOR J=1 TO N
D(I,J)=0
NEXT J
G(I)=0
B(I,I)<0
OUI
NON
FOR J=1 TO N
D(I,J)=A(I,J)
NEXT J
G(I)=B(I,2)
FOR J=1 TO N
D(I,J)=-A(I,J)
NEXT J
V(I)=1
G(I)=-B(I,2)
NEXT I
M1=M1+1
E=E+1
DIMENSIONNER
FOR I=1 TO M2
I≤M1
OUI
NON
3
4

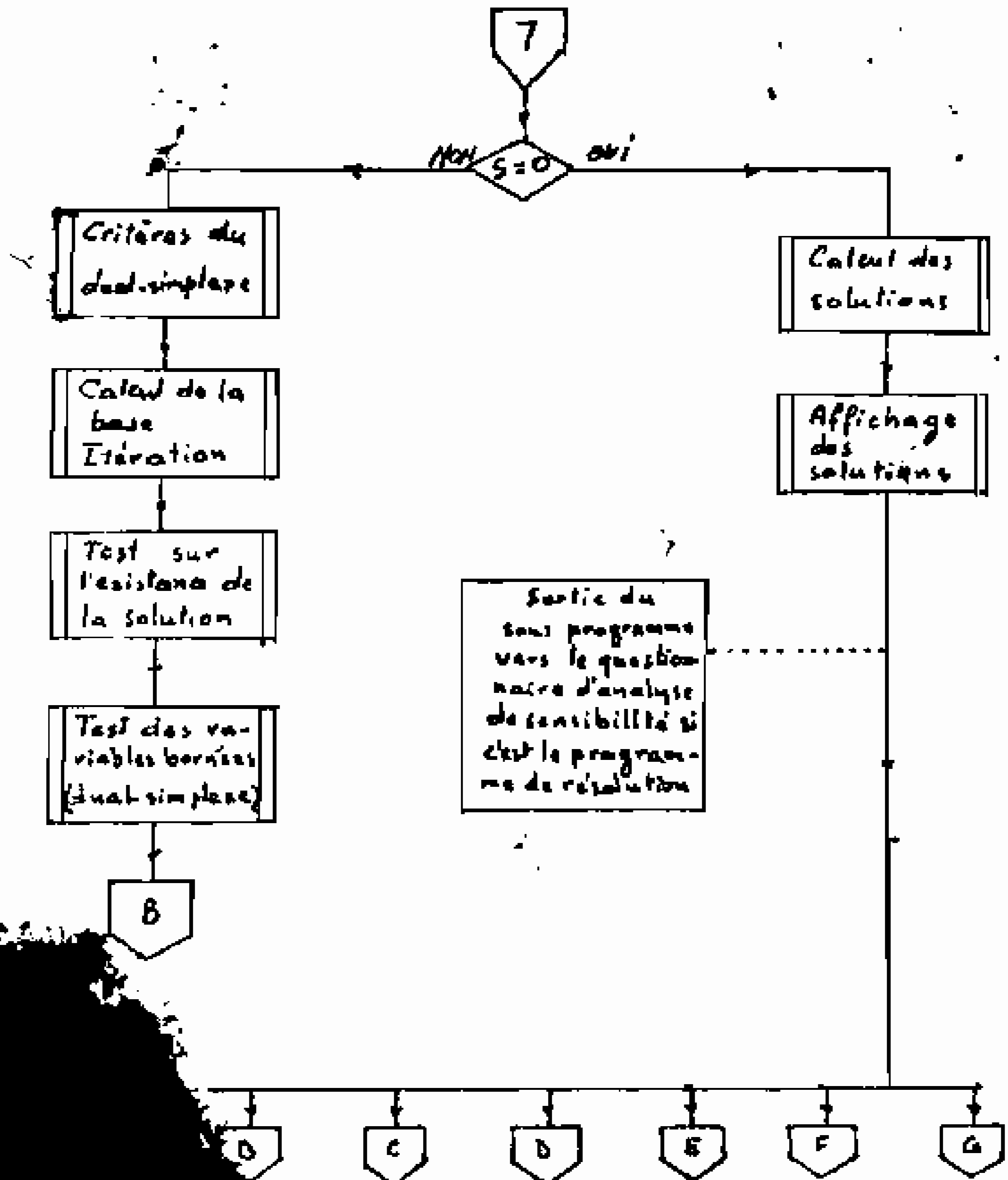

4
FOR J = 1 TO N1
J ≤ N
OUI
NON
T(I, J) = O(I, J)
J ≤ N+M1-2
NON
OUI
OUI
NON
J = N+M1+E+1
OUI
T(I, J) = G(I)
NON
OUI
T(I, J) = 1
OUI
T(I, J) = -1
NON
J = I + N
OUI
T(I, J) = 1
NEXT J
5


3
NON
OUI
FOR J=1 TO N2
FOR J=1 TO N2
NON
I=M1+1
OUI
OUI
NON
J≤N
OUI
OUI
T(I,J)=-1
J≤M1+N
NON
T(I,J)=C(J)
NON
OUI
NEXT J
T(I,J)=10^{100}
T(I,J)=20
NEXT J
5
NEXT I
FOR I=1 TO M1
P(I)=N+I
NEXT I
NEXT J
6

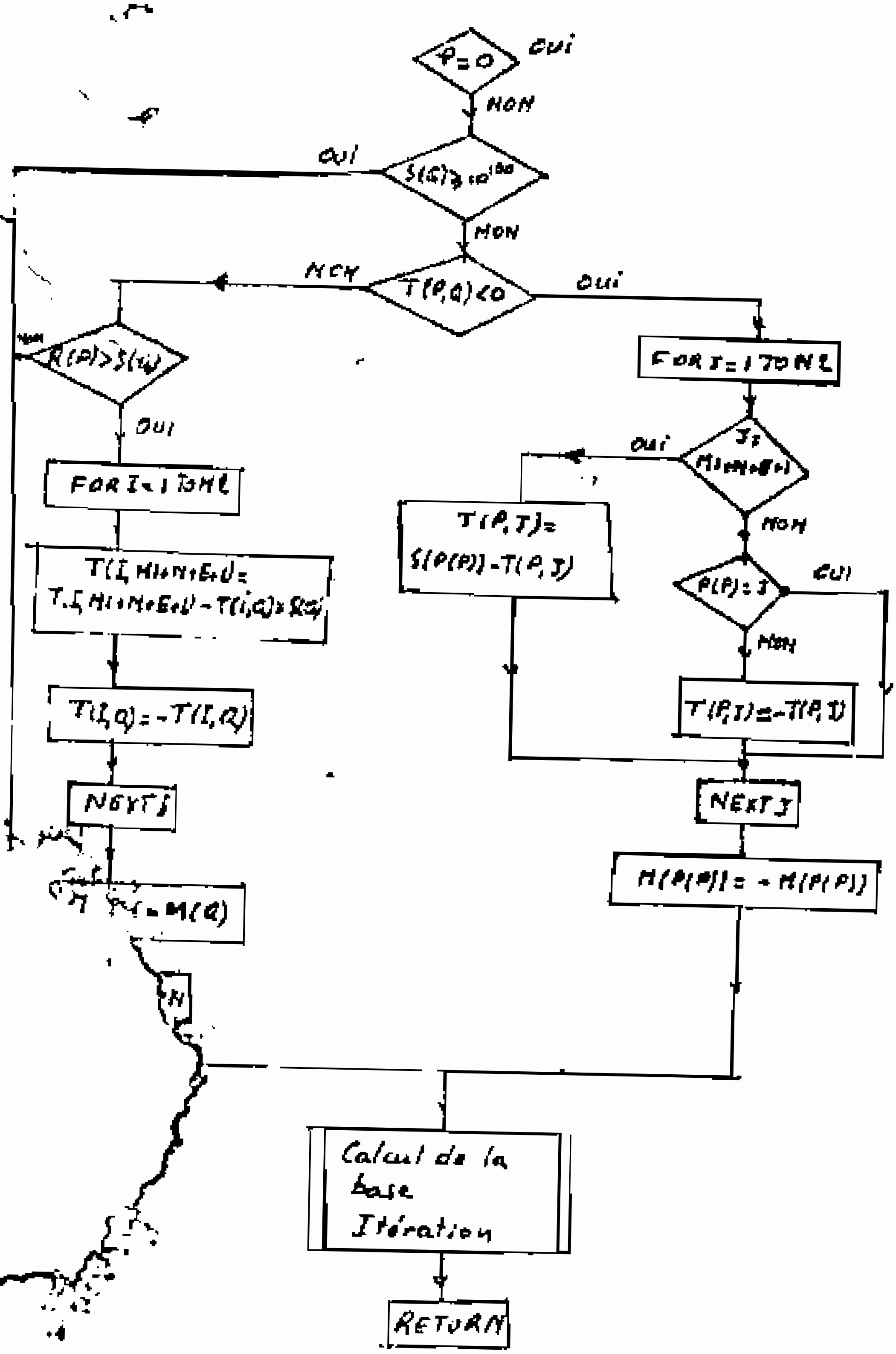

6
FOR J=1 TO M2
FOR I=1 TO M2
V(I)=1
NON
OUI
T(M1+2,J)=T(M1+2,J-T(I,J)*10^100
NEXT I
NEXT J
Critères du simplexe
Test des variables bornées
et itération

Sous-programme des critères (méthode simplexe)

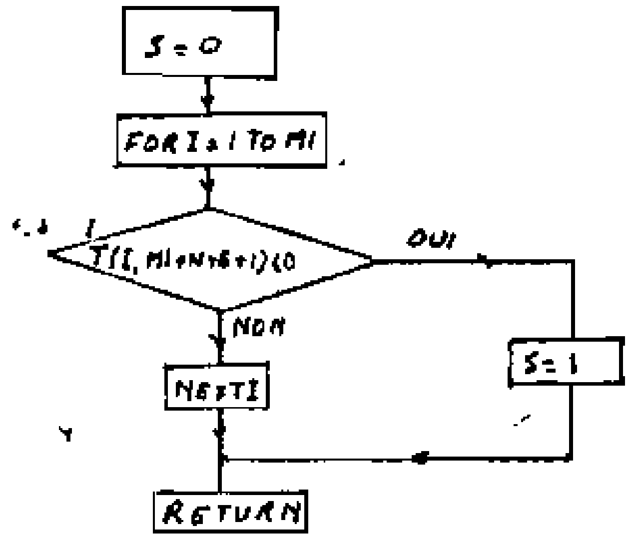



7
NON
S=0
OUI
Critères du dual-simplexe
Calcul de la base Itération
Test sur l'existence de la solution
Test des variables bornées (dual-simplexe)
B
Calcul des solutions
Affichage des solutions
Sortie du sous programme vers le questionnaire d'analyse de sensibilité si c'est le programme de résolution
C
D
E
F
G

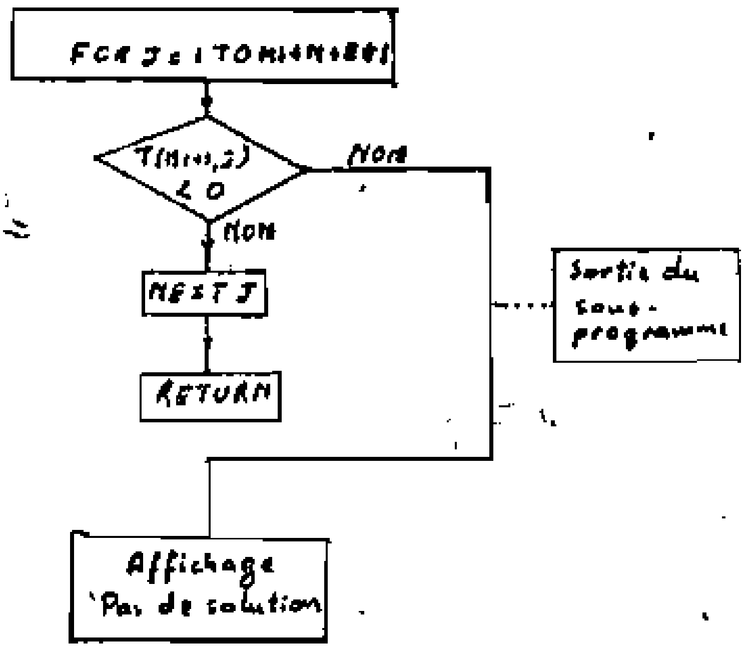
Sous-programme test des variables bornées (méthode simplexe)



