Réalisation de fonctions booléennes avec l'opérateur majorité

François Lustman

N° D'ORDRE

THÈSES

Présentées à la Faculté des Sciences

de l'Université de Grenoble

pour obtenir le grade de

DOCTEUR - INGENIEUR

Par

François Lustman

INGENIEUR I. E. G.
LICENCIE ES SCIENCES

PREMIERE THESE

Réalisation de Fonctions Booléennes avec l'opérateur Majorité

DEUXIEME THESE

PROPOSITIONS DONNEES PAR LA FACULTE

Thèse soutenue le 1966 devant la commission d'examen :

Monsieur J. KUNTZMANN	Président
Monsieur B. VAUQUOIS Madame F. BERTRANDIAS	Examinateurs

FACULTE DES SCIENCES

LISTE DES PROFESSEURS

DOYENS HONORAIRES M. FORTRAT P.

M. MORET L.

DOYEN M. WEIL L.

PROFESSEURS TITULAIRES

MM.	NEEL L.	MAGNETISME
	HEILMANN R.	CHIMIE ORGANIQUE
	KRAVTCHENKO J.	MECANIQUE RATIONNELLE
	CHABAUTY C.	MATHEMATIQUES PURES
	PARDE M.	POTAMOLOGIE
	BENOIT J.	RADIOELECTRICITE
	CHENE M.	CHIMIE PAPETIERE
	BESSON J.	ELECTROCHIMIE
	WEIL L.	THERMODYNAMIQUE
	FELICI N.	ELECTROSTATIQUE
	KUNTZMANN J.	MATHEMATIQUES APPLIQUEES
	BARBIER R.	GEOLOGIE APPLIQUEE
	SANTON L.	MECANIQUE DES FLUIDES
	OZENDA P.	BOTANIQUE
	FALLOT M.	PHYSIQUE INDUSTRIELLE
	GALVANI O.	MATHEMATIQUES
	MOUSSA A.	CHIMIE NUCLEAIRE ET RADIOACTIVITE
	TRAYNARD P.	CHIMIE GENERALE
	SOUTIF M.	PHYSIQUE GENERALE
	CRAYA A.	HYDRODYNAMIQUE
	REULOS R.	THEORIE DES CHAMPS
	AYANT Y.	PHYSIQUE APPROFONDIE
	GALISSOT F.	MATHEMATIQUES PURES
Mlle	LUTZ E.	MATHEMATIQUES GENERALES
MM.	BLAMBERT M.	MATHEMATIQUES
	BOUCHEZ	PHYSIQUE NUCLEAIRE
	LLIBOUTRY L.	GEOPHYSIQUE
	MICHEL R.	GEOLOGIE ET MINERALOGIE
	BONNIER E.	METALLURGIE
	DESSAUX G.	PHYSIOLOGIE ANIMALE
	PILLET E.	ELECTROTECHNIQUE
	DEBELMAS J.	GEOLOGIE GENERALE
	GERBER R.	MATHEMATIQUES PURES
	PAUTHENET R.	ELECTROTECHNIQUE
	VAUQUOIS B.	CALCUL ELECTRONIQUE
	SILBER R.	MECANIQUE DES FLUIDES
	MOUSSIEGT J.	ELECTRONIQUE
	BARBIER J.C.	PHYSIQUE EXPERIMENTALE

MM.	BUYLE-BODIN M.	ELECTRONIQUE
	KOSZUL J.L.	MATHEMATIQUES
	DREYFUS B.	THERMODYNAMIQUE
	VAILLANT F.	ZOOLOGIE
	KLEIN J.	MATHEMATIQUES PURES
	SENGEL P.	ZOOLOGIE
	ARNAUD P.	CHIMIE
	BARJON R.	PHYSIQUE NULCEAIRE
	BARNOUD F.	BIOSYNTHESE DE LA CELLULOSE

PROFESSEURS ASSOCIES.

MM.	WAGNER H.	BOTANIQUE
	NAPP-ZINN K.	BOTANIQUE

PROFESSEURS SANS CHAIRE

Mme	KOFLER L.	BOTANIQUE
	DEPASSEL R.	MECANIQUE
	PERRET R.	SERVOMECANISME
Mme	BARBIER M.J.	ELECTROCHIMIE
MM.	COHEN J.	PHYSIQUE
	GIDON P.	GEOLOGIE
Mme	SOUTIF J.	PHYSIQUE GENERALE
MM.	GIRAUD P.	GEOLOGIE
	GASTINEL N.	MATHEMATIQUES APPLIQUEES
	LACAZE A.	THERMODYNAMIQUE
	GLENAT R.	CHIMIE ORGANIQUE
	BRISSONNEAU P.	PHYSIQUE GENERALE
	DUCROS P.	MINERALOGIE
	ANGLES D'AURIAC	MECANIQUE DES FLUIDES
	ROBERT A.	CHIMIE PAPETIERE
	COUMES A.	ELECTRONIQUE
	PEBAY-PEROULA	PHYSIQUE
	DEGRANGE C.	ZOOLOGIE
	GAGNAIRE D.	CHIMIE PAPETIERE
	RASSAT A.	CHIMIE
	PERRIAUX J.	GEOLOGIE
	BARRA J.	MATHEMATIQUES APPLIQUEES

PROFESSEURS HONOAIRES

MM.	FORTIER A.	MECANIQUE DES FLUIDES
	BRELOT M.	MATHEMATIQUES
	WOLFERS F.	PHYSIQUE
	DORIER A.	ZOOLOGIE

MAITRES DE CONFERENCES

MM.	BIAREZ J.	MECANIQUE DES FLUIDES
	DODU J.	MECANIQUE DES FLUIDES

MM.	DOLIQUE J.M.	ELECTRONIQUE
	HACQUES G.	MATHEMATIQUES APPLIQUEES
	LANCIA R.	PHYSIQUE AUTOMATIQUE
	POULOUJADOFF M.	ELECTROTECHNIQUE
	KAHANE A.	PHYSIQUE
Mme	BONNIER J.	CHIMIE
Mme	KAHANE J.	PHYSIQUE
MM.	DEPORTES C.	CHIMIE MINERALE
	DEPOMMIER P.	PHYSIQUE NUCLEAIRE
	CAUQUIS G.	CHIMIE GENERALE
	BONNET G.	PHYSIQUE
Mme	BOUCHE L.	MATHEMATIQUES
MM.	COLOBERT L.	PHYSIOLOGIE ANIMALE
	PAYANT J.J.	MATHEMATIQUES
	CAUBET J.P.	MATHEMATIQUES
	LAURENT P.	MATHEMATIQUES APPLIQUEES
	BERTRANDIAS J.P.	MATHEMATIQUES APPLIQUEES
	BRIERE G.	PHYSIQUE
	LAJZEROWICZ J.	PHYSIQUE
	VALENTIN J.	PHYSIQUE
	DESRE P.	METALLURGIE
	BONNETAIN L.	CHIMIE MINERALE

MAITRE DE CONFERENCE ASSOCIE

MM.	RADELLI L.	GEOLOGIE

Le présent travail a été réalisé dans le cadre d'un contrat passé entre la Délégation Générale à La Recherche Scientifique et technique et le Laboratoire de Calcul de l'Université de GRENOBLE.

Je tiens à exprimer ici toute ma reconnaissance à Monsieur le Professeur KUNTZMANN, Directeur du Service de Mathématiques Appliquées de l'Université de Grenoble dont la bienveillante autorité et les conseils précieux m'ont permis de mener à bien cette étude. Je suis particulièrement sensible à l'honneur qu'il m'a fait en acceptant de présider le jury.

Je remercie vivement Monsieur le Professeur VAUQUOIS,
Madame BERTRANDIAS, Maître de Conférences,
d'avoir bien voulu accepter de faire partie du jury.

Je tiens enfin à remercier les membres du Laboratoire de Calcul et en particulier ceux de l'Equipe de Logique et de l'Equipe de Programmation dont l'aide a été si souvent sollicitée au cours de ce travail.

I N T R O D U C T I O N

Réaliser la synthèse d'une fonction Booléenne incomplète avec un ou plusieurs opérateurs donnés consiste à trouver un réseau dont les organes sont uniquement les opérateurs spécifiés et dont la sortie est compatible avec la fonction.

Il y a deux voies possibles pour aborder le problème de la synthèse :

- l'une consiste à partir de la fonction f à réaliser, à rechercher 3 fonctions g_1, g_2, g_3 telles que :

$$\text{Maj}\ (g_1, g_2, g_3) = f$$

Puis à recommencer le procédé jusqu'à obtention des variables. Nous avons appelé ces méthodes : méthodes de fractionnement. La deuxième voie consiste à élaborer à partir des variables des fonctions de plus en plus complexes jusqu'à obtention de la fonction cherchée; nous avons appelé ces méthodes, méthodes de composition.

Les principales méthodes de fractionnement sont chronologiquement celles de Cohn et Lindaman, celle de MYATA, celle de THOMA.

La méthode de Cohn et Lindaman est résumée dans une formule permettant par réductions de passer d'une fonction de n variables à 3 fonctions de $n-1$ variables. C'est une méthode intéressante quand le nombre de variables n'est pas trop élevé. Elle conduit à des réalisations dont le nombre de couches est assez élevé.

La méthode proposée par MYATA est une variante de l'application de la formule de Lagrange; elle repose sur la remarque suivante :

si $f = A(B+C)$ et si $B \cdot C = 0$ alors $f = \text{Maj}\ (A,B,C)$

si $f = A+BC$ si $B + C = 1$ alors $f = \text{Maj}\ (A,B,C)$.

$f(x,y,X) = x\, f(1,y,X) + x' f(0,y,X)$

on fera d'abord $f = \mathrm{Maj}\lfloor x\, f(1,y,X),\ x' f(0,y,X),\ 1 \rfloor$

puis :

$$x\, f(1,y,X) = x\,[y\, f(1,1,X) + y' f(1,0,X)]$$

on pose :

$$x = A \quad ; y\, f\,(1,1,X) = B; \quad y' f(1,0,X) = C$$

on a bien

$$x\, f\,(1,y,X) = A(B+C) \text{ avec } BC = 0.$$

Comme toutes les formules déduites de la structure canonique, la méthode est rapide, simple mais conduit à des réalisations coûteuses.

Citons encore la méthode proposée par THOMA, et consistant à supprimer une variable si la fonction est monotone par rapport à cette variable (sinon on se ramène d'abord à ce cas).

La principale méthode de composition est celle d'Ackers et Robbins.

Ackers et Robbins présentent la fonction sous forme d'un tableau. Ils remarquent que ce tableau doit satisfaire à certaines conditions pour que la fonction soit réalisable; ces conditions sont en fait les conditions de sous-imparité. Par adjonction de colonnes nouvelles (les compléments des variables, 0 et 1) ils obtiennent un tableau satisfaisant. Une fonction est alors représentée par une colonne de 1 et de 0 et une solution par une colonne de 1.

Si les algorithmes proposés par Ackers et Robbins sont difficilement applicables dès que le nombre de lignes et de colonnes du tableau est élevé, cette notion de tableau, les règles d'élimination introduites sont à la base de toutes les méthodes par composition.

Citons encore les travaux d'Amarel Cooke et Winder recherchant par fractionnement des réseau pour les fonctions du type S_{2p+1}^{p+1}.

En ce qui concerne la recherche d'un réseau de coût minimal le problème se trouve résolu dans le cas des réseaux arborescents par le théorème d'amélioration du à Monsieur le Professeur KUNTZMANN. L'algorithme qui en découle a été programmé et se trouve exposé au chapitre 2.

Le but du travail fait ici est double :

- Rechercher des propriétés des réseaux de coût minimal et borner supérieurement ce coût.
- Mettre au point, programmer et comparer des algorithmes de synthèse permettant de traiter un assez grand nombre de variables.

Dans le chapitre 1, on s'est efforcé de dégager certaines propriétés des fonctions impaires croissantes, générées par composition de l'opérateur majorité. Toute fonction n'appartenant pas à la famille impaire croissante sera transformée en une fonction de la famille.

Les chapitres2 et 3 sont consacrés à l'étude d'algorithmes de synthèse et à la comparaison des programmes correspondants.

Dans le chapitre 4 on a utilisé les propriétés du chapitre 1 et les algorithmes du chapitre 2 pour :

- rechercher des propriétés des réseaux de coût minimal
- obtenir pour certaines fonctions particulières des réseaux de coût aussi faible que possible.
- obtenir pour le nombre de couches et le coût des bornes supérieures aussi faibles que possible.

-:-:-:-:-

C H A P I T R E - I -

G E N E R A L I T E S

I - DEFINITIONS ET PROPRIETES

I - 1 - DEFINITIONS DE L'OPERATEUR MAJORITE.

I - 1 - a - Définition algébrique.

L'opérateur majorité est un opérateur à 3 entrées et une sortie qui réalise la fonction Booléenne :

$$f = xy + xz + yz \qquad (1)$$

Il y a d'autres expressions Booléennes équivalentes à la formule (1) :

- $f = (x+y)(x+z)(y+z) \qquad (2)$
- $f = xy + z \cdot (x \oplus y) \qquad (3)$

Les formules (1) et (2) duales l'une de l'autre font immédiatement apparaître des propriétés de l'opérateur.

- les entrées x, y, z sont permutables : l'opérateur donne une fonction symétrique par rapport à ses 3 entrées.
- la fonction de sortie est croissante par rapport à chacune des entrées.
- Si l'on fait en particulier y = x ou a

  $$f = x.x + xz + x.z = x$$
- On ne peut obtenir le complément d'une entrée par l'opérateur majorité.

I - 1 - b - <u>Définition par les opérateurs à seuil.</u>

Un opérateur à seuil est un opérateur à p entrées, une sortie, caractérisé par un nombre arithmétique S appelé seuil et p nombres $\alpha_1, \alpha_2, \ldots \alpha_p$ respectivements affectés aux entrées 1, 2, ... p, et appelés poids.

L'opérateur à seuil ainsi défini a pour sortie f une fonction des entrées $x_1, x_2, \ldots, x_p$ définie de la manière suivante :

$$- \ f = 1 \text{ si } \sum_{i=1}^{p} \alpha_i . x_i \geq S$$

$$- \ f = 0 \text{ dans les autres cas.}$$

$\sum$ désigne une somme arithmétique :

$$\alpha_i x_i = \alpha_i \text{ si } x_i = 1, \quad = 0 \text{ si } x_i = 0$$

L'opérateur majorité est alors défini comme un opérateur à seuil à 3 entrées, une sortie, dont le seuil est 2 et les poids :

$\alpha_1 = \alpha_2 = \alpha_3 = 1$

$$\text{Soit :} \quad \left.\begin{array}{ll} f = 1 & \text{si } x + y + z \geq 2 \\ f = 0 & \text{si } x + y + z < 2 \end{array}\right\} \text{Il s'agit ici d'une somme arithmétique des valeurs des variables.}$$

On retrouve la forme algébrique : pour que $x+y+z > 2$ il faut et il suffit que deux au moins des trois variables soient égales à 1; ce qui s'exprime en Booléen par : $xy + xz + yz = 1$.

<u>Notation</u> - Dans toute la suite on désignera par "Maj (u,v,w)" la fonction sortie d'un opérateur majorité dont les entrées sont u, v, w.

I - 2 - FONCTIONS IMPAIRES - [12]

I - 2 - a - Définition : Une fonction $f(X)$ est impaire si

$$\boxed{f(X) = f^{*}(X) = f'(X')}$$

Ces fonctions sont encore dites auto-duales.

I - 2 - b - Propriétés :

- De deux points symétriques de l'hypercube, l'un appartient à l'ensemble représentatif de f, l'autre pas.

  En effet : si $f(X) = 1$ on a $f^{*}(X) = 1 = f'(X')$

  d'où $f(X') = 0$

- Une fonction impaire de n variables compte toujours 2^{n-1} points dans son ensemble représentatif.

- Conséquence de la propriété précédente :

  2 fonctions impaires ne sont jamais liées par une inégalité $f \geqslant g$ ou $g \geqslant f$.

Exemples de fonctions impaires :

$f(x) = x$ est impaire ; $f(x) = x'$ également

$f(x,y,z,t) = x(y' + z' + t') + y'\,z'\,t'$ est impaire.

I - 2 - c - Ecriture des fonctions impaires :

Soit $f(x,X)$ une fonction impaire; comme pour toute fonction on peut écrire :

$$f(x, X) = xf(1, X) + x'\,f(0, X)$$

Ecrivons la conditions d'imparité pour f :

$$f(x, X) = f^*(x, X)$$

Soit : $x\, f\,(1, X) + x' f(0, X) = [x + f^*(1, X)][x' + f^*(0, X)]$

$$x\, f\,(1, X) + x' f(0, X) = x\, f^*(0, X) + x' f^*(1, X)$$

faisons $x = 1$; il vient :

$$\boxed{f\,(1, X) = f^*\,(0, X)}$$

Réciproque :

Si $f(x, X)$ est une fonction de la forme :

$$f(x, X) = x\, g(X) + x' g^*(X)$$

Alors $f(x, X)$ est impaire.

En effet :

$$f^*(x, X) = [x + g^*(X)][x' + g\,(X)]$$

$$= x\, g(X) + g^*(X)\,.\,x' = f\,(x, X).$$

I - 2 - d - Développement en φ d'une fonction impaire.

Soit $f(X)$ une fonction impaire et $\varphi(Y)$ une fonction quelconque donnée; aucune hypothèse n'est faite sur X et Y; les deux ensembles peuvent avoir des composantes communes ou non, l'un être inclus dans l'autre ou non.

On se propose de chercher quelles conditions devront remplir deux fonctions $\psi(X, Y)$, $\theta(X, Y)$ pour que l'une des deux écritures ci-dessous soit possible :

1	$f(X) = \varphi(Y)\,.\,\psi(X, Y) + \varphi'(Y)\,\theta(X, Y)$
2	$f(X) = \varphi(Y)\,.\,\psi(X, Y) + \theta\,(X, Y)$

Pour cela nous dresserons un tableau comportant les colonnes f, φ, φ', Ψ, Θ; sur chaque ligne on met les valeurs respectives de f et φ et on en déduit les valeurs que doivent prendre ψ et Θ, lorsque la valeur de ψ (ou de Θ) est sans importance nous indiquerons cela par le symbole habituel d'indétermination c'est-à-dire $\emptyset$.

Cas de l'écriture 1

f	φ	φ'	Ψ	Θ
1	1	0	1	$\emptyset$
1	0	1	$\emptyset$	1
0	1	0	0	$\emptyset$
0	0	1	$\emptyset$	0

$\psi(X, Y)$, $\Theta(X, Y)$ sont déterminées comme 2 fonctions incomplètes définies par :

$$\mathrm{I} \quad \begin{cases} \varphi(Y).\ f(X) \leqslant \psi\ (X, Y) \leqslant f(X) + \varphi'(Y) \\ \varphi'(Y).\ f(X) \leqslant \Theta\ (X, Y) \leqslant f(X) + \varphi\ (Y) \end{cases}$$

Ecriture 2

f	φ	Ψ	Θ
1	1	$\emptyset$	$\emptyset$
1	0	$\emptyset$	1
0	1	0	0
0	0	$\emptyset$	0

$\Psi + \Theta = 1$

On voit qu'ici la détermination de ψ et θ n'est pas complète; en effet sur tout point XY de l'hypercube correspondant à la 1ère ligne du tableau ($f(X) = \varphi(Y) = 1$) $\psi(X, Y)$ et $\theta(X, Y)$ sont déterminés l'un par l'autre.

