UNIVERSITE D'ALGER
ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ECONOMIE

THESE DE FIN D'ETUDES

OPTIMISATION FINANCIERE DES TOLERANCES EN FABRICATION MECANIQUE

SUJET

PROPOSÉ PAR :
Mr. A. GUERRIER

ET

ÉTUDIÉ PAR :
A. KECHOUANE
&
F. OUABDESSELAM

PROMOTION 71

optimisation financière des tolérances en fabrication mécanique

Nous voudrions exprimer nos vifs remerciements à Monsieur AÏT OUYAHIA, Professeur de Recherche Opérationnelle à l'ENP, qui a bien voulu nous faire l'honneur de présider le jury,

à Monsieur GUERRIER, Assistant à l'ENP, qui par son aide constante nous a permis de mener à bien cette étude,

à Monsieur BOUMAHRAT, Chef du Centre de Calcul de l'ENP, qui a bien voulu faire partie du jury,

à Monsieur POUGET, Professeur d'Analyse Numérique à la Faculté des Sciences, qui a accepté notre invitation.

Nous faisons hommage de cette étude à tous nos professeurs.

Table des matières

Introduction

La fabrication d'un produit se décide après l'examen de deux dossiers : le dossier technique et le dossier fabrication.

Le dossier technique, ouvert par le Bureau d'Etude sur une proposition du Service Commercial, sera complété lors de son passage aux Bureaux des Méthodes, d'Ordonnancement-Lancement puis au Service de Production. Ce circuit est parcouru plus d'une fois, car c'est tout à fait corrigé, et adapté aux possibilités de l'entreprise qu'il est discuté en réunion par tous les Services qui l'ont établi et la Direction Générale.

Grâce à ce premier dossier, s'il est accepté, le Bureau d'Ordonnancement Lancement constitue le dossier de fabrication avec la collaboration du Bureau d'Etudes, du Bureau des Méthodes et du Service Entretien. Le dernier examen de ce deuxième document est fait par le Service de Fabrication : il indique aux services déjà cités les mesures et vérifications à faire.

En fait un contrôle continuel est assuré en cours de fabrication et peut apporter éventuellement des rectifications.

Les entreprises, dans leur plus grand nombre, disposent pour ce travail d'équipes de spécialistes. Compte tenu des caractéristiques du produit, des délais et des quantités et enfin du parc machine, ces personnes savent choisir le nombre des passes, les surfaces de départ, les procédés d'usinage, l'ordre d'usinage des surfaces. Toutes ces opérations se résument dans l'établissement de la gamme d'usinage. Viennent ensuite les calculs de cote et la répartition des intervalles de tolérance.

Il faut savoir que l'analyse complète de la pièce, la décomposition de sa fabrication en une série d'opérations élémentaires, l'étude de ces opérations en détail n'est que très rarement faite. Les critères de décision sont bien trop souvent subjectifs.

Un grand nombre de vérifications, des choix multiples, l'homme n'est pas toujours assez rapide pour les traiter et assurer une étude totale, objective et rentable de la pièce à produire.

La solution est peut-être dans l'utilisation de l'ordinateur.

Au préalable il faudrait que toutes les phases du

travail commencé à partir des dessins du Bureau d'Etude aient pu être formalisés.

Ce vœu trouve réponse dans des travaux que nous allons citer.

Origine du problème de l'optimisation financière des tolérances

La vérification des cotes fabrication à l'aide de la méthode vectorielle, dite Citroën, est le prélude à un usage plus important de l'ordinateur dans les industries mécaniques.

I Travaux en cours (M M COLLET, GUERRIER, VERGNAUD)

Ces travaux ont débouché sur un programme de vérification des gammes basé sur la méthode Citroën.

En prenant pour sommets les surfaces et pour arcs les cotes on peut construire un graphe. Ce graphe est nécessairement un arbre car une cotation correcte lie obligatoirement deux surfaces quelconques par une seule cote et élimine ainsi tout circuit.

Comme les cotes se séparent en cotes à réaliser et cotes réalisées par la fabrication il y a deux graphes à N sommets et N-1 arcs.

Ce travail doit se poursuivre par l'établissement du nombre de passes l'ordre d'usinage des surfaces et le choix de la gamme économique.

La dernière partie le calcul des cotes et la répartition des intervalles de tolérance est actuellement utilisable sur l'ordinateur IBM 1130 de l'Ecole Nationale Polytechnique.

Pour l'usage de ce programme quelques hypothèses sont nécessaires :

- la gamme correcte est établie
- les machines pour réaliser cette gamme ont été choisies

Le programme s'effectue en 3 étapes:

La première la recherche des chaines de cotes vérifie la cohérence des données puis établit les liaisons entre cotes relation (ou à réaliser) et cotes fabrication (ou réalisées par la fabrication). Chaque chaîne de cotes fabrication "ferme" une cote relation et détermine ainsi une équation linéaire. Si une des chaînes est impossible le système est rejeté, le graphe des cotes fabrication n'étant pas un arbre.

Posons:

CF_j : cote fabrication de rang j ITF_j : intervalle de tolérance associé

CR_i : cote relation de rang i ITR_i : intervalle de tolérance associé

On obtient le système $(n-1) \times (n-1)$

$$(A)(CF) = (CR)$$

avec (CF) et (CR) vecteurs colonnes de composantes CF_j et CR_i

et (A) une matrice d'élément générateur A_{ij} tel que:

$$A_{ij} = \begin{cases} 0 & \text{la cote n'existe pas dans la chaîne} \\ +1 & \text{la cote } (\overrightarrow{S_1, S_2}) \text{ est dans la chaîne} \\ -1 & \text{la cote } (\overrightarrow{S_2, S_1}) \text{ est dans la chaîne} \end{cases}$$

Répartition des intervalles de tolérance : deuxième étape.

Les ITR sont imposés et le problème est de rechercher les ITF tels que:

$$(B)(ITF) = (ITR)$$

où (B) représente la matrice dont les éléments sont les valeurs absolues de ceux de (A)

Un premier essai est tenté en posant

$$(ITF) = (ITECO)$$

avec ITECO vecteur colonne de composantes les intervalles de tolérance de chaque cote travail (ou fabrication) (voir définition chap **I**, § 2)

Soit $(B)(ITECO) = (ITR)$ et la gamme est optimale, soit $(B)(ITECO) \neq (ITR)$ auquel cas il faut réajuster les composantes de (ITF). Dans cette deuxième alternative on réduit les composantes de (ITECO) à partir d'un coefficient de réduction définit comme le rapport

$$\frac{\sum_j B_{ij}\, ITECO_j}{ITR_i}$$

pour toutes les équations du système non satisfaites. Le programme sépare les contraintes satisfaites des autres qu'il traite par blocs de chaînes constituées du même nombre d'arcs. En prenant le coefficient de réduction maximum pour chaque bloc il répète ces opérations de vérification et d'ajustement.

II Conditions économiques de la fabrication.

Une étude des dispersions est toujours effectuée en début de série. On distingue :

Dp : la dispersion de production

Da : la dispersion accidentelle.

qui sont : soit calculées après dépouillement statistique

soit estimées par analogie ou par expérience

Ces dispersions ont une influence sur la détermination des intervalles de tolérance des cotes travail (ITCT).

On a les renseignements suivants :

Sans réglage entre chaque changement d'outil :

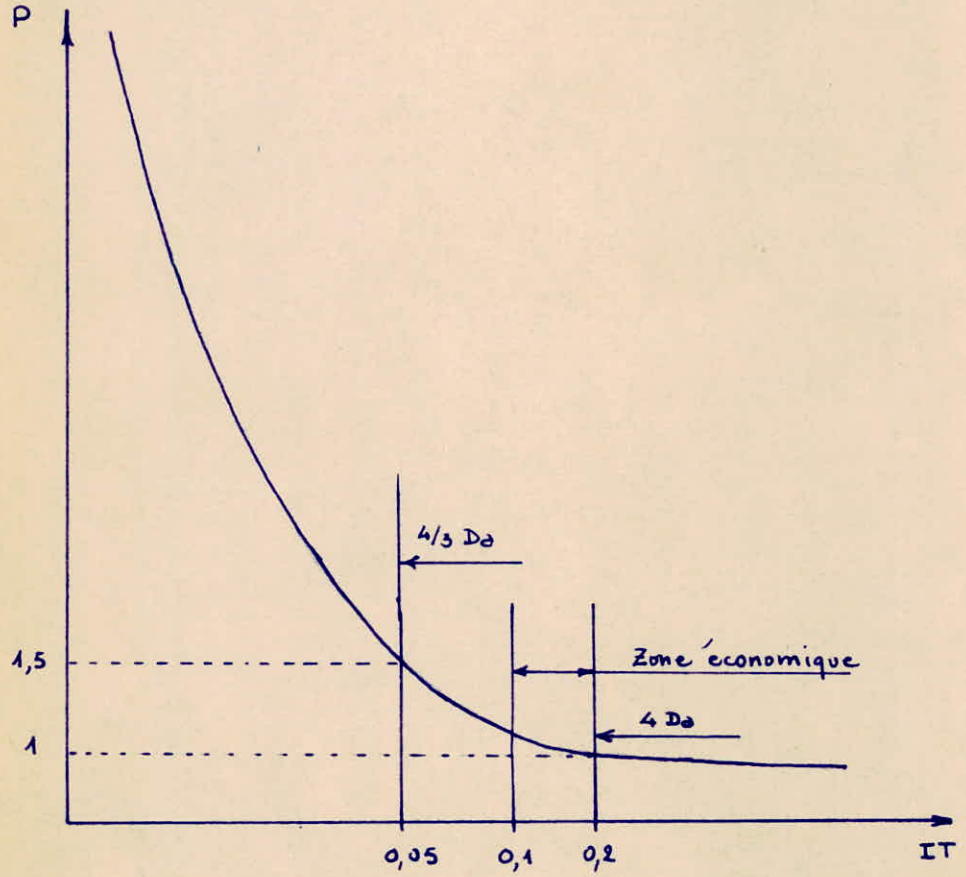
ITCT = 4 Da c'est l'intervalle de tolérance économique nommé auparavant ITECO

Avec réglage entre les changements d'outil

$IT \geq \frac{4}{2} Da$ c'est la limite de la fabrication.

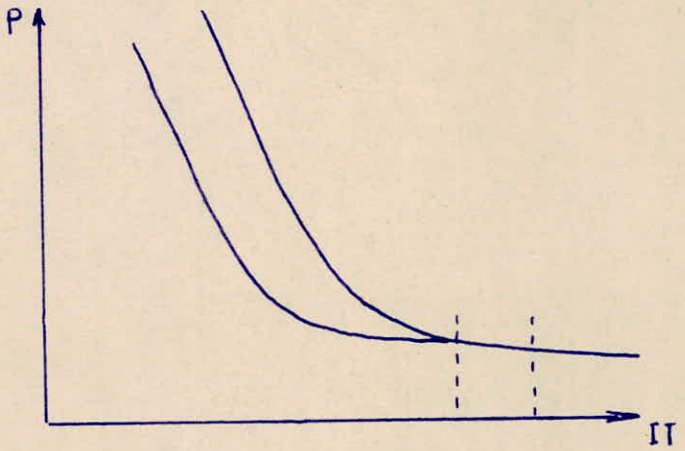
Si $IT < \frac{4}{3} Da$ le procédé et la gamme doivent être modifiés

Comme la répartition des intervalles de tolérance pour être économique doit tenir compte des coûts on doit observer l'évolution du prix de revient par pièce en fonction de la précision



c'est à partir de cette courbe qu'apparaissent les limites économiques d'un procédé. Pour $IT > 4 Da$ le prix de revient à peu près constant : il vaut donc mieux au même coût

travailler avec une meilleure précision et se fixer à IT = 4Da. quand IT < $\frac{4}{2}$ Da les coûts croissent rapidement et on peut toujours trouver un procédé plus adapté qui pour la même précision demandera des dépenses moins grandes (représentation graphique ci dessous)



III OPTIMISATION FINANCIERE

La méthode actuellement programmée, essaie au mieux de fournir peu d'"ITF" inférieurs aux ITEC∅. Pendant les opérations de réduction il n'y a aucune règle de priorité entre les surfaces.

Cependant certaines d'entre elles demandent pour leur réalisation de plus grandes quantités de travail, une qualité supérieure, des machines et un appareillage plus précis. Ce sont ces surfaces qui ~~doivent~~ supportent les dépenses les plus fortes, et les intervalles de tolérance qui leur sont associés

doivent être les moins réduits possible si on cherche à produire au moindre coût.

Une amélioration peut donc être ici apportée en cherchant à rendre minimum le prix de revient de la fabrication tout en attribuant à chaque intervalle de tolérance une valeur de sa zone économique.

Construction du modèle mathématique

I Formulation du problème.

On s'intéresse aux équations en ITF et ITR non satisfaites par ITF = ITECO. Ces équations ont la forme suivante.

$$\sum_{i=1}^{N} B_{ji}\, ITF_i = ITR_j \qquad (j = 1, \dots, M)$$

Posons $ITF_i = x_i$, $ITR_j = b_j$, $B_{ji} = a_{ji}$

Nouvelle écriture des équations :

$$\sum_{i=1}^{N} a_{ji}\, x_i = b_j \qquad (j = 1, \dots, M)$$

Chaque intervalle de tolérance prend une valeur de sa zone économique soit :

$$\frac{\alpha_i}{2} \leq x_i \leq \alpha_i \qquad (i = 1, \dots, N)$$

L'équation de la courbe d'évolution du prix en fonction de la précision est :

$$g_i(x_i) = 0{,}25\, P_i \left(3 + \frac{\alpha_i}{x_i}\right) \qquad (i = 1, \dots, N)$$

Nous cherchons à fixer la valeur de l'intervalle de tolérance de la cote fabrication de façon à travailler au moindre coût.

Le problème s'énonce donc ainsi :

Minimiser $G(x_1, \dots, x_N) = \sum_{i=1}^{N} g_i(x_i)$

sous les contraintes

$$\begin{cases} a_{11} x_1 + \dots + a_{1N} x_N = b_1 \\ \vdots \\ a_{M1} x_1 + \dots + a_{MN} x_N = b_M \end{cases}$$

et $\quad \frac{\alpha_i}{2} \leq x_i \leq \alpha_i \quad (i = 1, \dots, N)$

II Le problème est un programme convexe.

Les conditions :

- l'ensemble des contraintes est un ensemble convexe
- la fonction à optimiser est une fonction convexe

exigées pour un programme convexe sont vérifiées.

Montrons que les contraintes forment un ensemble convexe.

Soient $X^{(1)} = (x_1^{(1)}, \dots, x_N^{(1)}) \in \mathbb{R}^N$ et $X^{(2)} = (x_1^{(2)}, \dots, x_N^{(2)}) \in \mathbb{R}^N$

vérifiant $\frac{\alpha_i}{2} \leq x_i^{(1)} \leq \alpha_i$, $\frac{\alpha_i}{2} \leq x_i^{(2)} \leq \alpha_i \quad (i = 1, \dots, N)$

que peut-on dire de $X = \lambda X^{(1)} + (1-\lambda) X^{(2)}$ avec $0 \leq \lambda \leq 1$?

De $x_i^{(1)} \geq \frac{\alpha_i}{2}$ et $x_i^{(2)} \geq \frac{\alpha_i}{2} \quad (i = 1, \dots, N)$ on obtient

$$\lambda x_i^{(1)} \geq \lambda \frac{\alpha_i}{2} \text{ et } (1-\lambda) x_i^{(2)} \geq (1-\lambda) \frac{\alpha_i}{2}$$

et finalement :

$$\lambda x_i^{(1)} + (1-\lambda) x_i^{(2)} \geq \frac{\alpha_i}{2}$$

L'espace des points X dont les coordonnées vérifient

$$x_i \geq \frac{\alpha_i}{2} \quad (i = 1, \dots, N)$$

est donc un ensemble convexe.

De la même façon on montre que l'espace des points X dont les coordonnées vérifient

$$x_i \leq \alpha_i \qquad (i = 1, \ldots, N)$$

est un ensemble convexe.

De l'intersection de ces deux espaces naît l'espace des points X dont les coordonnées vérifient :

$$\frac{\alpha_i}{2} \leq x_i \leq \alpha_i \qquad (i = 1, \ldots, N)$$

qui est donc convexe.

Les demi - espaces fermés $a_{j1} x_1 + \ldots + a_{jN} x_N \leq b_j$ et $a_{j1} x_1 + \ldots + a_{jN} \geq b_j$ $(j = 1, \ldots, M)$ sont convexes.

Leurs intersections, les hyperplans

$$a_{j1} x_1 + \ldots + a_{jN} x_N = b_j \qquad (j = 1, \ldots, M)$$

sont donc convexes.