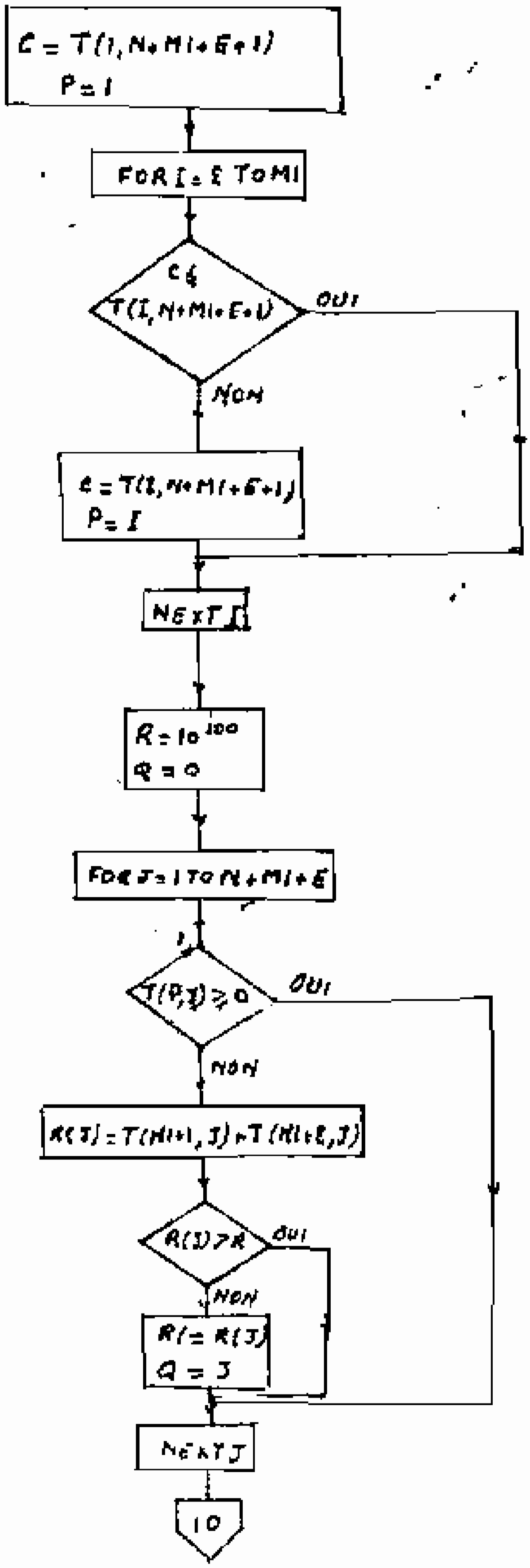
Sous-programme test de solution réalisable

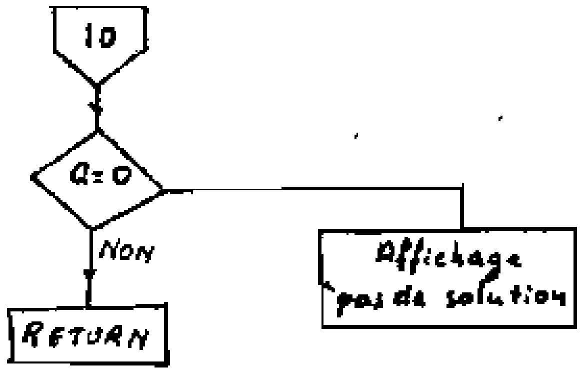


Sous programme test de l'existence de la solution

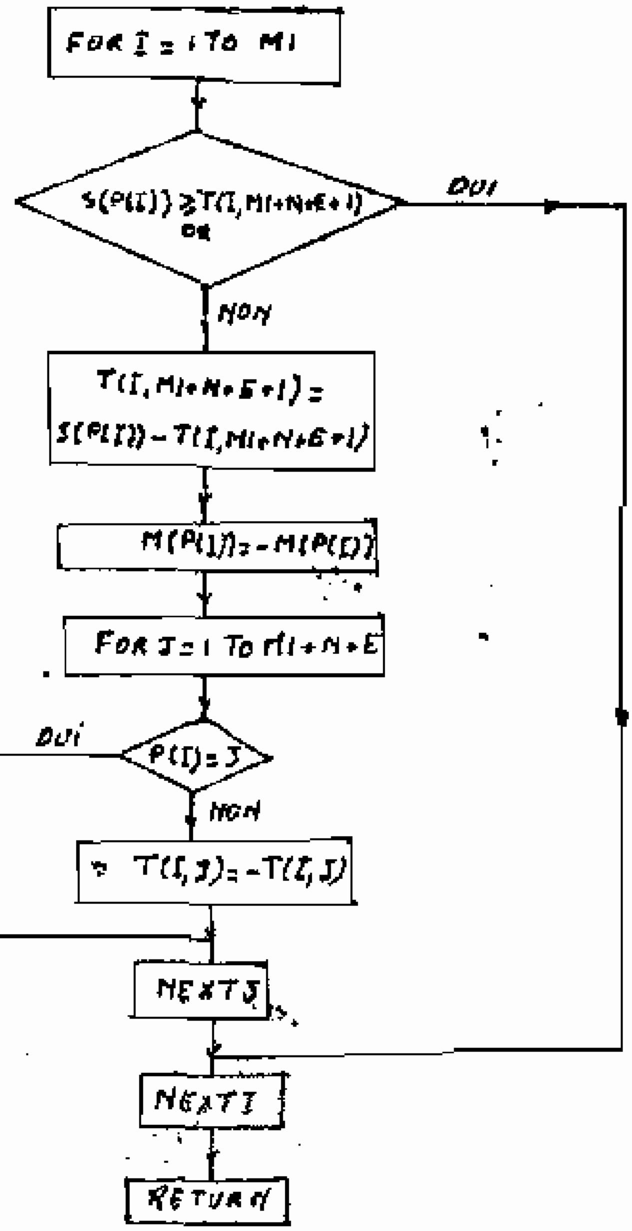


Sous-programme test des critères (dual-simplexe)

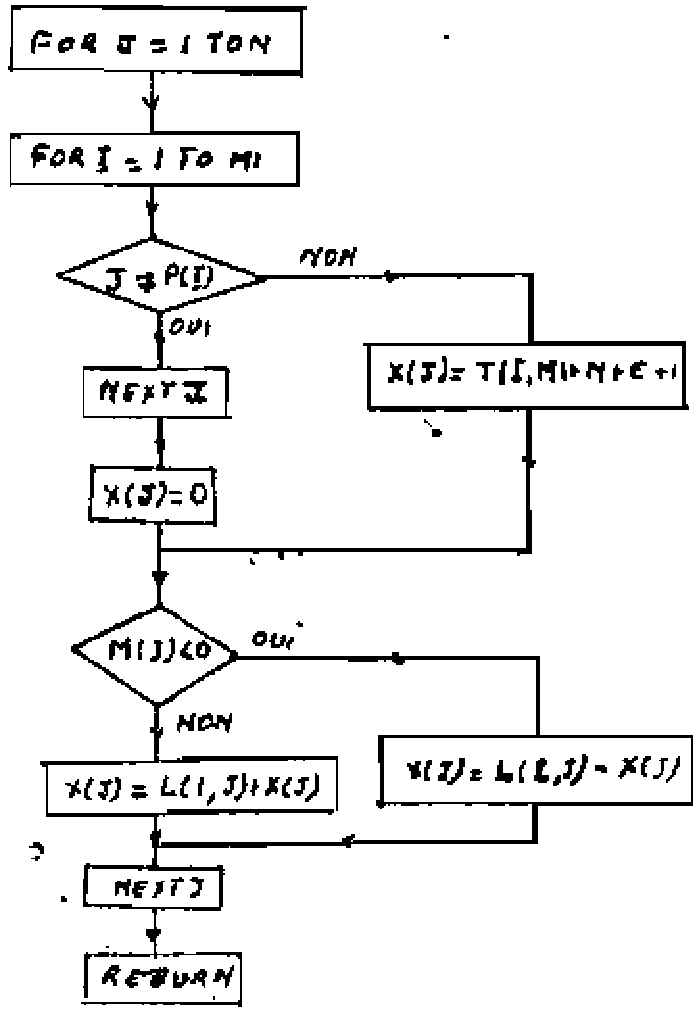



10
Q=0
NON
RETURN
Affichage
pas de solution

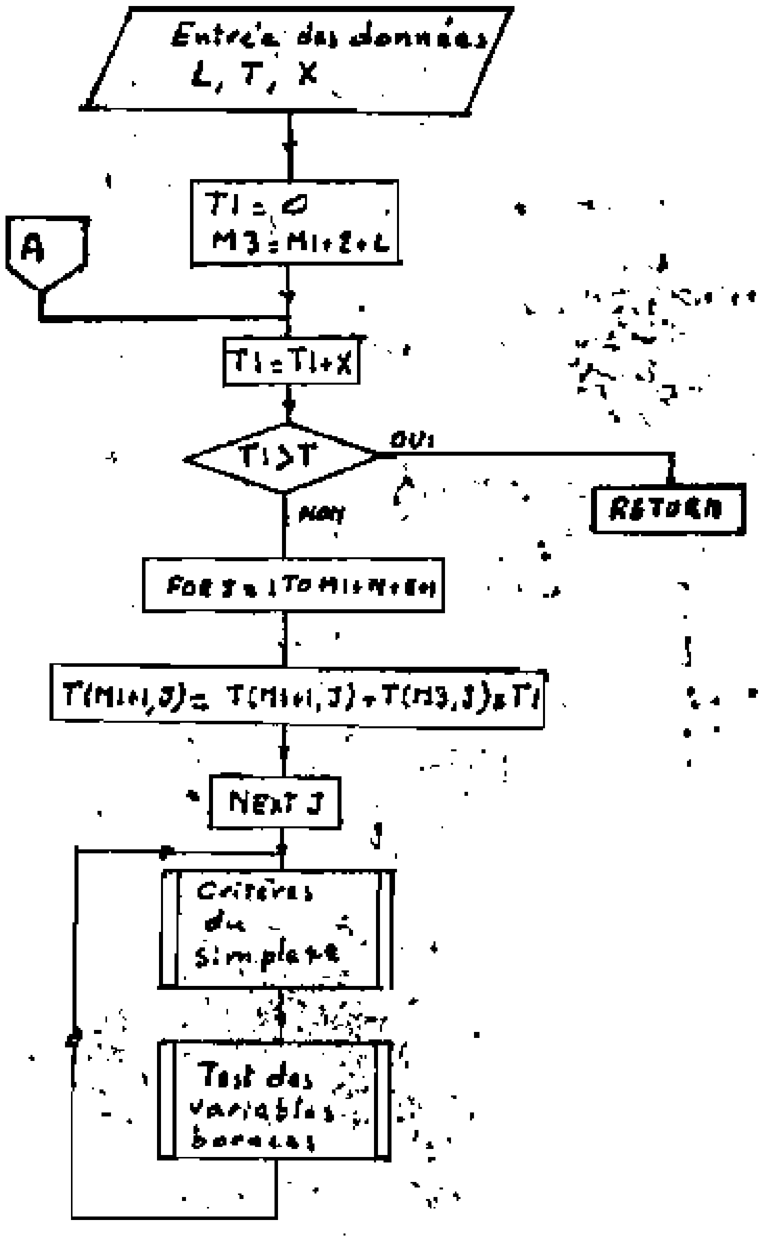
Sous-programme test des variables bornées (dual-simplexe)