Si X et Y sont tels que l'on choisit $\psi(X, Y) = 0$ il faut $\theta(X, Y) = 1$ ce qui, joint aux conditions de la deuxième ligne donne :

$$\psi(X, Y) + \theta(X, Y) \geqslant f(X)$$

$\theta(X, Y)$ est déterminé par ses deux bornes;

$\psi(X, Y)$ est déterminé seulement par une borne supérieure,

ce qui nous donne les relations définissant $\psi(X, Y)$ et $\theta(X, Y)$

$$\text{II} \quad \boxed{\begin{array}{lll} \varphi'(Y)\, f(X) \leqslant & \theta(X, Y) & \leqslant f(X) \\ & \psi(X, Y) & \leqslant f(X) + \varphi'(Y) \\ f(X) \leqslant \psi(X, Y) + \theta(X, Y) & & \end{array}}$$

<u>Cas particulier : $\varphi(Y)$ est impaire</u> :

Si $\varphi(Y)$ est impaire on peut prendre

$$\psi(X, Y) = \theta^*(X, Y)$$

<u>En effet</u> :

<u>Ecriture 1</u> : $\theta(X, Y) \leqslant \varphi(Y) + f(X)$ entraine $\theta^*(X, Y) \geqslant \varphi(Y).f(X)$

$\theta(X, Y) \geqslant \varphi'(Y) . f(X)$ entraine $\theta^*(X,Y) \leqslant \varphi'(Y) + f(X)$

soit $\boxed{\varphi(Y).f(X) \leqslant \theta^*(X, Y) \leqslant \varphi'(Y) + f(X)}$

En se reportant à la 2ème inégalité du système I on voit que $\theta^*(X,Y)$ est une solution pour ψ (X, Y)

$$\text{III} \quad \boxed{\begin{array}{l} f(X) = \varphi(Y) \;.\; \theta^*(X, Y) + \varphi'(Y)\, \theta\, (X, Y) \\ \varphi'(Y) \;.\; f(X) \leqslant \theta(X, Y) \leqslant \varphi\,(Y) + f(X) \end{array}}$$

Ecriture 2 : (a) $\theta\,(X, Y) \leqslant f(X)$ entraine $\theta^*(X, Y) \geqslant f(X)$

(b) $\theta\,(X, Y) \geqslant \varphi'(Y)\;.\; f(X)$ entraine $\theta^*(X, Y) \leqslant f(X) + \varphi'(Y)$

d'après (b) θ^* satisfait à la condition de borne supérieure sur ψ (X, Y);

d'après (a) $\theta^*(X, Y) \geqslant f(X)$ entraine $\theta(X,Y) + \theta^*(X, Y) \geqslant f(X)$

$\theta^*(X, Y)$ satisfait bien à toutes les conditions sur ψ (X, Y).

$$\text{IV} \quad \boxed{\begin{array}{l} f(X) = \varphi\,(Y) \quad \theta^*(X, Y) + \theta(X, Y) \\ \hline \varphi'(Y).\; f(X) \leqslant \theta\,(X, Y) \leqslant f(X) \end{array}}$$

Exemple $\quad f(x, y, z, t) = xy' + xz + xt' + y'zt'$

$$\varphi(x, y, z) = xy + xz + yz$$

$f + \varphi = x + yz + zt'$

$f + \varphi' = y' + xz + z't' + x'z' + xt'$

$\varphi f = xz + xyt'$

$\varphi' f = xy'z' + x'y'zt'$

Ecriture 1 $\quad xy'z' + x'y'zt' \leqslant \theta \leqslant x + yz + zt'$

prenons $\theta = x + zt'$ $\qquad \theta^* = xz + xt'$

$xy' + xz + xt' + y'zt' = (xy + xz + yz)\,(xz + xt') + (x'y' + x'z' + y\,z')$
$(x + zt')$

Ecriture 2 $\quad xy'z' + x'y'zt' \leqslant \theta \leqslant xy' + xz + xt' + y'zt'$

prenons $\quad \theta = xy' + y'zt' \qquad \theta^* = y' + xz + xt'$

$$f = (xy + xz + yz)\,(y' + xz + xt') + xy' + y'zt'$$

Remarque : Dans le cas ou l'on utilise 2 fonctions ψ et θ est-il possible de choisir $\psi = \theta$?

C'est impossible; en effet posons $U = \psi = \theta$

On obtiendrait quelle que soit l'écriture :

$$f \leqslant U \leqslant f$$

Soit $\psi\,(X, Y) = \theta\,(X, Y) = f(X)$

I - 3 - FONCTIONS SOUS IMPAIRES [12].

I - 3 - a - Définition.

$f(X)$ est une fonction sous impaire si

$$f(X)\,.\,f(X') = 0$$

Conséquence : De deux points symétriques de l'hypercube, un au plus appartient à l'ensemble représentatif de $f(X)$.

- Si $f(X)$ est une fonction sous impaire il existe au moins une fonction impaire $h(X)$ telle que :

$$h(X) \geqslant f(X)$$

I - 3 - b - Condition nécessaire et suffisante de sous imparité.

Pour que la fonction $f(X)$ soit sous impaire il faut et il suffit que tout couple de monômes m_1, m_2, compatibles avec $f(X)$, ait au moins une lettre commune apparaissant sous le même aspect dans les deux monômes.

Démonstration.

-- Condition nécessaire --

Supposons que les deux monômes m_1 et m_2 n'aient pas de lettres communes sous le même aspect.

Appelons X1 les lettres communes à m_1 et m_2.
Elles apparaissent dans m_1 sous une forme que nous noterons X1 et dans m_2 sous la forme opposée que nous symboliserons par X'1.
X2 sont les lettres directes de m_1, n'apparaissant pas dans m_2.
X'3 sont les lettres accentuées de m_1, n'apparaissant pas dans m_2
Y2 sont les lettres directes de m_2, n'apparaissant pas dans m_1.
Y'3 sont les lettres accentuées de m_2, n'apparaissant pas dans m_1.
Z est l'ensemble des variables n'apparaissant ni dans m_1 ni dans m_2.
les ensembles de lettres X1, X2, X3, Y2, Y3, Z sont disjoints 2 à 2.

On a :

$$m_1 = X1 \, . \, X2 \, . \, X'3$$

$$m_2 = X'1 \; Y2 \; Y'3$$

Considérons les deux monômes μ_1 et μ_2

$$\mu_1 = X1 \, . \, X2 \; X'3 \; Y'2 \; Y3 \; Z$$

$$\mu_2 = X'1 \; X'2 \; X3 \, . \, Y2 \; Y'3 \; Z'$$

on a $\mu_1 < m_1 < f$

$\mu_2 < m_2 < f$

μ_1 et μ_2 sont compatibles avec f; or ils représentent 2 points symétriques de l'hypercube; il y aurait 2 points symétriques X et X' pour lesquels on aurait

$$f(X) = f(X') = 1$$

ce qui est impossible puisque f(X) est sous impaire.

-- Condition suffisante --

Soit $f(X)$ donnée par une somme de monômes tels que deux d'entre eux ont toujours une lettre commune sous le même aspect.

Soit X un point de l'hypercube où $f(X) = 1$.

Soit m_1 un monôme tel que $m_1(X) = 1$.

Soit alors m_2 un monôme quelconque compatible avec $f(X)$; puisque m_1 et m_2 ont une lettre au moins en commun sous le même aspect on a : $m_2(X') = 0$

Ceci étant vrai quel que soit m_2 compatible avec f, on en déduit :

$$f(X') = 0$$

La fonction f est donc sous impaire.

Corollaire.

Les fonctions impaires étant un cas particulier de fonctions sous-impaires, la propriété est vérifiée par les fonctions impaires.

I - 4 - FONCTIONS CROISSANTES (RAPPELS)

I - 4 - a - Définition $f(X)$ est une fonction croissante, si, pour

$$X2 \geqslant X1 \quad \text{on a} : f(X2) \geqslant f(X1)$$

Conséquences.

Une fonction croissante possède une écriture en sommes de produits, sans lettre accentuée.

Tous ses monômes premiers sont donc croissants et comme il n'y a pas de consensus possible, ils sont tous obligatoires ; d'où :

- Une fonction croissante possède une base première unique.

I - 4 - b - Points caractéristiques d'une fonction croissante [12].

On appelle points caractéristiques de première espèce d'une fonction croissante, les points de l'hypercube dont le monôme de Lagrange est obtenu en ajoutant à chaque monôme premier de la fonction, les lettres manquantes sous forme accentuée.

La donnée des points caractéristiques de première espèce suffit à déterminer entièrement la fonction.

I - 5 - FONCTIONS IMPAIRES ET SOUS IMPAIRES CROISSANTES.

I - 5 - a - Fonctions impaires croissantes - Définition.

$f(X)$ est impaire croissante si elle est à la fois impaire et croissante. Comme fonction croissante elle possède donc une écriture sommes de produits sans lettres accentuées et tous ses monômes premiers sont obligatoires.

I - 5 - b - Conséquences.

Ecriture développée en x d'une fonction impaire croissante.

Soit $f(x, X)$ une fonction impaire croissante; $f(x, X)$ peut s'écrire :

$$f(x, X) = x\, g(X) + h\,(X)$$

ou $g(X)$ et $h(X)$ sont des fonctions croissantes.

En vertu de l'imparité on a :

$$f(x, X) = f^*(x, X) \qquad \text{soit :}$$

$$xg(X) + h(X) = [x + g^*(X)]\; h^*(X) = xh^*(X) + g^*h^*.$$

Si on fait $x = o$ on obtient

$$h = g^*\, h^*$$

D'où une écriture possible :

(1) $\boxed{f(x, X) = xh^*(X) + h(X)}$ où $h(X)$ est sous impaire croissante puisque $h^*(X) \geqslant h(X)$

Cette propriété est caractéristique des fonctions impaires croissantes; en effet, si une fonction croissante peut s'écrire suivant la formule (1) avec $h(X)$ sous impaire croissante, elle est impaire croissante car par dualité (1) redonne $f^* = f$.

<u>Exemples de fonctions impaires croissantes.</u>

$f(x) = x$ est impaire croissante

$f(x,y,z,t,u) = xy + xzt + xu + yzu + ytu$ est également impaire croissante.

I - 5 - c - <u>Ecriture développée en φ d'une fonction impaire croissante.</u>

Soit $f(X)$ une fonction impaire croissante; si $\varphi(Y)$ est une fonction impaire croissante donnée (X et Y étant quelconques) on a vu au I - 2 - d qu'il était toujours possible de trouver une fonction θ (X, Y) telle que :

(1) $$f(X) = \varphi(Y)\ \theta^*(X, Y) + \theta\ (X, Y)$$

$\theta\ (X, Y)$ étant définie par : $\varphi'(Y)\ .\ f(X) \leqslant \theta\ (X, Y) \leqslant f(X)$

On remarque immédiatement que θ est sous impaire. Est-il possible de trouver $\theta\ (X, Y)$ croissant et satisfaisant à la formule 1 ?

Ceci est toujours possible; en effet appelons $g(X, Y)$ l'enveloppe supérieure croissante de $\varphi'(Y)\ f(X)$.

Par définition de l'enveloppe supérieure croissante on a :

$$g(X, Y) \geqslant \varphi'(Y) \cdot f(X)$$

$f(X)$ est croissante et $f(X) \geqslant \varphi'(Y) \cdot f(X)$; $g(X, Y)$ étant la plus petite fonction croissante supérieure à $\varphi'(X) \cdot f(X)$ on a :

$$g(X, Y) \leqslant f(X)$$

$g(X, Y)$ est une solution pour $\theta\ (X, Y)$.

Remarque 1.

Il est très facile de déterminer $g(X, Y)$ sans passer par $\varphi'(Y)$.

En effet, posons $k(X, Y) = \varphi(Y) \cdot f(X)$

$h(X, Y) = \varphi'(Y) \cdot f(X)$;

$\hat{h}(X, Y) = \widehat{\varphi'(Y) \cdot f(X)}$.

on a : $f(X) = k(X, Y) + h\ (X, Y)$

$f(X)$ étant croissante $\hat{f}(X) = f(X) = k(X) + \hat{h}(X, Y)$

$\hat{h}(X, Y)$ peut être considérée comme la somme des monômes à ajouter à $k(X, Y) = \varphi(Y) \cdot f(X)$ pour obtenir $f(X)$.

d'où : $\hat{h}\ (X,Y) = g(X, Y)$.

Remarque 2.

Aucune hypothèse n'a été faite sur X et Y; en particulier il pourrait très bien se produire le cas où $g(X, Y) = f(X)$

exemple : $f = xy + xz + xt + yzt$

$\varphi = uv + uw + vw$

$\varphi f = (uv + uw + vw)\ (xy + xz + xt + yzt)$

Pour obtenir f on doit ajouter à φ f tous les monômes de f. Un tel développement est trivial. Nous allons donner une condition pour que ce cas ne se produise pas.

Théorème.

Pour que dans le développement d'une fonction impaire croissante $f(X)$ par rapport à une fonction impaire croissante $\varphi(Y)$, de la forme

$$f(X) = \varphi(Y)\ \psi\ (X, Y) + \psi^{*}(X, Y)$$

on puisse trouver $\psi^{*} \not\equiv f$ il faut et il suffit qu'un monôme premier de φ soit diviseur d'un monôme premier de f ou égal à un monôme premier de f.

Démonstration.

--Condition nécessaire --

Nous avons vu que pour obtenir l'enveloppe supérieure croissante de $\varphi' f$ on pouvait prendre la somme des monômes qu'il faut ajouter à φf pour obtenir f.

Si aucun monôme premier de $\varphi(Y)$ n'est diviseur d'un monôme premier de $f(X)$ ou égal à un monôme premier de $f(X)$ le produit $\varphi(Y) . f(Y)$ sera formé de monômes tous divisés par des monômes premiers de f.

-- Conséquence --

Pour restituer f il faudra ajouter à φf tous les monômes premiers de f

d'où : $\widehat{\varphi' f} = g(X, Y) = f(X)$

Or toute fonction croissante ψ^{*} (X, Y) permettant l'écriture vue plus haut est définie par :

$$g(X, Y) \leqslant \psi^{*}\ (X, Y) \leqslant f(X)$$

on aurait ψ^{*} $(X, Y) = f(X)$.

--Condition suffisante --

Si un monônme n de $\varphi(Y)$ est, soit diviseur d'un monôme premier de $f(X)$, soit égal à un monôme premier de $f(X)$, le produit $\varphi(Y) . f(X)$ restitue ce monôme premier de $f(X)$ soit m; pour restituer la fonction $f(X)$, il ne sera pas nécessaire d'ajouter à $\varphi(Y) . f(X)$ le monôme m.

d'où :

$$g(X, Y) < f(X)$$

on peut donc en particulier prendre $\psi^*(X, Y) = g(X, Y)$ et on a bien $\psi^* < f$.

<u>Remarque.</u>

L'écriture ainsi obtenue est impaire croissante en φ, X, Y.

<u>Exemples de développements.</u>

α) - $f = xy + xz + xt + yzt$

$\varphi = xy + xz + yz$

$\varphi f = xy + xz + yzt$ d'où $f = \varphi f + xt$

$f = \varphi(x + t) + xt \quad = (xy + xz + yz)(x + t) + xt$

β) - $f = xy + xz + xt + yzt$

$\varphi = xy + xu + yu$

$\varphi f = xy + xzu + xtu + yztu$; $f = \varphi f + xz + xt + yzt$

soit : $f = \varphi(xy + xz + xt + zt) + xz + xt + yzt$

I - 5 - d - <u>Fonctions sous impaires croissantes.</u>

Nous avons vu qu'une condition nécessaire et suffisante pour qu'une fonction soit sous impaire est que deux monômes quelconques, inférieurs à la fonction, aient une variable en commun, sous le même aspect dans les deux monômes.

Dans le cas des fonctions sous impaires croissantes, cette propriété se traduit de la manière suivante :

<u>Théorème.</u>

Une condition nécessaire et suffisante pour qu'une somme de monômes sans lettre accentuée définisse une fonction sous impaire croissante est que tout couple de monômes compatibles avec cette somme ait une lettre <u>directe</u> commune au moins.

Condition nécessaire.

La condition de sous imparité est que 2 monômes quelconques aient une lettre commune; les monômes définissant la fonction étant croissants, cette lettre ne peut être que sous forme directe.

Si deux monômes sont compatibles avec la somme ce sont soit des monômes de la somme, soit des multiples des monômes de la somme; ils ont donc nécessairement au moins une lettre directe commune.

Condition suffisante.

La somme de monômes croissants définit une fonction croissante; cette somme étant telle que deux monômes quelconques compatibles avec elle ont une lettre commune, la fonction définie est bien sous impaire.

Corollaire.

Les fonctions impaires croissantes étant des fonctions sous impaires croissantes possèdent cette propriété.

II - DECOMPOSITION DES FONCTIONS IMPAIRES

II - 1 - DEFINITIONS ET PROPRIETES.

II - 1 - a - Définitions.

Décomposition disjonctive. Une fonction Booléenne $f(X)$ possède une décomposition disjonctive si on peut la mettre sous la forme :

$$\boxed{\begin{array}{l} f(X) = g\,[\,\varphi(Y),\, Z\,] \\ Y \cup Z = X \quad ; Y \cap Z = \emptyset \end{array}}$$

Si Y se réduit à une seule variable le cas est trivial et n'est pas considéré comme un cas de décomposition.

Somme directe. Une fonction $f(X)$ possède une décomposition en somme directe si on peut la mettre sous la forme :

$$\boxed{\begin{array}{l} f(X) = g(Y) + h(Z) \\ Y \cup Z = X \quad ; Y \cap Z = \emptyset \end{array}}$$

Produit direct. Une fonction $f(X)$ possède une décomposition en produit direct si on peut la mettre sous la forme :

$$\boxed{\begin{array}{l} f(X) = g(Y) \,.\, h(Z) \\ Y \cup Z = X \quad ; Y \cap Z = \emptyset \end{array}}$$

II - 1 - b - Propriétés.

Décomposition des fonctions sous impaires.

Théorème 1. -

Une fonction sous impaire ne possède pas de décomposition en somme directe.

En effet, supposons que la fonction sous impaire $f(X)$ puisse se mettre sous la forme :

$$f(X) = g(Y) + h(Z)$$

Soit m_1 un monôme inférieur à $g(Y)$;
Soit m_2 un monôme inférieur à $h(Z)$;
m_1 et m_2 sont tous deux inférieurs à $f(X)$; ils ont donc, puisque $f(X)$ est sous impaire, une lettre commune au moins sous le même aspect d'après I - 3 - b.
d'où :

$$Y \cap Z \neq \emptyset.$$

Corollaire 1. 1 -

Par dualité : une fonction sur impaire ne possède pas de décomposition en produit direct.