Ainsi l'ensemble des points admissibles du problème est un ensemble convexe.

L'ensemble $\mathcal{D}$ des points X $(x_1, \ldots, x_N)$ avec $x_i > 0$ $(i = 1, \ldots, N)$ sur lequel est définie la fonction $G(x_1, \ldots, x_N)$ est un ensemble convexe. Pour prouver que la fonction $G(X)$ est convexe il faut vérifier :

$$G\left(\lambda X^{(1)} + (1-\lambda) X^{(2)}\right) \leq \lambda G\left(X^{(1)}\right) + (1-\lambda) G\left(X^{(2)}\right) \qquad 0 \leq \lambda \leq 1$$

$X_i^{(1)}$, $X^{(2)} \in \mathbb{R}^N$.

Posons pour simplifier les calculs :

$$G(X) = K + \sum_{i=1}^{N} \frac{A_i}{x_i} = K + G'(X) \quad , K \in \mathbb{R}$$

Il faut donc montrer que $G'(X)$ est une fonction convexe.

En développant nous obtenons :

$$\underbrace{\frac{A_1}{\lambda x_1^{(1)} + (1-\lambda) x_1^{(2)}}}_{S_1} + \cdots + \frac{A_N}{\lambda x_N^{(1)} + (1-\lambda) x_N^{(2)}} \leq \underbrace{\lambda \frac{A_1}{x_1^{(1)}} + (1-\lambda) \frac{A_1}{x_1^{(2)}}}_{S_2} + \cdots + \lambda \frac{A_N}{x_N^{(1)}} + (1-\lambda) \frac{A_N}{x_N^{(2)}}$$

Comparons S_1 et S_2 en partant de :

$$\frac{A_1}{\lambda x_1^{(1)} + (1-\lambda) x_1^{(2)}} \leq \lambda \frac{A_1}{x_1^{(1)}} + (1-\lambda) \frac{A_1}{x_1^{(2)}}$$

on obtient

$$\frac{1}{\lambda x_1^{(1)} + (1-\lambda) x_1^{(2)}} \leq \frac{\lambda}{x_1^{(1)}} + \frac{1-\lambda}{x_1^{(2)}}$$

$$1 \leq \frac{\lambda}{x_1^{(1)}} + \frac{1-\lambda}{x_1^{(2)}} \left(\lambda x_1^{(1)} + (1-\lambda) x_1^{(2)}\right)$$

$$1 \leq \lambda^2 + \lambda(1-\lambda) \frac{x_1^{(1)}}{x_1^{(2)}} + \lambda(1-\lambda) \frac{x_1^{(2)}}{x_1^{(1)}} + (1-\lambda)^2$$

$$1 \leq \lambda^2 + 1 - 2\lambda + \lambda^2 + \lambda(1-\lambda) \left(\frac{x_1^{(1)}}{x_1^{(2)}} + \frac{x_1^{(2)}}{x_1^{(1)}}\right)$$

$$0 \leq 2\lambda(\lambda-1) + (1-\lambda)\lambda \left(\frac{x_1^{2(2)} + x_1^{2(1)}}{x_1^{(1)} \cdot x_1^{(2)}}\right)$$

$$0 \leq \lambda(1-\lambda)\frac{1}{x_1^{(1)}\,x_1^{(2)}}\left(x_1^{2(1)} + x_1^{2(2)} - 2\,x_1^{(1)}\,x_1^{(2)}\right)$$

Soit

$$0 \leq \lambda(1-\lambda)\frac{1}{x_1^{(1)}\,x_1^{(2)}}\left(x_1^{(1)} - x_1^{(2)}\right)^2$$

Comme $x_1^{(1)} > 0$, $x_1^{(2)} > 0$, $0 \leq \lambda \leq 1$ l'inéquation est vérifiée.

Notons que pour $0 < \lambda < 1$ on a une inégalité stricte.

Ainsi, en comparant terme à terme les deux expressions on prouve que:

$$G\left(\lambda X^{(1)} + (1-\lambda)X^{(2)}\right) < \lambda G\left(X^{(1)}\right) + (1-\lambda)\,G\left(X^{(2)}\right),\quad 0<\lambda<1$$

La fonction $G(X)$ est une fonction strictement convexe.

III Propriétés de l'optimum.

La stricte convexité de la fonction nous assure un minimum unique. Ce minimum existe si les contraintes sont compatibles et en particulier si

$$\frac{1}{2}\sum_{j=1}^{N}\alpha_i \leq b_j \leq \sum_{i=1}^{N}\alpha_i \qquad (j = 1,\ldots,N)$$

On pourra éliminer du système les contraintes à coefficient et second membre proportionnels.

Le minimum est à distance finie car le domaine des points admissibles est une variété linéaire.

Programmation dynamique

Nous avons choisi la programmation dynamique avant tout parce que les valeurs optimales des intervalles de tolérance doivent être discrètes. Il importe que les intervalles de tolérance n'aient pas des valeurs continues, car les approximations qu'il faut alors faire pour permettre à l'ouvrier d'avoir sur ses fiches des grandeurs normalisées, peuvent sérieusement affecter l'exactitude de la solution.

La programmation dynamique est par ailleurs le meilleur des procédés que nous connaissions pour lequel des contraintes critiques de la forme :

$$\alpha_i \leq x_i \leq \beta_i$$

ne soient pas du tout source de difficultés.

Les caractéristiques de notre problème satisfont aux hypothèses de base de la programmation dynamique : son emploi est donc possible.

En effet

- le coût d'exécution des "IT" sont mesurés avec une même unité
- la dépense résultant de la réalisation d'un "IT"

est indépendante des valeurs des autres "IT". Les fonctions $g_i(x_i)$ $(i=1,\dots,N)$ ne sont donc fonction que d'une seule variable.

- Le coût total est la somme des coûts partiels.

La technique employée en programmation déterministe repose sur un théorème et un principe qui apparaît comme un corollaire de ce théorème. Nous les énonçons.

Théorème d'optimalité :

Toute sous politique, extraite d'une politique optimale, est elle-même optimale.

Principe d'optimalité :

Parmi l'ensemble des politiques qui contiennent une sous politique donnée (optimale ou non) la meilleure est celle qui on obtient en complétant la sous-politique donnée par une sous-politique optimale.

Pour pouvoir faire des distinctions suivant la nature du système des contraintes nous allons parler de programmes unidimensionnel et multidimensionnel.

Programme unidimensionnel.

1. On cherche le minimum M de la fonction de N variables

$$G(x_1, \dots, x_N) = \sum_{i=1}^{N} g_i(x_i)$$

sous les contraintes :

$$\sum_{i=1}^{N} x_i = b \qquad x_i \geq 0$$

En vertu du principe d'optimalité la méthode de BELLMAN fournit des fonctions auxilliaires $f_1(x), \dots, f_{N-1}(x)$, $f_N(x)$ liées par une relation de récurrence :

$$\begin{cases} f_1(x) = g_1(x) \\ f_i(x) = \inf \left[g_i(x_i) + f_{i-1}(x - x_i) \;/\; x_i \in [0, x] \right] \quad (i = 2, \dots, N) \\ x \in [0, b] \end{cases}$$

C'est le dernier élément de la suite des fonctions $\{ f_i(x) \}$ $(i = 1, \dots, N)$ soit $f_N(b)$, résultat de sous-politiques toutes optimales, qui est le minimum de la fonction $G(x_1, \dots, x_N)$.

Pour pouvoir faire du calcul du minimum un calcul itératif nous avons dû transformer le problème "continu" en un problème discret. La discrétisation est introduite en imposant pour les variables x_i $(i = 1, \dots, N)$ une croissance dans l'intervalle $[0, x]$ par pas de Δ.

Les organigrammes A1 et A2 sont les applications de cette

Nous avons eu besoin de trois tables de valeurs : FA, FB, X respectivement pour f_{i-1}, f_i, x_i. La recherche de la solution en machine est représentée par A2.

Le programme testé sur 3 exemples a donné des résultats justes ou très approchés.

Nous avons obtenu la solution suivante :

$M = 48 \qquad x_1 = 4{,}8 \qquad x_2 = 2{,}4 \qquad x_3 = 1{,}6 \qquad x_4 = 1{,}2$

au problème $\quad \text{Min } x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2$

$$\text{avec} \quad x_1 + x_2 + x_3 + x_4 = 10$$

$$x_i \geq 0 \quad (i = 1, \ldots, 4)$$

qui est la solution exacte. Nous avions choisi un pas de 0,1.

En cherchant à résoudre

$$\text{Min } \frac{100}{x_1} + \frac{200}{x_2} + \frac{300}{x_3} + \frac{400}{x_4}$$

$$\text{avec} \quad x_1 + x_2 + x_3 + x_4 = 5 \qquad x_i > 0 \quad (i = 1, \ldots, 4)$$

nous avons étudié l'influence du pas.

Le choix du pas a surtout des conséquences sur le minimum de la fonction objectif. Les valeurs des variables ne sont que très peu affectées.

Solution analytique

$x_1 = 0,81 \quad x_2 = 1,15 \quad x_3 = 1,41 \quad x_4 = 1,63$

$M = 754,5$

Programmation dynamique

Pas = 0,05 $\quad M = 733,63$

$x_1 = 0,75 \quad x_2 = 1,15 \quad x_3 = 1,45 \quad x_4 = 1,65$

Pas = 0,025 $\quad M = 744,37$

$x_1 = 0,775 \quad x_2 = 1,15 \quad x_3 = 1,425 \quad x_4 = 1,65$

La "mémoire" nécessaire dépend du pas et de la valeur de b tout comme le nombre d'itérations. Si une variable x_i peut prendre k_i valeurs espacées de Δ le nombre total d'itérations pour résoudre un problème à N variables est:

$$k_1 + \sum_{i=2}^{N} \frac{k_i (k_i + 1)}{2}$$

2. Il faut quelques modifications pour trouver le minimum M de la fonction :

$$G(x_1, \dots, x_N) = \sum_{i=1}^{N} g_i(x_i)$$

sous les contraintes $\sum_{i=1}^{N} a_i x_i = b \quad , \quad x_i \geq 0$

A la première itération :

$$f_1(x) = g_1\left(\frac{x}{a_1}\right) \qquad x \in [0, b]$$

et $x_1 = \frac{x}{a_1}$

Pour les autres :

$$f_i(x) = \operatorname{Inf}\left[g_i(x_i) + f_{i-1}(x - a_i x_i) \,/\, x_i \in \left[0, \frac{x}{a_i}\right] \right]$$
$$x \in [0, b]$$

La recherche la solution également fait intervenir les coefficients a_i

3. Problème général de la répartition à une seule dimension.

$$\text{Min } G(x_1, \ldots, x_N) = \sum_{i=1}^{N} g_i(x_i)$$

sous les contraintes

$$\sum_{i=1}^{N} a_i x_i = b$$

$$\alpha_i \leq x_i \leq \beta_i \qquad (i = 1, \ldots, N)$$

On a ainsi, dans R, une suite donnée d'intervalles $[\alpha_i, \beta_i]$ $(i = 1, \ldots, N)$, des fonctions données $g_i(x_i)$. $(i = 1, \ldots, N)$ respectivement définies sur ces intervalles et b, un réel donné appartenant à l'intervalle $\left[\sum_{i=1}^{N} \alpha_i, \sum_{i=1}^{N} \beta_i\right]$.

Soient :

$$d_i = \sum_{k=1}^{i} \alpha_i \quad \text{et} \quad e_i = \sum_{k=1}^{i} \beta_i$$

Les fonctions auxiliaires $f_i(x)$ $(i = 1, \ldots, N)$ définies alors sur $[d_i, e_i]$ sont telles que :

$$f_i(x) = \text{Inf}\left[g_i(x_i) + f_{i-1}(x - a_i x_i) \,/\, x_i \in [\alpha_i, \beta_i],\ x_i \leq \frac{x}{a_i},\right.$$
$$\left. d_{i-1} \leq x - a_i x_i \leq e_{i-1}\right]$$
$$x \in \left[d_i, \text{Inf}\,[e_i, b]\right]$$

Tous les exemples traités montrent qu'il n'y a pas

de variables prioritaires, état qui aurait pu être la conséquence des bornes. Le nouveau type de contrainte $\alpha_i \leq x_i \leq \beta_i$ présente de grands avantages pour une réduction du temps de calcul et une diminution de la "mémoire" nécessaire. C'est ainsi qu'à l'inverse de la plupart des méthodes d'optimisation la programmation dynamique s'accommode très bien de ce genre de contrainte.

C'est à partir des organigrammes A3 et A4 que nous avons programmé nos exemples. T_1 représente f_{i-1}, T_2, f_i, $X(K,I)$ la table des valeurs des variables.

Programme multidimensionnel

Le nombre de contraintes de liaison augmente. Nous étudions deux possibilités de résoudre ce programme.

1. Méthode A

Le système de contrainte suivant :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_n \end{cases}$$

peut être mis sous forme matricielle $AX = B$

avec $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & & & \\ a_{m1} & a_{m2} & \cdots & a_{mN} \end{pmatrix}$ $\quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ $\quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$

Nous décomposons la matrice en blocs et obtenons ainsi une condensation du système et une représentation plus simple

Soient $A_1 = (a_{11}\ a_{21} \cdots a_{m1})$

$A_2 = (a_{12}\ a_{22} \cdots a_{m2})$

$\vdots$

$A_N = (a_{1N}\ a_{2N} \cdots a_{mN})$

$B = (b_1\ b_2 \cdots b_m)$

Nous écrivons symboliquement le système :

$$A_1 x_1 + A_2 x_2 + \cdots + A_N x_N = B$$

On introduit $Y\ (y_1\ y_2 \cdots y_m)$ pour résoudre le problème par programmation dynamique.

Les fonctions auxiliaires sont du type :

$$f_i(Y) = f_i(y_1, y_2, \cdots, y_m) = \operatorname{Inf} \left[g_i(x_i) + f_{i-1}(Y - A_i x_i) \,/\, x_i \leq \frac{y_j}{a_j},\ (j = 1, \cdots, M) \right]$$

avec $y_j \in [0, b_j]$ $\quad (j = 1, \cdots, M)$

$Y - A_i x_i$ est le bloc $\left((y_1 - a_{1i} x_i), \cdots, (y_m - a_{mi} x_i) \right)$

Considérons à part $f_1(Y)$.

$$\begin{cases} f_1(Y) = \operatorname{Inf} \left[g_1(x_1) \,/\, A_1 x_1 < Y \right] \\ y_j \in [0, b_j] \quad (j = 1, \cdots, M) \end{cases}$$

A l'itération i du système d'inégalités suivant:

$$a_{1i}\, x_i \leq y_1$$
$$a_{2i}\, x_i \leq y_2$$
$$\vdots$$
$$a_{mi}\, x_i \leq y_m$$

une seule est conservée : celle qui fournit $\underset{j}{\text{Inf}} \left[\frac{y_j}{a_{ji}}\right]$

Ainsi la méthode A correspond à un programme unidimensionnel. La reconsidération de la matrice par blocs colonnes représente un gain de place : deux seulement sont nécessaires à chaque itération ce sont A_i $(i=1,\dots,N)$ et B.

Dans le cas du problème général

$$\text{Min } G(x_1,\dots,x_n) = \sum_{i=1}^{N} g_i(x_i)$$

sous les contraintes
$$\begin{cases} \sum_{i=1}^{N} a_{ji}\, x_i = b_j & (j=1,\dots,M) \\ \alpha_i \leq x_i \leq \beta_i & (i=1,\dots,N) \end{cases}$$

nous nous ramenons à un problème unidimensionnel à l'aide de la fonction auxiliaire

$$f_i(Y) = \text{Inf}\left[g_i(x_i) + f_{i-1}(Y - A_i x_i) \,/\, x_i \in \left[\alpha_i, \text{Inf}\left[\beta_i, \frac{y_j}{a_j}\right]\right], \; (j=1,\dots,M) \right]$$

avec $y_j \in \left[\sum_{k=1}^{i} \alpha_i,\ \text{Inf}\left[\sum_{k=1}^{i} \beta_i, b_j\right]\right]$

L'organigramme B1, dans le cas particulier du problème de l'optimisation des tolérances où les coefficients a_{ji} sont des valeurs booléennes, indique les conditions auxquelles sont satisfaites les N contraintes.

A partir de l'organigramme B2 nous constatons que la programmation de la méthode A n'est différente du programme unidimensionnel du même type que pour la mise en mémoire des données. Les tables des valeurs des fonctions auxiliaires et des variables sont des tables à M entrées. Il faut imbriquer les boucles pour que une à une et toute seule les variables y_j prennent toutes les valeurs des intervalles et qu'à chaque pas on procède aux calculs.