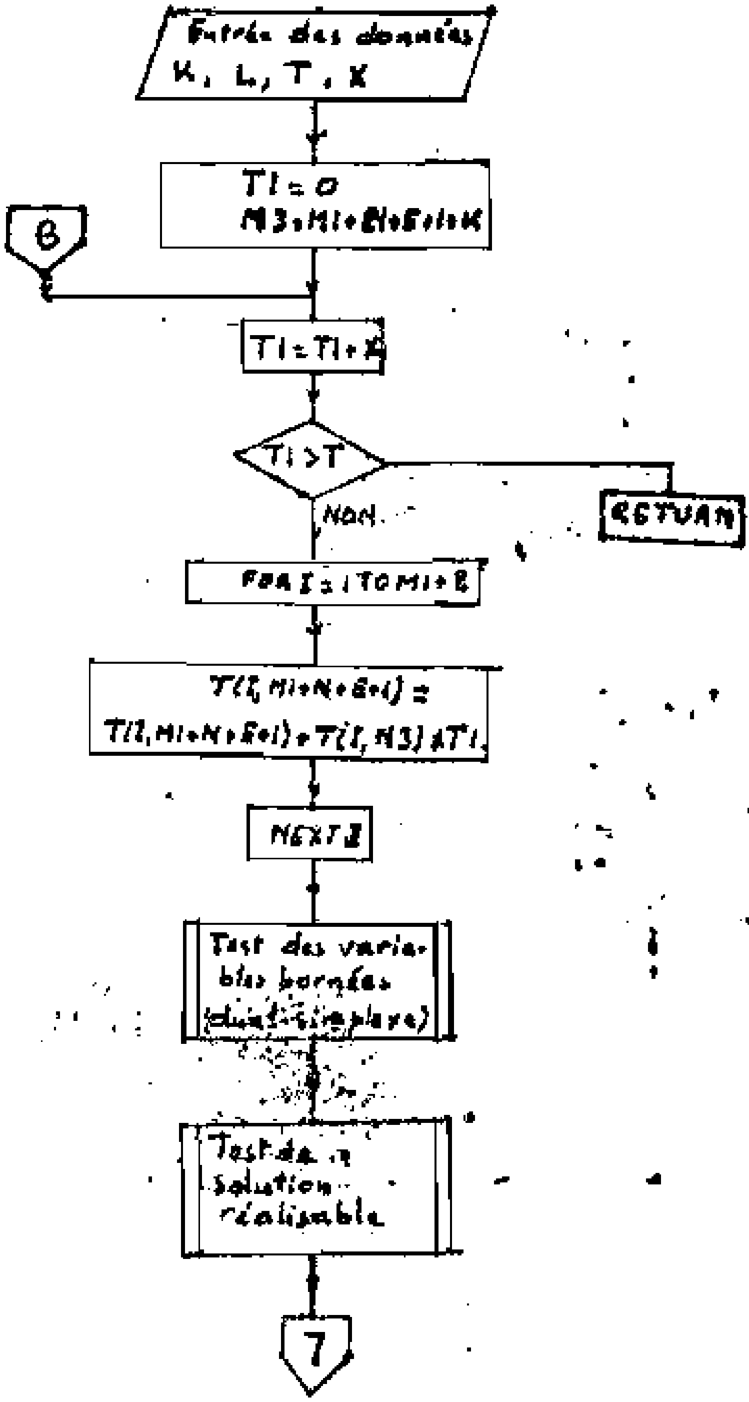
Sous-programme de calcul des solutions



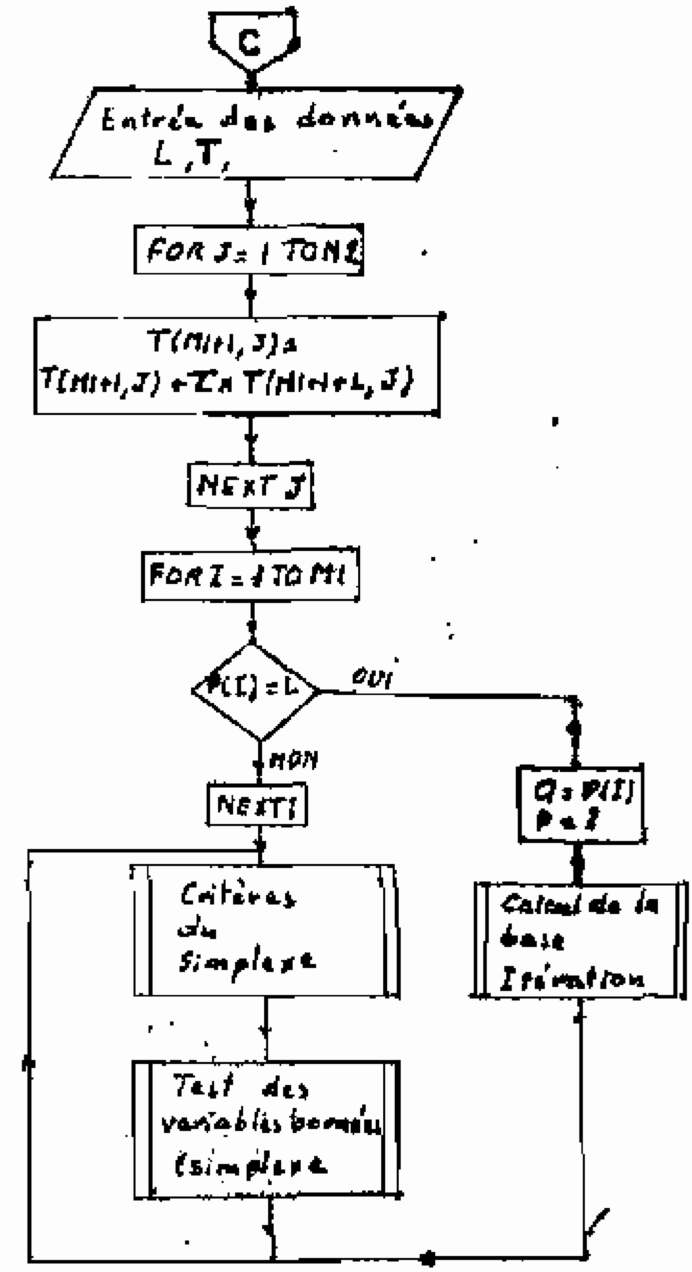
Paramétrage de la fonction économique



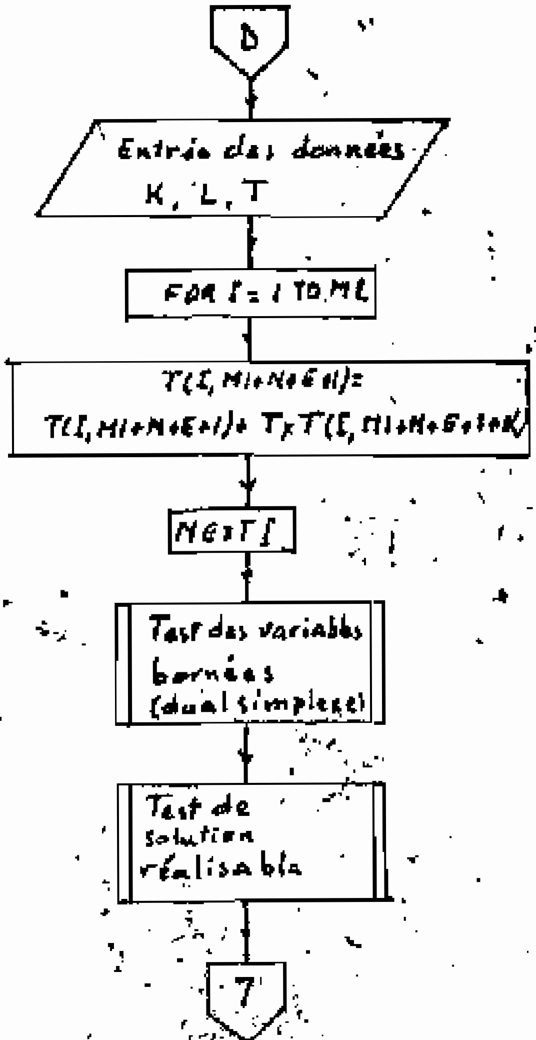
Paramétrage du second membre



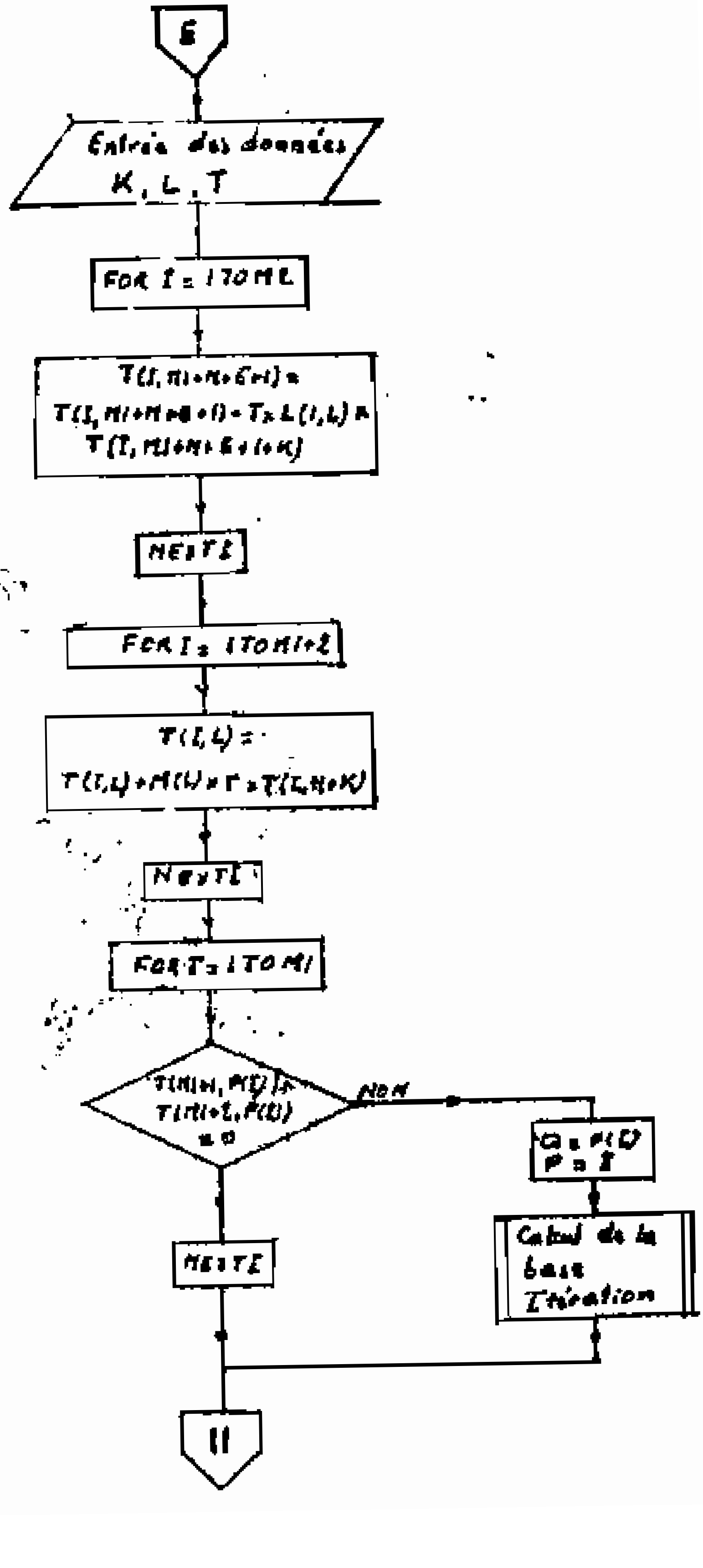
- Changement à la fonction économique

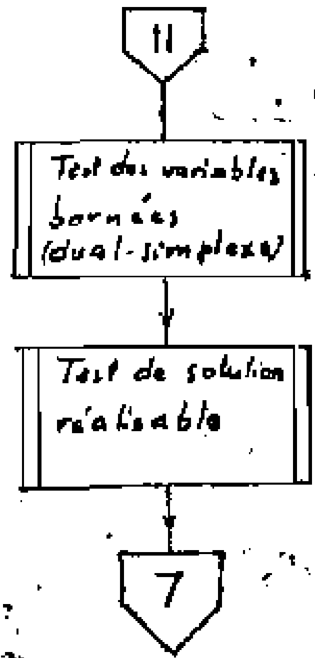


Changement au second membre

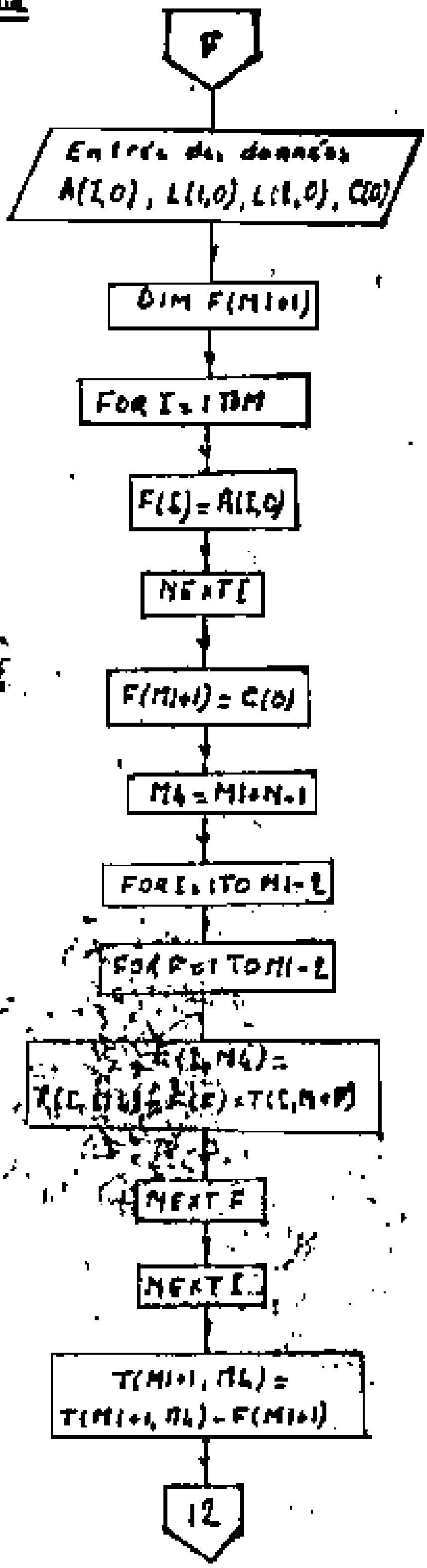


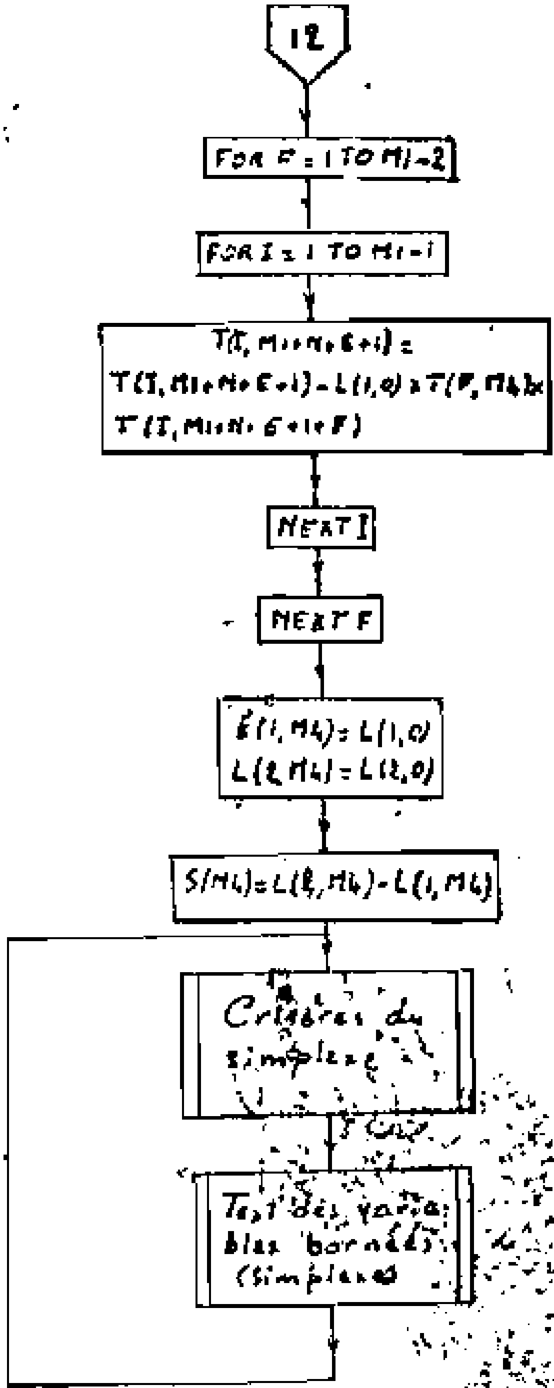