Corollaire 1. 2 -

Une fonction impaire étant à la fois sous impaire et sur impaire ne possède pas de décomposition ni en somme directe ni en produit direct.

Théorème 2. -

Si une fonction sous impaire $f(X)$ possède une décomposition en produit direct $f(X) = g(Y) \cdot h(Z)$ l'une au moins des deux fonctions g, h est sous impaire.

Démonstration.

Si $g(Y)$ n'est pas sous impaire il existe deux monômes premiers m_1 et m_2 de $g(Y)$ n'ayant aucune lettre commune sous le même aspect.

$$g(Y) = m_1 + m_2 + g_1(Y)$$

De même $h(Y) = n_1 + n_2 + h_1(Z)$ n_1 et n_2 n'ayant aucune lettre commune.

D'où : $f(X) = [\ m_1 + m_2 + g_1(Y)\ ]\ [\ n_1 + n_2 + h_1(Z)\ ]$

d'où : $m_1\ n_1$, $m_2\ n_2$ sont inférieurs à $f(X)$.

m_1 et m_2 n'ont aucune lettre commune par hypothèse

n_1 et n_2 n'ont aucune lettre commune par hypothèse.

m_1 et n_2 non plus car $Y \cap Z = \emptyset$; de même pour m_2 et n_1.

D'où : $m_1\ n_1$ et $m_2\ n_2$ n'ont pas de lettre commune ce qui est impossible puisque $f(X)$ est sous impaire.

Comme $Y \cap Z = \emptyset$ il faut que m_1 et m_2 (ou n_1 et n_2) aient au moins une lettre commune d'où contradiction.

Corollaire 2 - 1.

Par dualité : si une fonction sur impaire possède une décomposition en somme directe, une au moins des deux fonctions de cette somme est sur impaire.

Types de décompositions pour les fonctions impaires [14].

Si $f(X)$ est une fonction impaire décomposable de la forme $g\ [\varphi, Z]$ la décomposition est obligatoirement de l'un des deux types suivants :

- $g\ [\varphi, Z]$ impaire; $\varphi(Y)$ impaire.
- $g\ [\varphi, Z] = g'[\varphi, Z']$; $\varphi(Y)$ paire.

Nous allons étudier les cas de décomposition de fonctions impaires pour lesquelles $g(\varphi, Z)$, $\varphi(Y)$ sont impaires. Nous appellerons ces décompositions, décompositions impaires.

II - 2 - DECOMPOSITION IMPAIRE D'UNE FONCTION IMPAIRE.

II - 2 - a - Condition nécessaire et suffisante de décomposition impaire.

Pour que la fonction impaire $f(x, X)$ possède une décomposition impaire de la forme $g\,[\varphi\,(x, Y)\;Z]$ il faut et il suffit que $f(1, X)$ possède une décomposition de la forme

$$f(1, X) = U(Y).\, V(Z) + U'(Y)\, V^{*}(Z)$$

Dans ce cas

$$\varphi(x, Y) = x\, U(Y) + x'\, U^{*}(Y)$$
$$g(\varphi, Z) = \varphi\, V(Z) + \varphi'\, V^{*}(Z)$$

--Condition nécessaire --

Soit la fonction impaire $g(\varphi, Z)$ donnée ci-dessus avec $\varphi = \varphi(x, Y) = x\, U(Y) + x' U^{*}(Y)$.

Comme g est impaire, si on remplace dans $g(\varphi, Z)$, φ par une fonction impaire, $g\,[\varphi(x, Y), Z]$ reste impaire.

Pour que $g\,[\varphi(x, Y), Z] = f(x, X)$ il faut que ceci soit vérifié pour $x = 1$.

Soit $g\,[\varphi(1,Y), Z] = f(1, X)$

$$= U(Y)\, .\, V(Z) + U'(Y)\, V^{*}(Z)$$

Condition suffisante : $f(1, X) = U(Y)\, V(Z) + U'(Y)\, .\, V^{*}(Z)$

Posons $\varphi(x, Y) = xU(Y) + x'U^{*}(Y)$

et $g(\varphi, Z) = \varphi V(Z) + \varphi' V^{*}(Z)$

On vérifie aisément que $g\,[\varphi(1, Y), Z] = f(1, X)$;
Comme $f(x, X)$ et $g\,[\varphi(x, Y), Z]$ sont impaires cela suffit pour que l'on ait :

$$f(x, X) = g[\varphi(x, Y), Z]$$

II - 2 - b - Décomposition d'une fonction croissante en x.

Soit $f(x, X)$ une fonction impaire, croissante en x; supposons que $f(x, X)$ possède une décomposition impaire; on a

$$f(1, X) = U(Y) \, . \, V(Z) + U'(Y) \, . \, V^{*}(Z)$$

on a :

$$g[\varphi, Z] = \varphi \, V(Z) + \varphi' V^{*}(Z)$$

$$\varphi(x, Y) = xU(Y) + x'U^{*}(Y).$$

Nous allons montrer que si $f(x, X)$ est croissante en x alors $g\,[\varphi, Z]$ est croissante en φ et $\varphi(x, Y)$ est croissante en x.

En effet, on a

$$f(1, X) > f(0, X)$$

Mais $f(0, X) = f^{*}(1, X)$ d'où $f(1, X) > f^{*}(1, X)$

ce qui entraîne $f(1, X)$ est sur impaire.

$$f(1,X) = U(Y) \, V(Z) + U'(Y) \, . \, V^{*}(Z) = [U(Y) + V^{*}(Z)]$$

$[U'(Y) + V(Z)]$

$f(1, X)$ peut être considérée comme un produit de fonctions; pour que ce produit soit sur impair, il faut que chaque facteur soit sur impair.

D'où :

$U(Y) + V^{*}(Z)$ sur impair; $\quad U'(Y) + V(Z)$ sur impair

D'après un théorème vu en II - 1 - b $U(Y) + V^{*}(Z)$ sur impaire avec $Y \cap Z = \emptyset$ entraîne que l'une au moins des deux fonctions est surimpaire.

Supposons $U(Y)$ sur impaire; $U'(Y)$ est alors sous impaire et cela entraîne pour la même raison : $V(Z)$ sur impaire.

Soit : $U(Y) > U^*(Y)$; $V(Z) > V^*(Z)$

$$\varphi(x, Y) = xU(Y) + x'U^*(Y) + U(Y) \,.\, U^*(Y)$$

$$\varphi(x, Y) = xU(Y) + U^*(Y)$$

$\varphi(x, Y)$ est bien croissante en x.

$$g(\varphi, Z) = \varphi\, V(Z) + \varphi' V^*(Z) + V(Z) \,.\, V^*(Z)$$

$$= \varphi\, V(Z) + V^*(Z)$$

$g(\varphi, Z)$ est bien croissante en φ. D'où :

Théorème. Si la fonction impaire $f(x, X)$, croissante en x, possède une décomposition impaire $g\,[\varphi(x, Y), Z]$ alors $\varphi(x, Y)$ est croissante en x, $g(\varphi, Z)$ est croissante en φ.

Remarque 1 -

On a supposé $U(Y)$ sur impaire; si $V^*(Z)$ est sur impaire cela entraine $U'(Y)$ sur impaire et on se ramène au cas ci-dessus en posant $U_1(Y) = U'(Y)$; $V_1(Z) = V^*(Z)$

Remarque 2 -

Puisque $V(Z) > V^*(Z)$ on a en fait comme condition de décomposition :

$$f(1, X) = U(Y)\, V(Z) + V^*(Z)$$

$U(Y)$, $V(Z)$ sur impaires.

II - 2 - c - Décomposition impaire d'une fonction impaire croissante.

Nous avons vu que si $f(x, X)$, impaire et croissante en x possède une décomposition impaire $g\ [\ \varphi(x, X), Z]$, alors :

- $f(1, X) = U(Y)\ .\ V(Z) + V^*(Z)$
- $\varphi(x, Y)$ est croissante en x
- $g(\varphi, Z)$ est croissante en φ.

Nous allons montrer que si $f(x, X)$ est croissante alors $\varphi(x, Y)$ est croissante et $g(\varphi, Z)$ est croissante.

$f(x, X)$ étant croissante on a $f(1, X)$ croissante soit $U(Y)\ V(Z) + V^*(Z)$ croissante.

Puisque $U(Y)\ .\ V(Z) + V^*(Z)$ est croissante et que $Y \cap Z = \emptyset$, si $Z_1 \geqslant Z_2$ on aura :

$$U(Y)\ V(Z1) + V^*(Z_1) \geqslant U(Y)\ V(Z_2) + V^*(Z_2)$$

ceci étant vrai quel que soit Y prenons Y tel que

$$U(Y) = 0$$

on a alors :

$$V^*(Z_1) \geqslant V^*(Z_2) \text{ si } Z_1 \geqslant Z_2$$

$V^*(Z)$ est donc croissante; d'où $V(Z)$ croissante.

De même si $Y_1 \geqslant Y_2$ on a $U(Y_1)\ V(Z) + V^*(Z) \geqslant U(Y_2)\ V(Z) + V^*(Z)$

on peut prendre Z tel que :

$V(Z) = 1 \quad V^*(Z) = 0$; ceci est toujours possible puisque $V(Z)$ est sur impaire; sinon on aurait $V(Z) = V^*(Z)$

$$f(1,X) = V^*(Z) = V(Z) \text{ impaire}$$

et f serait indépendante de Y et X

$U(Y_1) \geqslant U(Y_2) \qquad U(Y)$ est croissante.

D'où :

Théorème :

Si la fonction impaire croissante $f(x, X)$ possède une décomposition impaire on a :

$f(1, X) = U(Y)\ V(Z) + V^*(Z)$

- $U(Y)$ sur impaire croissante.

- $V(Z)$ sur impaire croissante

$\varphi(x, Y) = xU(Y) + U^*(Y)$ est croissante

$g(\varphi, Z) = \varphi\ V(Z) + V^*(Z)$ est croissante en $\{\varphi, Z\}$

II - 2 - d - Fonctions croissantes. Une condition nécessaire de décomposition.

Supposons $f(x, X) = yk_1(X) + k_2(X)$ une fonction impaire, croissante par rapport à toutes les variables.

Supposons $f(x, X)$ décomposable sous la forme :

$g\,[\varphi, Z] = \varphi\ V(Z) + V^*(Z)$

$\varphi(x, Y) = xU(Y) + U^*(Y)$

$Y \cup Z = X \qquad Y \cap Z = \emptyset$

On a : $f(x, X) = x\ UV + U^*V + V^*$ (1)

Nous avons vu que U et V sont des fonctions sur impaires croissantes.

Pour rendre la forme (1) irréductible nous allons faire des éliminations ;

ces éliminations sont d'un seul type :

- un monôme de V^* est égal à un monôme de V ou un monôme de U^* est égal à un monôme de U.

Soit $m_1 < V^*$ et m_1 est aussi monôme premier de V; les monômes du produit $x\ U\ m_1$ seront éliminés par m_1 de V^*.

Nous pouvons donc dans xUV remplacer V(Z) par $V_1(Z)$ obtenu en supprimant m_1 dans V(Z). De même un monôme m_2 est premier de $U^*(Y)$ et de U(Y).

Alors $m_2V(Z)$ élimine $xm_2V(Z)$ de $xU(Y) . V(Z)$.

On peut donc également remplacer U(Y) par $U_1(Y)$.

D'où :

$$f(x, X) = x\, U_1(Y) . V_1(Z) + U^*(Y)\, V_1(Z) + V^*(Z)$$

Cette forme étant complètement réduite, comme il n'y a qu'une base première pour f(x, X) on en déduit :

$$k_1(X) = U_1(Y) . V_1(Z)$$

D'où une condition nécessaire de décomposition:

Théorème.

Pour qu'une fonction impaire croissante f(x, X) se décompose en g [φ(x, Y), Z] il faut que, lorsque la fonction est écrite sous forme de base première, le coefficient de x soit un produit direct.

Corollaire.

Si une fonction impaire croissante écrite sous forme de base première admet pour coefficient de x une fonction sur impaire elle ne possède pas de décomposition majoritaire de la forme g [φ(x, Y), Z].

Ceci est évident puisqu'une fonction sur impaire ne possède pas de décomposition en produit direct.

Exemple. $f(X) = xy + xzt + xzu + xtu + yzt + yzu + ytu$

Prenons le coefficient de x:

$y + zt + zu + tu$ cette fonction est sur impaire donc pas de produit direct possible.

x et y étant symétriques il est inutile d'examiner y.

Prenons le coefficient de z :

$$xt + xu + yt + yu = (x+t)\ (y+u)$$

Il y a une décomposition possible :

$$\begin{aligned} f(z = 1, x, y, t, u) &= (x+y)\ (t+u) + xy + xtu + ytu \\ &= (x+y)(t+u)+xy. \end{aligned}$$

La fonction est décomposable sous la fome :

$$g = \varphi(x + y) + xy$$

$$\varphi(z, t, u) = z(t + u) + tu.$$

III - GENERATION DE FONCTIONS PAR L'OPERATEUR MAJORITE

III - 1 - FAMILLE IMPAIRE CROISSANTE ET OPERATEUR MAJORITE.

III - 1 - a - Opération de composition [12].

L'opération de composition consiste à remplacer, dans l'expression d'une fonction Booléenne, une variable par une fonction.

Exemple : $f(x, y, z) = xyz' + x'y'z$.

Formons la fonction $h = f(g, y, z)$ avec $g = xy' + tu$

on obtient :

$h = yz'tu + x'y'zt' + x'y'zu'$

III - 1 - b - Opération de réduction [12].

L'opération de réduction consiste à égaler deux ou plusieurs variables dans l'expression d'une fonction Booléenne.

Exemple : $f(x, y, z, t) = xy + xz' + xt + yz't$

Si on fait la réduction $z = y$ on obtient :

$f_r = xy + xy' + xt = x$.

III - 1 - c - Famille de fonctions [12].

Une famille de fonctions est un ensemble de fonctions, fermé par rapport aux opérations de réduction et de composition.

Exemple : Famille de fonctions croissantes. La réduction d'une fonction croissante donne une fonction croissante; si dans l'expression d'une fonction croissante on remplace une variable par une fonction croissante on obtient évidemment une fonction croissante.

L'ensemble des fonctions impaires croissantes forme une famille. Nous allons montrer que par réduction et composition de la majorité on obtient la famille impaire croissante.

Hypothèses : Les seules quantités dont on dispose sont les variables directes.

Théorème 1 -

Toute fonction impaire croissante peut être générée par composition de l'opérateur majorité.

Démonstration.

Le théorème est vérifié par les fonctions impaires croissantes de 3 variables au plus : en effet les seules fonctions impaires croissantes de 3 variables au plus sont x et Maj (x,y,z).

Supposons le vrai pour toute fonction impaire croissante de n-1 variables au plus.

Soit $f(x, X)$ une fonction impaire croissante de n variables de la forme :

$$f(x, X) = x\,\psi\,(X) + \psi^*(X)$$

$\Psi(X)$ étant une fonction sur impaire croissante de n-1 variables, peut être considérée comme une somme de p fonctions impaires croissantes d'au plus n-1 variables et $p \leq n-1$. Supposons $p = 2$.

On a : $$f(x, X) = x\,[\Psi_1(X) + \Psi_2(X)] + \psi_1(X)\,.\,\psi_2(X)$$
$$= \mathrm{Maj}\,[x, \psi_1(X), \psi_2(X)\,]$$

Supposons que la propriété soit vraie si $p \leq p_o$ et supposons $p = p_o + 1$.

on a $f_{p_{o+1}}(x, X) = x\ [\ \Psi_1(X) + \Psi_2(X) + \dots + \Psi_{p_o}(X) + \Psi_{p_{o+1}}(X)\ ]$

$$+ \Psi_1(X) \dots \Psi_{p_{o+1}}(X) = x \left\{ x[\Psi_1(X)+\dots+\Psi_{p_o}(X)] + \Psi_1(X)\dots\Psi_{p_o}(X) + \Psi_{p_{o+1}}(X) \right\}$$

$$+ \Psi_{p_{o+1}}(X) \left\{ x[\Psi_1(X)+\dots+\Psi_{p_o}(X)] + \Psi_1(X)\dots\Psi_{p_o}(X) \right\}$$

en posant $f_{p_o}(x, X) = x\ [\Psi_1(X)+\dots+\Psi_{p_o}(X)] + \Psi_1(X)\dots\Psi_{p_o}(X)$ on a :

$$f_{p_{o+1}}(x, X) = x\ [f_{p_o}(x, X) + \Psi_{p_{o+1}}(X)] + f_{p_o}(x, X)\ .\ \Psi_{p_o}(X)$$

$$f_{p_{o+1}}(x, X) = \mathrm{Maj}\ [x, f_{p_o}(x, X), \Psi_{p_o}(X)]$$

$\Psi_{p_o}(X)$ est impaire croissante de n-1 variables donc réalisable par hypothèse;

$f_{p_o}(x, X)$ est de n variables; le coefficient de x étant une somme d'au plus p_o fonctions impaires d'au plus n-1 variables est réalisable; d'où $f_{p_{o+1}}(x, X)$ est réalisable.

Théorème 2.

Moyennant les hypothèses faites (variables directes comme seules entrées de l'opérateur majorité), la composition de majorités ne génère que des fonctions impaires croissantes.
Il est évident que toutes les fonctions générées sont croissantes puisqu'elles peuvent s'écrire sans lettre accentuée.

Pour démontrer la propriété il suffira de démontrer que la composition, par l'opérateur majorité, de fonctions impaires croissantes donne une fonction impaire croissante.

Soient φ_1, φ_2, φ_3 - 3 fonctions impaires croissantes; d'après le théorème 1 elles peuvent être obtenues par composition de majorités.

Pour que Maj $(\varphi_1, \varphi_2, \varphi_3)$ soit impaire il faut et il suffit que

Maj $(\varphi_1, \varphi_2, \varphi_3) = [\text{ Maj } (\varphi_1, \varphi_2, \varphi_3)]^*$

$$[\text{Maj } (\varphi_1, \varphi_2, \varphi_3)]^* = (\varphi_1 \varphi_2 + \varphi_1 \varphi_3 + \varphi_2 \varphi_3)^* =$$

$$(\varphi_1^* + \varphi_2^*)\ (\varphi_1^* + \varphi_3^*)(\varphi_2^* + \varphi_3^*)$$

or φ_1, φ_2, φ_3 étant impaires, on a :

$$\varphi_1^* = \varphi_1 \quad ; \quad \varphi_2^* = \varphi_2 \quad ; \quad \varphi_3^* = \varphi_3.$$

d'où : $[\text{ Maj } (\varphi_1,\varphi_2,\varphi_3)]^* = (\varphi_1 + \varphi_2)\ (\varphi_1 + \varphi_3)(\varphi_2 + \varphi_3) =$

Maj $(\varphi_1,\varphi_2,\varphi_3)$

<u>Remarque</u> : Comme on le voit facilement la réduction n'apporte rien de plus puisque c'est une composition particulière.

<u>Conclusion</u> : Avec les variables directes comme données disponibles, on pourra réaliser par composition de majorités toutes les fonctions impaires croissantes et elles seules.

III - 2 - GENERATION DE FONCTIONS QUELCONQUES.

III - 2 - a - Génération de fonctions impaires non croissantes.