La recherche de la solution est une recherche ascendante. En connaissant les contraintes où intervient la variable x_i, soit par exemple p et q, on détermine la valeur de x_{i-1} en se référant à la table des valeurs de x_{i-1} avec point $x_{i-1}\left(b'_1, b'_2, \dots, b'_p - a_{pi}x_i, b'_{p+1}, \dots, b'_q - a_{qi}x_i, \dots, b'_M\right)$

avec $b'_j \leq b_j \quad (j = 1, \dots, M)$

Exemple d'application : Min $\dfrac{4}{x_1} + \dfrac{16}{x_2} + \dfrac{9}{x_3}$

sous les contraintes
$$\begin{cases} x_i > 0 & (i = 1,2,3) \\ x_1 + x_2 = 1,2 \\ x_1 + x_3 = 1,2 \end{cases}$$

En utilisant les multiplicateurs de Lagrange et après dérivations nous avons :

$$\frac{x_1}{\sqrt{4}} = \frac{x_2}{\sqrt{16+9}} = \frac{x_1+x_2}{\sqrt{4}+\sqrt{16+9}} = \frac{1,2}{\sqrt{4}+\sqrt{16+9}}$$

et la solution est : $M = 40,83 \quad x_1 = 0,34 \quad x_2 = x_3 = 0,86$

Solution "dynamique" avec un pas de 0,1 :

$M = 44 \quad x_1 = 0,5 \quad x_2 = x_3 = 0,7$

Plus le pas est petit plus on tend vers la solution exacte mais alors il faut disposer d'une place importante en mémoire.

L'inconvénient majeur de cette méthode est la saturation trop rapide de la mémoire.

2. Méthode B

On tente ici la réduction de la dimension du problème en utilisant les coefficients de Lagrange.

En utilisant λ un paramètre réel déterminer les valeurs de

$$M = \text{Inf} \sum_{i=1}^{N} g_i(x_i)$$

tous les contraintes $\sum_{i=1}^{N} x_i = b$

$$\alpha_i \le x_i \le \beta_i \quad (i = 1, \dots, N)$$

avec $b \in \left[\sum_{i=1}^{N} \alpha_i , \sum_{i=1}^{N} \beta_i\right]$

et de

$$M' = \text{Inf}\left[\sum_{i=1}^{N} g_i(x_i) - \lambda x_i \;/\; x_i \in [\alpha_i, \beta_i] , \sum_{i=1}^{N} x_i = b\right]$$

tout un même problème

Considérons $M'' = \text{Inf}\left[\sum_{i=1}^{N} g_i(x_i) - \lambda x_i \,/\, x_i \in [\alpha_i, \beta_i] \ (i=1,\dots,N)\right]$

nous avons :

$$M'' = \sum_{i=1}^{N} \text{Inf}\left(g_i(x_i) - \lambda x_i\right) \,/\, x_i \in [\alpha_i, \beta_i] \ (i=1,\dots,N)$$

Soit $x_i(\lambda)$ la valeur de x_i fournissant le minimum de la fonction $g_i(x_i) - \lambda x_i$, $(i = 1,\dots,N)$ respectivement.

La somme $\sum_{i=1}^{N} x_i(\lambda)$ définie pour λ réel prend ses valeurs dans l'intervalle $\left[\sum_{i=1}^{N} \alpha_i, \sum_{i=1}^{N} \beta_i\right]$.

Nous avons le théorème suivant:

Une condition nécessaire et suffisante pour que la suite $\{x_i(\lambda)\}$ $(i=1,\dots,N)$ soit solution du 1er problème énoncé et qu'il existe une valeur λ_0 de ~~la~~ λ telle que $x_i(\lambda_0)$ soit solution en u de

$$\text{Inf}\left[g_i(u) - \lambda_0 u \,/\, u \in [\alpha_i, \beta_i]\right] \quad \text{pour } i=1,\dots,N$$

et que la relation suivante soit vérifiée :

$$\sum_{i=1}^{N} x_i(\lambda_0) = b$$

Une méthode de résolution du premier problème est donc : ajuster la valeur de λ de façon à ce que $\sum_{i=1}^{N} x_i(\lambda) = b$

Notons que :

$$M' = \text{Inf}\left[\sum_{i=1}^{N} g_i(x_i) - \lambda x_i\right]$$

$$M' = \text{Inf}\left[\sum_{i=1}^{N} g_i(x_i) - \lambda \sum_{i=1}^{N} x_i\right]$$

Or $M' \geq 0$

Donc $M' = \text{Inf}\left[\sum_{i=1}^{N} g_i(x_i)\right] - \text{Max}\left[\lambda \sum_{i=1}^{N} x_i\right]$

$$M' = M - \lambda \sum_{i=1}^{N} x_i \qquad \text{avec } x_i = x_i(\lambda) \quad (i=1,\dots,N)$$

A l'optimum en particulier :

$$\hat{M}' = \hat{M} - \lambda_0 b$$

Soit $\underline{\hat{M} = \hat{M}' + \lambda_0 b}$

Dans le cas de fonctions $g_i(x_i)$ strictement convexes, à dérivées premières et secondes continues la somme $\sum_{i=1}^{N} x_i(\lambda)$ est une fonction de λ strictement croissante depuis $\sum_{i=1}^{N} \alpha_i$ jusqu'à $\sum_{i=1}^{N} \beta_i$.

Envisageons le problème P_1.

Minimiser $G(x_1,\dots,x_N) = \sum_{i=1}^{N} g_i(x_i)$

sous les contraintes :

$$\begin{cases} \begin{cases} a_{11}x_1 + \dots + a_{1N}x_N = b_1 \\ a_{21}x_1 + \dots + a_{2N}x_N = b_2 \end{cases} \\ \alpha_i \le x_i \le \beta_i \qquad (i=1,\dots,N) \end{cases}$$

Par analogie avec les résultats précédents nous considérons le problème équivalent P_2 :

Minimiser $G'(x_1,\dots,x_N) = \sum_{i=1}^{N} g_i(x_i) - \lambda \sum_{i=1}^{N} a_{1i}x_i$

sous les contraintes :

$$\begin{cases} a_{21}x_1 + \dots + a_{2N}x_N = b_2 \\ \alpha_i \le x_i \le \beta_i \qquad (i=1,\dots,N) \end{cases}$$

La recherche de l'optimum par programmation dynamique est alors basée sur les fonctions auxiliaires suivantes:

$$f_i(x) = \text{Inf}\left[g_i(x_i) - \lambda a_{1i} x_i + f_{i-1}(x - a_{2i} x_i) \;/\; x_i \in [\alpha_i, \beta_i],\right.$$
$$\left. x_i \leq \frac{x_{\max}}{a_{2i}},\ \sum_{k=1}^{i} \alpha_k \leq x - a_i x_i \leq \sum_{k=1}^{N} \beta_k \right]$$
$$x \in \left[\sum_{i=1}^{N} \alpha_i,\ \text{Inf}\left[\sum_{k=1}^{i} \beta_k, b_2 \right] \right]$$

Pour une valeur λ' de λ nous obtenons une solution

$$f_N(b_2) + \lambda' h(\lambda') \qquad \text{où } h(\lambda') = \sum_{i=1}^{N} x_i(\lambda')$$

qui n'est pas nécessairement la solution optimale de P_1.

Nous savons qu'il faut réussir à ajuster la valeur de λ de sorte que $h(\lambda) = b_1$. On détermine généralement cette valeur par interpolation de la fonction $h(\lambda)$ à partir de deux valeurs de départ $h(\lambda_1)$ et $h(\lambda_2)$.

Les méthodes d'interpolation ne sont pas toujours les même.

1er exemple (problème 9)

Minimiser $x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2$

sous les contraintes

$$\begin{cases} 2x_1 + x_2 + x_3 + x_4 = 13{,}5 \\ x_1 + x_2 + x_3 + x_4 = 10 \end{cases} \qquad x_i \geq 0 \ (i = 1, \ldots, 4)$$

Pour des valeurs différentes de λ nous avons calculé par programmation dynamique l'optimum du problème

Minimiser $x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 - \lambda(2x_1 + x_2 + x_3 + x_4)$

dans le domaine

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 10 \\ x_i \geq 0 \quad (i = 1, \dots, 4) \end{cases}$$

Le graphe de la fonction $h(\lambda)$ est représenté en G. La courbe expérimentale vérifie l'équation de $h(\lambda)$ calculée analytiquement : $h(\lambda) = \frac{13}{50}\lambda + \frac{74}{5}$

Ici une interpolation linéaire avec des valeurs de λ comprises entre 0 et 20 nous mène à l'optimum de P avec une bonne précision.

2ème exemple.

$$P_1 \begin{cases} \text{Min} \quad \frac{10}{x_1} + \frac{20}{x_2} + \frac{30}{x_3} + \frac{40}{x_4} \\ \text{dans le domaine} \\ x_2 + x_4 = 4 \\ x_1 + x_2 + x_3 + x_4 = 10 \\ x_i > 0 \quad (i = 1, \dots, 4) \end{cases}$$

$$P_2 \begin{cases} \text{Min} \quad \frac{1,2}{x_1} + \frac{2,4}{x_2} + \frac{2,88}{x_3} + \frac{0,24}{x_4} \\ \text{dans le domaine} \\ x_1 + x_2 = 2,88 \\ x_1 + x_2 + x_3 + x_4 = 7,3 \\ x_i > 0 \quad (i = 1, \dots, 4) \end{cases}$$

Les graphes G1 et G2 des $h(\lambda)$ pour P_1 et P_2 respectivement sont des courbes en S qui présentent un point d'inflexion pour la valeur $\lambda = 0$ et qui sont à asymptotes horizontales : ε et 10 pour G_1, ε et 7,3 pour G_2 (ε dépend du pas)

Nous obtenons les solutions optimales en procédant à une interpolation de type hyperbolique à asymptote horizontale :

$$h = A\left(1 + \frac{\lambda_0}{\lambda}\right)$$

Les valeurs initiales de λ sont calculées de la façon suivante : nous associons les paramètres λ et μ respectivement à la 1ère et à la 2ème contrainte. Nous calculons les dérivés partielles et nous obtenons le système :

(cas de $\mathcal{P}_2$) $\lambda+\mu = -\frac{1,2}{x_1^2}$, $\lambda+\mu = -\frac{2,4}{x_2^2}$, $\lambda = -\frac{2,88}{x_3^2}$, $\lambda = -\frac{0,24}{x_4^2}$

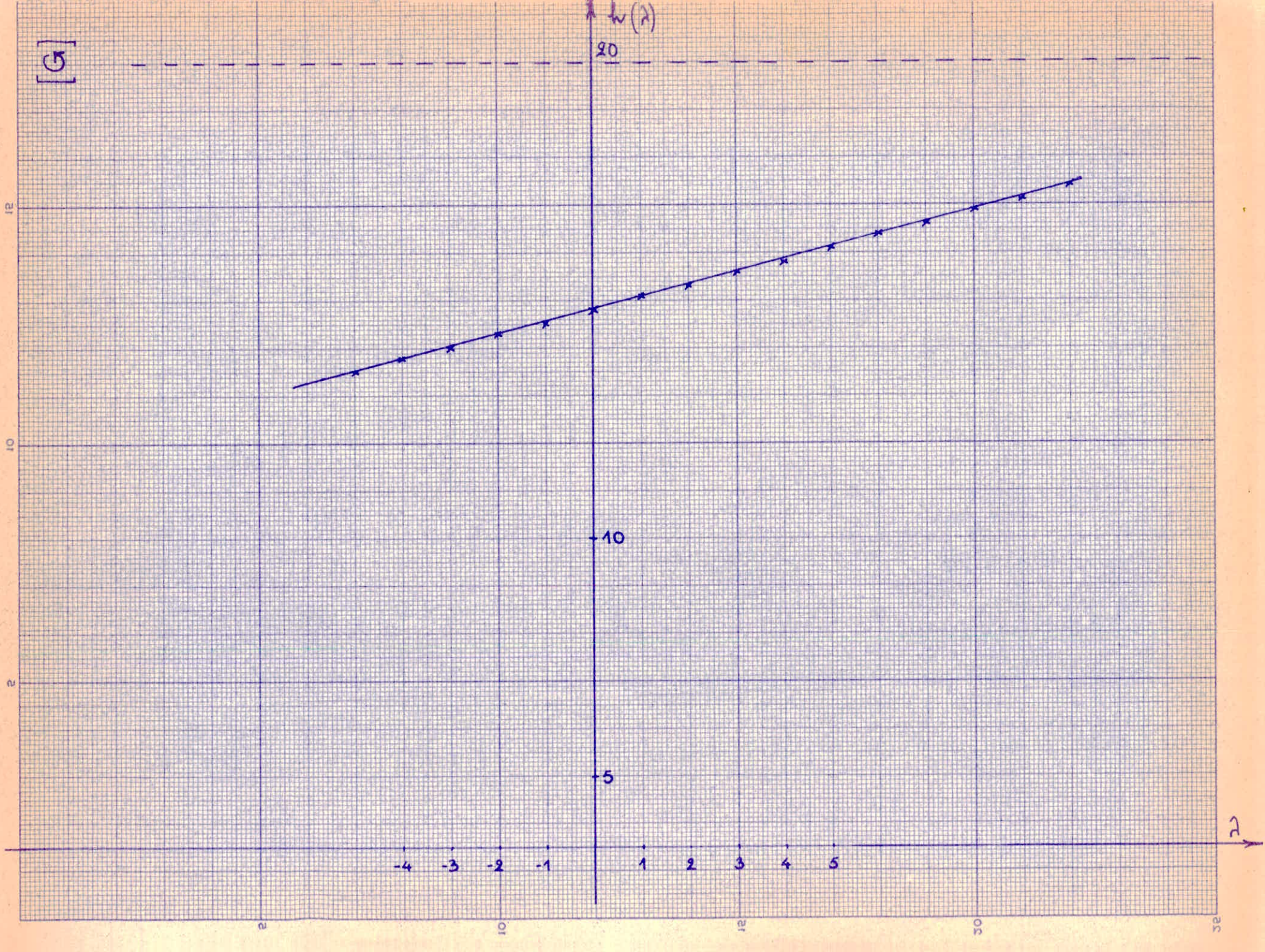
Nous posons successivement $x_i = \alpha_i$ et $x_i = \frac{\alpha_i}{\sum_{i=1}^{4} \alpha_i} b_2$ $(i = 1, \dots 4)$

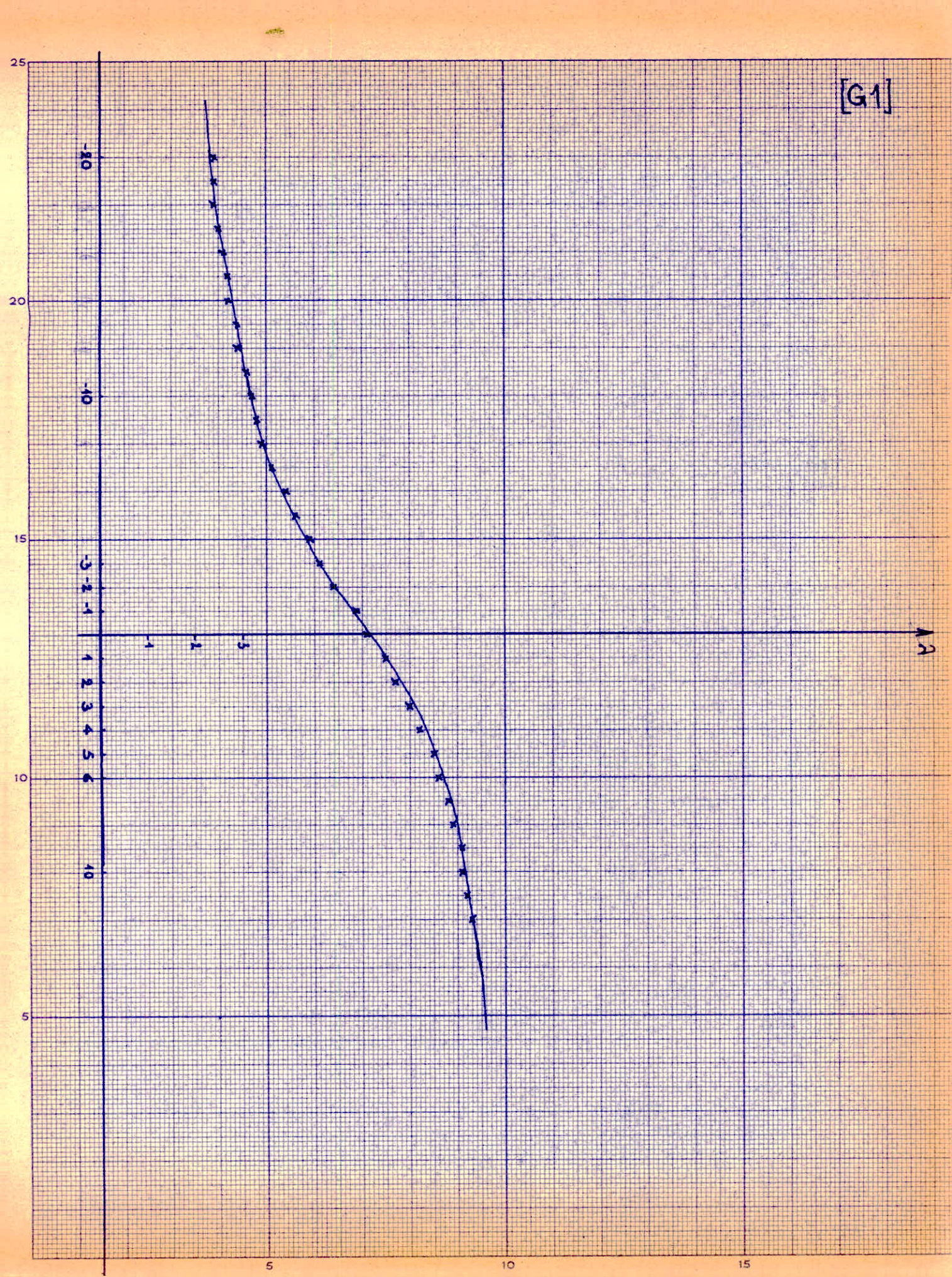
Nous résolvons le système en λ et μ après sélectionné parmi les contraintes liant les mêmes paramètres celles dont le second membre est le plus petit.