Changement aux coefficients des contraintes




11
Test des variables bornées (dual-simplexe)
Test de solution réalisable
7

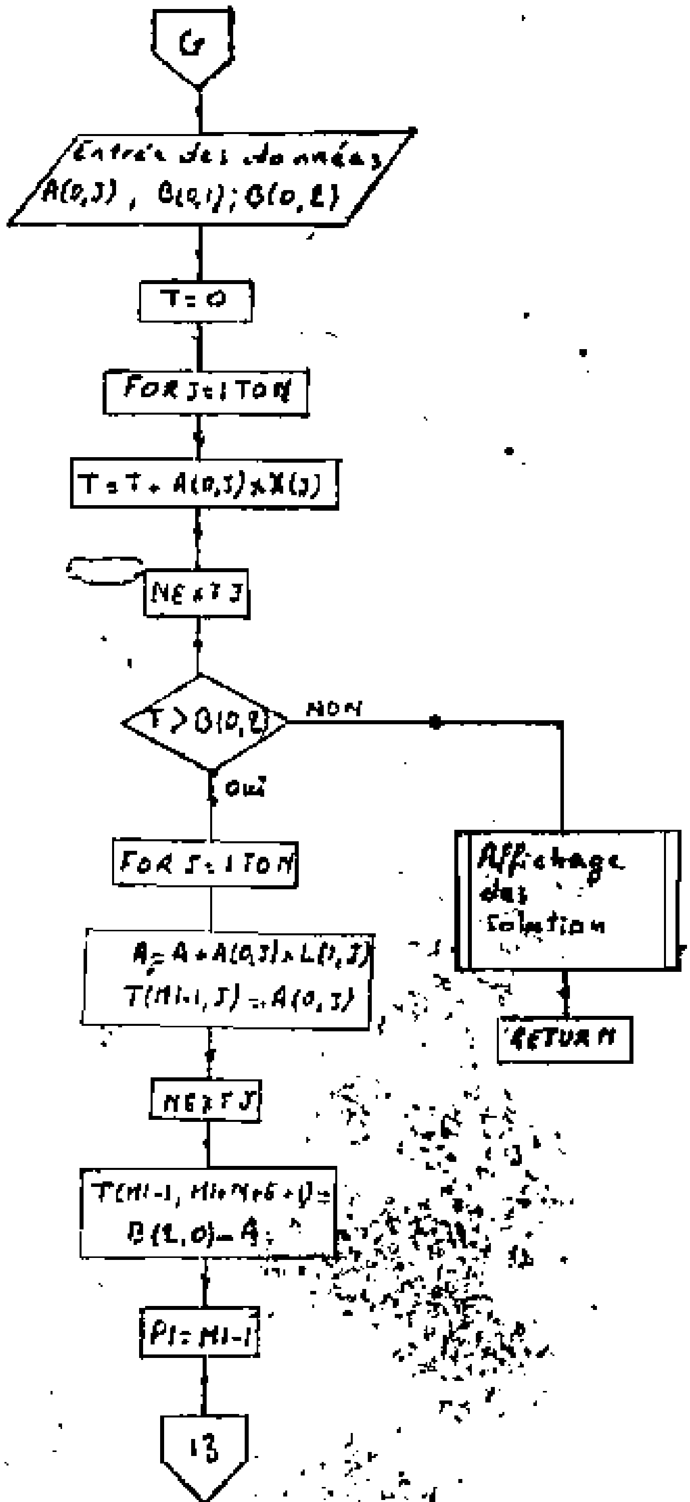
Variable additionnelle

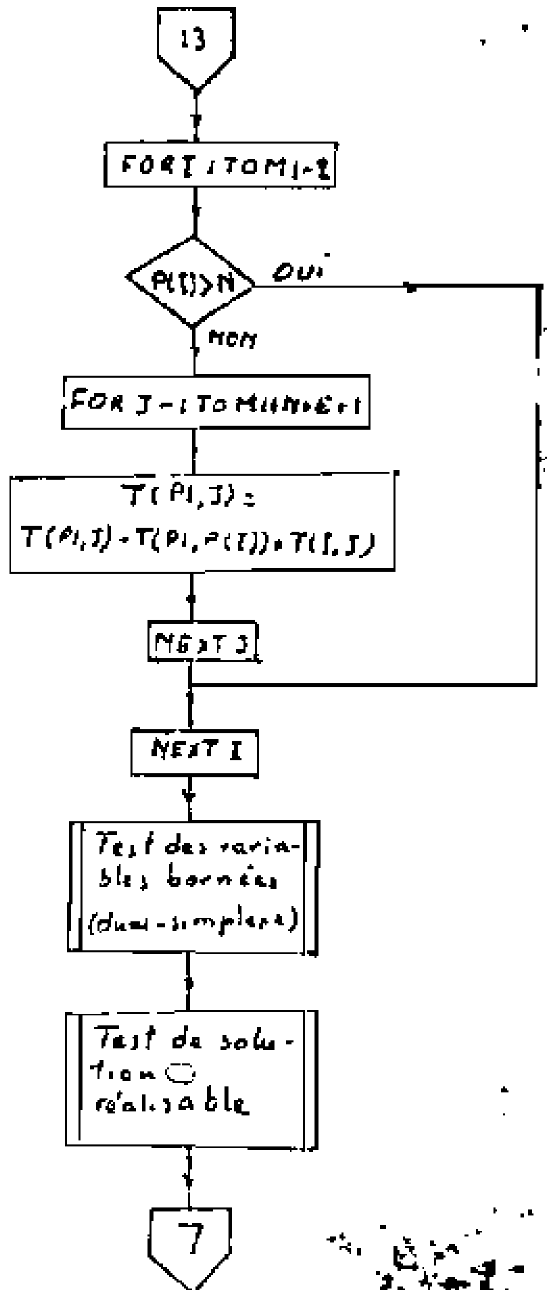




12
FOR F = 1 TO M1 - 2
FOR I = 1 TO M1 - 1
T(I, M1+N+E+1) =
T(I, M1+N+E+1) - L(1,0) x T(F, M4) x
T (I, M1+N+ E+1+F)
NEXT I
NEXT F
L(1, M4) = L(1,0)
L(2, M4) = L(2,0)
S(M4) = L(2, M4) - L(1, M4)
Critères du
simplexe
Test des varia-
bles bornées
(simplexe)