Nous avons vu au début de ce chapitre que l'opérateur majorité ne pouvait réaliser le complément; en d'autres termes la sortie réalise une fonction croissante par rapport à l'entrée.

Si on met x à l'entrée la sortie sera croissante en x.

Si on met x' à l'entrée la sortie sera décroissante en x.

Etant donné une fonction impaire non croissante en x soit f(x,X) elle s'écrit :

$$f(x, X) = x\, g(X) + x'\, g^*(X)$$

Si l'on pose $x' = u$ on a :

$$k(x, u, X) = x\, g(X) + ug^*(X)$$

et $$k(x, x', X) = f(x, X).$$

Si l'on effectue cette opération pour toutes les variables non directes de X on aboutit à une fonction :

$$H(x, u, X, U) = x\, G(X, U) + u\, G_1(X, U)$$

cette fonction est sous impaire croissante; en effet, le dédoublement des variables l'a rendue croissante.
2 monômes de $x\, G(X, U)$ ont en commun la variable x.
2 monômes de $u\, G_1(X, U)$ ont en commun la variable u.
Soit 2 monômes l'un de $xG(X, U)$, l'autre de $uG_1(X, U)$; f(x, X) est impaire :2 monômes ont toujours une variable commune sous le même aspect; les compléments des variables X ayant été remplacés par U 2 monômes ont toujours une lettre commune directe.

Remarque : On peut toujours ajouter à H des multiples arbitraires de $x_i\, u_i$ sans détruire la sous imparité et en conservant le résultat; en effet :

$$H(x, x', X, X') = f(x, X). \qquad x_i\, u_i\, A = 0 \text{ si } u_i = x'_i.$$

H(x, u, X, U) étant sous impaire croissante il existe une fonction impaire croissante au moins $\varphi(x, u, X, U)$ telle que

$$\varphi(x, u, X, U) \geqslant H(x, u, X, U)$$

d'où :

$$\varphi(x, x', X, X') \geqslant H(x, x', X, X')$$

soit puisque :

$$H(x, x', X, X') = f(x, X)$$

$$\varphi(x, x', X, X') \geqslant f(x, X)$$

$\varphi(x, u, X, U)$ étant impaire $\varphi(x, x', X, X')$ l'est aussi (la réduction et composition de fonctions impaires donne des fonctions impaires) d'où :

$$\varphi(x, x', X, X') = f(x, X).$$

Conclusion.

Si l'on dispose comme données d'entrée des variables directes et des variables complémentées, on pourra réaliser toutes les fonctions impaires.

Dans toute la suite nous nous préoccuperons donc seulement de la réalisation de fonctions croissantes. Si une fonction n'est pas croissante, on lui associera une fonction croissante obtenue par dédoublement des variables comme ci-dessus.

III - 2 - b- Génération de fonction sous-impaires croissantes (sur impaires croissantes).

Soit $f(X)$ une fonction sous impaire croissante. Nous avons vu que si l'on se donne uniquement les variables X, il est impossible de la réaliser avec un réseau majoritaire.

Par contre, si l'on se donne, en plus des variables, la constante 0 comme entrée disponible, alors il est possible de réaliser $f(X)$ comme intersection de fonctions impaires et par suite au moyen de l'opérateur majorité puisque :

$$\text{Maj}\ (u, v, o) = uv.$$

Par dualité la donnée des variables et de la constante 1 permettra de réaliser des fonctions sur impaires comme somme de fonctions impaires puisque :

$$\text{Maj}\ (u, v, 1) = u + v$$

<u>Remarque</u>.

Pour pouvoir réaliser une fonction quelconque, il sera donc nécessaire de disposer simultanément des deux constantes 0 et 1.

III - 3 - FORMULES DE TRANSFORMATION POUR UNE FONCTION CROISSANTE INCOMPLETE.

L'idée est de considérer les constantes 0 et 1 comme deux variables nouvelles u et v et d'associer à la fonction croissante incomplète $f(X)$, une fonction croissante incomplète $g(u, v, X)$ qui possède des représentations majoritaires et telle que :

- Toute représentation majoritaire de $g(u, v, X)$ devient une représentation de $f(X)$ si on y fait : $u = 1$; $v = 0$.
- Toute solution $\varphi(X)$ de la fonction incomplète $f(X)$ se traduit, par les mêmes formules de transformation en une fonction $\Psi(u, v, X)$ qui est une solution de $g(u, v, X)$.

III - 3 - a - Recherche de fonctions impaires croissantes compatibles avec une fonction incomplète croissante donnée.

Etant donné une fonction incomplète croissante $f(X)$, donnée par sa borne inférieure $\underline{f}(X)$ et sa borne supérieure $\overline{f}(X)$, une condition nécessaire et suffisante pour qu'il existe une fonction impaire croissante compatible avec $f(X)$ est que :

- $\underline{f}(X)$ soit sous impaire croissante.
- $\overline{f}(X)$ soit sur impaire croissante.

Condition nécessaire.

Elle est évidente : toute fonction $\varphi(X)$ compatible avec $f(X)$ satisfait par définition à la double inégalité :

$$\underline{f}(X) \leqslant \varphi(X) \leqslant \overline{f}(X)$$

Si l'on veut que $\varphi(X)$ soit impaire croissante, $f(X)$ étant croissante il faut que $\underline{f}(X)$ soit sous impaire croissante, $\overline{f}(X)$ sur impaire croissante.

Condition suffisante.

Considérons la fonction $\underline{f}(X) + \overline{f}^*(X)$; cette fonction est sous impaire croissante.

En effet, si elle ne l'était pas, il existerait un couple de points, symétriques sur l'hypercube, X et X' tels que

$$\underline{f}(X) + \overline{f}^*(X) = \underline{f}(X') + \overline{f}^*(X') = 1$$

Mais $\underline{f}(X)$ étant sous impaire croissante, $\overline{f}^*(X)$ également (car $\overline{f}(X)$ est sur impaire croissante) ceci ne peut se produire que si :

$$\underline{f}(X) = \overline{f}^*(X') = 1$$

$\overline{f}^*(X') = 1 = \overline{f}'(X)$; soit $\overline{f}(X) = 0$ ce qui entraine $\underline{f}(X) = 0$ ce qui est en contradiction avec l'hypothèse $\underline{f}(X) = 1$.

Il existe donc une fonction impaire croissante $\varphi(X)$ telle que :

$$\varphi(X) \geqslant \underline{f}(X) + \overline{f}^*(X)$$

soit $\quad \varphi(X) \geqslant \underline{f}(X)$

Par dualité on a : $\varphi(X) \leqslant \underline{f}^*(X) \,.\, \overline{f}(X)$ soit

$$\varphi(X) \leqslant \overline{f}(X)$$

d'où : $\quad \underline{f}(X) \leqslant \varphi(X) \leqslant \overline{f}(X).$

III - 3 - b - Fonctions croissantes incomplètes quelconques.

Introduisons la variable supplémentaire t et son complément t'
(t est en fait la constante 1 et t' la constante 0).
$f(X)$ est donné par ses bornes $\underline{f}(X)$ et $\overline{f}(X)$
Considérons les deux fonctions impaires :

$$g_1(t, X) = t\underline{f}(X) + t'\underline{f}^*(X)$$

$$g_2(t, X) = t\overline{f}(X) + t'\overline{f}^*(X)$$

et soient Ψ_1 Ψ_2 :

$$\Psi_1 = g_1\, g_2 \qquad \Psi_2 = \Psi_1^* = g_1 + g_2$$

Ψ_1 étant un produit de fonctions impaires est sous impaire. Il existe donc une fonction impaire croissante $g(t, u = t', X)$ avec

$$g \geqslant \Psi_1 \quad \text{ce qui entraine par dualité } g \leqslant \Psi_2$$

$g \leqslant \Psi_2 = g_1 + g_2 = t\,\overline{f} + t'\underline{f}^*$ soit $g\,[t,u, X] \leqslant t\,\overline{f}(X) + t'\underline{f}^*(X)$

Si dans les 2 membres on fait $t = 1$ $t' = u = 0$ on a :

$$g(1, 0, X) \leqslant \overline{f}(X).$$

D'autre part :

$$g(t, u, X) \geqslant \Psi_1 = g_1\, g_2 = t\underline{f} + t'\overline{f}^*$$

soit : $g(1, 0, X) \geqslant \underline{f}(X)$

Toute solution de Ψ_1 est donc une solution de f.

Soit maintenant $\varphi(X)$ une solution de $f(X)$. Considérons la fonction impaire

$$h(t, X) = t\,\varphi + t'\,\varphi^*$$

on a : $\underline{f} \leqslant \varphi \leqslant \overline{f}$ d'où $\varphi^* \geqslant \overline{f}^*$

or $\Psi_1 = g_1 \cdot g_2 = t\underline{f} + t'\overline{f}^*$ d'où : $h(t, X) \geqslant \Psi_1(t, X)$

et comme h est impair, par dualité on a h $\leqslant \Psi_1^* = \Psi_2$.
Toute solution de f admet une transformée qui est solution de la fonction incomplète, transformée de f.

III - 4 - CONCLUSIONS.

Le problème de la synthèse des fonctions Booléennes par des réseaux comportant uniquement des opérateurs majorité se posera donc pour nous de la manière unique suivante :

Trouver une fonction impaire croissante compatible avec une fonction croissante donnée dont les bornes sont :

- une fonction sous impaire croissante g pour la borne inférieure.
- la constante 1 pour la borne supérieure.

Par dédoublement des variables et par les formules de transformation vues plus haut, toute fonction f(X) sera transformée en une fonction g ayant ces propriétés.

Définition.

Nous appellerons fonction réalisable une fonction possédant les propriétés suivantes :
- sa borne inférieure est sous impaire croissante
- sa borne supérieure n'est pas spécifiée.

Nous voyons que les transformations vues plus haut associent à toute fonction, une fonction réalisable.
Il est bien évident qu'une fonction impaire croissante est réalisable.

IV - INTRODUCTION AUX METHODES DE COLONNES

TABLEAU ASSOCIE A UNE FONCTION REALISABLE.

L'origine de ces méthodes est l'article de S.B. Ackers [3]. Il introduit la notion de tableau et de transformation d'une fonction en une fonction réalisable.

Nous ne faisons que rappeler ici sa méthode en abrégeant l'exposé et en nous servant des notions algébriques vues plus haut.

IV - 1 - Tableau associé à une fonction réalisable.

Soit une fonction réalisable donnée par sa borne inférieure $\underline{f}(X)$. Le tableau associé est en fait la liste des sommets caractéristiques de 1ère espèce de $f(X)$. Il s'agit d'un tableau dont les colonnes représentent les variables et les lignes les sommets caractéristiques de $f(X)$.

Exemples. 1 - $f = xy + xz + xt + yzt$

On obtient le tableau ci-contre :

x	y	z	t
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1

2 - $\underline{f} = xy + x'y'$ $\qquad \overline{f} = x + y'$

On a vu qu'il suffit de fabriquer la fonction :

$$g = t\underline{f} + t'\overline{f}^{*} \quad \text{soit}$$

$$g = t(xy + x'y') + t'xy'$$

d'où le tableau

x	y	t	u=x'	v=y'	w=t'
1	1	1	0	0	0
0	0	1	1	1	0
1	0	0	0	1	1

IV - 2 - Propriétés du tableau d'une fonction réalisable.

Nous allons en fait retrouver sous une autre forme les propriétés des fonctions sous impaires. La forme initiale de ces propriétés (exprimée en termes de lignes et de colonnes) est due à Ackers

Théorème 1.

Deux lignes du tableau ont toujours au moins un 1 en commun dans une même colonne.

Cela traduit le fait que pour une fonction sous impaire croissante deux monômes premiers ont toujours une variable directe commune.

Théorème 2.

Si deux lignes L_i et L_j sont telles que $L_i > L_j$, on peut supprimer la ligne L_i.

En effet cette ligne contenant plus de 1 que L_j correspond à un monôme non premier donc à un sommet non caractéristique.

Théorème 3.

Pour pouvoir supprimer une colonne x il suffit d'avoir un ensemble de colonnes ayant la propriété suivante :

- sur tout couple de lignes du tableau où x est la seule variable valant 1 sur les deux lignes du couple, l'une au moins des colonnes nouvelles vaut 1 sur ces deux lignes.
- cet ensemble de colonnes peut toujours être réalisé par des fonctions une couche.

En effet, supposons $Y = \{y_1, y_2, \dots y_p\}$ l'ensemble des colonnes nouvelles.

Si cet ensemble satisfait à la condition correspondant à x, on peut supprimer x du tableau sans que celui ci perde la propriété de représenter une fonction réalisable; en effet soit $l_i\ l_j$ un couple de lignes :

- ou bien $x_k = 1$ sur ces deux lignes ($x_k \neq x$)
- ou bien il existe un r $1 \leqslant r \leqslant p$ avec $y_r = 1$ sur ces deux lignes.

D'autre part, soit $l_i\ l_j$ un couple de lignes pour lequel x est la seule variable à valoir 1 sur les deux lignes du couple.

Alors, il y a une variable x_i valant 1 sur l_i, 0 sur l_j il y a une variable x_j valant 0 sur l_i, 1 sur l_j.
$y_k = \text{Maj}\ (x, x_i, x_j)$ vaut 1 sur les 2 lignes du couple et est une fonction une couche.

Remarque :

Un cas particulier de cette propriété est la règle d'élimination introduite par Ackers :

- Si une colonne s_i et une colonne x_j sont telles que :

$$x_i > x_j$$

alors on peut supprimer x_j.

Exemple : $f = xyz + xyt + xyu + xzt + xzu + xtu + yzt + yzu + ytu + ztu$

	x	y	z	t	u	v_1	v_2
1	1	1	1	0	0	1	1
2	1	1	0	1	0	1	1
3	1	1	0	0	1	1	1
4	1	0	1	1	0	1	0
5	1	0	1	0	1	0	1
6	1	0	0	1	1	1	1
7	0	1	1	1	0	1	0
8	0	1	1	0	1	0	1
9	0	1	0	1	1	1	1
10	0	0	1	1	1	0	0

Proposons nous l'élimination de x

$l_1.l_6$: v_1 = Maj(x, y, t) ou v_2 = Maj(x, y, u) ou v_3 = Maj(x, z, t) ou v_4 = Maj(x, z, u).

$l_2.l_5$: v_5 = Maj(x, y, z) ou v_2 ou v_3 ou v_6 = Maj(x, t, u)

$l_3.l_4$: v_1 ou v_4 ou v_5 ou v_6

v_1 et v_2 suffisent car figurant dans les 3 couples.

L'adjonction de v_1 et v_2 permet alors de supprimer x. (ici on peut aussi supprimer y).

C H A P I T R E - 2 -

M E T H O D E S D E S Y N T H E S E

INTRODUCTION

RESEAU MAJORITAIRE

Définition : On appellera réseau majoritaire un réseau sans boucles, ayant une sortie unique et dont les organes actifs sont uniquement des opérateurs réalisant la fonction majorité.

Réseau arborescent. Un réseau majoritaire sera dit arborescent si chaque sortie d'opérateur est entrée d'un opérateur au plus.

Coût d'un réseau majoritaire. On appellera coût d'un réseau majoritaire le nombre d'opérateurs majoritaires entrant dans la composition du réseau.

COUT D'UNE FONCTION

Etant donné une fonction $\varphi(X)$ on peut se proposer de rechercher un réseau majoritaire dont la sortie soit compatible avec $\varphi(X)$. (Si $\varphi(X)$ est une fonction complète la sortie réalise $\varphi(X)$).

Cependant nous savons qu'une même fonction peut être représentée par plusieurs réseaux majoritaires de coûts différents.

On appellera coût d'une fonction le minimum du coût d'un réseau dont la sortie est compatible avec cette fonction.

Coût arborescent.

Ce sera le minimum du coût d'un réseau arborescent dont la sortie soit compatible avec la fonction.

I - RECHERCHE D'UN RESEAU ARBORESCENT DE COUT MINIMUM

La méthode proposée est une généralisation par Monsieur le Professeur KUNTZMANN d'une règle d'élimination introduite par Ackers à propos des réseaux majoritaires.

I - 1 - AMELIORATION D'UNE SORTIE D'OPERATEUR.

Soit un réseau arborescent quelconque et θ_i un opérateur de ce réseau.

La sortie de θ_i réalise une fonction $f_i(X)$ pour un coût arborescent c_i.

Améliorer la sortie de θ_i par rapport à $\varphi(X)$ consiste à remplacer la partie de réseau située en amont de cette sortie par un réseau arborescent de coût σ_i réalisant la fonction $g_i(X)$ et satisfaisant aux conditions suivantes :

- $f_i(X) = \varphi(X)$ entraine $g_i(X) = \varphi(X)$
- $\sigma_i < c_i$ ou $\sigma_i = c_i$ et $g_i(X) \neq f_i(X)$ pour une

valeur au moins de X telle que : $\varphi(X) \neq \emptyset$

<u>Théorème d'amélioration.</u>

Dans un réseau arborescent l'amélioration d'une sortie conserve ou améliore toutes les sorties situées en aval.

<u>Démonstration.</u>

Soit $f_i(X)$ la sortie de l'opérateur θ_i

Soit $g_i(X)$ son amélioration.

Soit θ_j un opérateur situé en aval de θ_i

$f_j(X)$ est la sortie de θ_j avant amélioration

$g_j(X)$ est la sortie de θ_j améliorée.

Comme il s'agit d'opérateurs croissants on a :

$$f_j(X) = A\ f_i(X) + B$$

Le réseau étant arborescent A et B sont indépendants du réseau θ_i; l'amélioration de sa sortie ne change ni A ni B; d'où :

$$g_j(X) = A\ g_i(X) + B$$

Montrons que $f_j(X) = \varphi(X)$ entraine $g_j(X) = \varphi(X)$

soit : $A\ f_i(X) + B = \varphi(X)$ entraine $A\ g_i(X) + B = \varphi(X)$

Ceci est vrai si $A = 0$, ou si $B = 1$.

Reste le cas $A = 1$, $B = 0$ soit :

$$f_i(X) = \varphi(X) \text{ entraine } g_i(X) = \varphi(X)\ ,$$

ce qui est vérifié par hypothèse puisque $g_i(X)$ est une amélioration de $f(X)$.

Conséquence.

Si la sortie de l'opérateur θ_i est non améliorable, on peut obtenir cette sortie par un réseau arborescent de même coût et dont aucune sortie intermédiaire n'est améliorable.

I - 2 - Application à la recherche du minimum arborescent.

Supposons connus tous les réseaux de coût $\leqslant k-1$, non améliorables par rapport à $\varphi(X)$.

Construisons un réseau de coût k; ce réseau est formé de sous réseaux non améliorables par rapport à $\varphi(X)$.

Soit $f(X)$ la sortie du réseau.

Si $f(X)$ est améliorable son amélioration est un réseau de coût inférieur ou égal à k.
On trouvera donc l'amélioration d'une fonction en la comparant à toutes celles déjà fabriquées.
Si $f(X)$ est compatible avec $\varphi(X)$ alors $f(X)$ représente une solution arborescente minimale de $\varphi(X)$; en effet, car si $f(X)$ est améliorable, son amélioration est également compatible avec $\varphi(X)$ et aurait été trouvée d'abord.