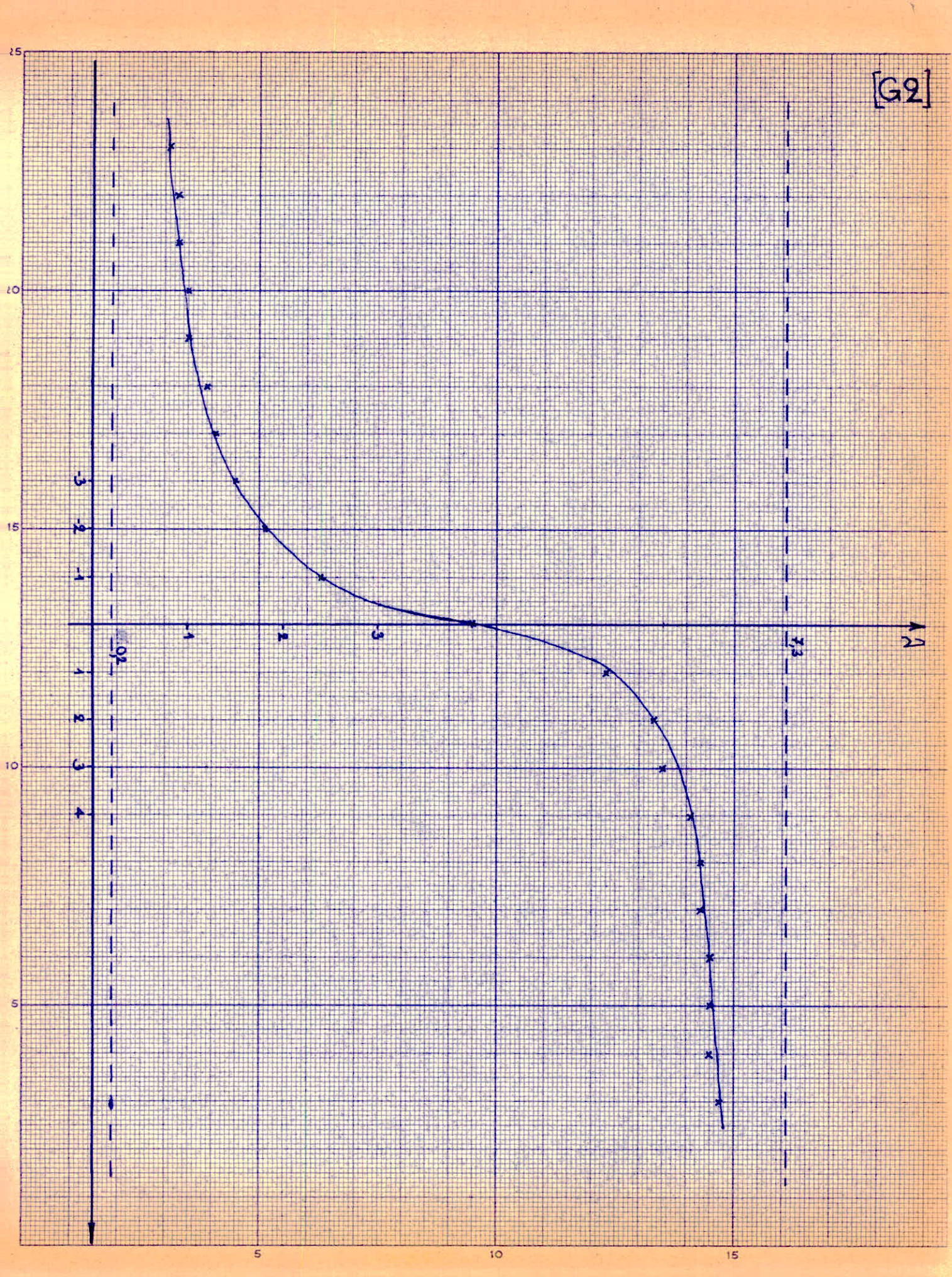
Nous atteignons l'optimum de $\mathcal{P}_1$ après une interpolation alors que l'optimum de $\mathcal{P}_2$ est fourni directement pour la valeur λ solution du 2ème système en λ, μ.

Le test d'arrêt de l'interpolation porte sur la précision de $h(\lambda)$: on décide que l'on est à l'optimum lorsque $|h(\lambda) - b_2| = \Delta$ Δ étant le pas du programme dynamique.

Notons que les systèmes de contraintes de $\mathcal{P}_1$ et $\mathcal{P}_2$ sont


[5]
$L_w(\lambda)$
20
10
5
λ
-4
-3
-2
-1
1
2
3
4
5


[G1]
25
20
15
10
5
-20
-10
-3 -2 -1
1 2 3 4 5 6
10
-1
-2
-3
5
10
15


[G2]
25
20
15
10
5
5
10
15
x
y
x=3
y=0

des systèmes en fait à variables séparées.

$$\left.\begin{aligned} x_2 + x_4 &= 4 \\ x_1 + x_2 + x_3 + x_4 &= 10 \end{aligned}\right\} \text{ est identique à } \begin{cases} x_2 + x_4 = 4 \\ x_1 + x_3 = 6 \end{cases}$$

En bornant les variables par α_i et β_i $h(\lambda)$ ne peut plus prendre que des valeurs de l'intervalle $\left[\sum_{i=1}^{N} \alpha_i , \sum_{i=1}^{N} \beta_i\right]$. Cette restriction fait que son graphe présente une moins forte courbure. On peut conserver l'interpolation hyperbolique, sur l'intervalle de définition de la fonction $h(\lambda)$; pour les autres valeurs de λ la fonction garde une valeur constante qui est soit $\sum_{i=1}^{N} \alpha_i$, soit $\sum_{i=1}^{N} \beta_i$.

3ème exemple

$$\text{Min} \quad \frac{1,2}{x_1} + \frac{2,4}{x_2} + \frac{2,88}{x_3} + \frac{0,24}{x_4}$$

$$\text{sous les contraintes} \quad \begin{cases} x_1 + x_2 + 2x_3 + x_4 = 8,2 & (1) \\ x_1 + x_2 + x_3 + x_4 = 7,3 & (2) \\ x_j > 0 \quad (j = 1, \dots, 4) \end{cases}$$

On associe le paramètre λ à l'équation (1). La fonction $x_1(\lambda) + x_2(\lambda) + 2x_3(\lambda) + x_4(\lambda)$ est du même genre que celle de l'exemple précédent. On conserve la même technique d'approximation de la valeur optimale de λ ; ici encore, il suffit d'"interpoler" une fois.

Généralisation à M contraintes.

Le problème Minimiser $\sum_{i=1}^{N} g_i(x_i)$

dans le domaine $\begin{cases} \sum_{i=1}^{N} a_{ji} x_i = b_j & (j = 1, \dots, M) \\ \alpha_i \leq x_i \leq \beta_i & (i = 1, \dots, N) \end{cases}$

est transformé en un problème unidimensionnel

Minimiser $\sum_{i=1}^{N} g_i(x_i) - \sum_{j=1}^{M-1} \lambda_j \left(\sum_{i=1}^{N} a_{ji} x_i \right)$

sous les contraintes $\begin{cases} a_{M1} x_1 + \dots + a_{MN} x_N = b_M \\ \alpha_i \leq x_i \leq \beta_i & (i = 1, \dots, N) \end{cases}$

Les courbes $h(\lambda)$ sont devenues $h(\lambda_1, \dots, \lambda_{M-1})$. Nous avons remarqué, à travers nos tentatives, que la résolution du système en λ_j $(j = 1, \dots, M)$, obtenu en égalant à zéro les dérivées partielles de la fonction :

$$\sum_{i=1}^{N} g_i(x_i) - \sum_{j=1}^{M} \lambda_j \left(\sum_{i=1}^{N} a_{ji} x_i \right)$$

par rapport à chacune des variables x_i $(i = 1, \dots, N)$, et en posant ensuite $x_i = \dfrac{\alpha_i}{\sum_{i=1}^{N} \alpha_i} b_i$, donne des valeurs des paramètres qui se rapprochent assez des valeurs optimales.

Nous choisissons une interpolation de type hyperbolique qui préjuge donc de la forme des courbes $h(\lambda_1, \dots, \lambda_{M-1})$.

Nous procédons alors de la façon suivante :

soient $\left(\lambda_1^{(0)}, \dots, \lambda_{M-1}^{(0)}, \lambda_M^{(0)}\right)$ solution du système quand $x_i = \alpha_i$

$\left(\lambda_1^{(1)}, \dots, \lambda_{M-1}^{(1)}, \lambda_M^{(1)}\right)$ solution du système quand $x_i = \frac{\alpha_i}{\sum_{i=1}^{N} \alpha_i} b_M$

Nous conserverons les $m-1$ uples $\left(\lambda_1^{(0)}, \dots, \lambda_{M-1}^{(0)}\right)$ et $\left(\lambda_1^{(1)}, \dots, \lambda_1^{(M-1)}\right)$ car λ_M est le paramètre associé à la contrainte de liaison et il n'intervient pas dans la recherche de l'optimum par programmation dynamique. A partir de $\left(\lambda_1^{(0)}, \dots, \lambda_{M-1}^{(0)}\right)$ et $\left(\lambda_1^{(1)}, \dots, \lambda_1^{(M-1)}\right)$ après recherche de l'optimum par programmation dynamique et interpolation de la 1ère contrainte par rapport à λ_1 nous obtenons λ'_1. Nous calculons un nouvel optimum pour $\left(\lambda'_1, \lambda_2^{(0)}, \lambda_3^{(1)}, \dots, \lambda_{M-1}^{(1)}\right)$ et pour $\left(\lambda'_1, \lambda_2^{(1)}, \lambda_3^{(1)}, \dots, \lambda_{M-1}^{(1)}\right)$. L'interpolation fournit λ'_2. On répète les mêmes opérations à partir de $\left(\lambda'_1, \lambda'_2, \lambda_3^{(0)}, \lambda_4^{(1)}, \dots, \lambda_{M-1}^{(1)}\right)$ et $\left(\lambda'_1, \lambda'_2, \lambda_3^{(1)}, \lambda_4^{(1)}, \dots, \lambda_{M-1}^{(1)}\right)$.

Si l'ensemble des valeurs $\left(\lambda'_1, \lambda'_2, \dots, \lambda'_{M-1}\right)$ n'est pas satisfaisant nous répétons les opérations.

Application de la programmation dynamique à la répartition des tolérances

Les intervalles de tolérance sont compris entre 1 micron et 1 millimètre. En prenant un pas de 5 microns il nous faut au maximum 200 incréments.

S'il y a au plus 150 équations à 150 inconnues nous avons déjà une mesure de la place en mémoire nécessaire. La table des valeurs des variable x_i exige la réservation de $150 \times 200 = 30\,000$ mots-mémoire, en faisant de cette table une table à valeurs entières.

Si nous décidons d'appliquer la méthode A le bloc des fonctions auxiliaires nécessite 2×400^{150} mots mémoire nombre qui dépasse la capacité de l'ordinateur IBM 1130.

Dans la méthode B les tableaux des fonctions auxiliaires (valeurs réelles) mobiliseront $2 \times 400 = 800$ mots mémoire. Cette méthode pourrait donc être programmable

Nous avons pris comme exemple d'application la programmation d'un problème où les contraintes sont à variables séparées.

En entrée le COMMON est organisé de la façon suivante :

	K, LI, LJ, NS, NT
150 mots	ITECΦ
150 mots	ITR
150 mots	C
150 mots	MARQF
150 mots	MARQR

avec :

K = variable de contrôle du fichier disque

NS = nombre de surface

NT = nombre de côtes fabrication

MARQF : marqueur qui indique les ITF non réductibles

MARQR : marqueur qui indique les équations de contraintes

C : vecteur condensé qui contient les coefficients de la matrice A ligne par ligne.

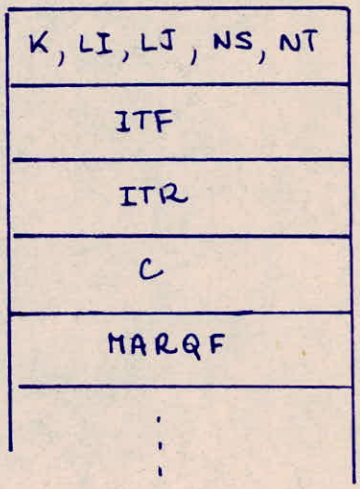
Dans MARQF et C il y a NT valeurs, dans MARQR NS; ce sont des valeurs binaires

MARQF (I) = { 0 : l'ITF peut être réduit
1 : l'ITF ne doit pas être réduit (on retranche sa valeur à l'ITR de la contrainte dans laquelle il figure)

$$MARQR(I) = \begin{cases} 0 & \text{Cette équation n'est pas une contrainte} \\ 1 & \text{l'équation est à traiter} \end{cases}$$

$$C(I) = \begin{cases} 0 & \text{l'ITF ne figure pas dans la contrainte} \\ \pm 1 & \text{l'ITF est dans la contrainte.} \end{cases}$$

En sortie nous aurons



L'organigramme [C] permet la résolution du problème REDUC est le programme "dynamique".