Addition d'une contrainte




13
FOR I 1 TO M1-2
P(I)>N
OUI
NON
FOR J-1 TO M1+N+E+1
T(PI,J)=
T(PI,J)-T(PI,P(I))*T(I,J)
NEXT J
NEXT I
Test des variables bornées (dual-simplexe)
Test de solution réalisable
7

```
00004 CLOSE (6) , OPEN (6) "LP"
00005 PRINT 'CS',"**************RESOLUTION DE PROGRAMME LINEAIRE *************
00005 **"
00010 PRINT 'CS',"VOUS VOULEZ", PRINT "1   LIRE LES DONNEES SUR DISQUE"; PRIN
00010 T "2   INTRODUIRE LES DONNEES AU CLAVIER", LET O=0, INPUT (0,ERR=00014)
00010  "INSCRIVEZ LE NUMERO DE LA PROCEDURE CHOISIE",O (2); GOTO 00020
00014 PRINT "  VOUS AVEZ FAIT UNE ERREUR", PRINT "    RECOMMNCEZ", GOTO 000
00014 10
00020 ON O GOTO 00014,05000,00050
00060 CLEAR ; LET H=0
00070 PRECISION 13
00080 PRINT 'CS'
00090 PRINT "INTRODUIRE LE NOMBRE DE CONTRAINTES ET LE NOMBRE DE VARIABLES"
00100 INPUT @(10,3),M, INPUT @(20,3),N
00102 GOSUB 00272
00104 LET A$=" ", ON C GOTO 00110,00080
00110 GOSUB 05300
00120 REM FONCTION OBJ
00130 PRINT 'CS',"COEFF DE LA FN  ECONOMIQUE"
00140 FOR J=1 TO N; INPUT @(5*J,3),C(J), LET W(J)=C(J), NEXT J
00142 GOSUB 00272
00144 LET A$=" ", ON C GOTO 00150,00130
00150 PRINT 'CS',"COEFF  DES CONTRAINTES"
00160 FOR I=1 TO M
00170 FOR J=1 TO N, INPUT @(6*J,3*I),A(I,J), PRINT "A(",I,J,") =",A(I,J),"
00170 ", NEXT J
00180 NEXT I
00182 GOSUB 00272
00184 LET A$=" ", ON C GOTO 00190,00150
00190 PRINT 'CS',"COEFF  DU SECOND MEMBRE"
00200 FOR I=1 TO M
00210 FOR J=1 TO 2, INPUT @(10*J,3*I),B(I,J), PRINT "B(",I,J,") =",B(I,J),"
00210 ", NEXT J
00220 NEXT I
00222 GOSUB 00272
00224 LET A$=" ", ON C GOTO 00230,00190
00230 PRINT 'CS',"BORNES DES VARIABLES"
00240 FOR J=1 TO N
00250 FOR I=1 TO 2; INPUT @(10*I,3*J),L(I,J), PRINT "L(",I,J,") =",L(I,J),"
00250 ", NEXT I
00255 IF J>N THEN LET S(J)= 1E+120
00260 LET S(J)=L(2,J)-L(1,J)
00270 NEXT J, GOSUB 00272
00271 LET A$=" ", ON C GOTO 00280,00230
00272 PRINT " AVEZ VOUS ENTRE LES BONNES VALEURS ?", PRINT "   REPONDEZ PAR O
00272 UI OU NON", INPUT R$
00274 IF R$<>"OUI" AND R$<>"NON" THEN GOTO 00272 ELSE GOTO 00276
```

```
00276 IF R$="OUI" THEN GOTO 00279
00278 LET G=1; RETURN
00279 LET G=0; RETURN
00283 PRINT "                    PROGRAMME  EN  EXECUTION"
00289 REM ******TRAITEMENT DES MATRICES********
00290 FOR I=1 TO M; LET A=0
00300 FOR J=1 TO N; LET A=A+A(I,J)*L(1,J); NEXT J
00310 LET D(I,1)=D(I,1)-A
00320 LET A=0
00330 FOR J=1 TO N; LET A=A+A(I,J)*L(1,J); NEXT J
00340 LET D(I,2)=D(I,2)-A
00350 NEXT I
00360 FOR J=1 TO N; LET Z0=Z0+C(J)*L(1,J); NEXT J
00370 LET M1=M
00380 FOR I=1 TO M
00390 IF D(I,1)>-.1E+16 THEN GOTO 00440
00400 IF D(I,2)<.1E+15 THEN GOTO 00490
00410 FOR J=1 TO N; LET O(I,J)=0; NEXT J
00420 LET G(I)=0
00430 GOTO 00670
00440 LET M1=M1+1
00450 IF D(I,1)<0 THEN GOTO 00550
00460 FOR J=1 TO N; LET O(M1,J)=A(I,J); NEXT J
00470 LET V(M1)=1; LET G(M1)=D(I,1); LET E=E+1; LET D(I,1)=M1
00480 GOTO 00400
00490 IF D(I,2)<0 THEN GOTO 00610
00500 FOR J=1 TO N
00510 LET O(I,J)=A(I,J)
00520 NEXT J
00530 LET G(I)=D(I,2)
00540 GOTO 00670
00550 FOR J=1 TO N
00560 LET O(M1,J)=-A(I,J)
00570 NEXT J
00580 LET G(M1)=-D(I,1)
00590 LET D(I,1)=-M1
00600 GOTO 00400
00610 FOR J=1 TO N
00620 LET O(I,J)=-A(I,J)
00630 NEXT J
00640 LET V(I)=1
00650 LET G(I)=-D(I,2)
00660 GOTO 00670
00670 NEXT I
00675 LET M1=M1+2; LET E=E+2
00680 LET M2=M1+N+2
00690 LET N2=2*M1+N+E+1
00700 DIM T(M2,N2),U(M2,N2),R(M1),Q(M1+N)
00710 FOR I=1 TO M2
00720 IF I<=M1 THEN GOTO 00780
00730 IF I<=M1+2 THEN GOTO 00930
00740 FOR J=1 TO N2
00750 IF I=J+M1+2 THEN GOSUB 03440 ELSE GOSUB 03400
00760 NEXT J
00770 GOTO 01030
00780 FOR J=1 TO N2
00790 IF J<=N THEN GOTO 00900
00800 IF J<=N+M1-2 THEN GOTO 00880
00810 IF J<=N+M1+E-2 THEN GOTO 00860
00820 IF J=N+M1+E+1 THEN GOSUB 03480 ELSE GOTO 00840
00830 GOTO 00910
00840 IF J=I+N+M1+E+1 THEN GOSUB 03420 ELSE GOSUB 03400
00850 GOTO 00910
00860 IF V(I)=1 AND J=I+N+M1 THEN GOSUB 03420 ELSE GOSUB 03400
00870 GOTO 00910
00880 IF J=I+N AND V(I)=1 THEN GOSUB 03440 ELSE GOTO 00885
00882 GOTO 00910
00885 IF J=I+N THEN GOSUB 03420 ELSE GOSUB 03400
00887 GOTO 00910
```

```
00890 GOTO 00910
00900 LET T(I,J)=O(I,J)
00910 NEXT J
00920 GOTO 01030
00930 FOR J=1 TO N2
00940 IF I=M1+1 THEN GOTO 00990
00950 IF J<=M1+N THEN GOSUB 03400 ELSE GOTO 00970
00960 GOTO 01020
00970 IF J<=M1+N+E+2 THEN GOSUB 03520 ELSE GOSUB 03400
00980 GOTO 01020
00990 IF J<=N THEN GOSUB 03500 ELSE GOTO 01010
01000 GOTO 01020
01010 IF J=N+M1+E+1 THEN GOSUB 03540 ELSE GOSUB 03400
01020 NEXT J
01030 NEXT I
01040 REM ----------------------------------------------------------------------
01080 GOTO 01260
01090 REM AFFICHAGE DU TABLEAU
01120 FOR J=1 TO N+M1+E; PRINT A(10*J-1),"!   X",J,"   ",; NEXT J; PRINT "!    C
01120:1    !"; GOSUB 01200
01130 FOR I=1 TO M2
01140 IF I<=M1 THEN PRINT "   X",P(I),
01145 IF I=M1+1 THEN PRINT " Z0",
01150 FOR J=1 TO N2; PRINT A(10*J-1),"!",A(10*J),T(I,J)I"-&&&0.&&&",
01160 NEXT J
01170 PRINT "!"
01180 IF I=M1 OR I=M1+2 THEN GOSUB 01200 ELSE GOSUB 01220
01190 GOTO 01240
01200 FOR J=1 TO N2; PRINT A(10*J-1),"!---------",; NEXT J
01210 PRINT "!"; RETURN
01220 FOR J=1 TO N2; PRINT A(10*J-1),"!                ",; NEXT J
01230 PRINT "!"; RETURN
01240 NEXT I
01250 PRINT . PRINT ; PRINT ; RETURN
01260 REM CALCUL   ITERATIONS
01270 DIM P(M1),M(N2),N(N2),X(N2),Y(N2),K(N),R(M1),E(M2)
01280 FOR I=1 TO M1
01290 LET P(I)=N+I
01300 NEXT I
01310 FOR J=1 TO N+M1+E
01320 LET M(J)=1
01330 NEXT J
01340 REM ----ELIMINATION DES M ----
01350 FOR J=1 TO N2
01360 FOR I=1 TO M2
01370 IF V(I)<>1 THEN GOTO 01390
01380 LET T(M1+2,J)=T(M1+2,J)-T(I,J)* 1E+15
01390 NEXT I
01400 NEXT J
01420 GOSUB 01450
01430 GOSUB 01740
01440 GOTO 01420
01441 GOTO 20060
01442 GOSUB 02720
01443 GOSUB 02200
01444 IF S=0 THEN GOTO 01447 ELSE GOSUB 02810
01445 GOSUB 02010: GOSUB 02202
01446 GOTO 01442
01447 GOSUB 02290
01448 ON H,GOTO 01567,21070,20780,20955,21700,22400,20060,23120
01450 REM ***********CRITERES 1 & 2 DU SIMPLEXE*****************
01460 REM ----1ier CRITERE------ D'ENTREE -----
01470 LET C=T(M1+1,1)+T(M1+2,1)
01480 LET O=1
01490 FOR J=2 TO M1+N+E
01500 IF T(M1+1,J)+T(M1+2,J)-C>.0000000001 THEN GOTO 01530
01510 LET C=T(M1+1,J)+T(M1+2,J)
01520 LET O=J
01530 NEXT J
```

```
01540 IF C>=0 OR ABS(C)<=.000000001 THEN GOSUB 02200 ELSE GOTO 01570
01550 IF S=0 THEN GOTO 01560 ELSE EXITTO 01442
01560 GOSUB 02390
01562 IF H<=5 THEN GOSUB 02430 ELSE EXITTO 01441
01566 ON H GOTO 01567,03010,03020,03021,03022;03023,03024,03025
01567 GOSUB 02520; GOSUB 07000
01568 GOSUB 02480; GOTO 03000
01570 REM ---------------2ieme CRITERE---------------------
01580 LET R=.1E+138; LET P=0
01590 FOR I=1 TO M1
01600 IF T(I,O)<=.000000001 THEN GOTO 01670
01605 IF T(I,M1+N+E+1)<0 THEN GOTO 01720
01610 LET R(I)=T(I,M1+N+E+1)/T(I,O)
01620 IF R(I)<.000000001 THEN GOTO 01720
01630 IF R(I)>=R THEN GOTO 01720
01640 LET R=R(I)
01650 LET P=I
01660 GOTO 01720
01670 IF ABS(T(I,O))< 0000000001 THEN GOTO 01720
01690 IF S(P(I))>=.1E+15 THEN GOTO 01720
01700 LET R(I)=(S(P(I))-T(I,M1+N+E+1))/(-T(I,O))
01710 GOTO 01620
01720 NEXT I
01730 RETURN
01740 REM &&&&&&&&&&&TEST SUR LES VARIABLES BORNEES &&&&&&&&&BORITEST&&&&&&
01740;&&&&&&&&
01750 IF P=0 THEN GOSUB 30000 ELSE GOTO 01780
01755 ON H GOTO 03000,03010,03020,03021,03022,03023,03024,03025
01780 IF S(O)>=.1E+15 THEN GOSUB 02010 ELSE GOTO 01800
01790 GOTO 02000
01800 IF T(P,O)<0 THEN GOTO 01830
01810 IF R(P)>S(O) THEN GOTO 01930 ELSE GOSUB 02010
01820 GOTO 02000
01830 FOR J=1 TO N2
01840 IF J=M1+N+E+1 THEN LET T(P,J)=S(P(P))-T(P,J) ELSE GOTO 01870
01860 GOTO 01870
01870 IF F(P)=J THEN GOTO 01890
01880 LET T(P,J)=-T(P,J)
01890 NEXT J
01900 LET M(P(P))=-M(P(P))
01910 GOSUB 02010
01920 GOTO 02000
01930 FOR I=1 TO M2
01950 LET T(I,M1+N+E+1)=T(I,M1+N+E+1)-T(I,O)*S(O)
01970 LET T(I,O)=-T(I,O)
01980 NEXT I
01990 LET M(O)=-M(O)
02000 RETURN
02010 REM ------CREATION DE LA BASE-----
02020 FOR I=1 TO M2
02030 IF I=P THEN GOTO 02060
02040 LET E(I)=(-T(I,O))/T(P,O)
02050 GOTO 02070
02060 LET E(I)=1/T(P,O)
02070 NEXT I
02080 REM ---------ITERATIONS---------------
02090 FOR I=1 TO M2
02100 IF I=P THEN GOTO 02140
02110 FOR J=1 TO N2
02120 LET T(I,J)=T(I,J)+T(P,J)*E(I)
02130 NEXT J
02140 NEXT I
02150 FOR J=1 TO N2
02160 LET T(P,J)=T(P,J)*E(P)
02170 NEXT J
02180 LET F(P)=O
02190 RETURN
02200 REM ----------SOLUTION REALISABLE?--------------
02210 LET S=0
```

```
02220 FOR I=1 TO M1
02230 IF SGN(T(I,M1+N+E+1))=-1 THEN GOTO 02250
02240 GOTO 02270
02250 LET S=1
02260 EXITTO 02280
02270 NEXT I
02280 RETURN
02282 FOR J=1 TO M1+N+E+1
02283 IF T(M1+1,J)<-.0000000001 THEN EXITTO 02286
02284 NEXT J, GOTO 02288
02286 LET O=0; GOTO 02960
02288 RETURN
02290 REM -----------EDITION DES SOLUTIONS------------
02300 FOR J=1 TO N
02310 FOR I=1 TO M1
02320 IF J<>P(I) THEN GOTO 02340
02330 EXITTO 02370
02340 NEXT I
02350 LET X(J)=0
02360 GOTO 02380
02370 LET X(J)=T(I,N+M1+E+1)
02380 IF M(J)<0 THEN GOTO 02410
02390 LET X(J)=L(1,J)+X(J)
02400 GOTO 02420