Il suffit donc de construire, relativement à $\varphi(X)$ les fonctions de coût 0, 1, 2, ... etc...
On ne conservera que les fonctions non améliorables et pour cela il suffit de comparer une fonction de coût k avec celles déjà existantes; 2 cas sont possibles si $f(X)$ est améliorable :

- l'amélioration de $f(X)$ est de coût inférieur à celui de $f(X)$; on le constate immédiatement et $f(X)$ est éliminé.
- l'amélioration de $f(X)$ est de coût égal à celui de $f(X)$ mais n'a pas encore été fabriqué.
  $f(X)$ est alors considérée comme non améliorable et retenue.
  Si $g(X)$ est l'amélioration de $f(X)$, on remplacera $f(X)$ par $g(X)$; d'où la règle:

Soit $f(X)$ une colonne de coût k : elle sera supposée non améliorable uniquement si aucune colonne de coût k existante ne l'élimine.
Elle ne sera considérée comme définitivement retenue que lorsque toutes les colonnes de coût k auront été fabriquées et qu aucune ne l'aura éliminée.

<u>Mise en oeuvre pour les réseaux majoritaires.</u>

Si $f(X)$ est amélioré par $g(X)$ relativement à $\varphi(X)$ on obtient, grâce aux conditions de sous imparité la règle d'amélioration.

Si f(X) est améliorée par g(X) alors :

- si $\varphi(X) = 1$ $\quad f(X) = 1$ entraine $g(X) = 1$

Soit si f et g sont les colonnes correspondantes du tableau :

- si coût f > coût g $\qquad f \leq g$
- si coût f = coût g $\qquad f < g$

D'où la règle de mise en place d'une colonne f de coût k:

Pour toutes les colonnes c_i pour lesquelles le coût est inférieur à k voir si l'inégalité :

$$c_i \geqslant f$$

est vérifiée; s'il existe un i_o tel que $c_{i_o} > f$, f est améliorable, c_{i_o} est son amélioration et f n'est pas retenue.

- Pour toutes les colonnes c_j de coût égal à k on vérifie
- - s'il existe un j_o tel que :

$$c_{j_o} > f$$

Si oui c_{j_o} est une amélioration de f et f est éliminée.

Sinon : on cherche s'il existe un j_1 tel que :

$$c_{j_1} \leqslant f$$

Si c_{j_1} existe elle est amélioré par f et on la remplace par celle-ci.

I - 3 - Exemple de mise en oeuvre.

Soit à réaliser la fonction :

$$f = xy + xz + xtu + yzt + yzu$$

Nous formons le tableau de f et fabriquons les réseaux de coût 1 comme combinaisons de variables :

m_1 = Maj (x, y, z) ; m_2 = Maj (x, y, t) ; m_3 = Maj (x, y, u) ;
m_4 = Maj (x, z, t) ; m_5 = Maj (x, z, u) ; m_6 = Maj (x, t, u) ;
m_7 = Maj (y, z, t) ; m_8 = Maj (y, z, u) ; m_9 = Maj (y, t, u) ;
m_{10} = Maj (z, t, u).

x	y	z	t	u	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
1	1	0	0	0	1	1	1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	1	1	0	0	0	0	0
1	0	0	1	1	0	1	1	1	1	1	0	0	1	1
0	1	1	1	0	1	1	0	1	0	0	1	1	1	1
0	1	1	0	1	1	0	1	0	1	0	1	1	1	1

m_6, améliorée par x, t, u, m_2, m_3, m_4, m_5 est éliminée.

m_7 et m_8 améliorées par y, z, m_1 sont éliminées.

m_{10} égale à m_9 est éliminée.

Il reste donc 6 colonnes de coût 1, non améliorables.

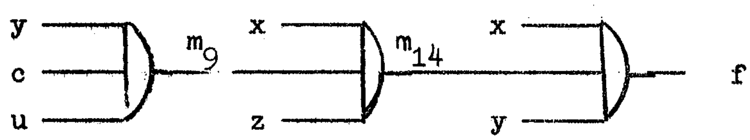
Nous allons fabriquer les colonnes de coût 2 en combinant 2 variables avec une colonne de coût 1.
Afin de ne pas perdre trop de place, nous avons placé dans le tableau ci-dessous, uniquement les colonnes de coût 2 non améliorables et qui sont au nombre de 4.

x	y	z	t	u	m_1	m_2	m_3	m_4	m_5	m_9	m_{11}	m_{12}	m_{13}	m_{14}
1	1	0	0	0	1	1	1	0	0	0	1	1	1	0
1	0	1	0	0	1	0	0	1	1	0	1	1	0	1
1	0	0	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	0	1	1	0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	1	0	1	1	0	1	1	1

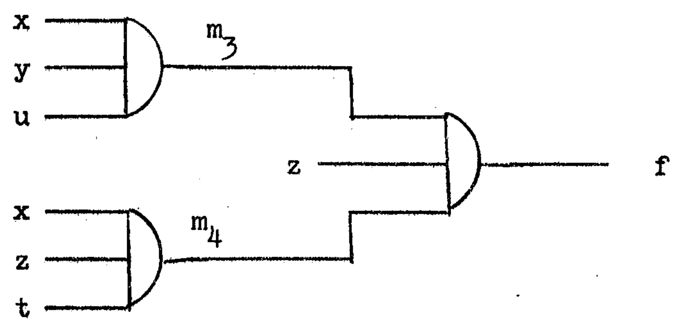
On passe aux réseaux de coût 3; ils sont obtenus de 2 manières :
- combinaison de deux variables avec un réseau de coût 2.
- combinaison de deux réseaux de coût 1 avec une variable.

Dans les 2 cas on obtient plusieurs solutions; donnons en une de chaque type :

$$f = \text{Maj}\ (x,\ y,\ m_{14})$$



$$f = \text{Maj}\ (z,\ m_3,\ m_4)$$



II - METHODES DE SYNTHESE HEURISTIQUE PAR COMPOSITION

Principe commun à toutes les méthodes par composition.

Une fonction réalisable $f(X)$ de n variables sera représentée par son tableau associé tel qu'il est défini au chapitre I.

Si $\varphi(Y)$, $Y \subseteq X$, est susceptible d'entrer dans la réalisation de $f(X)$, $\varphi(Y)$ sera considérée comme une variable nouvelle et adjointe au tableau comme colonne supplémentaire; cette colonne vaudra 1 sur la ligne L_i représentant le monôme m_i si $\varphi(X) > m_i$, 0 dans le cas contraire.

On étudiera alors la fonction $g(\varphi, X)$ telle que si

$$\varphi = \varphi(Y) \quad \text{alors} \quad g\,[\varphi(Y), X] = f(X).$$

Classification des méthodes par composition.

A - Méthodes par élimination de variables

B - Recherche de colonnes obéissant à des critères de distance.

II - A - METHODES PAR ELIMINATION DE VARIABLES.

Définition de la majorité composée.

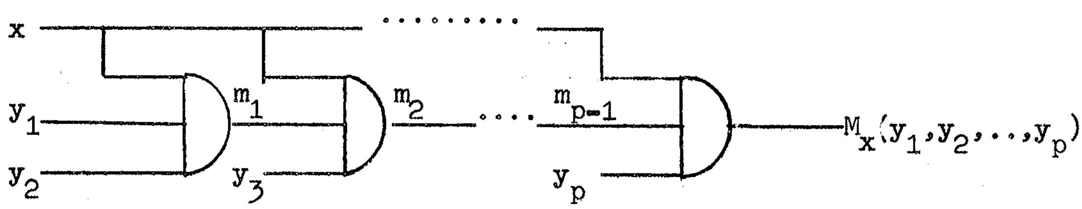
On appelle majorité composée à p + 1 variables la fonction impaire croissante :

$$M_x(y_1, y_2, \ldots, y_p) = x\,(y_1 + y_2 + \ldots + y_p) + y_1\, y_2 \ldots y_p.$$

Il existe pour une telle fonction une réalisation de coût p.

En effet, soit :

$$m_1 = \mathrm{Maj}(x, y_1, y_2)$$
$$m_2 = \mathrm{Maj}\,(x, m_1, y_3)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$m_p = \mathrm{Maj}\,(x, m_{p-1}, y_p)$$



Propriété d'élimination des majorités composées.

Etant donné une fonction f(X), représentée par son tableau, il est possible avec une majorité composée à au plus n-1 variables, d'obtenir une colonne qui par rapport à une colonne x_i choisie du tableau :

- vaut 1 sur toutes les lignes où $x_i = 1$
- vaut 1 sur une ligne déterminée où $x_i = 0$

Soit f(X) représentée par le tableau ci-dessous.

	x_1	x_2	...	x_p	....	x_{n-1}	x_n
L_i	1	1	...	1	...	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
L_j	1	.	.	.	.	.	1
	0	.	.	.	.	.	1
	0	.	.	.	.	.	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	1	.	.	.	.	.	1

Choisissons par exemple la variable x_n et la ligne L_i sur laquelle $x_n = 0$

Soit p le nombre des variables $x_1, x_2, \dots x_p$ valant 1 sur L_i

On a $p \geqslant 2$

Posons $Y = \{x_1, x_2, \dots x_p\}$

Formons $M_{x_n}(Y) = x_n(x_1 + x_2 + \dots + x_p) + x_1 x_2 \dots x_p$

$M_x(Y) = 1$ sur la ligne L_i puisque $x_1 = x_2 = \dots = x_p = 1$ entraine

$x_1 . x_2 \dots x_p = 1$

Soit une ligne L_j sur laquelle $x_n = 1$

D'après la propriété de sous imparité les deux lignes L_i et L_j ont un 1 au moins en commun; supposons que c'est x_1.

D'où : $x_n x_1 = 1$ sur L_j et $M_{x_n}(Y)$ aussi.

$M_{x_n}(Y) > x_n$ et satisfait donc bien aux conditions posées.

II - A - 1 - MAJORITE COMPOSEE : MINIMISATION LOCALE.

II - A - 1 - a - Principe.

On se propose de trouver, pour une fonction f de n variables, un réseau formé d'opérateurs majorité composée et obéissant aux deux lois suivantes :

- Si majorité composée de u_1, u_2, ...u_p intervient dans le réseau alors

$$\text{I} \left\{ \begin{array}{l} \text{1)- Majorité composée de } u_1, u_2, \ldots u_p \text{ élimine } u_1 \\ \text{2)- Cette majorité composée est la colonne de coût minimum} \\ \quad\;\; \text{éliminant } u_1. \end{array} \right.$$

On conviendra que les majorités composées entrant dans le réseau d'une fonction de n variables ont au plus n entrées et sont donc de coût inférieur ou égal à n-2.

Définition d'un pas de l'Algorithme.

On appellera pas de l'algorithme l'ensemble des opérations suivantes :

- S'il y a p colonnes dans la table on recherche l'ensemble de toutes les colonnes de coût minimal éliminant au moins une des p colonnes et ce, de manière à ce que chacune des p colonnes soit éliminée par une colonne nouvelle au moins. Les colonnes nouvelles (au nombre de q par exemple) sont alors adjointes au tableau qui comporte alors p + q colonnes.

Les deux conditions de I vont nous permettre d'obtenir des propriétés très intéressantes car elles diminueront le nombre d'essais et le nombre de colonnes à conserver.

II - A - 1 - b - Convergence.

Soit x une des colonnes initiales du tableau.
Supposons que $x = 0$ sur p lignes.
A la fin du 1er pas il existe une colonne m_1 avec

$$m_1 > x$$

m_1 est une colonne ayant au plus $p - 1$ des p zéros de x. Au bout de p pas au plus il y a ura une colonne n'ayant plus aucun zéro; en effet, au 2ème pas m_2 élimine m_1, donc a au moins deux des zéros de x en moins, etc...

II - A - 1 - c - Propriétés d'élimination.

Théorème 1 -

Si $M_x(Y_1)$ élimine x alors quel que soit $Y_2 \supset Y_1$

on a :

$M_x(Y_2) \leqslant M_x(Y_1)$ sur toute ligne du tableau.

Démonstration.

Soit : $Y_1 = \{y_1, y_2, \dots y_p\}$

$Y_2 = \{y_1, y_2, \dots y_p, y_{p+1}, \dots y_q\}$

Soit une ligne ou $M_x(Y_2) = 1$

2 possibilités : $x = 0$ ou $x = 1$

Si $x = 0$ il faut, pour que $M_x(Y_2) = 1$ que l'on ait :

$$y_1 = y_2 = \dots = y_p = y_{p+1} = \dots = y_q = 1$$

d'où $y_1 = y_2 = \dots = y_p = 1$ entraine

$M_x(Y_1) = 1$ sur cette ligne

Si $x = 1$ alors par hypothèse :

$$M_x(Y_1) = 1$$

Corollaire. Si $M_x(Y_1)$ élimine x alors il est inutile de fabriquer $M_x(Y_2)$ pour tout $Y_2 \supset Y_1$.

En effet, $M_x(Y_1) \geqslant M_x(Y_2)$

et comme $Y_2 \supset Y_1$ (inclusion stricte) on a :

Coût : $[M_x(Y_2)]$ > Coût $[M_x(Y_1)]$

Théorème 2.

Si $M_x(Y_1, Y_2)$ élimine Y_1 alors $M_x(Y_1)$ élimine également Y_1 et coûte moins cher.

Démonstration.

$$Y_1 = \{y_1, y_2, \dots , y_p\}$$

$$Y_2 = \{y_{p+1}, \dots , y_q\}$$

Soit une ligne où $y_j = 1$ $\quad 1 \leqslant j \leqslant p$

cela entraine $M_x(Y_1, Y_2) = 1$ sur cette ligne.

Si $x = 1$ alors $M_x(Y_1) = 1$ également sur cette ligne puisque $y_j \in Y_1$ et $y_j = 1$ sur cette ligne.

Si $x = 0$ alors $M_x(Y_1, Y_2) = 1$ sur cette ligne entraine $y_1 = y_2 = \dots y_p = y_{p+1} = \dots = y_q$ sur cette ligne d'où :

$y_1 = y_2 = \dots y_p = 1$ entraine

$M_x(Y_1) = 1$ sur cette ligne.

Théorème 3.

Si $M_x(Y)$ élimine y_j seul $(y \in Y)$ alors il existe un $y_k \in Y$ tel que Maj (x, y_j, y_k) élimine y_j et coûte moins cher que $M_x(Y)$.

Il s'agit d'un cas particulier de la propriété précédante, le cas où $p = 1$.

Soit une ligne où $y_j = 1$

2 cas possibles : $x = 1$, $x = 0$

Si $x = 1$ alors $Maj(x, y_j, y_k) = 1$ sur cette ligne quelle que soit y_k

Si $x = 0$ alors $M_x(Y) = 1$ entraine

$y_1 = \ldots = y_j = \ldots = y_q = 1$ sur cette ligne et ceci est vrai pour toute ligne où $x = 0$, $y_j = 1$.

Conséquence : Quelle que soit $y_k \in Y$ et $y_k \neq y_j$

on a sur une ligne où $x = 0$ $y_j = 1$:

$$Maj\ (x, y_j, y_k) = 1$$

Enfin, il existe au moins une ligne où $y_j = 0$ et $M_x(Y) = 1$ cela entraine

- $x = 1$ sur cette ligne
- il existe un $y_k \in Y$ et valant 1 sur cette ligne;

en choisissant cet y_k on a :

$$Maj\ (x, y_j, y_k) > y_j$$

II - A - 1 - d - Application : mise en oeuvre de la méthode.

L'utilisation systématique du théorème 2 conduirait à des structures différentes de celles fixées dans l'hypothèse : $M_x(Y)$ figure dans le réseau si elle élimine x.

Par contre l'utilisation du théorème 1 va nous permettre d'éviter des essais inutiles.

En effet, soit $M_x(Y)$ une colonne.

1) - Si $M_x(Y)$ élimine x il est inutile de calculer la valeur de $M_x(Y, Z)$.

2) - Si $M_x(Y)$ est éliminée par $M_x(Z)$, $M_x(Z)$ éliminant x il est inutile de fabriquer $M_x(Y, Y_1)$.

En effet, le seul cas où $M_x(Y, Y_1)$ pourrait figurer dans le tableau est :

$M_x(Y, Y_1)$ élimine x.

Si $x = 1$ $M_x(Y, Y_1) = 1$ et $M_x(Z) = 1$

Si $x = 0$ $M_x(Y, Y_1) = 1$ entraine $Y = 1$ soit $M_x(Y) = 1$
Mais $M_x(Z) > M_x(Y)$ entraine pour cette ligne $Z = 1$ d'où :

$$M_x(Z) \geqslant M_x(Y, Y_1)$$

Pour éviter que le coût de $M_x(Y, Y_1)$ ne soit inférieur à celui de $M_x(Z)$ on fabriquera les colonnes par coûts croissants.

Fabrication des colonnes : liste de référence.

Pour toute colonne x_i du tableau initial on fabriquera les majorités composées $M_{x_i}(Y)$ par coûts croissants.

On placera dans le tableau les colonnes $M_{x_i}(Y)$ éliminant x_i et étant de coût minimal.

D'autre part, on constituera une liste de référence. Cette liste comportera les noms des colonnes $M_{x_i}(Y)$:

- qui éliminent x_i
- qui sont éliminées par une colonne éliminant x_i.

D'après ce qui a été vu plus haut, soit à fabriquer $M_{x_i}(Z)$. Si $M_x(Y)$ avec $Y \subset Z$ figure dans la liste de référence, il est inutile de fabriquer $M_x(Z)$.

Au pas suivant, il est inutile de fabriquer les majorités composées relatives aux variables initiales. On fabrique seulement celles relatives aux colonnes nouvelles trouvées etc... jusqu'à obtenir une colonne de 1.

II - A - 1 - e - Exemple :

$$f = xyt + xzt + yzu + xzu + tu$$

x	y	z	t	u	m_1	m_2	m_3	m_4
1	1	0	1	0	1	1	1	0
1	0	1	1	0	1	1	0	1
0	1	1	0	1	1	0	1	1
1	0	1	0	1	1	1	0	1
0	0	0	1	1	0	1	1	1

Majorités composées relatives à x :

-M(x, y, z) élimine x et est à mettre dans le tableau donc dans la liste de référence également.

$m_1 = M\,(x, y, z)$

- M(x, y, t) est éliminée par m_1, on la met dans la liste de référence.

- M(x, y, u), M(x, z, t), M(x, z, u) sont éliminées par m_1 et prennent place dans la liste de référence.

- M(x, t, u) élimine x et est à retenir tant dans le tableau que dans la liste de référence :

$m_2 = M\,(x, t, u)$

Relativement à x, toutes les majorités de coût 1 figurent dans la liste de référence; il est donc inutile de fabriquer celles de coût supérieur.

Majorités composées relatives à y.

Toutes celles de coût 1 contenant x ont déjà été examinées. Reste pour le coût 1 :

$M(y, z, t)$, $M(y, z, u)$ sont éliminées par m_1.

$m_1 = M(x, y, z) = M_y\ (x, z)$

m_1 figure donc également dans la liste de référence relative à y; d'où $M(y, z, t)$, $M(y, z, u)$ également puisque $M_y(x, z)$ élimine y.