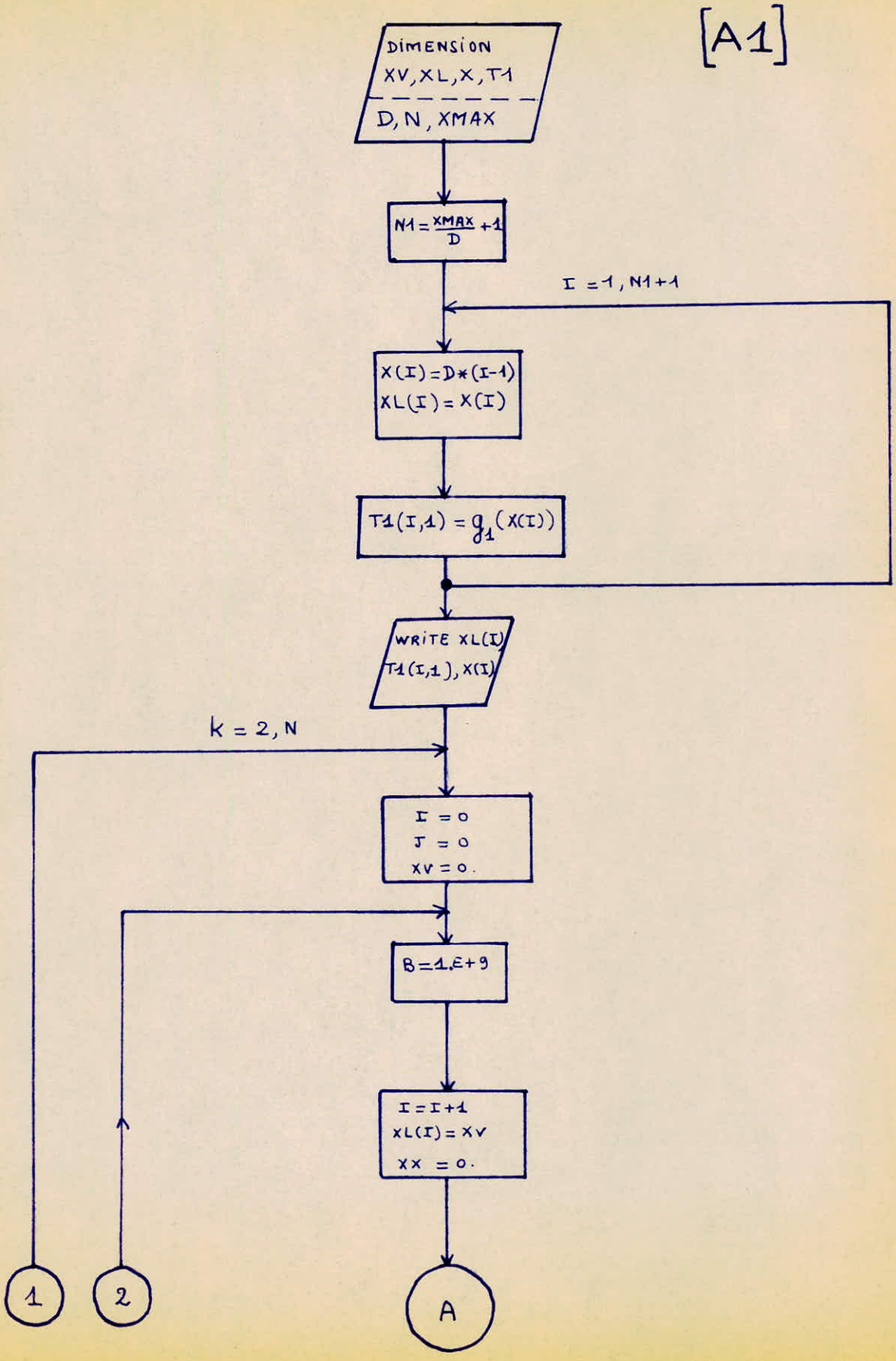
Nous venons de voir que l'application de la programmation dynamique à l'optimisation financière des tolérances est très limitée.

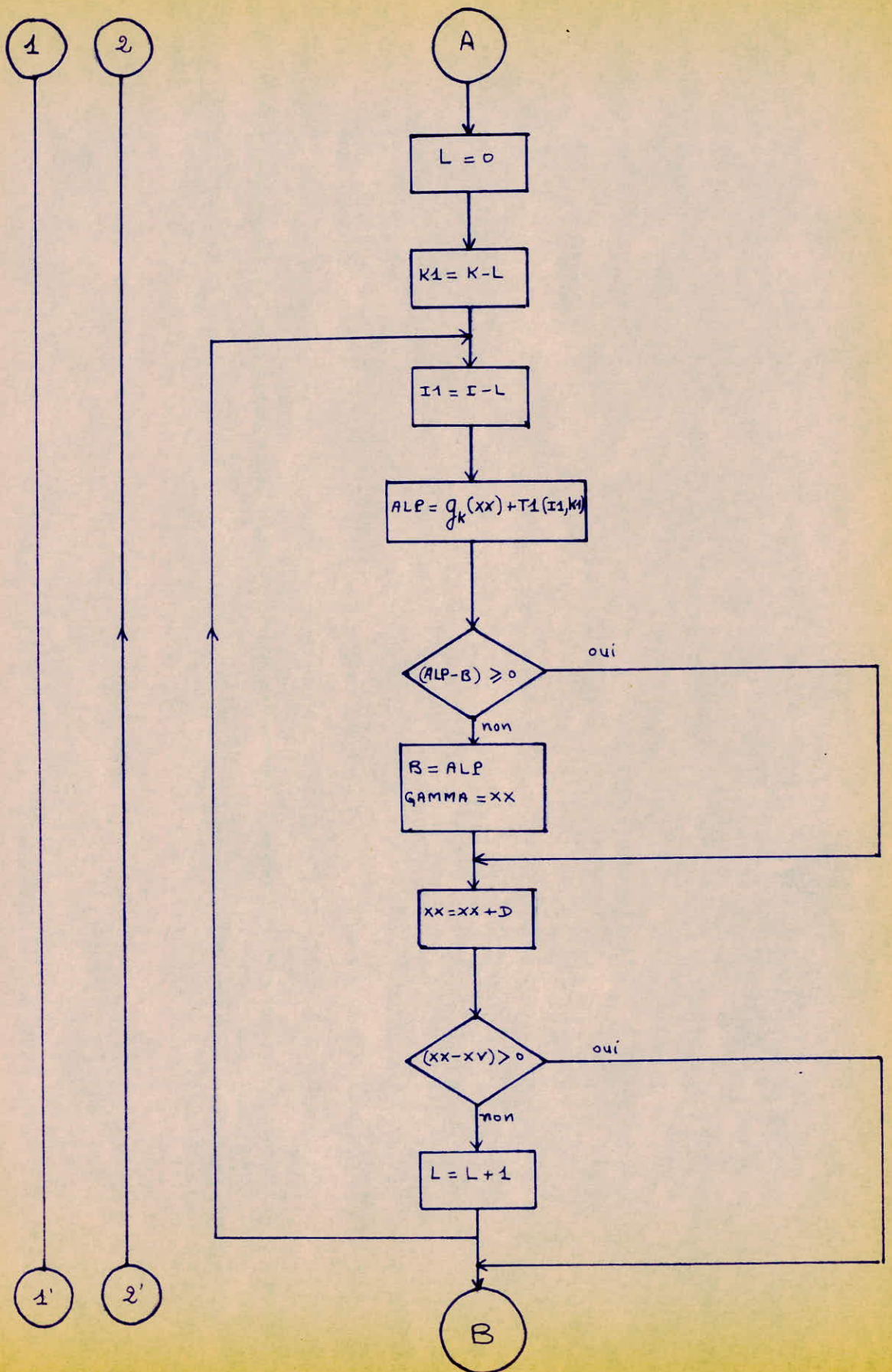
Nous pouvons actuellement traiter les cas de contraintes à variables séparées ou de contraintes à variables séparées avec une équation de liaison à coefficients tous égaux à 1.

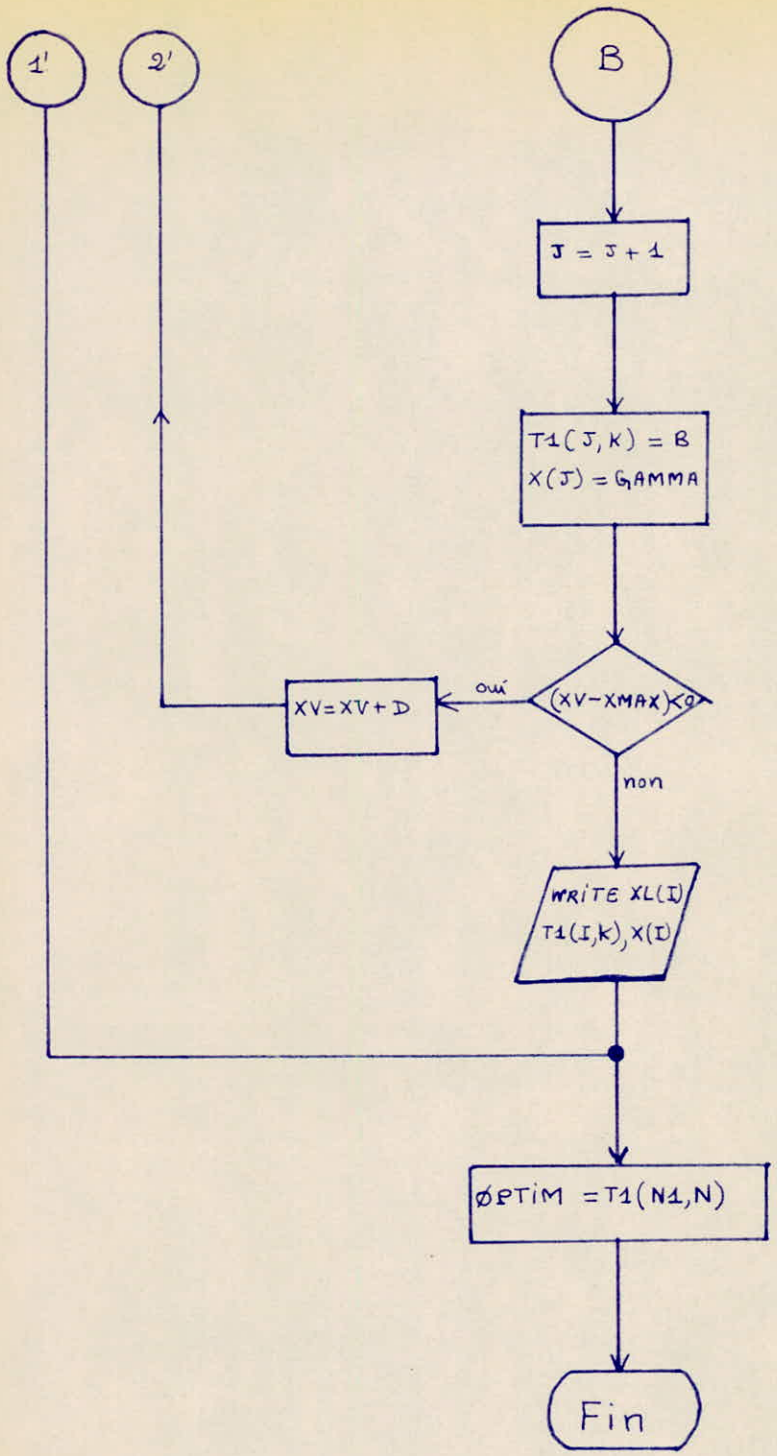
La méthode B qui est la plus facilement applicable ne pourra être employée que lorsque le problème de l'interpolation aura été résolu.

Organigrammes

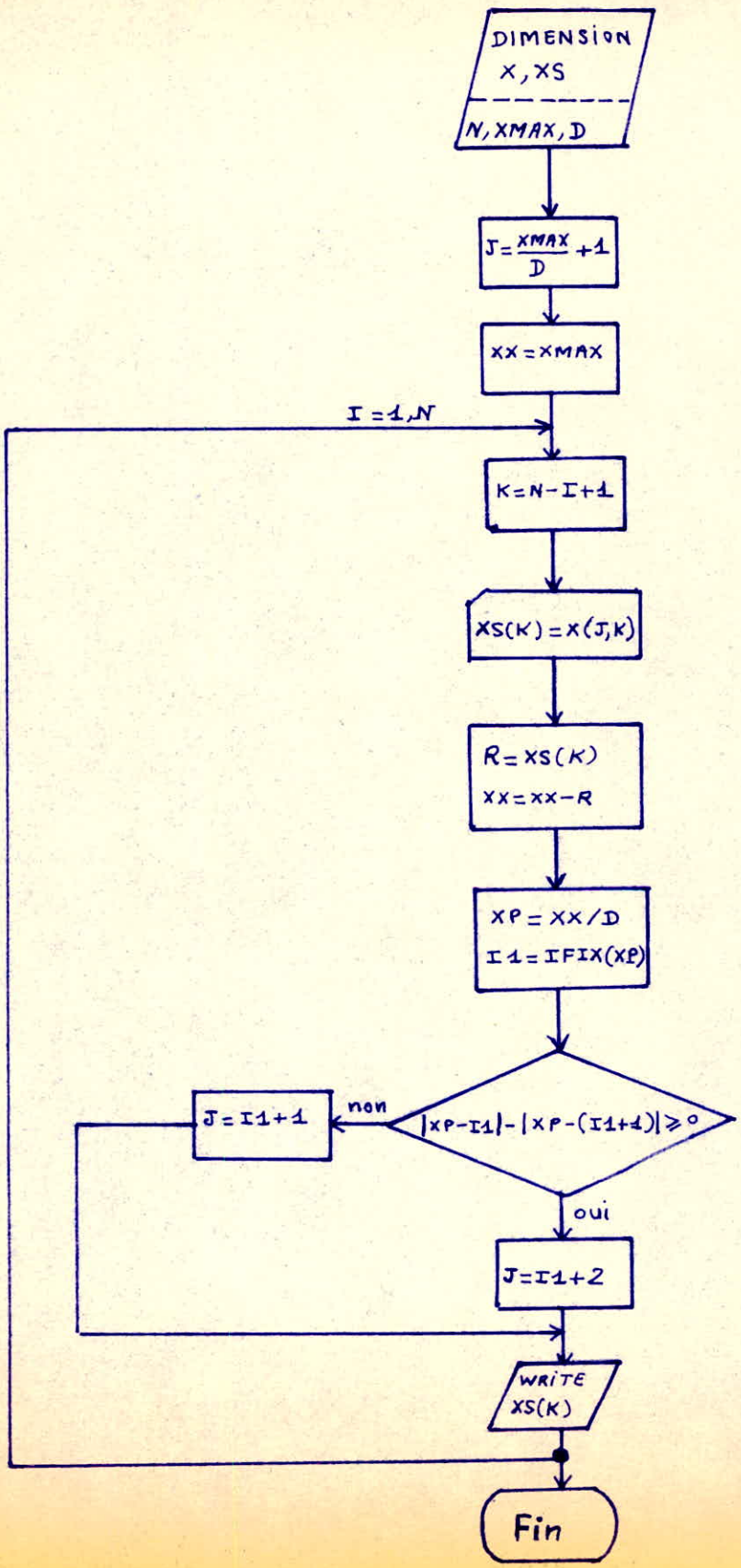
[A1]


DIMENSION
XV, XL, X, T1
D, N, XMAX
N1 = XMAX/D + 1
I = 1, N1+1
X(I) = D*(I-1)
XL(I) = X(I)
T1(I,1) = g1(X(I))
WRITE XL(I)
T1(I,1), X(I)
k = 2, N
I = 0
J = 0
XV = 0.
B = 1.E+9
I = I+1
XL(I) = XV
XX = 0.
1
2
A

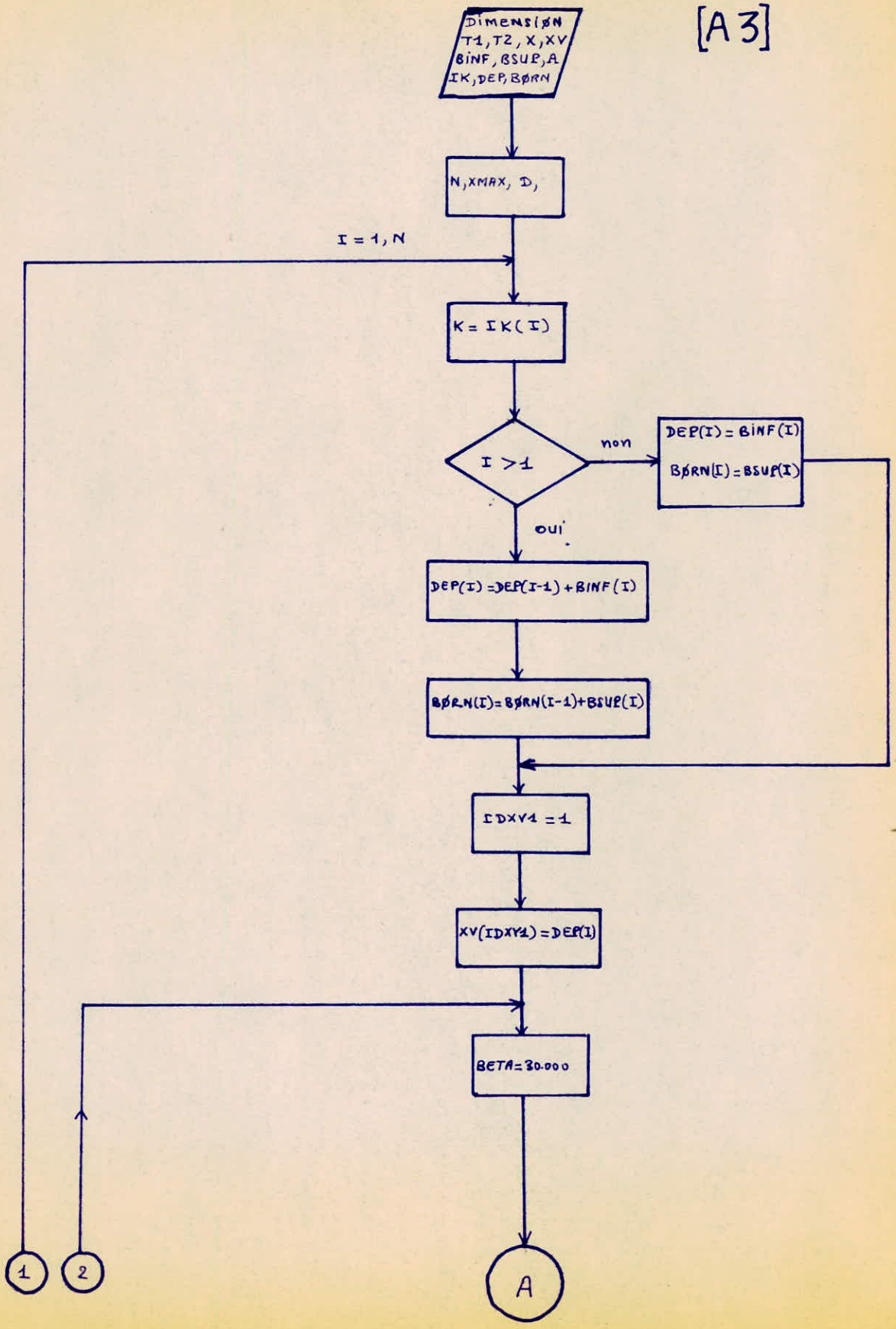

1
2
A
L = 0
K1 = K-L
I1 = I-L
ALP = g_k(XX) + T1(I1,K1)
(ALP-B) ≥ 0
oui
non
B = ALP
GAMMA = XX
XX = XX + D
(XX-XV) > 0
oui
non
L = L + 1
1'
2'
B