02410 LET X(J)=L(2,J)-X(J)
02420 NEXT J
02425 RETURN
02430 REM ***********AFFICHAGE DES SOLUTIONS***********
02445 PRINT "SOLUTIONS"; PRECISION 5
02450 FOR J=1 TO N
02460 PRINT "X",J,"=      ",X(J)
02470 NEXT J
02475 RETURN
02480 FOR I=1 TO M2
02490 FOR J=1 TO N2; LET U(I,J)=T(I,J); NEXT J
02500 NEXT I
02505 FOR I=1 TO M1; LET Q(I)=P(I); NEXT I
02510 FOR J=1 TO N2; LET N(J)=M(J); LET Y(J)=X(J); NEXT J; RETURN
02520 LET Z=0
02530 FOR J=1 TO N; LET Z=Z+X(J)*C(J), NEXT J
02540 IF H=4 THEN LET Z=Z+X(M4)*C(0)
02550 PRINT "Z   =   ",Z; PRECISION 13; RETURN
02720 REM ---------TEST DES VARIABLES BORNEES ------
02730 FOR I=1 TO M1
02740 IF S(P(I))-T(I,M1+N+E+1)>0 OR ABS(S(P(I))-T(I,M1+N+E+1))<.0000000001 TH
02740:EN GOTO 02800
02750 LET T(I,N+M1+E+1)=S(P(I))-T(I,N+M1+E+1); LET M(P(I))=-M(P(I))
02760 FOR J=1 TO N+M1+E
02770 IF P(I)=J THEN GOTO 02790
02780 LET T(I,J)=-T(I,J)
02790 NEXT J
02800 NEXT I
02805 RETURN
02810 REM -------1er CRITERE -- - - DE SORTIE ------------
02820 LET C=T(1,N+M1+E+1); LET P=1
02830 FOR I=2 TO M1
02840 IF T(I,N+M1+E+1)>=C THEN GOTO 02860
02850 LET C=T(I,N+M1+E+1); LET P=I
02860 NEXT I
02870 REM ---------2ieme - - - D'ENTREE --------------
02880 DIM R(N+M1+E)
02890 LET R=.1E+16; LET O=0
02900 FOR J=1 TO N+M1+E
02910 IF T(P,J)>=0 OR ABS(T(P,J))<.0000000001 THEN GOTO 02950
02920 LET R(J)=(T(M1+1,J)+T(M1+2,J))/(-T(P,J))
02930 IF R(J)>R THEN GOTO 02950
02940 LET R=R(J); LET O=J
02950 NEXT J
02960 IF O=0 THEN GOSUB 30000 ELSE GOTO 03030
```

```
02970 ON H GOTO 03000,03010,03020,03021,03022,03023,03024,03025
03000 EXITTO 09999
03010 EXITTO 21090
03020 EXITTO 20790
03021 EXITTO 20960
03022 EXITTO 21710
03023 EXITTO 22400
03024 EXITTO 20205
03025 EXITTO 23160
03030 RETURN
03050 REM --SOR POUR ANALYSE DE SENSIBILITE -----
03060 IF O(1,1)<0 THEN GOTO 03080
03070 LET K=D(K,1); GOTO 03090
03080 LET K=-D(K,1), LET T=-T
03090 RETURN
03100 IF D(K,2)>=0 THEN GOTO 03120
03110 LET T=-T
03120 RETURN
03400 LET T(I,J)=0
03410 RETURN
03420 LET T(I,J)=1
03430 RETURN
03440 LET T(I,J)=-1
03450 RETURN
03460 LET T(I,J)=O(1,J)
03470 RETURN
03480 LET T(I,J)=G(I)
03490 RETURN
03500 LET T(I,J)=-C(J)
03510 RETURN
03520 LET T(I,J)=,1E+15
03530 RETURN
03540 LET T(I,J)=20
03550 RETURN
05000 REM ***********UTILISATION DE DISQUE****LECTURE DES DONNEES**********
05010 PRINT "NBRE DE CONTRAINTES   NBRE DE VARIABLES"; CLOSE (1) ; OPEN (1) "
05010:363DATA"
05020 READ (1) M,N: PRINT &(10),M,&(35),N: GOSUB 05300
05030 PRINT "COEFF. DE LA FONCTION OBJECTIF"
05040 FOR J=1 TO N; READ (1) C(J); PRINT &(6*J),C(J),; NEXT J; PRINT
05060 PRINT "COEFF. DES CONTRAINTES"
05070 FOR I=1 TO M
05080 FOR J=1 TO N; READ (1) A(I,J); PRINT &(6*J),A(I,J),; NEXT J; PRINT
05100 NEXT I
05110 PRINT "COEFF. DU SECOND MEMOMBRE"
05120 FOR I=1 TO M
05130 FOR J=1 TO 2; READ (1) B(I,J); PRINT &(6*J),B(I,J),; NEXT J; PRINT
05140 NEXT I
05160 PRINT "BORNES DES VARIABLES"
05170 FOR J=1 TO N
05180 FOR I=1 TO 2; READ (1) L(I,J); PRINT &(6*I),L(I,J),; NEXT I; PRINT
05190 NEXT J
05200 GOTO 00280
05300 DIM A(M,N),C(N),B(M,2),L(2,3*M+N+1),S(3*M+N+1),D(M,2),G(2*M),V(2*M+N+2)
05300:,O(2*M+2,N),W(N); RETURN
05400 REM *****AFFICHAGE DU PROBLEME***********
05600 REM ******IMPRESSION DES RESULTATS***********
05605 PRECISION 5
05610 CLOSE (6) ; OPEN (6) "LP"
05620 PRINT (6) "****S O L U T I O N****"
05630 FOR J=1 TO N; PRINT (6) "X",J,"  =  ",X(J); NEXT J
05640 IF M=4 THEN PRINT (6) "X",N+1,"  =  ",X(M4)
05650 PRINT (6) "Z     =  ",Z; PRECISION 13; RETURN
05700 REM **IMPRESSION DES RESULTATS :PROG. PARAMETRIQUE***
05710 OPEN (1) "RESULTAT"
05720 CLOSE (6) ; OPEN (6) "LP"
05730 PRINT (6) "   T    ",
05740 FOR J=1 TO N; PRINT (6) &(8*J),"I    X",J,, NEXT J
05750 PRINT (6) &((N+1)*8),"I    Z    I"
```

```
05760 FOR J=1 TO N; PRINT (6) 8*J,"!--------",; NEXT J; PRINT (6) A((N+1)*8),"/
05760:!---------!"
05765 FOR R=1 TO INT(T/X)
05770 READ (1) T1, PRINT (6) T1:"-&0 &&",
05780 FOR J=1 TO N; READ (1) X(J); PRINT (6) A(8*J),"!",X(J):"-&&0.&&&",; NEX
05780 T J; READ (1) Z; PRINT (6) A((N+1)*8),Z:"-&&&&0.&&&"; NEXT R; CLOSE (1)
05780: , ERASE "RESULTAT"
05790 RETURN
06000 PRINT (6) "**PARAMETRAGE DE LA FONCTION ECONOMIQUE**"; PRINT (6) "Param
06000:etrage du coefficient C",L, PRINT (6) "Valeur maxi du parametre",T, RET
06000.URN
06050 PRINT (6) "**PARAMETRAGE DU SECOND MEMBRE****"; PRINT (6) "Parametrage
06050 ducoefficient B",X,L; PRINT (6) "Valeur maxi du parametre",T; RETURN
06100 PRINT (6) "CHANGEMENT DE C",L; PRINT (6) "Valeur du changement :",T; RE
06100.TURN
06200 PRINT (6) "CHANGEMENT DE A",K,L; PRINT (6) "Valeur du chagement :",T; R
06200:ETURN
06300 PRINT (6) "CHANGEMENT DE B",K,L, PRINT (6) "Valeur du changement :",T;
06300.RETURN
06400 PRINT (6) "ADDITION D'UNE VARIABLE"
06410 FOR I=1 TO N; PRINT (6) "Contrainte ",I,"  Coeff. de Xn+1 :",A(I,0); NE
06410:XT 1; RETURN
06500 PRINT (6) "ADDITION D'UNE CONTRAINTE"
06510 FOR J=1 TO N; PRINT (6) "Coeff. de X",J," :",A(0,J); NEXT J; PRINT (6)
06510:"B 1=",B(0,1),"      B2=",B(0,2); RETURN
06800 PRINT (6) "SOLUTION DU PROGRAMME LINEAIRE"; PRINT (6) "Max :",
06810 FOR J=1 TO N-1; PRINT (6) C(J),"*X",J," + ",; NEXT J, PRINT (6) C(N),"*
06810:X",N; PRINT (6) : PRINT (6) "sc:",
06820 FOR I=1 TO M; PRINT (6) B(I,1),"<= ",
06830 FOR J=1 TO N-1; PRINT (6) A(I,J),"*X",J,"  +  ",, NEXT J; PRINT (6) A(I
06830:,N),"*X",N,"< = ",B(I,2), NEXT 1
06840 FOR J=1 TO N; PRINT (6) L(1,J),"<= X",J,"<=",L(2,J); NEXT J
06850 GOSUB 05600
07000 INPUT "VOULEZ VOUS IMPRIMER LES RESULTATS",A$
07010 IF A$<>"OUI" AND A$<>"NON" THEN GOTO 07000
07020 IF A$="NON" THEN GOTO 07030
07022 ON H GOSUB 06800,06100,06200,06300,06400,06500
07024 GOSUB 05600
07030 RETURN
09999 REM *****ANALYSE DE SENSIBILITE ET PROGRAMMATION PARAMETRIQUE**********
09999:**********************************INTERACT******************************
09999:*******
10000 PRINT "POUR EXECUTER L'UN DES PROGRAMMES SUIVANTS,INSCRIVEZ SON NUMERO
10000 AU CLAVIER ET APPUYER SUR (RETURN)"; PRINT "1   ANALYSE DE SENSIBILITE"
10000 ; PRINT "2   PROGRAMMATION PARAMETRIQUE"; PRINT "3   EXECUTION D'UN AUT
10000 RE PROGRAMME LINEAIRE"; LET O=0
10010 INPUT (0,ERR=10000) O:(3)
10020 ON O GOTO 10000,10050,10100,00020
10050 PRINT "**********************ANALYSE  DE  SENSIBILITE******************"
10055 PRINT 'CS',"INSCRIVEZ AU CLAVIER LE NUMERO DE L'ANALYSE QUE VOUS VOULEZ
10055: FAIRE"; PRINT "1   CHANGEMENT A LA FONCTION OBJECTIF"; PRINT "2   CHAN
10055 GEMENT AUX COEFF. DES CONTRAINTES", PRINT "3   CHANGMENT AU SECOND MEMB
10055:RE"; PRINT "4   ADDITION D'UNE VARIABLE"; PRINT "5   ADDITION D'UNE CON
10055:TRAINTE"; LET O=0
10070 INPUT (0,ERR=10055) O:(5)
10080 ON O GOTO 10055,10400,10450,10500,10550,10600
10100 PRINT "******************PROGRAMMATION  PARAMETRIQUE******************"
10105 PRINT 'CS',"INSCRIVEZ AU CLAVIER LE NUMERO DU PARAMETRAGE QUE VOUS VOUL
10105.EZ EFFECTUER"; PRINT "1   PARAMETRAGE DE LA FONCTION OBJECTIF"; PRINT "
10105 2   PARAMETRAGE DU SECOND MEMBRE"; PRINT "3   ANALYSE DE SENSIBILITE";
10105:LET O=0
10110 INPUT (0,ERR=10105) O.(3)
10120 ON O GOTO 10105,10700,10750,10050
10220 GOSUB 15000
10230 ON O GOTO 10220,10055,10240,10260
10240 GOSUB 15500
10250 GOTO 10050
10260 GOSUB 15500
10270 GOTO 10000
```

```
10400 PRINT "CHANGEMENT A LA FONCTION OBJECTIF"
10410 GOSUB 21000
10420 GOSUB 15500, GOTO 10105
10450 PRINT "CHANGEMENT AUX COEFF  DES CONTRAINTES"
10460 GOSUB 20599
10470 GOSUB 15500, GOTO 10105
10500 PRINT "CHANGEMENT AU SECOND MEMBRE"
10510 GOSUB 20900
10520 GOSUB 15500, GOTO 10105
10550 PRINT "ADDITION D'UNE VARIABLE"
10560 GOSUB 21500
10570 GOSUB 15500, GOTO 10105
10600 PRINT "ADDITION D'UNE CONTRAINTE"
10610 GOSUB 22000
10620 GOSUB 15500, GOTO 10105
10700 PRINT "PARAMETRAGE DE LA FONCTION OBJECTIF"
10710 GOSUB 20000
10720 GOSUB 15500
10730 GOTO 10105
10750 PRINT "PARAMETRAGE DU SECOND MEMBRE"
10760 GOSUB 23000
10770 GOSUB 15500
10780 GOTO 10105
15000 PRINT "VOUS VOULEZ CONTINUER L' ANALYSE DE SENSIBILITE", PRINT "1   AVE
15000 C LES DERNIERS RESULTATS OBTENUS", PRINT "2   AVEC LES RESULTATS DU PRO
15000 GRAMME INITIAL", PRINT "INSCRIVEZ AU CLAVIER LE NUMERO QUI VOUS INTERES
15000 SE OU APPUYEZ SUR (RETURN) POUR CHANGER", LET O=0
15010 INPUT (0,ERR=15000) O (2), RETURN
15500 FOR I=1 TO M2
15510 FOR J=1 TO N2; LET T(I,J)=U(I,J), NEXT J
15520 NEXT I
15525 FOR I=1 TO M1, LET P(I)=Q(I), NEXT I
15530 FOR J=1 TO N2, LET M(J)=N(J), LET X(J)=Y(J), NEXT J, RETURN
20000 REM ******PARAMETRAGE FONCTION OBJECTIF*********
20005 PRINT 'CS',"PARAMETRAGE DE LA FONCTION OBJECTIF", LET H=6
20010 INPUT "INTRODUIRE L'INDICE DU COUT A PARAMETRER",L, INPUT "INTRODUIRE L
20010 A VALEUR MAXI DE LA VARIATION",T, INPUT "INTRODUIRE LE TAUX DE VARIATIO
20010 N",X, LET T1=0, PRINT "PARAMETRAGE DU COEFF  C",L, PRINT "VALEUR MAXI D
20010 EMANDEE T=",T, PRINT "TAUX DE VARIATION(INCREMENT) X=",X; GOSUB 20012
20011 GOTO 20020
20012 PRINT "    T1",