$M(y, t, u) = m_3$ élimine y et sera retenue.

Toutes les majorités de coût 1 relatives à y, figurant dans la liste de référence, il est inutile de calculer celles de coût supérieur.

Majorités composées relatives à z.

Toutes celles contenant x et y ont été vues: ce sont m_1 qui élimine z et les autres sont éliminées par m_1; elles figurent donc toutes dans la liste de référence relative à z.

Reste Maj (z, t, u) qui élimine z et est à retenir

$m_4 = M\ (z, t, u)$

Relativement à z toutes les majorités de coût 1 figurent dans la liste de référence; il est inutile de fabriquer les autres.

Majorités composées relatives à t.

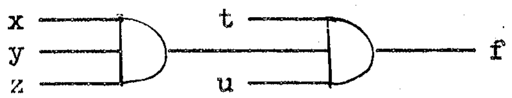
m_2 élimine t et est de coût 1.
C'est la seule majorité de coût 1 éliminant t ; toutes les autres sont à rejeter comme étant de coût plus élevé.

Majorités composées relatives à u.

m_4 élimine u; c'est la seule majorité de coût 1 éliminant u; il est inutile d'examiner les autres.

Pas suivant.

Parmi les majorités composées relatives à m_1 on trouve Maj (m_1, t, u) qui est une solution pour f.



II - A - 2 - METHODE PAR SELECTION DE MAJORITES COMPOSEES : PROGRAMME MAJOCO

Introduction :

La méthode précédante offre le même inconvénient pratique que celle de minimisation arborescente : au cours de la synthèse on est amené à conserver et par suite à manipuler un nombre considérable de colonnes ce qui conduit à des programmes lents et pouvant traiter un petit nombre de variables seulement.

La méthode présentée ci-dessous, conduit à des solutions peut être plus coûteuses que celles fournies par la méthode précédente mais conduit à un programme beaucoup plus rapide et permettant de traiter un plus grand nombre de variables.

II - A - 2 - a - Principe de la méthode.

Soit une fonction $f(X)$ et son tableau; on construira des colonnes nouvelles de la manière suivante :

- si $x_1, x_2, \ldots, x_m$ sont les colonnes du tableau à un pas donné, pour tout i, $1 \leqslant i \leqslant m$, on recherche la colonne $M_{x_i}(Y_i)$ qui élimine x_i et qui est de coût minimal; on forme donc ainsi m colonnes.

A l'aide d'un critère de coût et de qualité on choisit l'une de ces colonnes soit $M_{x_j}(Y_j)$. Cette colonne est alors adjointe au tableau comme colonne nouvelle; par construction elle élimine x_j; on fait toutes les éliminations possibles sur le tableau puis le cycle recommence jusqu'à obtention d'une colonne de 1.

Convergence.

Comme dans la méthode précédente, au bout d'un nombre de pas au plus égal au nombre de zéros du tableau initial, on obtient une colonne de 1.

II - A - 2 - b - Mise en oeuvre de la méthode.

Soit la fonction réalisable $f(X)$ représentée par son tableau T.

Considérons le sous tableau T_i correspondant à la variable x_i et obtenu de la manière suivante:

T_i est formé de l'ensemble des lignes du tableau T pour lesquelles $x_i = 1$; on fait $x_i = 0$ sur ces lignes dans le tableau T_i, c'est-à-dire qu'on supprime la colonne x_i.

Sur le sous-tableau T_i, on cherche l'ensemble de colonnes $x_1, x_2, \ldots, x_p$ satisfaisant aux conditions suivantes :

- la fonction $x_1 + x_2 + \dots + x_p$ vaut 1 sur chaque ligne de T_i.

- il existe au moins une ligne du tableau T sur laquelle la fonction $x_1 . x_2 \dots . x_p = 1$

- il n'existe pas d'ensemble $x_{i_1}, x_{i_2}, x_{i_q}$ satisfai- aux deux conditions ci-dessus avec $q < p$.

Si l'ensemble des colonnes $x_1, x_2, \dots x_p$ satisfait à ces conditions on a :

$$[M_{x_i}(x_1, \dots x_p)] . f > x_i f ;$$

la colonne x_i est éliminée par $M_x(x_1, \dots, x_p)$.

Toutefois, nous n'allons pas adjoindre au tableau T les colonnes M_{x_i} pour toutes les valeurs de i.

<u>Introduction d'un critère de coût et de qualité.</u>

On prend comme coût de la majorité, celui trouvé au début de I - 1, c'est-à-dire p-1 si la majorité composée est à p + 1 variables.

Nous avons vu que, relativement à la colonne x_i du tableau T, la majorité composée $M_{x_i}(Y_i)$ avait la propriété suivante :

- c'est la majorité composée de coût minimal éliminant x_i.

- soit alors k_i le coût de cette colonne

- soit l_i le nombre de colonnes du tableau T éliminées par $M_{x_i}(Y_i)$, on a toujours $l_i \geqslant 1$ puisque $M_{x_i}(Y_i)$ élimine x_i;

à la colonne M_{x_i} on attache la quantité :

$$c_i = \frac{k_i}{l_i}$$

Ceci étant fait pour tout i on choisit la colonne M_{x_j} pour laquelle c_j est minimum.

Cette colonne est la seule colonne nouvelle que l'on adjoint au tableau T à ce pas. On fait sur le tableau T toutes les éliminations possibles et le cycle recommence jusqu'à obtention d'une colonne de 1.

II - A - 2 - c - Exemple.

Soit la fonction réalisable suivante dont on ne donne que la borne inférieure :

$$f(x, y, z, t, u) = xzt + xyt + yzu + xzu + tu$$

On la représente par son tableau ci-dessous.

x	y	z	t	u	
1	0	1	1	0	L_1
1	1	0	1	0	L_2
0	1	1	0	1	L_3
1	0	1	0	1	L_4
0	0	0	1	1	L_5

Nous allons rechercher les majorités composées éliminant x, y, z, t, u par la méhode décrite ci-dessus.

- Elimination de x; le tableau est formé des lignes L_1, L_2, L_4

y	z	t	u	
0	1	1	0	L_1
1	0	1	0	L_2
0	1	0	1	L_4

y + z couvre le tableau; de plus yz = 1 sur L_3 où x = 0 .

On retiendra donc :

$m_x = \text{Maj}(x, y, z)$ coût 1

On constate en adjoignant cette colonne que x, y, z sont éliminés ; d'où $c_x = \frac{1}{3}$

Pour l'élimination de y et de z on trouve la même fonction.

- Elimination de t, lignes L1, L2, L5

x	y	z	u
1	0	1	0
1	1	0	0
0	0	0	1

On trouve $x + u$ qui convient d'où :

$$m_t = \text{Maj}\,(x,t,u) \text{ - Coût } 1$$

On constate sur le tableau de f que m_t élimine x et t d'où :

$$c_t = \frac{1}{2}$$

- Elimination de u : lignes L3, L4, L5

x	y	z	tt
0	1	1	0
1	0	1	0
0	0	0	1

On trouve $z + t$ qui convient :

$$m_u = \text{Maj}\,(z,\ t,\ u) \text{ - Coût } 1$$

m_u élimine z et u d'où :

$$c_u = \frac{1}{2}$$

$m_x = m_y = m_z = \text{Maj}\,(x,\ y,\ z)$ a un facteur de qualité minimum.

On pose $m_1 = \text{Maj}\,(x,\ y,\ z)$

x	y	z	t	u	m_1	
1	0	1	1	0	1	L1
1	1	0	11	0	1	L2
0	1	1	0	1	1	L3
1	0	1	0	1	1	L4
0	0	0	1	1	0	L5

Après élimination de x, y, z, on a L1 = L2; L3 = L4.

Il reste t, u, m_1 et

$\text{Maj}\,(t,\ u,\ m_1)$ couvre le tableau; c'est donc une solution :

$$f < \text{Maj}\,[t,\ u,\ \text{Maj}\,(x,\ y,\ z)]$$

II - B - ADJONCTION DE COLONNES OBEISSANT A DES CRITERES DE DISTANCE.

Préliminaire. Dans la représentation d'une fonction réalisable par son tableau, une solution est représentée par une colonne de 1. Nous allons sélectionner par différents critères des colonnes ayant le plus de 1 possible.

Définition . Soit x une colonne du tableau d'une fonction réalisable f; on appellera distance de x à f le nombre de lignes de f pour lesquelles :

$$x = 0$$

Une solution de f sera donc une colonne à distance 0.

II - B - 1 - Meilleure Majorité : Programme MAJMAX.

Cette méthode, qui conduit à d'excellents résultats offre l'avantage d'être très simple tant dans son principe que dans la mise en oeuvre.

II - B - 1 - a - Principe de la méthode.

Etant donné un ensemble de colonne $\{c_1, c_2, \ldots c_p\}$ du tableau d'une fonction f on adjoindra une colonne nouvelle c_{p+1} formée de la manière suivante :

$$c_{p+1} = \mathrm{Maj}\ (c_i, c_j, c_k)$$

$$1 \leqslant i \leqslant p \qquad 1 \leqslant j \leqslant p; \quad 1 \leqslant k \leqslant p.$$

avec les 2 conditions suivantes :

a) quel que soit $1 \leq r \leq p$ on n'a pas

$$c_r \geq c_{p+1}$$

b) i, j, k sont choisis de telle sorte que c_{p+1} soit à une distance minimale de f parmi les colonnes nouvelles formées.

Si $c_{p+1} = f$ c_{p+1} est une solution.

Sinon on recommence pour $p + 1$.

II - B - 1 - b - Mise en oeuvre de la méthode.

Soient $x_1, x_2, \ldots, x_n$ les colonnes du tableau de f représentant les variables initiales.

On forme toutes les majorités de 3 colonnes :

$u_1 = \text{Maj}\ (x_1, x_2, x_3)$

....

$u_k = \text{Maj}\ (x_{n-2}, x_{n-1}, x_n)$

Pour chaque colonne u_k on calcule la distance de u_k à f soit $d(u_k)$.

On sélectionne alors la colonne u_{k1} de distance minimum à f.

u_{k1} est comparée à toutes les colonnes du tableau soit à $x_1, x_2, \ldots, x_n$.

S'il existe une colonne x_r telle que

$$x_r \geq u_{k1}$$

u_{k1} est rejetée et on prend u_{k2}, venant immédiatement après dans l'ordre des distances croissantes.

Soit u_{kj} la colonne telle que :

- u_{kj} n'est éliminé par aucune colonne du tableau.

- s'il existe une colonne u_r dont la distance à f est inférieure à celle de u_{kj} alors u_r est éliminé par une colonne x_s du tableau de f ($x_s \geqslant u_r$).

La colonne u_{kj} est alors adjointe au tableau de f; elle est appellée x_{n+1}.

Si x_{n+1} a des 1 sur toutes les lignes du tableau c'est une solution de f.

Sinon on remplace n par $n+1$ et le cycle recommence jusqu'à obtention d'une colonne de 1.

II - B - 1 - c - Convergence.

Nous avons vu que toute colonne nouvelle inférieure à une colonne du tableau est éliminée; si une colonne nouvelle couvre seulement 2 lignes du tableau elle sera éliminée par une variable.

D'autre part, nous avons vu qu'étant donné 3 lignes il y a toujours une majorité simple qui les couvre.

Conséquence :

Si le tableau de f est à P lignes, au bout de c_p^3 pas au plus, tout ensemble de 3 lignes sera couvert; au pas suivant on trouvera au moins une colonne couvrant 4 lignes, etc...

Exemple : $f = xy + xu + xzt + yzu + ytu$

Dressons le tableau de f.

x	y	z	t	u	m_1
1	1	0	0	0	1
1	0	0	0	1	1
1	0	1	1	0	0
0	1	1	0	1	1
0	1	0	1	1	1

Considérons la liste de toutes les majorités et donnons leurs distances à f.

Fonctions	distances à f
Maj (x, y, z)	2
Maj (x, y, t)	2
Maj (x, y, u)	1
Maj (x, z, t)	4
Maj (x, z, u)	2
Maj (x, t, u)	2
Maj (y, z, t)	2
Maj (y, z, u)	3
Maj (y, t, u)	3
Maj (z, t, u)	2

On constate que Maj (x, y, u) est à distance minimum de f et n'est pas éliminé par une des variables.

On pose m_1 = Maj (x, y, u)

et on l'adjoint au tableau (à droite du double trait, tableau ci-dessus).

m_1 élimine y et u; après réduction des lignes il reste le tableau ci-dessous:

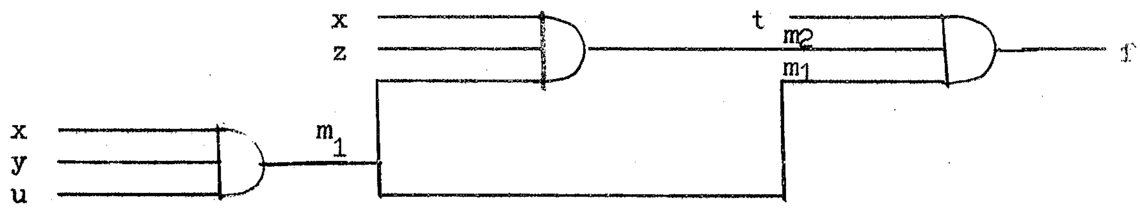
x	z	t	m_1	Majorités	Distance à f
1	0	0	1	Maj(x, z, t)	3
1	1	1	0	Maj(x, z, m_1)	1
0	1	0	1	Maj(x, t, m_1)	1
0	0	1	1	Maj(z, t, m_1)	1

Il y a 3 majorités équivalentes; le programme choisit la première.

m_2 = Maj (x, z, m_1)

En adjoignant m_2 au tableau précédent, x et z sont éliminés et il reste :

$$f = \text{Maj}\ (t,\ m_1,\ m_2)$$



Les performances de cette méthode seront vues dans la partie programmation. On peut prévoir que le programme sera assez simple; les résultats sont tout à fait raisonnables et les temps d'exécution très acceptables.

II - B - 2 - SOMMES ET PRODUITS - PROGRAMME MAJORF.

II - B - 2 - a - Principe de la méthode.

Soit f(x, y, X) une fonction réalisable donnée par son tableau T. On se propose de rechercher une colonne u telle que m = Maj (x, y, u) élimine simultanément x et y.

Pour que m satisfasse à cette condition, il faut et il suffit que : u = 1 sur les lignes de T sur lesquelles :

$$\begin{cases} x + y = 1 \\ x\ y = 0 \end{cases}$$

II - B - 2 - b - <u>Mise en oeuvre de la méthode.</u>

Appelons T_o le tableau de $f(X)$.

Pour tout couple de colonnes x_{o_i} , y_{o_i} on forme

$\sigma_{o_i} = x_{o_i} + y_{o_i}$ et on calcule la distance d_{o_i} de σ_{o_i} à f.

On choisit le couple x_o, y_o pour lequel d_o est minimum. Si plusieurs couples satisfont à ce critère, on choisit le couple x_o, y_o pour lequel la fonction $x_o \cdot y_o$ est à distance minimale de f. (En particulier tout couple x_o, y_o tel que $x_o \cdot y_o = 0$ sur toutes les lignes de T_o est rejeté).

Soit alors x_o, y_o le couple choisi;

On extrait de T_o, le tableau T_1 formé des lignes de T_o pour lesquelles :

$$\begin{cases} x_o + y_o = 1 \\ x_o \cdot y_o = 0 \end{cases}$$

S'il existe z tel que z = 1 sur toutes les lignes de T_1 alors on forme m = Maj (x_o, y_o, z) que l'on adjoint au tableau T_o ;
m élimine x_o et y_o.

Sinon on recherchera sur T_1 un couple x_1, y_1 par les mêmes règles que celles qui ont permis de trouver x_o, y_o.

On recommence ce processus jusqu'à ce qu'on ait un tableau T_p défini par $x_{p-1} + y_{p-1} = 1$; $x_{p-1} \cdot y_{p-1} = 0$ et tel qu'il existe $z_p = 1$ sur toutes les lignes de T_p.

On forme alors $m = \text{Maj}\ (x_{p-1}, y_{p-1}, z_p)$ que l'on adjoint au tableau T_o.

On procède à toutes les éliminations possibles. Puis le cycle recommence jusqu'à obtension d'une colonne valant 1 sur toutes les lignes de T_o.

II - B - 2 - c - Convergence.

Appelons T_i le tableau obtenu au pas i de la manière suivante :

Sur T_i le choix du couple x_i, y_i conduit à un tableau couvert par une colonne z_i.

D'après ce qui a été dit plus haut on forme alors

$m_i = \text{Maj}\ (x_i, y_i, z_i)$

et sur le tableau T_i, m_i élimine x_i et y_i; (mais pas nécessairement sur le tableau de f).

Soit l_i le nombre de lignes du tableau T_i et C_i le nombre de lignes de T_i sur lesquelles $m_i = 1$.

On a évidemment $l_i \geqslant c_i$.

a) - Quel que soit i on peut affirmer :

$l_i \geqslant c_i \geqslant 3$

b) - Si q est le nombre de lignes du tableau de f on aura :

si $i > C_q^3 \Longrightarrow l_i \geqslant c_i > 3$

si $i \geqslant C_q^3 + C_q^4 \Longrightarrow l_i \geqslant c_i > 4$

au bout d'un nombre fini de pas, on aura :

$$l_i = c_i = q.$$

il y aura une colonne valant 1 sur toutes les lignes du tableau de f.

II - B - 2 - d - Exemple.

$$f = xyz + xvw + yuw + zuv$$

Formons ci-dessous le tableau de f.

	x	y	z	u	v	w	m_1
L_1	1	1	1	0	0	0	1
L_2	1	0	0	0	1	1	1
L_3	0	1	0	1	0	1	1
L_4	0	0	1	1	1	0	0

Les meilleures sommes sont $x + u$, $y + v$, $z + w$; mais les produits correspondant valant 0 sur toutes les lignes, ces couples sont rejetés.

Les autres couples sont alors équivalents.

Prenons le couple x, y. Le tableau défini par $x + y = 1$; $xy = 0$ est formé des lignes L_2 et L_3 sur lesquelles $w = 1$.

On pose donc $m_1 = \text{Maj}(x, y, w)$ que l'on adjoint au tableau de f. x et y doivent alors être éliminés mais on constate que w est éliminé également. (colonne à droite du double trait sur le tableau de f ci-dessus).

On obtient le tableau ci-dessous :

	z	u	v	m_1	m_2
L_1	1	0	0	1	0
L_2	0	0	1	1	1
L_3	0	1	0	1	1
L_4	1	1	1	0	1

On trouve trois sommes maximales : $z + m_1$, $u + m_1$, $v + m_1$; ces trois sommes sont équivalentes au point de vue produits. Prenons $z + m_1$.

Le tableau défini par les lignes où $z + m_1 = 1$; $z \cdot m_1 = 0$ est formé des lignes L_2, L_3, L_4.

	z	u	v	m_1
L_2	0	0	1	1
L_3	0	1	0	1
L_4	1	1	1	0

z est éliminé sur ce tableau.