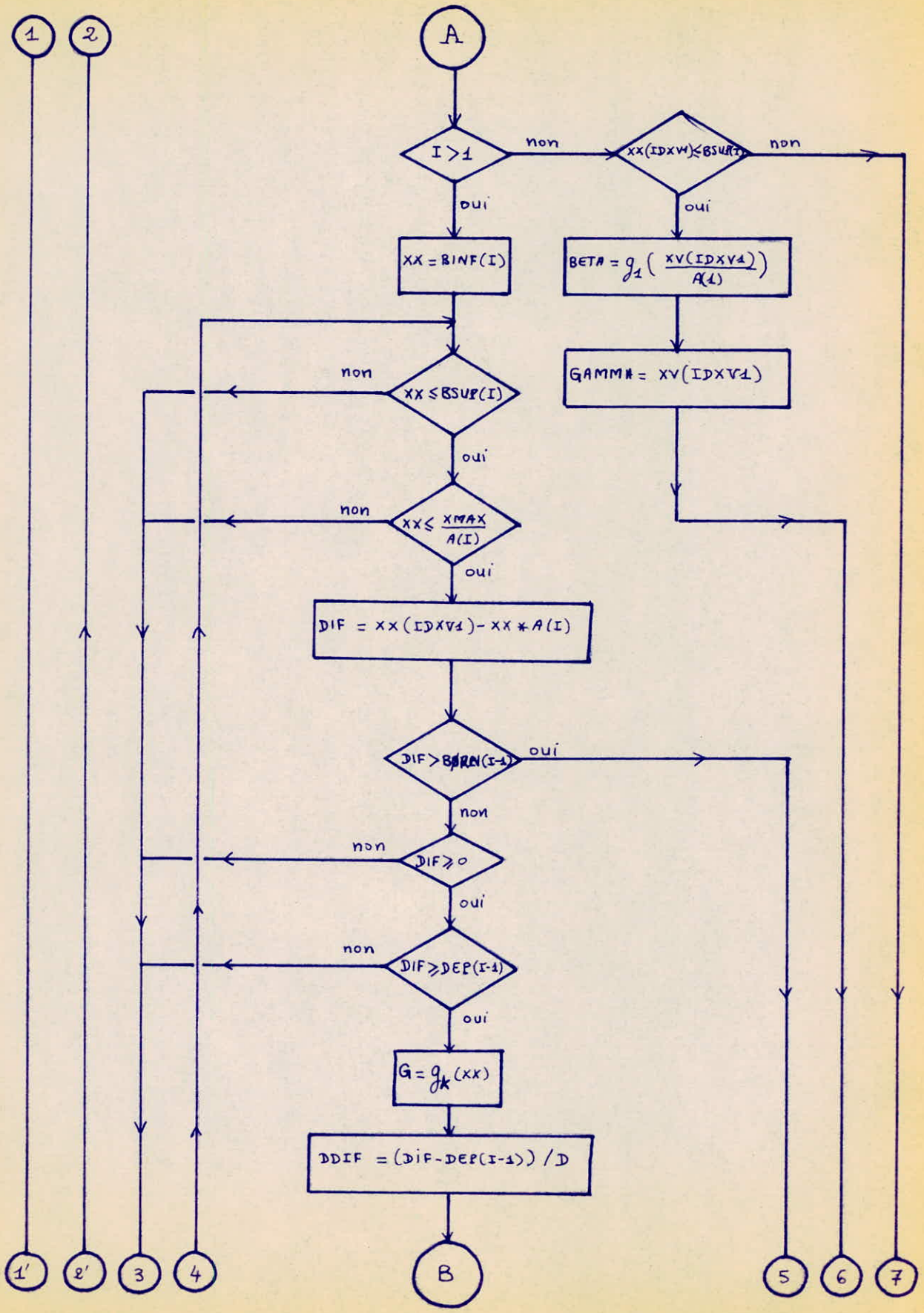

1'
2'
B
J = J + 1
T1(J,K) = B
X(J) = GAMMA
(XV-XMAX)<0
oui
XV=XV+D
non
WRITE XL(I)
T1(I,K), X(I)
OPTIM =T1(N1,N)
Fin

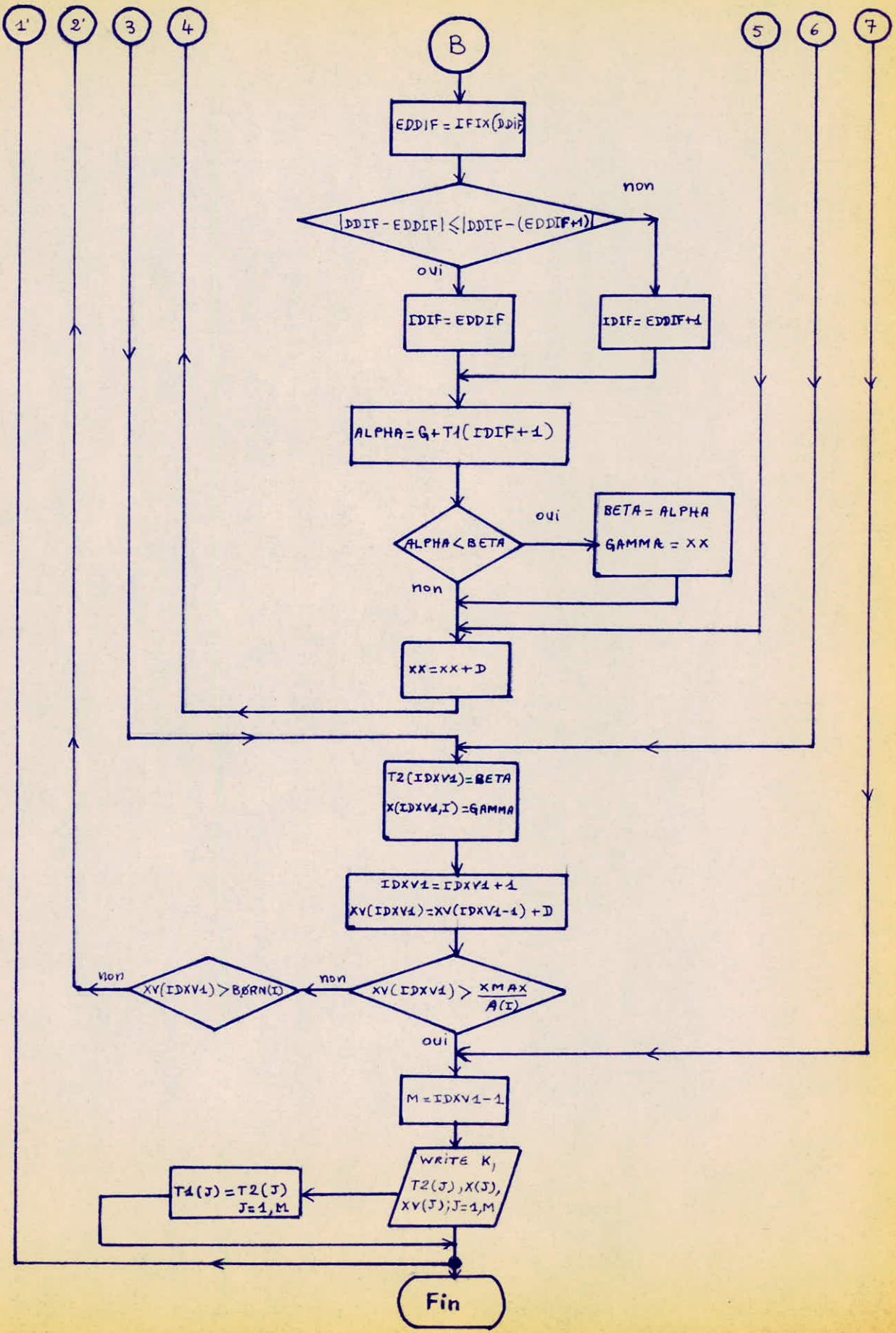
[A2]


DIMENSION
X, XS
N, XMAX, D
J = XMAX/D + 1
XX = XMAX
I = 1, N
K = N - I + 1
XS(K) = X(J,K)
R = XS(K)
XX = XX - R
XP = XX / D
I1 = IFIX(XP)
|XP-I1| - |XP-(I1+1)| ≥ 0
non
J = I1 + 1
oui
J = I1 + 2
WRITE
XS(K)
Fin

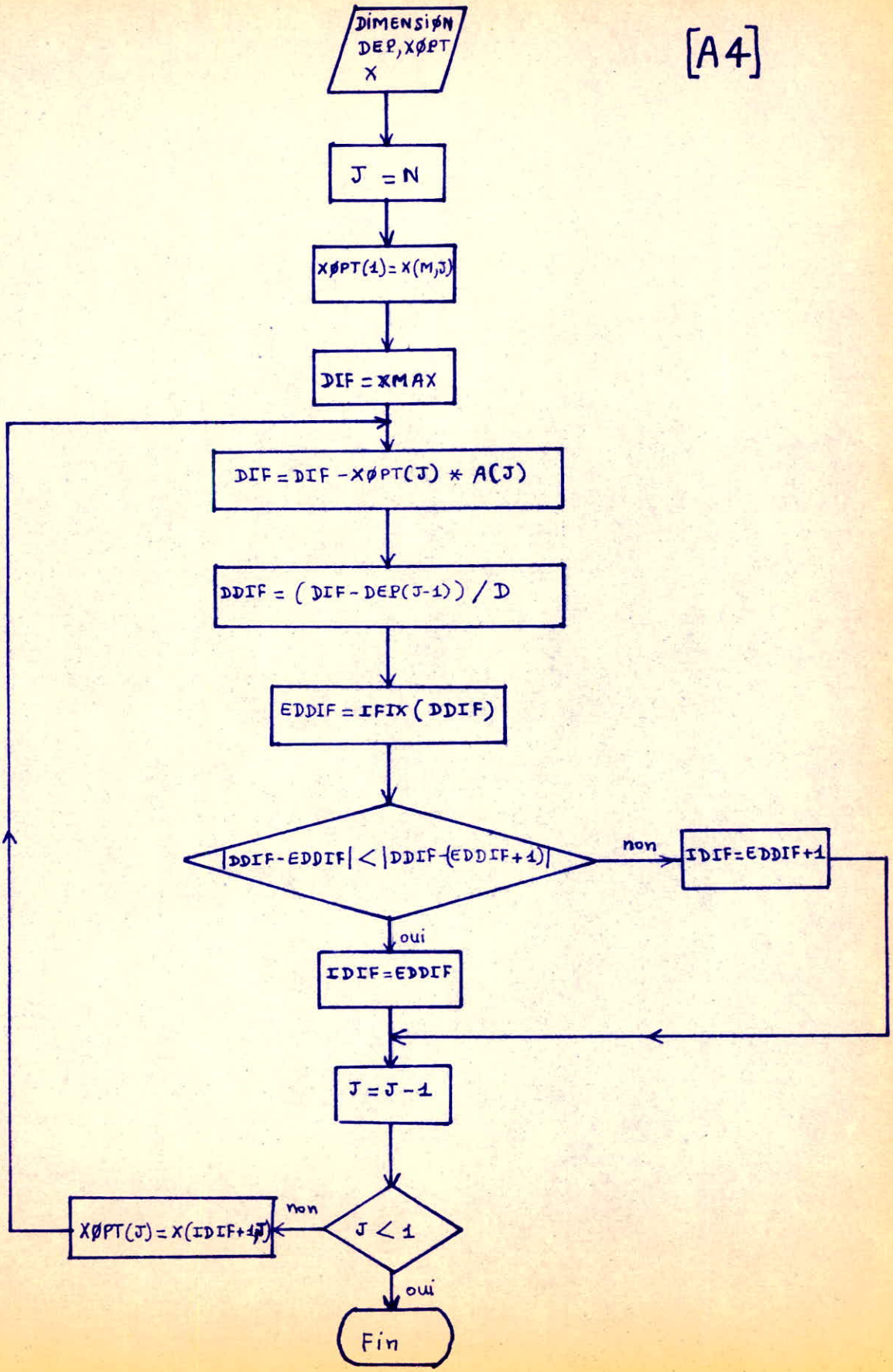
[A3]


DIMENSION T1, T2, X, XV BINF, BSUP, A IK, DEP, BØRN
N, XMAX, D,
I = 1, N
K = IK(I)
I > 1
non
DEP(I) = BINF(I)
BØRN(I) = BSUP(I)
oui
DEP(I) = DEP(I-1) + BINF(I)
BØRN(I) = BØRN(I-1) + BSUP(I)
IDXV1 = 1
XV(IDXV1) = DEP(I)
BETA = 30.000
1
2
A

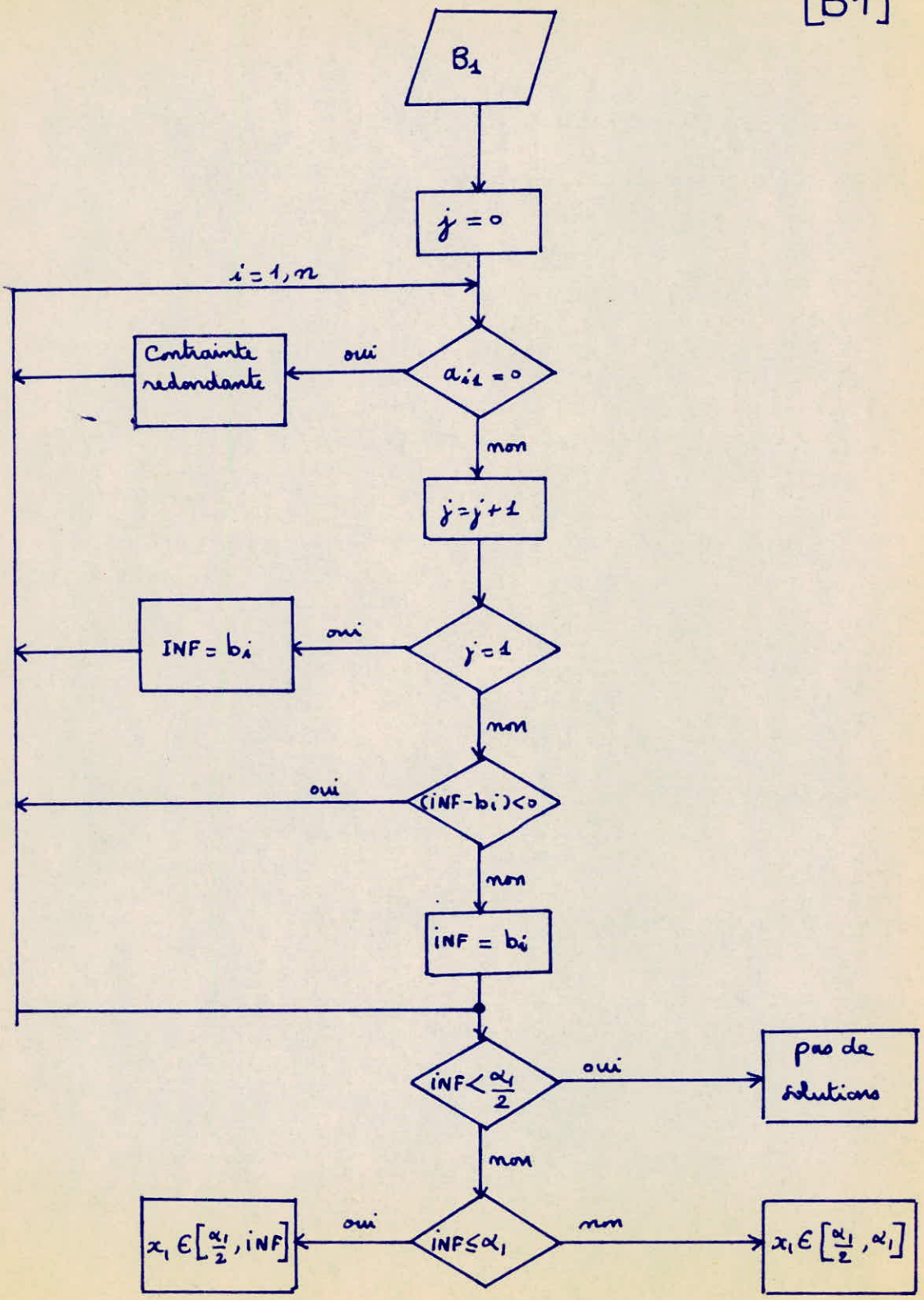

1
2
A
I > 1
non
oui
XX = BINF(I)
non
XX ≤ BSUP(I)
oui
non
XX ≤ XMAX / A(I)
oui
DIF = XX(IDXV1) - XX * A(I)
DIF > BSUP(I-1)
oui
non
non
DIF ≥ 0
oui
non
DIF ≥ DEP(I-1)
oui
G = gk(XX)
DDIF = (DIF - DEP(I-1)) / D
B
XX(IDXV1) ≤ BSUP(I)
non
BETA = g1(XV(IDXV1) / A(1))
GAMMA = XV(IDXV1)
1'
2'
3
4
5
6
7