20014 FOR J=1 TO N, PRINT A(8*J),"!","    X",J,, NEXT J
20016 PRINT A((N+1)*8),"!      Z      ", PRINT "--------",
20017 FOR J=1 TO N, PRINT A(8*J),"!--------",, NEXT J, PRINT "!---------!", I
20017 NDEXED "RESULTAT",INT(T/X)*(N+2),17,1,0, CLOSE (1) , OPEN (1) "RESULTAT
20017 "
20018 GOSUB 20200
20019 RETURN
20020 LET M3=M1+2+L
20030 IF T1>T THEN GOSUB 20210 ELSE GOTO 20035
20033 RETURN
20035 LET W(L)=C(L)+T1
20040 FOR J=1 TO M1+N+D+1, LET T(M1+1,J)=T(M1+1,J)+T1*T(M3,J), NEXT J
20050 GOTO 01420
20060 REM ***********S O L U T I O N*****************
20070 GOSUB 20200
20075 LET W(L)=C(L)
20080 GOTO 20030
20200 WRITE (1) T1, PRINT T1 "#0 ##",
20201 FOR J=1 TO N, WRITE (1) X(J), PRINT A(8*J),"!",X(J) "-##0 ###",, NEXT J
20201 , LET Z=0
20202 FOR J=1 TO N, LET Z=Z+W(J)*X(J), NEXT J, WRITE (1) Z, PRINT A((N+1)*8),
20202 Z "!-####0 ##!", IF T1=T THEN WAIT 15
20203 LET T1=T1+X; GOSUB 15500
20205 RETURN
20210 INPUT "VOULEZ VOUS L'IMPRESSION DES RESULTATS",A$
20220 IF A$<>"OUI" AND A$<>"NON" THEN GOTO 20210
20230 IF A$="NON" THEN GOTO 20300
20240 RETURN
```

```
20250 ON H-4 GOSUB 06000,06050
20260 GOSUB 05700
20300 RETURN
20599 REM ***********CHANGEMENT AUX COEFF. DES CONTRAINTES*********
20600 PRINT 'CS',"CHANGEMENT AUX COEFF. DES CONTRAINTES:"; INPUT (0,ERR=20605
20600:) "INTRODUISEZ L'INDICE DE LA LIGNE (CONTRAINTE)",K:(M); INPUT (0,ERR=2
20600:0605) "INTRODUISEZZ L'INDICE DE LA COLONNE (VARIABLE) QUE VOUS VOULEZ A
20600:FFECTER",L:(N); INPUT "INTRODUISEZ LA GRANDEUR DE LA VARIATION",T; LET
20600:H=2; GOTO 20606
20605 PRINT "    VOUS AVEZ FAIT UNE ERREUR "; PRINT "      RECOMMENCEZ"; GOTO
20605:20600
20606 IF K<=0 OR L<=0 THEN GOTO 20605
20610 IF D(K,1)<0 THEN GOTO 20660 ELSE LET K1=D(K,1)
20630 FOR I=1 TO M2; LET T(I,M1+N+E+1)=T(I,M1+N+E+1)-T*L(1,L)*T(I,M1+N+E+1+K1
20630:); NEXT I
20635 FOR I=1 TO M1+2; LET T(I,L)=T(I,L)+M(L)*T(I,N+K1)*T; NEXT I
20640 GOTO 20700
20660 IF D(K,1)<-.1E+16 THEN GOTO 20700 ELSE LET K1=-D(K,1)
20680 FOR I=1 TO M2; LET T(I,M1+N+E+1)=T(I,M1+N+E+1)+T*L(1,L)*T(I,M1+N+E+1+K1
20680:)
20685 FOR I=1 TO M1+2; LET T(I,L)=T(I,L)+M(L)*T(I,N+K1)*(-T); NEXT I
20700 IF D(K,2)>,1E+16 THEN GOTO 20730
20710 FOR I=1 TO M2; LET T(I,M1+N+E+1)=T(I,M1+N+E+1)-T*L(1,L)*T(I,M1+N+E+1+K)
20710:; NEXT I
20720 FOR I=1 TO M1+2; LET T(I,L)=T(I,L)+M(L)*T(I,N+K)*T; NEXT I
20730 FOR I=1 TO M1
20740 IF ABS(T(M1+1,P(I))+T(M1+2,P(I)))<.00000000001 THEN GOTO 20770
20750 LET O=P(I); LET P=I
20760 GOSUB 02010; REM ***BASE--ITERATION****
20765 EXITTO 20775
20770 NEXT I
20775 GOTO 01442
20780 GOSUB 02430; GOSUB 02520
20785 GOSUB 07000
20790 RETURN
20900 REM ******CHANGEMENT AU 2nd MEMBRE**********
20910 PRINT 'CS',"CHANGEMENT AU SECOND MEMBRE:"; LET H=3
20920 INPUT (0,ERR=20922) "INTRODUISEZ L'INDICE DE LA LIGNE A CHANGER",K:(M);
20920: GOTO 20924
20922 PRINT "   VOUS AVEZ FAIT UNE ERREUR"; PRINT "      RECOMMENCEZ"; GOTO 2
20922:0920
20924 IF K<=0 THEN GOTO 20922
20925 INPUT (0,ERR=20925) "INTRODUISEZ 1OU 2 POUR CHANGER LE Bi INF. OU LE Bi
20925: SUP (RESPECT.):",L:(2)
20926 ON L GOTO 20925,20927,20929
20927 GOSUB 03050
20928 GOTO 20930
20929 GOSUB 03100
20930 INPUT "INTRODUISEZ LA GRANDEUR DE LA VARIATION:",T
20940 FOR I=1 TO M2; LET T(I,M1+N+E+1)=T(I,M1+N+E+1)+T(I,M1+N+E+1+K)*T; NEXT
20940:I
20950 GOTO 01442
20955 GOSUB 02430; GOSUB 02520
20956 GOSUB 07000
20960 RETURN
21000 REM ***CCHANGEMENT A LA FONCTION OBJECTIF ****
21010 PRINT 'CS',"CHANGEMENT A LA FONCTION OBJECTIF"; LET H=1
21020 INPUT (0,ERR=21024) "INTRODUISEZ L'INDICE DU COEFF. A CHANGER",L:(N); I
21020:NPUT "INTRODUISEZ LA GRANDEUR DE LA VARIATION",T; GOTO 21026
21024 PRINT "   VOUS AVEZ FAIT UNE ERREUR"; PRINT "      RECOMMENCEZ"; GOTO 2
21024:1020
21026 IF L<=0 THEN GOTO 21024
21028 FOR J=1 TO M2; LET T(M1+1,J)=T(M1+1,J)+T*T(M1+1+L,J); NEXT J
21030 FOR I=1 TO M1
21040 IF P(I)=L THEN EXITTO 21085
21050 NEXT I
21060 GOTO 01420
21070 GOSUB 02430
21080 LET C(L)=C(L)+T; GOSUB 02520; GOSUB 07000; RETURN
```

```
21085 LET Q=P(I); LET P=I: GOTO 21070
21090 GOSUB 02010: GOTO 21060
21500 REM ***********ADDITION D'UNE VARIABLE****************
21510 PRINT 'CS',"ADDITION D'UNE VARIABLE"; LET H=4: DIM F(M1+1)
21520 PRINT "INTRODUISEZ LES COEFF. DE LA VARIABLE"; LET M4=M1+N-1
21530 FOR I=1 TO M; INPUT A(6*I),A(I,0); NEXT I
21540 GOSUB 00272
21550 LET A$=" "; ON C GOTO 21552,21510
21552 PRINT "INTRODUISEZ LES BORNES DE LA VARIABLE"; INPUT A(8),L(1,0),A(16),
21552:L(2,0); GOSUB 00272; LET A$=" "; ON C GOTO 21560,21552
21560 INPUT "INTRODUISEZ LE COEFF. DE LA FONCTION OBJECTIF",C(0); GOSUB 00272
21570 LET A$="-"; ON C GOTO 21580,21560
21580 FOR I=1 TO M
21583 IF D(I,1)<0 AND D(I,1)>-M1 THEN LET F(-D(I,1))=-A(I,0)
21586 IF D(I,2)<0 THEN LET F(I)=-A(I,0) ELSE LET F(I)=A(I,0)
21590 IF D(I,1)>=0 THEN LET F(D(I,1))=A(I,0)
21600 NEXT I
21610 LET F(M1+1)=C(0)
21620 FOR I=1 TO M1-2
21630 FOR F=1 TO M1-2; LET T(I,M4)=T(I,M4)+F(F)*T(I,N+F); NEXT F; NEXT I
21650 LET T(M1+1,M4)=T(M1+1,M4)-F(M1+1)
21660 FOR F=1 TO M1-2
21670 FOR I=1 TO M1-1; LET T(I,M1+N+E+1)=T(I,M1+N+E+1)-L(1,0)*T(F,M4)*T(I,M1+
21670:N+E+1+F); NEXT I
21680 NEXT F
21690 LET L(1,M4)=L(1,0); LET L(2,M4)=L(2,0); LET S(M4)=L(2,M4)-L(1,M4); LET
21690:M(M4)=1; GOTO 01420
21700 GOSUB 02430
21710 FOR I=1 TO M1
21720 IF M4=P(I) THEN EXITTO 21740
21730 NEXT I; LET X(M4)=0; GOTO 21750
21740 LET X(M4)=T(I,M1+N+E+1)
21750 IF M(M4)<0 THEN LET X(M4)=L(2,M4)-X(M4) ELSE LET X(M4)=L(1,M4)+X(M4); P
21750:RINT "X",N+1,"  =  ",X(M4); GOSUB 02520
21755 GOSUB 07000
21760 RETURN
22000 REM ****************ADDITION D'UNE CONTRAINTE*******************
22010 PRINT 'CS',"ADDITION D'UNE CONTRAINTE"
22020 PRINT "INTRODUISEZ LES COEFF. DES VARIABLES"
22030 FOR J=1 TO N; INPUT A(8*J),A(0,J); NEXT J
22040 GOSUB 00272; LET A$=" "; ON C GOTO 22050,22020
22050 INPUT "INTRODUISEZ LES COEFF. DU SECOND MEMBRE B:1 ET B:2",A(8),B(0,1),
22050:A(16),B(0,2)
22070 GOSUB 00272; LET A$=" "; ON C GOTO 22080,22050
22080 LET T=0
22090 FOR J=1 TO N; LET T=T+A(0,J)*X(J); NEXT J
22100 IF T<B(0,2) AND T>B(0,1) THEN GOTO 22120
22105 IF T>B(0,2) THEN GOTO 22150
22110 IF T<B(0,1) THEN GOTO 22230 ELSE GOTO 22310
22120 GOSUB 02290; GOSUB 02430; GOTO 22400
22150 LET A=0
22160 IF B(0,2)<0 THEN GOTO 22200
22170 FOR J=1 TO N; LET A=A+A(0,J)*L(1,J); LET T(M1-1,J)=A(0,J); NEXT J
22180 LET T(M1-1,M1+N+E+1)=B(2,0)-A; LET T(M1-1,M1+N-1)=1; GOTO 22230
22200 FOR J=1 TO N; LET A=A+A(0,J)*L(1,J); LET T(M1-1,J)=-A(0,J); NEXT J
22210 LET T(M1-1,M1+N+E+1)=(-B(0,2))+A; LET T(M1-1,M1+N-1)=-1; LET T(M1-1,M1+
22210:N+E-1)=1; LET T(M1+2,M1+N+E-1)=.1E+101; LET V(M1-1)=1
22220 LET P1=M1-1; GOTO 22110
22230 LET A=0
22240 IF B(0,1)<0 THEN GOTO 22280
22250 FOR J=1 TO N; LET A=A+A(0,J)*L(1,J); LET T(M1,J)=A(0,J); NEXT J
22260 LET T(M1,M1+N+E+1)=B(1,0)-A; LET T(M1,M1+N)=-1; LET T(M1,M1+N+E)=1; LET
22260: T(M1+2,M1+N+E)=.1E+101; LET V(M1)=1
22270 GOTO 22300
22280 FOR J=1 TO N; LET A=A+A(0,J)*L(1,J); LET T(M1,J)=-A(0,J); NEXT J
22290 LET T(M1,M1+N)=1; LET T(M1,M1+N+E+1)=(-B(0,1))+A
22300 LET P2=M1
22310 FOR I=1 TO M1-2
22320 IF P(I)>N THEN GOTO 22350
22340 FOR J=1 TO M1+N+E+1; LET T(P1,J)=T(P1,J)-T(P1,P(I))*T(I,J); LET T(P2,J)
```

```
22340 =T(P2,J)-T(P2,P(I))*T(I,J), NEXT J
22350 NEXT I
22360 FOR I=M1-1 TO M1, IF V(I)<>1 THEN GOTO 22380
22370 FOR J=1 TO M1+N+E+1, LET T(M1+2,J)=T(M1+2,J)-T(I,J)* 1E+101, NEXT J
22380 NEXT I
22390 GOTO 01442
22395 GOSUB 02430, GOSUB 02520
22396 GOSUB 07000
22400 RETURN
23000 REM *****P**PARAMETRAGE OU 2nd MEMBRE*****
23010 PRINT 'CS',"PARAMETRAGE OU SECOND MEMBRE ", LET H=7
23020 INPUT (0,ERR=23024) "INTRODUISEZ L'INDICE OU Bi QUE VOUS VOULEZ PARAMET
23020 RER",K (M), GOTO 23026
23024 PRINT "  VOUS AVEZ FAIT UNE ERREUR", PRINT "      RECOMMENCEZ", GOTO 230
23024 20
23026 IF K<=0 THEN GOTO 23024
23030 INPUT (0,ERR=23030) "INTRODUISEZ 1 OU 2 POUR CHANGER LE Bi INF. OU LE B
23030 i SUP  (RESPECT )",L (2)
23040 ON L GOTO 23030,23050,23070
23050 GOSUB 03050
23060 GOTO 23080
23070 GOSUB 03100
23080 INPUT "INTRODUISEZ LA VALEUR MAXI DE LA VARIATION ",T; INPUT "INTRODUIS
23080 EZ LE TAUX DE VARIATION ",X, LET T1=0, PRINT "PARAMETRAGE DE B",K,L, PR
23080 INT "VALEUR MAXI DEMANDEE T=",T, PRINT "TAUX DE VARIATION (INCREMENT) X
23080 =",X, GOSUB 20012
23090 LET N3=M1+N+E+I+K
23100 IF T1>T THEN GOSUB 20210 ELSE GOTO 23105
23103 RETURN
23105 FOR I=1 TO M1+2, LET T(I,M1+N+E+1)=T(I,M1+N+E+1)+T1*T(I,N3), NEXT I
23110 GOTO 01442
23120 REM ****************S O L U T I O N*****************
23140 GOSUB 20200
23150 GOTO 23100
23160 RETURN
24999 REM *************************************************************
25000 GOSUB 02200
25010 ON S GOSUB 02290,02720
25015 IF S=0 THEN GOTO 25030
25017 GOSUB 02810
25020 GOSUB 02010
25025 GOTO 25000
25030 RETURN
30000 IF H<4 THEN GOTO 30010 ELSE GOTO 30020
30010 PRINT "CE PROBLEME N'A PAS DE SOLUTION", GOTO 30030
30020 PRINT "APRES CETTE VALEUR IL N'Y APLUS DE SOLUTION REALISABLE", GOSUB 1
30020 5500
30030 RETURN
34463 END
```