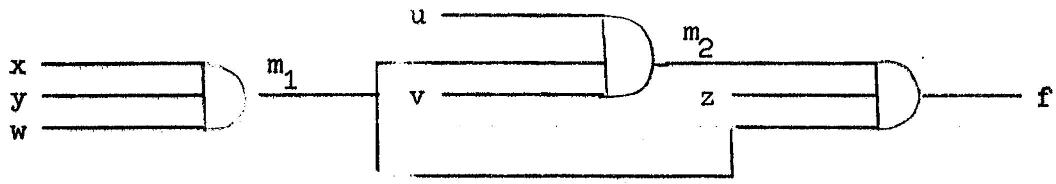
On trouve les 3 couples équivalents $(u.v)$,$(u.m_1)$,$(v.m_1)$.

Choisissons u, v; les lignes où $u + v = 1$, $u \cdot v = 0$ sont L_2 et L_3 sur lesquelles $m_1 = 1$.

On forme donc $m_2 = \text{Maj}\ (u, v, m_1)$ que l'on adjoint au tableau de f. (à droite du double trait).

u et v sont éliminés et il reste 3 colonnes : z, m_1, m_2; on a :

$$f = \mathrm{Maj}\ (z,\ m_1,\ m_2)$$



II - B - 2 - e - Modification de la méthode : Programme MAJORF.

La méthode exposée ci-dessus nécessite en fait une place considérable en machine par suite du nombre de sous-tableaux nécessaires; il a donc paru préférable de la modifier de la manière suivante :

- Si T est le tableau de f on définira 2 colonnes x_o y_o par les mêmes règles que précédemment ce qui nous conduit à un tableau T_1.

Sur T_1 on cherchera la majorité de 3 colonnes située à distance minimale de T_1; cette majorité sera adjointe à T_o et le cycle recommence.

La convergence sera assurée de la même manière : il suffit d'appeler l_i le nombre de lignes de T_1 et c_i le nombre de lignes de T_1 couvertes par la majorité choisie.

Exemple : $f = xy + xz + xtu + yzt + yzu$

Dressons le tableau ci-dessous de la fonction :

	x	y	z	t	u	m_1
L_1	1	1	0	0	0	1
L_2	1	0	1	0	0	1
L_3	1	0	0	1	1	0
L_4	0	1	1	1	0	1
L_5	0	1	1	0	1	1

Les meilleurs couples sont $x + y$ et $x + z$ équivalents au point de vue produits.

Choisissons $\{x, y\}$; T_1 est alors défini par les lignes L_2, L_3, L_4, L_5 ;

x	y	z	t	u
1	0	1	0	0
1	0	0	1	1
0	1	1	1	0
0	1	1	0	1

Les meilleures majorités sont à distance 1. Choisissons par exemple :

$$m_1 = \mathrm{Maj}\ (x, y, z)$$

Adjoignons m_1 au tableau de f (à droite du double trait). On constate que m_1 élimine y et z. Il reste le tableau ci- dessous :

	x	t	u	m_1	m_2
l_1	1	0	0	1	1
l_2	1	1	1	0	1
l_3	0	1	0	1	1
l_4	0	0	1	1	0

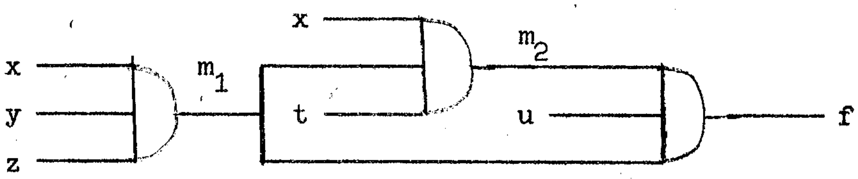
On trouve plusieurs couples équivalents :

$x + m_1$, $t + m_1$, $u + m_1$; Choisissons par exemple $u + m_1$.

Le tableau défini par u, m_1 est formé des lignes l_1, l_2, l_3 couvertes par

$$m_2 = \mathrm{Maj}\ (x, t, m_1)$$

On a alors $f = \mathrm{Maj}\ (u, m_1, m_2)$



III - METHODES DE FRACTIONNEMENT.

Rappelons que l'on appelle méthode de fractionnement une méthode consistant à trouver pour une fonction réalisable f(X), un ensemble de 3 fonctions φ_1, φ_2, φ_3 telles que Maj (φ_1, φ_2, φ_3) soit une représentation de f(X) et telles que la même méthode, appliquée à φ_1, φ_2, φ_3 etc ... conduise aux variables.

III - 1 - FORMES SYSTEMATIQUES.

Il s'agit de formules analogues aux formules de Lagrange.

Exemple : Les formules de Cohn et Lindaman dont nous donnons ici un exemple [2].

$$f(x, y, z, X) = \text{Maj}\,\{\text{Maj}[x,y,f(x,x',z,X)],\ \text{Maj}[x',y',f(x,x'z,X)],\ f(x, x, z, X)\}.$$

III - 1 - a - Forme systématique pour fonctions croissantes.

Soit f(x, y, z, X) une fonction non forcément impaire; si elle est croissante (ou monotone) en x, y, z on peut toujours la mettre sous la forme :

$$f(x,y,z,X) = \text{Maj}\,[f(x,x,z,X),\ f(x,y,y,X),\ f(z,y,z,X)] \qquad (1)$$

où

$$f(x,y,z,X) = \text{Maj}\,[f(y,y,z,X),\ f(x,z,z,X),\ f(x,y,x,X)] \qquad (2)$$

Nous ferons la démonstration uniquement pour la formule (1)

La formule étant respectée par toute permutation circulaire de x, y, z, il suffit d'examiner les 2 cas :

$$x = y = z \; ; \qquad x = y = z'.$$

a) - $x = y = z$ la formule est évidente.

b) - $x = y = z'$; $f(x,y,z,X)$ étant croissante en x, y, z on a :

$$f(x, y, y, X) = f(x,x,x,X) > f(x,x,x',X) > f(x',x,x,X)$$

d'où : $\mathrm{Maj}[f(x,x,x,X),\ f(x,x,x',X),\ f(x',x,x,X)] = f(x,x,x',X) = f(x,y,z,X).$

On voit que si $f(x,y,z,X)$ est croissante par rapport à toutes ses variables la répétition de cette formule conduit aux variables.

<u>Exemple</u> : $f = xy + xz + xtu + yzt + yzu$

$y = x$	$f(x,x,z,t,u) = x$
$x = z$	$f(z,y,z,t,u,) = z$
$z = y$	$f(x,y,y,t,u,) = xy + xtu + yt + yu.$

$$f = \mathrm{Maj}\,[x,\ z,\ (yx + yt + yu + xtu)]$$

$$yx + yt + yu + xtu = \mathrm{Maj}\,[y,\ x,\ \mathrm{Maj}\,(y,t,u)]$$

$$f = \mathrm{Maj}\,\{x,z,\ \mathrm{Maj}\,[x,\ y,\ \mathrm{Maj}(y,tu,)]\,\}$$

III - 1 - b - <u>Forme systématique pour fonctions quelconques.</u>

Soit $f(x, y, z, X)$ une fonction quelconque.

Considérons les 3 fonctions φ_1, φ_2, φ_3 obtenues de la manière suivante :

Dans l'expression de f on considère x et x' comme 2 variables indépendantes; de même pour y et y', z et z'.

φ_1 est obtenue en remplaçant x par y, y' par x' ce que nous noterons :

$$x \rightarrow y$$
$$y' \rightarrow x'$$

φ_2 est obtenue en faisant :

$$y \rightarrow z$$
$$z' \rightarrow y'$$

φ_3 est obtenue en faisant :

$$z \rightarrow x$$
$$x' \rightarrow z'$$

On a alors :

$$\text{Maj}\ (\varphi_1,\ \varphi_2,\ \varphi_3) = f$$

<u>Démonstration</u> :

$$f = xyz\ A_1 + xyz'A_2 + xy'zA_3 + x'yzA_4 + xy'z'A_5 + x'yz'A_6 + x'y'zA_7 + x'y'z'A_8$$

$$\varphi_1 = yzA_1 + yz'A_2 + x'yzA_3 + x'yzA_4 + x'yz'A_5 + x'yz'A_6 + x'zA_7 + x'z'A_8$$

$$\varphi_2 = xzA_1 + xy'zA_2 + xy'zA_3 + x'zA_4 + xy'A_5 + x'y'zA_6 + x'y'zA_7 + x'y'A_8$$

$$\varphi_3 = xyA_1 + xyz'A_2 + xy'A_3 + xyz'A_4 + xy'z'A_5 + yz'A_6 + xy'z'A_7 + y'z'A_8.$$

En raison de la symétrie circulaire par rapport à x, y, z, il suffit d'étudier les cas :

$$|x,\ y,\ z| = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{vmatrix}$$

Si $x = y = z = 1$ $\quad f = A_1$ $\quad$ le cas est évident

Si $x = y = z = 0$ $\quad f = A_8$ $\quad$ de même;

$x = 0;\ y = z = 1.$ $\quad f = A_4$ $\quad \varphi_1 = A_1 + A_3 + A_4 + A_7$

$$\varphi_2 = A_4$$

$$\varphi_3 = 0$$

$$\text{Maj}\ (\varphi_1, \varphi_2, \varphi_3) = \varphi_1\ \varphi_2 = A_4 = f.$$

$x = y = 0;\ z = 1$ $\quad f = A_7;$ $\quad \varphi_1 = A_7$

$$\varphi_2 = A_4 + A_6 + A_7 + A_8$$

$$\varphi_3 = 0$$

$$\text{Maj}\ (\varphi_1, \varphi_2, \varphi_3) = \varphi_1\ \varphi_2 = A_7$$

Remarques.

1) - φ_1 est croissante en y décroissante en x

φ_2 est croissante en z décroissante en y

φ_3 est croissante en x décroissante en z.

2) - si $f(x,y,z,X)$ en monotone en $x_i \subset X$ alors $\varphi_1, \varphi_2, \varphi_3$ sont également monotones en x_i.

Application à la synthèse.

Partant de $f(x, y, z, X)$ on pourra en appliquant cette forme aboutir à un ensemble de fonctions monotones par rapport à toutes les variables sauf une.

Si pour chaque fonction on dédouble la variable non monoforme, on obtient un ensemble de fonctions monotones; à ces fonctions monotones on applique la formule vue plus haut pour les fonctions croissantes; on aboutit ainsi à un ensemble de fonctions croissantes de 2 variables au plus qui sont alors de la forme :

$$xy = \mathrm{Maj}(x,y,0) \quad \text{ou} \quad x + y = \mathrm{Maj}\,(x,\ y,\ 1)$$

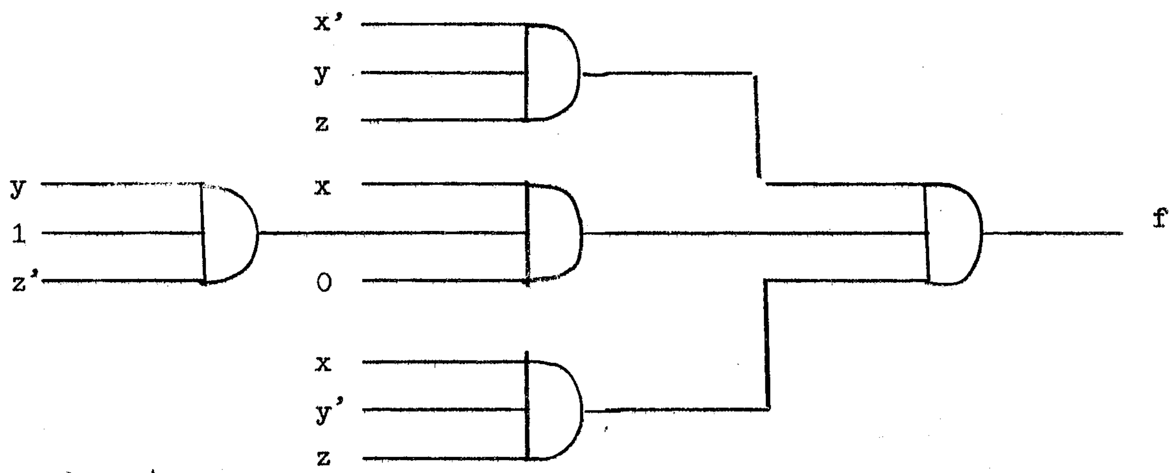
Exemple :

$$f = xyz + xy'z' + x'y'z$$

$$\left.\begin{array}{l} x \rightarrow y \\ y' \rightarrow x' \end{array}\right\} \quad \varphi_1 = yz + x'yz' + x'z \quad = \quad x'y + x'z + yz = \mathrm{Maj}(x',y,z)$$

$$\left.\begin{array}{l} y \rightarrow z \\ z' \rightarrow y' \end{array}\right\} \quad \varphi_2 = xz + xy' + x'y'z \quad = \quad xz + xy' + y'z = \mathrm{Maj}(x,y',z)$$

$$\left.\begin{array}{l} z \rightarrow x \\ x' \rightarrow z' \end{array}\right\} \quad \varphi_3 = xy + xy'z' + xy'z' = xy + xz' = x(y + z')$$



III - 2 - SYNTHESE PAR MAJORANTS DE MONOMES PREMIERS

III - 2 - a - Principe de la méthode.

Soit f(X) une fonction réalisable représentée par son tableau. On se propose de rechercher 3 fonctions sous impaires croissantes :

$\varphi_1 < f(X)$, $\varphi_2 < f(X)$, $\varphi_3 < f(X)$ telles que

$$\text{Maj}\ (\varphi_1, \varphi_2, \varphi_3) = f(X)$$

Pour que 3 fonctions φ_1, φ_2, φ_3 satisfassent à la question il faut et il suffit que toute ligne du tableau de f soit dans deux au moins des 3 fonctions.

Recherche de trois fonctions par des sommes de variables.

On se propose de rechercher φ_1, φ_2, φ_3 de la manière suivante :

Si σ_1, σ_2, σ_3 sont 3 sommes de variables:

φ_1 = somme des lignes de f contenant au moins une variable de σ_1

φ_2 = somme des lignes de f contenant au moins une variable de σ_2

φ_3 = somme des lignes de f contenant au moins une variable de σ_3.

Pour que σ_1, σ_2, σ_3 satisfassent à la condition sur les φ_i, il suffit que tout monôme 1er de f soit dans deux au moins des trois sommes; ce qui revient à dire :

$$\sigma_1 + \sigma_2 > f$$

$$\sigma_1 + \sigma_3 > f$$

$$\sigma_2 + \sigma_3 > f$$

III - 2 - b - Mise en oeuvre de la méthode.

On considérera des sommes de variables Σ_1, Σ_2, Σ_3 et l'on posera :

σ_1 = lettres communes à Σ_1 et Σ_2

σ_2 = - - à Σ_1 et Σ_3

σ_3 = - - à Σ_2 et Σ_3

Théorème : Si 3 sommes de variables Σ_1, Σ_2, Σ_3 sont telles que :

a) - $\Sigma_1 > f(X)$; $\Sigma_2 > f(X)$; $\Sigma_3 > f(X)$

b) - $\mathrm{Maj}\ (\Sigma_1, \Sigma_2, \Sigma_3) = x_1 + x_2 + \ldots + x_n$

alors σ_1, σ_2, σ_3 déduites de Σ_1, Σ_2, Σ_3 par la règle précédente conviennent :

$$\mathrm{Maj}\ (\Sigma_1, \Sigma_2, \Sigma_3) = \Sigma_1\ \Sigma_2 + \Sigma_1\ \Sigma_3 + \Sigma_2\ \Sigma_3$$

$$= \sigma_1 + k_1 + \sigma_2 + k_2 + \sigma_3 + k_3$$

où k_1, k_2, k_3 représentent des sommes de monômes à 2 lettres; comme :

$$\mathrm{Maj}(\Sigma_1, \Sigma_2, \Sigma_3) = x_1 + \ldots + x_n$$

on a :

$$\sigma_1 + \sigma_2 + \sigma_3 = x_1 + \ldots + x_n$$

Soit alors m un monôme premier de $f(X)$; une au moins des 3 sommes σ_1, σ_2, σ_3 est supérieure à m puisque :

$$\sigma_1 + \sigma_2 + \sigma_3 = x_1 + \ldots + x_n$$

Supposons que $\sigma_1 > m$

Si $\sigma_2 > m$ ou $\sigma_3 > m$, le théorème est démontré.

Sinon, puisque $\sigma_1 + \sigma_2 + \sigma_3 = x_1 + \ldots + x_n$ on a : $m^* < \sigma_1$

d'où :
$$\Sigma_1 = m^* + h_1$$
$$\Sigma_2 = m^* + h_2$$

Σ_3 ne contient aucune lettre de m; en effet, s'il en contenait une soit x_i $\Sigma_1 \cup \Sigma_3$ contiendrait x_i donc σ_3 également.

Si Σ_3 ne contient aucune lettre de m on n'a pas $\Sigma_3 > f$ ce qui est contraire à l'hypothèse.

<u>Application :</u> $\Sigma_1 > f(X)$ entraine $\Sigma_1^* < f^*(X)$

On obtient toutes les sommes majorant f en prenant les duaux de tous les monômes croissants compatibles avec f^*.

La recherche de 3 monômes Σ_1, Σ_2, Σ_3 se traduit alors par le problème suivant :

- Etant donné la fonction $g(X) = x_1 + x_2 + \ldots + x_n$ trouver une solution majoritaire pour g en disposant pour entrées, de colonnes représentant des sommes de lettres.

Le tableau de g(X) a pour lignes $x_1, x_2, \ldots x_n$ et pour colonnes les sommes de lettres supérieures à f(X); il y a un 1 en ligne i colonne j si Σ_j contient x_i.

On constate que ce tableau est en fait la table de vérité des points ou f(X) = 1, dont on a permuté lignes et colonnes.

III - 2 - c - Convergence.

Si $\varphi_1 < f(X)$; $\varphi_2 < f(X)$; $\varphi_3 < f(X)$ alors la convergence est évidente: en effet, φ_1, φ_2, φ_3 comptent respectivement moins de lignes que $f(X)$; en les traitant par le même procédé on finira par aboutir à des tableaux à 3 lignes, que l'on couvre par une majorité au plus.

Pour démontrer la convergence il suffit de montrer que le procédé utilisé conduit à 3 fonctions φ_1 ,φ_2,φ_3 strictement inférieures à $f(X)$.

Pour cela nous allons construire une telle solution en utilisant les deux propriétés suivantes.

Théorème 1. Si $f(X)$ est une fonction sous impaire croissante et si m_1 et m_2 sont deux monômes premiers de $f(X)$ ayant une seule lettre commune alors il existe $m > m_1$ m n'étant pas monôme premier de $f^*(X)$.

- m_1 et m_2 existent toujours, sinon tout couple de monômes premiers de $f(X)$ aurait au moins 2 lettres communes et on pourrait supprimer des variables

Soit $m_1 = xn1$; $m_2 = xn2$; n_1 et n_2 n'ayant aucune lettre commune n_1 n'est pas premier de $f^*(X)$; en effet, s'il l'était, on aurait

$n_1^* > f^*(X) \geqslant f(X)$.

or n_1 ne contenant aucune des lettres de n_2 on ne peut avoir $n_1^* > f(X)$.