1'
2'
3
4
B
5
6
7
EDDIF = IFIX(DDIF)
|DDIF - EDDIF| ≤ |DDIF - (EDDIF+1)|
non
oui
IDIF = EDDIF
IDIF = EDDIF+1
ALPHA = G + T1(IDIF+1)
ALPHA < BETA
oui
BETA = ALPHA
GAMMA = XX
non
XX = XX + D
T2(IDXV1) = BETA
X(IDXV1,I) = GAMMA
IDXV1 = IDXV1 + 1
XV(IDXV1) = XV(IDXV1-1) + D
XV(IDXV1) > XMAX / A(I)
non
XV(IDXV1) > BORN(I)
non
oui
M = IDXV1 - 1
WRITE K, T2(J), X(J), XV(J); J=1,M
T1(J) = T2(J) J=1,M
Fin

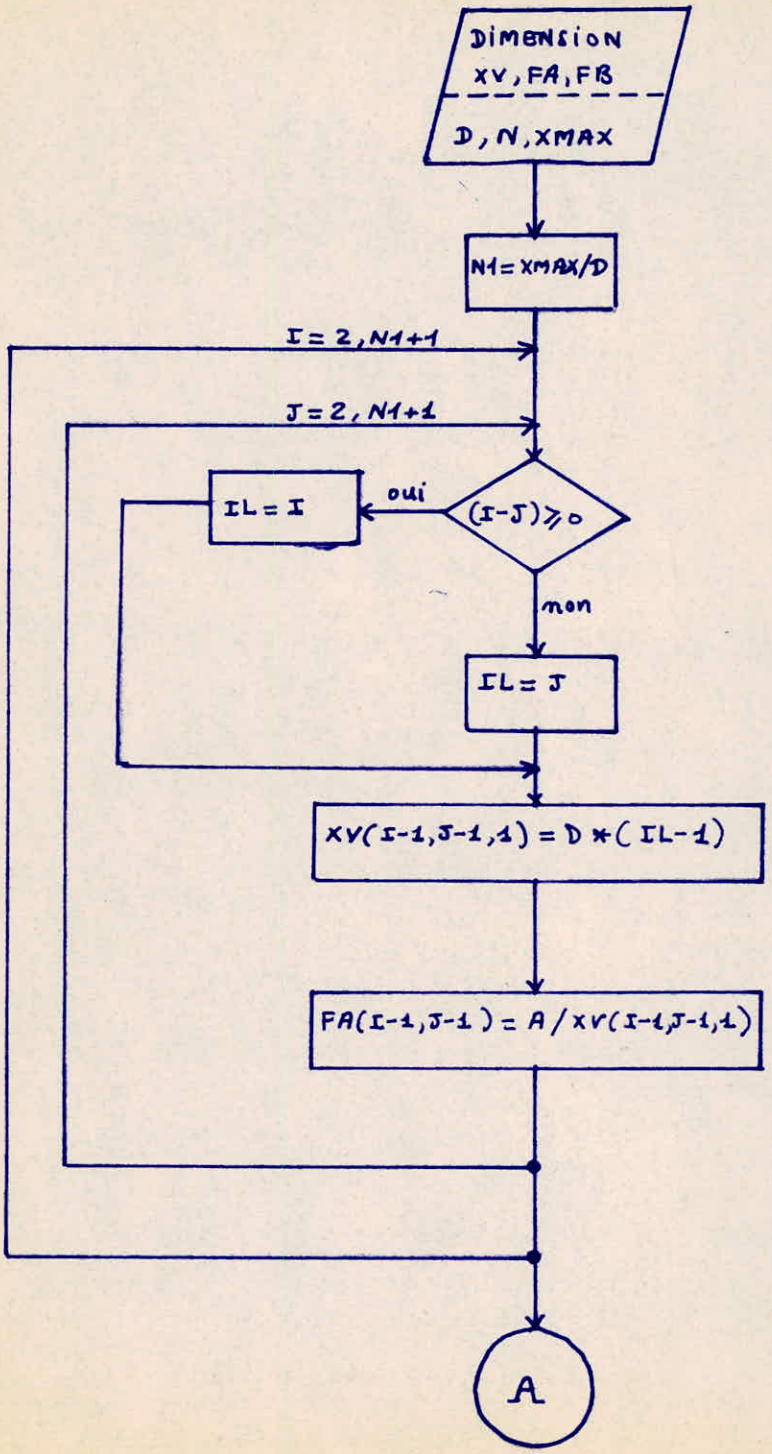
[A4]

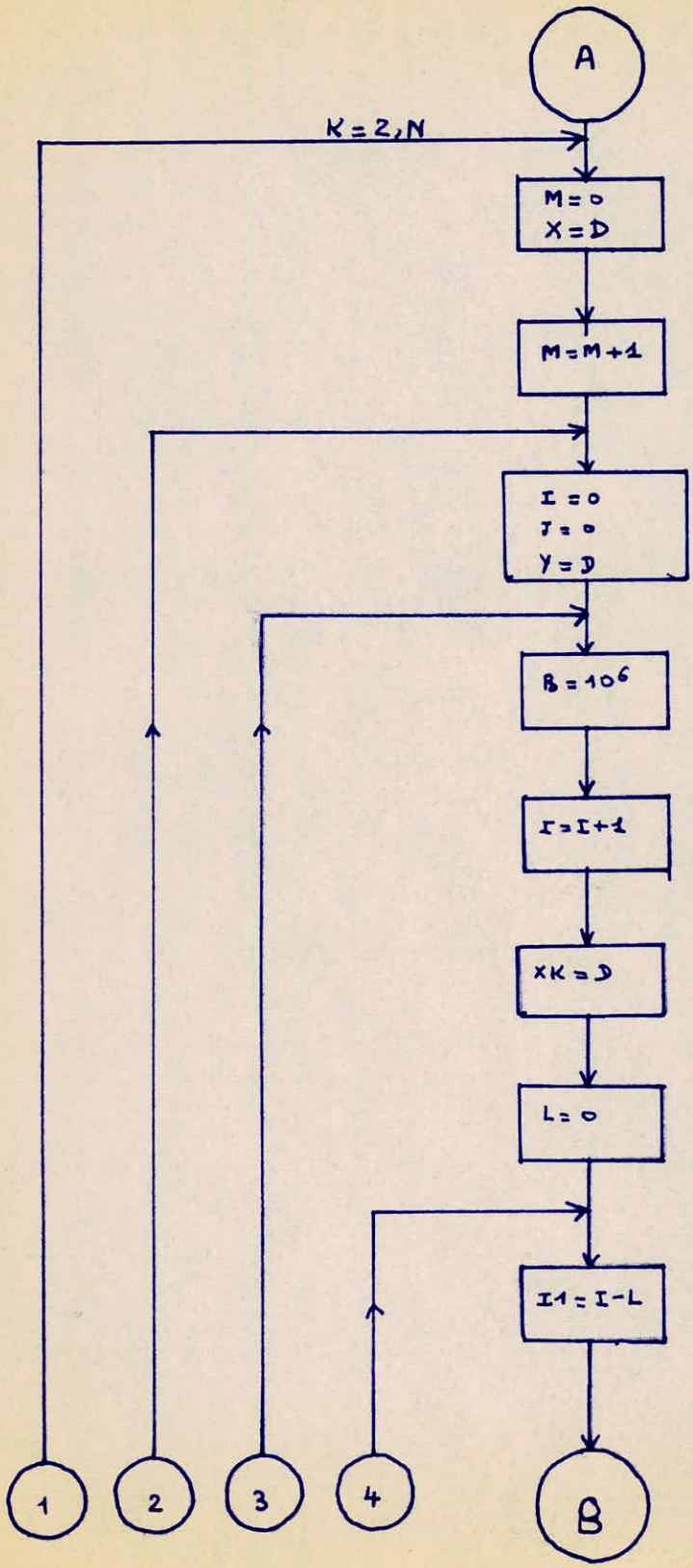

DIMENSION DEP, XOPT X
J = N
XOPT(1) = X(M,J)
DIF = XMAX
DIF = DIF - XOPT(J) * A(J)
DDIF = (DIF - DEP(J-1)) / D
EDDIF = IFIX (DDIF)
|DDIF - EDDIF| < |DDIF - (EDDIF+1)|
non
IDIF = EDDIF+1
oui
IDIF = EDDIF
J = J - 1
J < 1
non
XOPT(J) = X(IDIF+1,J)
oui
Fin

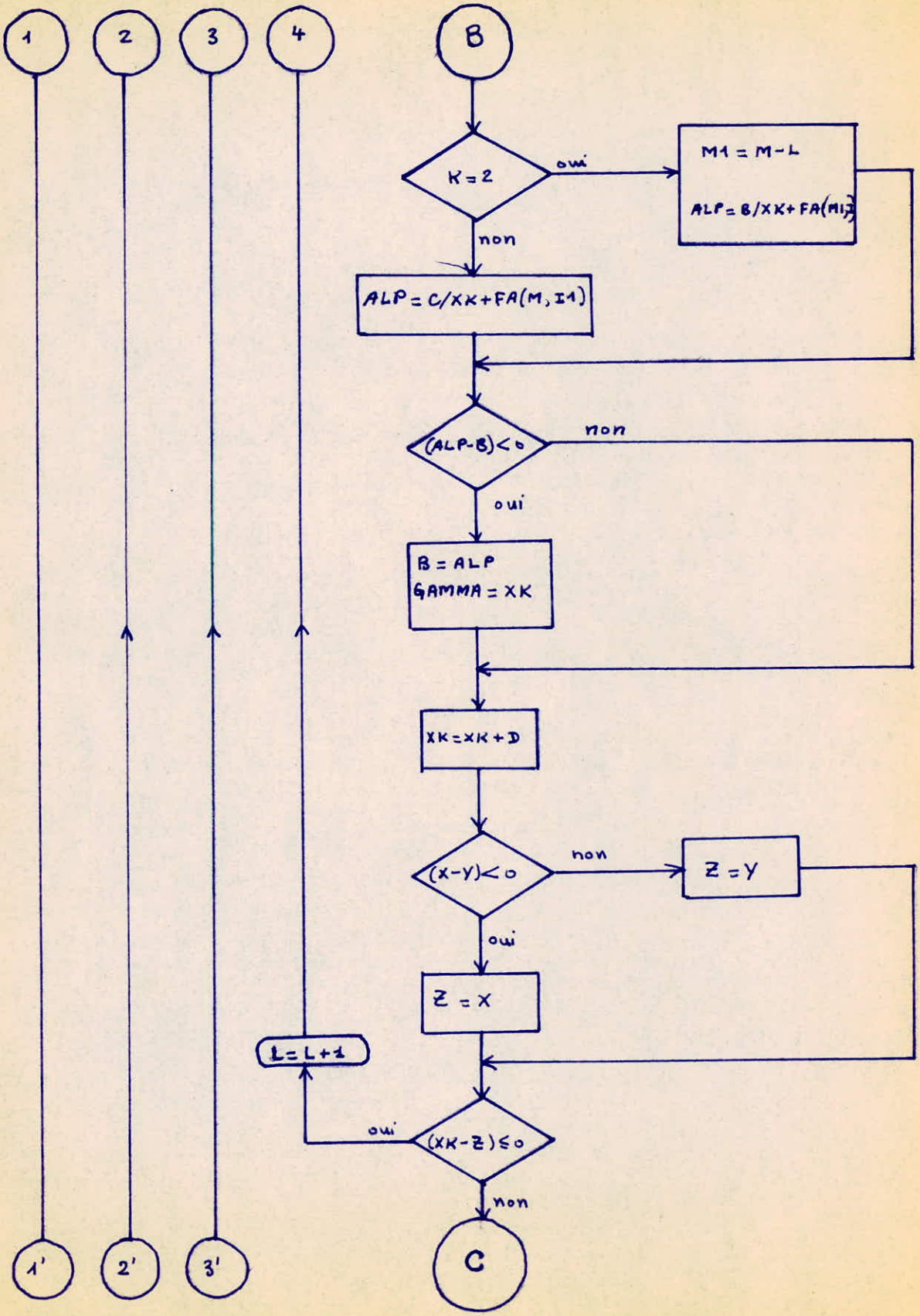
[B1]

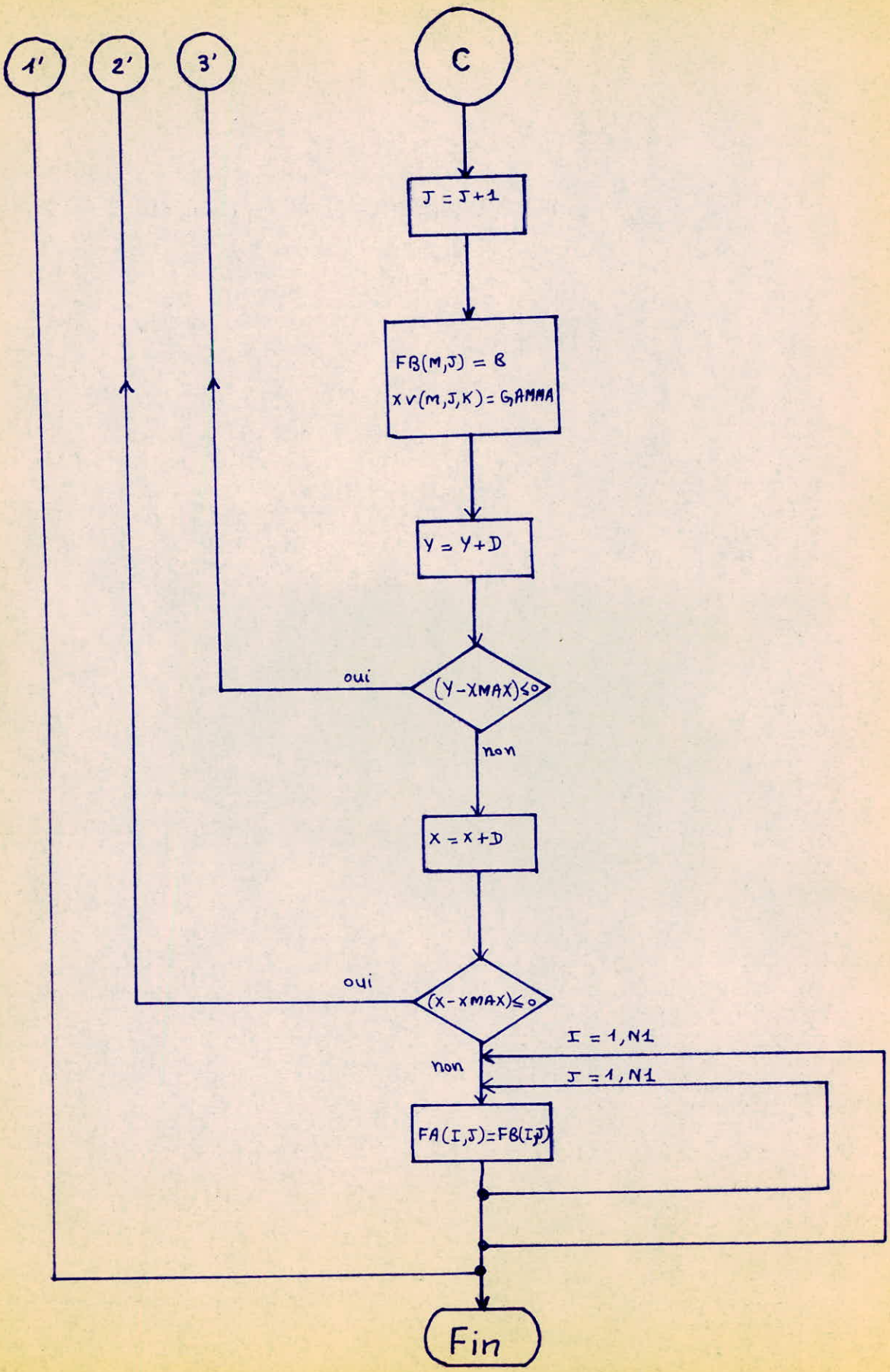

B1
j = 0
i = 1, n
Contrainte redondante
oui
$a_{i1} = 0$
non
j = j + 1
INF = b_i
oui
j = 1
non
oui
(INF - b_i) < 0
non
INF = b_i
INF < $\frac{\alpha_1}{2}$
oui
pas de solutions
non
$x_1 \in \left[\frac{\alpha_1}{2}, INF\right]$
oui
INF ≤ α_1
non
$x_1 \in \left[\frac{\alpha_1}{2}, \alpha_1\right]$