CONCLUSION

Nous ne reviendrons pas en détail sur les objectifs de cette étude.

Toujours est-il qu'il faut préciser que la section programmation paramétrique a été ajoutée a la definition initiale du sujet.

La programmation paramétrique permet de mieux rendre compte des details de la variation des résultats en fonction des coefficients

Les plus grandes difficultés rencontrées se situent au niveau de l'analyse de sensibité. Ces difficultés ont été induites par l'existence des bornes des variables qui rendaient l'utilisation des coefficients des variables d'écart inadéquate. Et ce problème n'a pu être résolu qu'en utilisant le principe de la programmation parametrique.

Au dela de la resolution des programmes linéaires qui s'est révélée satisfaisante après les essais effectués, il faut souligner l'apparition d'incertitudes de dernière heure dues à l'intégration d'une option imprimante pour l'impression des résultats.

En effet des contraintes de temps ne nous ont pas permis de faire des essais avec cette option imprimante.

D'autre part, un des objectifs de l'étude n'a pu être atteint. Il s'agit de la détermination du temps d'opération du programme.

Ceci est dû à l'affluence au niveau du centre de calcul et à la faiblesse de la machine qui perd beaucoup de sa vitesse d'exécution lorsqu'elle est sollicitée sur les trois consoles.

Pour ce qui est des limitations de mémoire et de la dimension des problèmes résolus, elles ne sont pas satisfaisantes.

En effet, l'ordinateur a une capacité limitée à 29000 octets. Le programme à lui seul occupe les 19000 octets. Un problème à deux contraintes et trois variables occupe 3000 octets sur les 11000 qui restent. Ceci nous permet d'estimer la capacité de l'ordinateur à la résolution d'un problème équivalent à six contraintes et sept variables ce qui est assez faible.

Pour cette raison nous suggérons la possibilité de séparer le programme en deux, ce qui permettrait de disposer de plus de mémoire pour la résolution.

Ce principe de la séparation pourrait aussi permettre éventuellement de faire un programme capable de varier simultanément plusieurs coefficients du même type pendant l'analyse de sensibilité.

Pour terminer, nous nous excuserons auprès des utilisateurs pour la présentation du programme, la renumérotation et la liste des variables n'ayant pu être faites, toujours en raison des contraintes de temps.

ANNEXE

MANUEL DE L'UTILISATEUR

Le programme mis au point est sauvé en mémoire sous le nom de EP383SPL

Il est prêt à être exécuté lorsqu'il est chargé par l'instruction LOAD "EP383SPL".

L'utilisateur doit alors inscrire RUN au clavier puis appuyer sur la touche RETOUR.

A partir de ce moment il devra fournir exactement les instructions demandées sur l'écran et appuyer sur RETOUR après chaque donnée introduite.

Avant d'exécuter le programme, l'utilisateur prendra soin d'écrire le programme linéaire sous la forme suivante :

$$\text{Max} \quad c_1 x_1 + c_2 x_2 + \dots$$

s.c.

$$b_{11} \leq a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots \leq b_{12}$$

$$b_{21} \leq a_{21} x_1 + \dots \leq b_{22}$$

$$l_{11} \leq x_1 \leq l_{21} \;,\; l_{12} \leq x_2 \leq l_{22} \;, \dots$$

(Attention : si b_{11} n'existe pas il faut lui donner la valeur $-\infty$ (pour l'ordinateur -1.10^{29}), de même pour b_{12} ou l_{12} qui auront la valeur $+\infty$ (1.10^{29}).)

L'option impression des résultats n'ayant pas été testée, il est conseillé à l'utilisateur d'exécuter d'abord le programme entièrement sans demander l'impression.

Lecture des données sur fichier.

L'enregistrement des données d'un problème sur disque s'étant révélé un travail fastidieux, nous proposons à l'utilisateur le programme ci-dessous permettant un enregistrement plus aisé.

```
10 REM --PROGRAMME D'ENREGISTREMENT DE DONNEES--
20 ERASE "383 DATA"; PRINT "Entrer les données demandées"
30 INPUT "Nombre de contraintes et nombre de variables"; M,N
40 INDEXED "383DATA", M*N+2*M+3*N+1, 34, 1, 0; OPEN(1) "383DATA"
50 WRITE(1) M,N
60 PRINT "COEFFICIENTS DE LA FONCTION ECONOMIQUE"
70 FOR J=1TON; INPUT C ; WRITE(1) C ; NEXT J
80 PRINT "COEFFICIENTS DES CONTRAINTES"
90 FOR I=1TOM
100 FOR J=1TON ; INPUT A ; WRITE(1) A ; NEXT J : NEXT I
110 PRINT " COEFFICIENTS DU SECOND MEMBRE"
120 FOR I = 1TOM
130 FOR J = 1TO2 ; INPUT B ; WRITE(1) B ; NEXT J; NEXT I
```

```
140 PRINT "BORNES DES VARIABLES."
150 FOR J = 1 TO N
160 FOR I = 1 TO 2 : INPUT L : WRITE(I) L ; NEXT I ; NEXT J
170 END
```

L'exécution de ce programme permet l'enregistrement des données d'un problème.

Après chaque valeur introduite l'utilisateur devra appuyer sur RETOUR.

Les coefficients des contraintes devront être introduits un par un et par ligne (par contrainte).

Il en est de même pour les coefficients du second membre.

Les bornes des variables seront introduites une par variable.

DEFINITION DE VARIABLES DANS LE PROGRAMME

A(M,N) : matrice des contraintes

C(N) : matrice de la fonction économique

B(M,1), : matrice du second membre

D(M,1) ; G(1,M) matrices de transfert (second membre)

L(1,N), S(N); matrices des bornes des variables

V(1+M+M+1) : vecteur d'indices des variables artificielles

O(1+M,N) : Matrice de transfert des coefficients des contraintes

T(M1,N1) : matrice de travail

U(M1,N1) : matrice du tableau final

R(N1) : matrice des ratios

P(M1) : vecteur d'indices des variables de base

M(N) : vecteur d'indices des variables ayant [illegible]

X(M) : vecteur des résultats

E(M1) : Vecteur de calcul des itérations

P, Q, : indices du pivot

C : variable de test des coûts réduits

R : variable de comparaison des ratios.

ZO : valeur minimum de la fonction économique

M : nombre de contraintes

N : nombre de variables

M1, N1 : dimensions finales du tableau du P.L

M1 : nombre de contraintes final

E : nombre de variables artificielles

BIBLIOGRAPHIE

Alain MARTEL	"Techniques et applications de la recherche operationnelle" Gaëtan Morin
HILLIER & LIEBERMAN	"Introduction to opérations research" Holden-Day
George B. DANTZIG	"Applications et prolongements de la programmation linéaire" Dunod
M. SIMONNARD	"Programmation linéaire" Dunod
Gérard CHARET	"Méthodologie [illegible] et de la programmation" Sedas-Informatique