[B2]

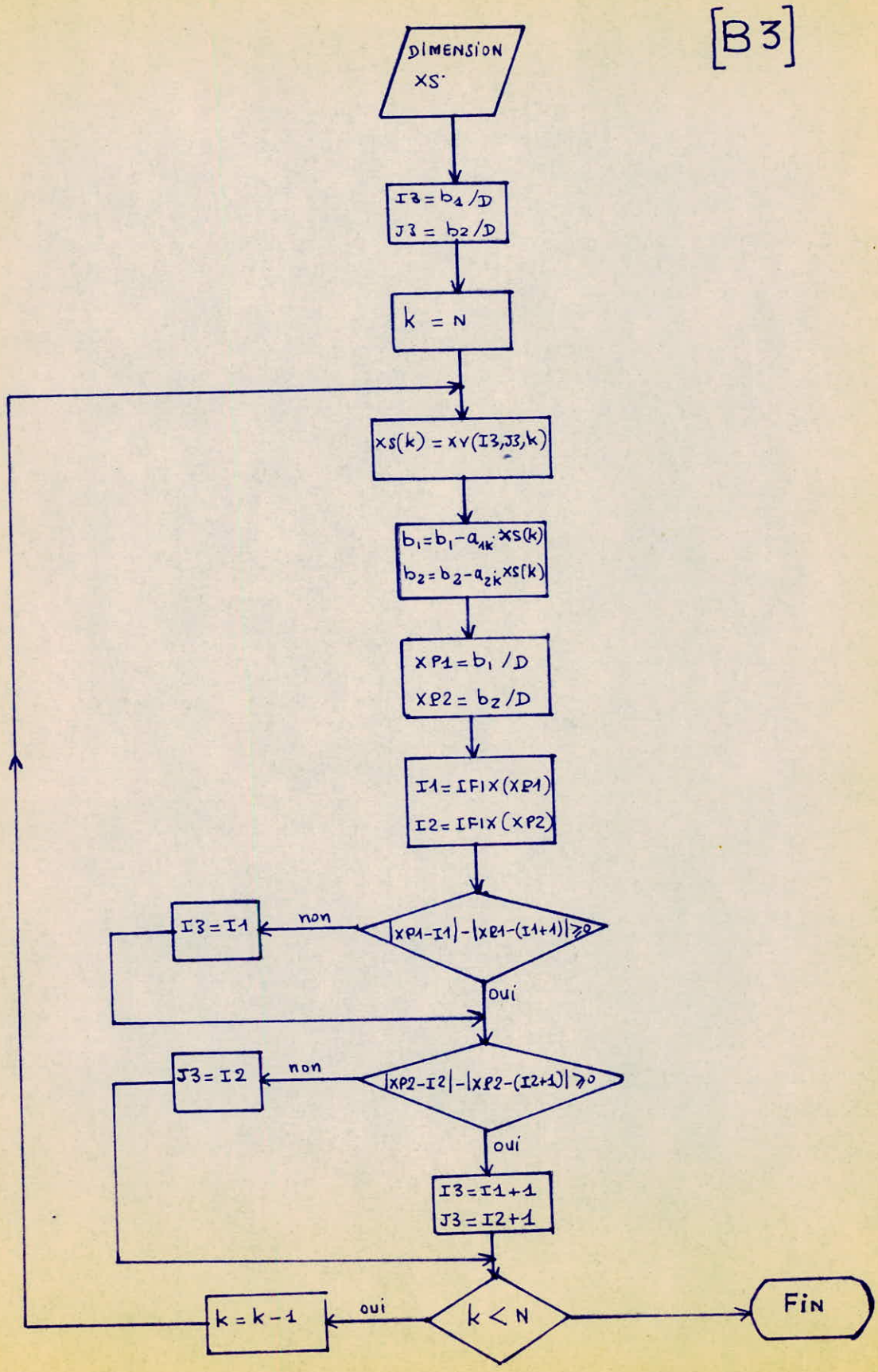

DIMENSION
XV, FA, FB
D, N, XMAX
N1 = XMAX/D
I = 2, N1+1
J = 2, N1+1
(I-J) ≥ 0
oui
IL = I
non
IL = J
XV(I-1, J-1, 1) = D * (IL-1)
FA(I-1, J-1) = A / XV(I-1, J-1, 1)
A


A
K=2,N
M=0
X=D
M=M+1
I=0
J=0
Y=D
B=10^6
I=I+1
XK=D
L=0
I1=I-L
1
2
3
4
B

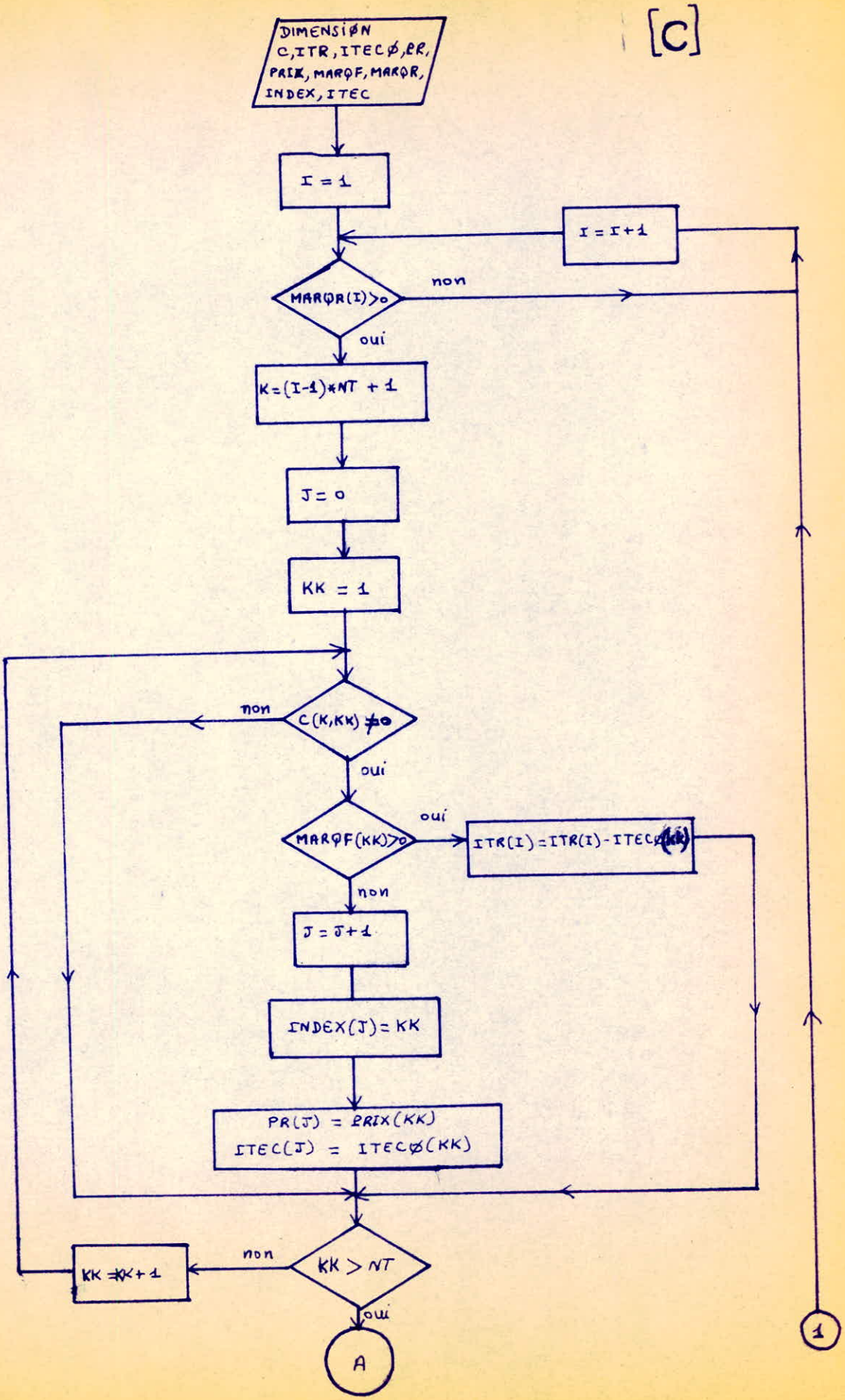

1
2
3
4
B
K = 2
oui
M1 = M - L
ALP = B/XK + FA(M1)
non
ALP = C/XK + FA(M, I1)
(ALP-B) < 0
non
oui
B = ALP
GAMMA = XK
XK = XK + D
(X-Y) < 0
non
Z = Y
oui
Z = X
L = L + 1
oui
(XK-Z) ≤ 0
non
1'
2'
3'
C

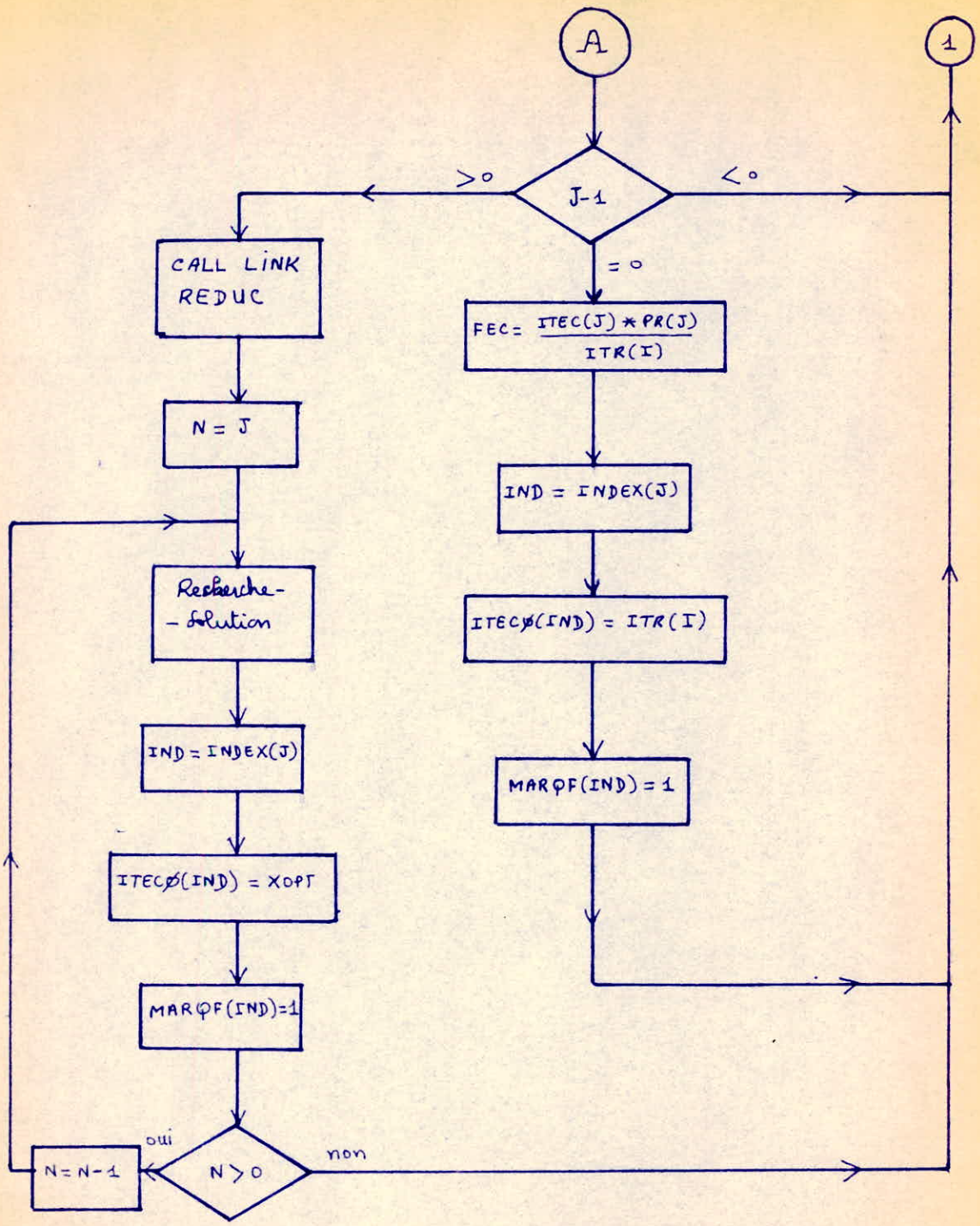

1'
2'
3'
C
J = J+1
FB(M,J) = B
XV(M,J,K) = GAMMA
Y = Y+D
oui
(Y-XMAX)≤0
non
X = X+D
oui
(X-XMAX)≤0
non
I = 1,N1
J = 1,N1
FA(I,J)=FB(I,J)
Fin

[B3]


DIMENSION XS
I3 = b1 / D
J3 = b2 / D
k = N
xs(k) = xv(I3,J3,k)
b1 = b1 - a1k · xs(k)
b2 = b2 - a2k · xs(k)
xP1 = b1 / D
xP2 = b2 / D
I1 = IFIX(xP1)
I2 = IFIX(xP2)
|xP1-I1| - |xP1-(I1+1)| ≥ 0
non
I3 = I1
oui
|xP2-I2| - |xP2-(I2+1)| ≥ 0
non
J3 = I2
oui
I3 = I1+1
J3 = I2+1
k < N
oui
k = k-1
FIN

[C]


DIMENSION C,ITR,ITECØ,PR, PRIX, MARQF, MARQR, INDEX, ITEC
I = 1
I = I+1
MARQR(I)>0
non
oui
K=(I-1)*NT + 1
J= 0
KK = 1
non
C(K,KK) ≠0
oui
MARQF(KK)>0
oui
ITR(I) = ITR(I) - ITECØ(KK)
non
J = J+1
INDEX(J) = KK
PR(J) = PRIX(KK)
ITEC(J) = ITECØ(KK)
KK = KK+1
non
KK > NT
oui
A
1


A
1
J-1
> 0
< 0
= 0
CALL LINK
REDUC
N = J
Recherche-
-solution
IND = INDEX(J)
ITECØ(IND) = XOPT
MARQF(IND)=1
N=N-1
oui
N > 0
non
FEC= ITEC(J) * PR(J) / ITR(I)
IND = INDEX(J)
ITECØ(IND) = ITR(I)
MARQF(IND) = 1

Bibliographie

1. R FAURE — Eléments de Recherche Opérationnelle.

2. P GORDAN — Introduction à la Programmation Dynamique.

3. R.E BELLMAN & S.E DREYFUS — La Programmation Dynamique et ses applications.

4. A KAUFMANN — La Programmation Dynamique (Gestion scientifique séquentielle).

5. S VAJDA — Leçons sur la Programmation Dynamique

6. Revue AFCET — Modèle de minimisation d'une fonction économique.

7. H.P KUNZI & W. KRELLE — La Programmation non linéaire.

8. M. BAISSAC — Application des méthodes de la Programmation Dynamique au problème de la répartition optimale d'une activité. (Thèse de 3ème cycle - Université de Grenoble)

9. COLLET - GUERRIER VERGNAUD — Travaux en cours de publication

10. BRUNAND - COLLET VERGNAUD — Méthodes (Cours IUT Génie Mécanique à Lyon)