Théorème 2. Si $f = m + g$ est une fonction sous impaire croissante écrite sous forme de somme de ses monômes premiers, si m est un monôme premier, alors si n est le monôme produit de toutes les lettres absentes de m la fonction

$$h = n + g$$

est également sous impaire croissante.

En effet, si h n'est pas sous impaire croissante un monôme de g et n n'ont aucune lettre commune; ce monôme serait alors formé d'un sous ensemble des lettres de m et m ne serait pas un monôme premier de f.

Application.

Soit Σ_1^* un monôme premier de $f(X)$.
D'après le théorème 1 , on peut toujours trouver $x_i \subset \Sigma_1$ tel que

$$\Sigma_1 = x_i + \Sigma_{11}$$

et Σ_{11}^* n'est pas premier de $f^*(X)$.

Remarque.

Si x_i était premier de $f^*(X)$ on aurait :
$f(X) = x_i\, g$ et le réseau réduit à x_i serait une solution pour $f(X)$; un tel cas sera considéré comme trivial.

Appelons alors m^* la somme de toutes les lettres ne figurant pas dans Σ_1

Posons : $\Sigma_2 = m^* + x_i$

$\Sigma_3 = m^* + \Sigma_{11}$

on a : $f(X) = \Sigma_1^* + g(X)$.

D'après le théorème 2 on a :

$$m^* > g(X)$$

d'où
$$\begin{cases} m^* + x_i > g(X) + \Sigma_1^* \\ m^* + \Sigma_{11}^* > g(X) + \Sigma_1^* \end{cases}$$

Soit : $\Sigma_2 > f(X)$; $\Sigma_3 > f(X)$

<u>Conséquence</u> : Les 3 sommes Σ_1, Σ_2, Σ_3 existent effectivement et satisfont bien à la première condition.

On pose alors :

$$\sigma_1 = \text{lettres communes à } \Sigma_1 \text{ et } \Sigma_2$$

$$\sigma_2 = \text{lettres communes à } \Sigma_1 \text{ et } \Sigma_3$$

$$\sigma_3 = \text{lettres communes à } \Sigma_2 \text{ et } \Sigma_3$$

Soit : $\sigma_1 = x_i$ et puisque x_i n'est pas compatible avec f^* on n'a pas $x_i > f$ d'où :

$$\varphi_1 < f$$

$\sigma_2 = \Sigma_{11}$ Par construction Σ_{11} n'est pas compatible avec $f^*(X)$ donc on n'a pas $\sigma_2 > f(X)$ d'où :

$$\varphi_2 < f$$

$\sigma_3 = m^*$ qui ne contient aucune lettre de Σ_1 donc φ_3 ne contient pas Σ_1^* et

$$\varphi_3 < f.$$

<u>Exemple</u> :

$$f = xy + xtu + xtv + ytv + yuv$$

$$f^* = xy + yt + xtu + yuv + xv$$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}
x	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0
y	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
t	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1	0
u	0	0	1	0	1	0	1	1	0	1	0	0	1	0	1	1	1
v	0	0	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1

On a la solution possible Maj (a_1, a_{10}, a_{15})

$$\Sigma_1 = x + y \qquad \Sigma_{10} = y + t + u + v \qquad \Sigma_{15} = x + t + u + v$$

$$\sigma_1 = y \qquad \sigma_2 = x \qquad \sigma_3 = t + u + v$$

d'où :

$$\varphi_1 = y$$
$$\varphi_2 = x$$
$$\varphi_3 = xtu + xtv + ytv + yuv$$

Avant de reprendre le procédé sur φ_3, essayons de le réduire

x	y	t	u	v
1	0	1	1	0
1	0	1	0	1
0	1	1	0	1
0	1	0	1	1

x est éliminé par t,
y est éliminé par v

d'où :
Maj(t, u, v) est une solution pour φ_3

Soit la solution $f \leqslant$ Maj [x, y, Maj (t,u,v)]

III - 3 - DECOMPOSITION PAR CHANGEMENT DE VARIABLE.

III - 3 - a - Principe de la méthode.

Soit f(X) une fonction réalisable. Rappelons que l'on appelle réalisation de f(X) un réseau dont la sortie :

- est égale à f(X) si f(X) est impaire croissante.
- est supérieure à f(X) si f(X) est sous impaire croissante.

On se propose de remplacer l'étude de la synthèse de la fonction réalisable $f(x, X)$ par celle de la fonction impaire croissante $g[\varphi(x, Y), Z]$ telle que :

$\varphi(x, Y)$ impaire croissante.

$$Y \subset X \; ; \quad Z \subset X ; \quad x \cap Z = \emptyset .$$

La méthode consistera à choisir $\varphi(x, Y)$ de manière que l'étude de $g(\varphi, Z)$ soit plus simple que celle de $f(x, X)$. Pour cela on procèdera de manière à utiliser la remarque suivante :

Si $h(x, X)$ est une fonction réalisable, on peut l'écrire :

$$h(x, X) = x\,\psi_1(X) + \psi_2(X)$$

avec

$$\psi_1^*(X) \geqslant \psi_2(X).$$

Si $\psi_2(X)$ est formé d'un seul monôme alors :

$$x\,\psi_2^*(X) + \psi_2(X)$$

est une réalisation de $h(x, X)$ et c'est précisément la majorité composée dont on sait trouver une réalisation.

III - 3 - b - <u>Mise en oeuvre de la méthode</u>.

$$f(x, X) = x\,\psi_1(X) + \psi_2(X)$$

et on a puisque $f(x, X)$ est réalisable :

$$\psi_2^*(X) > \psi_1(X)$$

Si $\psi_2(X)$ est formé d'un seul monôme on sait trouver une solution. Sinon, on peut toujours séparer en deux les monômes de $\psi_2(X)$; par exemple :

$$\psi_2(X) = k_2(Y_1) + h_2(Y_2) \qquad Y_1 \in X ; \quad Y_2 \in X .$$

On a , $k_2^*(Y_1) \cdot h_2^*(Y_2) \geqslant \psi_1(X)$.

Si l'on pose $\varphi(x, Y_1) = x\, k_2^*(Y_1) + k_2(Y_1)$

φ est impaire croissante.

Soit alors $g(\varphi, Y_2) = \varphi\, h_2^*(Y_2) + h_2(Y_2)$

g est également impaire croissante.

$$g[\varphi(x, Y_1), Y_2] = [xk_2^*(Y_1) + k_2(Y_1)]\, h_2^*(Y_2) + h_2(Y_2)$$
$$= xk_2^*(Y_1) \cdot h_2^*(Y_2) + k_2(Y_1) \cdot h_2^*(Y_2) + h_2(Y_2)$$

Mais $k_2(Y_1) < \psi_2(X) < \psi_2^*(X) < h_2^*(Y_2)$

d'où : $k_2(Y_1) < h_2^*(Y_2)$ soit $k_2(Y_1) \cdot h_2^*(Y_2) = k_2(Y_2)$.

Soit $g[\varphi(x, Y_1), Y_2] = xk_2^*(Y_1) \cdot h_2^*(Y_2) + k_2(Y_1) + h_2(Y_2)$

on a puisque $k_2^*(Y_1) \cdot h_2^*(Y_2) \geqslant \psi_1(X)$

$$g[\varphi(x, Y_1), Y_2] \geqslant f(x, X)$$

$g[\varphi(x, Y_1), Y_2]$ est bien une solution.

Si l'on impose à $k_2(Y_1)$ d'être à n-2 variables au plus alors $\varphi(x, Y_1)$ est une fonction impaire croissante de n-1 variables au plus.

<u>Convergence</u>. $\varphi(x, Y_1)$ étant de n-1 variables au plus, s'étudie par la même méthode.

Quant à $g(\varphi, Y_2) = \varphi\, h_2^*(Y_2) + h_2(Y_2)$;

Si $h_2(Y_2)$ compte un seul monôme le problème est résolu. Sinon il suffit de traiter g comme f en remarquant que $h_2(Y_2)$ compte moins de monômes que $\psi_2(X)$.

Remarque. Si la coupure faite sur ψ_2 est telle que $k_2(Y_2)$ est réduit à un monôme, cette méthode s'apparente à la majorité composée. En fait, il est possible de faire la coupure comme on le veut et l'on verra au chapitre 4 une manière de pratiquer la coupure, permettant de calculer une borne supérieure du coût des fonctions impaires croissantes.

III - 3 - d - Exemple.

$f = xyz + xyt + xyv + yzt + yzu + ytu + ytv + xzuv + xtu + xzt + ztv$

$f = y\,[xz + xt + xv + zt + zu + tu + tv] + xzuv + xtu + xzt + ztv$

Posons :

$$k_2 = z.t.v \qquad k_2^* = z + t + v$$
$$h_2 = xzuv + xtu + xzt \qquad h_2^* = x + zu + zt + tu + tv$$
$$\varphi_1 = y(z + t + v) + ztv \qquad \text{d'où :}$$

$$g_1(\varphi_1, X) = \varphi_1(x + zu + zt + tu + tv) + xzuv + xtu + xzt$$
$$= x\,(\varphi_1 + zuv + tu + zt)\ \varphi_1 zu + \varphi_1 zt + \varphi_1 tu + \varphi_1 tv$$

Prenons :

$$k_2 = \varphi_1(zt + zu + tu) \qquad k_2^* = \varphi_1 + zt + zu + tu$$

Il reste alors $h_2 = \varphi_1 tv$

$\varphi_2 = x\,(\varphi_1 + zu + zt + tu) + \varphi_1 zu + \varphi_1 zt + \varphi_1 tu$

$g_2 = \varphi_2(\varphi_1 + t + v) + \varphi_1 tv = \mathrm{Maj}\,[\varphi_2, \varphi_1, \mathrm{Maj}(\varphi_2, t, v)\,]$

et on a coût f = coût g_2 + coût φ_2 + coût φ_1

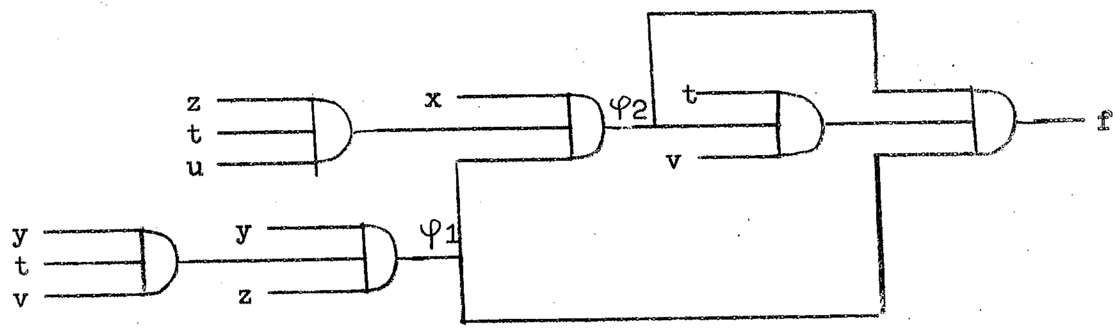
Coût $g_2 = 2$.

$$\varphi_2 = x(\varphi_1 + zt + zu + tu) + \varphi_1 zt + \varphi_1 zu + \varphi_1 tu$$

$$= \text{Maj}\ [x, \varphi_1, \text{Maj}(z, t, u)]$$ qui est de coût 2

$$\varphi_1 = y(z + t + v) + ztv = \text{Maj}\ [y, z, \text{Maj}\ (y, t, v)]$$ Coût 2

D'où au total : coût 6



CHAPITRE -3-

PROGRAMMATION

I - INTRODUCTION

II - PRESENTATION DE PROGRAMMES

III - RESULTATS

I - INTRODUCTION.

I - 1 - CHOIX DU LANGAGE DE PROGRAMMATION.

Le but poursuivi n'était pas tant d'obtenir des programmes d'exploitation que de pouvoir comparer les performances et qualités des Algorithmes.

D'autre part, il importait de ne pas consacrer un travail trop important à l'écriture et à la mise au point des programmes.

Pour ces raisons, le langage ALGOL a été choisi; en effet, ce langage présente les deux avantages suivants :

- généralité : ALGOL est susceptible d'être exploité sur des calculateurs différents.
- rapidité de l'écriture et de la mise au point des programmes.

I - 2 - PROBLEMES POSES PAR LA PROGRAMMATION BOOLEENNE.

I - 2 - a - Problèmes inhérents au calculateur.

Le calculateur à notre disposition est un IBM 7044 possédant 32768 mots machine de 36 positions binaires chacun; le mot est l'unité de mémoire directement accessible.

Une fonction Booléenne se présente comme une expression littérale; l'unité d'information est la lettre ou le chiffre (représentant une variable) ou le symbole (union, intersection, complément, etc...).

Or, le mot machine est beaucoup trop grand pour recevoir un symbole : en machine on peut faire tenir une lettre sur un bloc de 6 positions binaires : le caractère; le caractère permet donc l'enregistrement d'expressions booléennes. Mais il n'est adressable que par l'intermédiaire du mot machine; le traitement d'une information de ce type conduit à une programmation lourde et délicate.

Il est beaucoup plus intéressant d'adopter un ordre lexicographique pour les variables.

On peut alors choisir comme unité d'information le monôme que l'on place dans un mot machine. Une variable étant caractérisée par son rang occupe une position déterminée dans un mot machine représentant un monôme; l'absence de variable se traduira par l'absence d'information (c'est-à-dire l'information 0) dans la position correspondant au rang de cette variable.

Les problèmes de synthèse entrainent en général la manipulation et le stockage d'une quantité considérable d'informations. Le problème de la place nécessaire est fondamental. Ce point est si important qu'il conditionne les méthodes de synthèse : pour qu'une méthode ait de l'intérêt il faut qu'elle permette de conserver en cours de calcul le minimum d'information compatible avec l'obtention d'une solution.

I - 2 - b - Problèmes posés par le langage ALGOL.

Une variable en Algol n'est pas "banale"; suivant sa nature (réelle, entière, Booléenne) certaines opérations sont possibles d'autres non.

En particulier :

Sur les mémoires correspondant à des variables entières (ou réelles) les opérations Booléennes (union, intersection, complément) ne sont pas possibles;

Sur les mémoires Booléennes les opérations arithmétiques ne sont pas possibles.

Or, une variable déclarée Booléenne ne contient qu'une des deux informations <u>VRAI</u> ou <u>FAUX</u>; ceci ne permet pas de traiter une mémoire contenant par exemple un monôme booléen.

Toutefois la structure du compilateur utilisé permet les tolérances suivantes:

Si A est une mémoire correspondant à une variable Booléenne, on peut placer dans cette mémoire par une artifice (procédures en autres langages) des données quelconques.

A est alors considéré comme <u>VRAI</u> si $A \neq 0$

A est alors considéré comme <u>FAUX</u> si $A = 0$

D'autre part, si A et B sont deux variables Booléennes et C une variable Booléenne l'instruction.

$C := A \cup B;$

a pour effet de placer dans la mémoire correspondant à C l'union des contenus des mémoires correspondant à A et B. (de même pour l'intersection).

Par contre pour la complémentation et la comparaison il n'y a pas cette tolérance :

si A est Booléen C = <u>NON</u> A vaut VRAI si $A_{(35)}$ est faux $(A = 0)$

C = <u>NON</u> A vaut <u>FAUX</u> si $A_{(35)}$ est vrai $(A \neq 0)$

$A \equiv B$ ne considère que les valeurs logiques VRAI ou FAUX correspondant aux deux mémoires et ne compare pas leurs contenu position à position.

Il a donc été nécessaire d'adjoindre un certain nombre de procédures permettant d'effectuer toutes les manipulations qui sont nécessaires.

Citons les principales dans le cas de fonctions croissantes :

- comparaison de deux mémoires Booléennes
- Complément d'une mémoire Booléenne
- Elimination de multiples
- Comptage du nombre de lettres d'un monôme.

II - PRESENTATION DE PROGRAMMES.

Pour chaque programme nous rappellerons de manière succinte le principe de la méthode, celle-ci étant exposée plus en détail dans le chapitre 2.

II - 1 - PROGRAMME 'ARBRE'.

Ce programme fournit, pour une fonction impaire croissante (ou une fonction incomplète réalisable) une représentation arborescente de coût minimal.

Principe de la méthode.

Formation de colonnes représentant des fonctions de coûts arborescents croissants; élimination des colonnes améliorables (Voir chapitre 2 - I - 2).

Caractéristiques du programme.

Langages utilisés :

Corps du programme en ALGOL
Entrées sorties FORTRAN
5 procédures en code machine IBM 7044.

Dimensions.

3068 unités syntaxiques
Programme compilé : environ 3200 instructions machine

Durée de compilation : 1 mn.

Procédures en code :

CALCULOT (U, V) :

Permet de connaitre le nombre exact de mémoires disponibles pour les tableaux, compte tenu de l'encombrement du programme et des sous-programmes nécessaires.

<u>TROU(A)</u>;

Fabrique 35 mémoires ayant la configuration suivante : la $i^{ème}$ mémoire contient un 1 dans le $i^{ème}$ digit, des 0 partout ailleurs.

Ces 35 mémoires (sous forme d'un tableau appelé MASQUE [1:35]) permettent de sélectionner un digit quelconque de la mémoire Booléenne A par l'opération :

C := A ∩ MASQUE [I];

place le $i^{ème}$ digit de A en C.

<u>CARAC (A, B, C)</u>

Place dans la mémoire C, au $5^{ème}$ caractère, le $B^{ème}$ caractère de la mémoire A.

Permet d'entrer les données en machine sous forme littérale sans aucun autre codage et de former à partir de la chaine représentant cette expression, le tableau d'Ackers correspondant, à l'aide d'une séquence d'instructions ALGOL utilisant cette procédure.

<u>COMPAR (U, V, A, B, C, F)</u>

Booléens U,V, F; entiers A, B, C.

Cette procédure dont le rôle est assez complexe permet d'accélerer le déroulement du programme.

Il s'agit d'une "entière procédure".

U symbolise une colonne du tableau des colonnes déjà existantes.

V symbolise la colonne nouvelle calculée.

A symbolise l'adresse de 1ère colonne du tableau des colonnes existantes à traiter.

B symbolise l'adresse de dernière colonne du tableau des colonnes existantes à traiter.

C vaut 0 si l'on traite des colonnes de coût < coût de V

C vaut 1 si l'on traite des colonnes de coût égal à coût de V

F représente la colonne du résultat à obtenir.

Déroulement de la procédure :

- Comparaison de V à F

Si V = F COMPAR = 3 Fin procédure.

- Si V $\neq$ F

Comparaison de V aux colonnes U[A], U[A+1], ... U[A+B]

1er cas : C = 0

Si V $\leqslant$ U [I] COMPAR = 0 fin procédure

sinon I = I+1 ; Si I = B COMPAR = 1 Fin procédure.

2ème cas : C = 1

Si V $\leqslant$ U [I] COMPAR = 0 Fin procédure

Si V > U [I] COMPAR = I Fin procédure

sinon si I = B COMPAR = 1 Fin procédure.

TRANS(A); entier A. "entière procédure"

Place en TRANS l'image DCB du nombre entier A.
Utilisation : permet de sortir par le même ordre d'impression, tantôt des lettres, tantôt des nombres.
Utilisée dans la sortie des résultats.

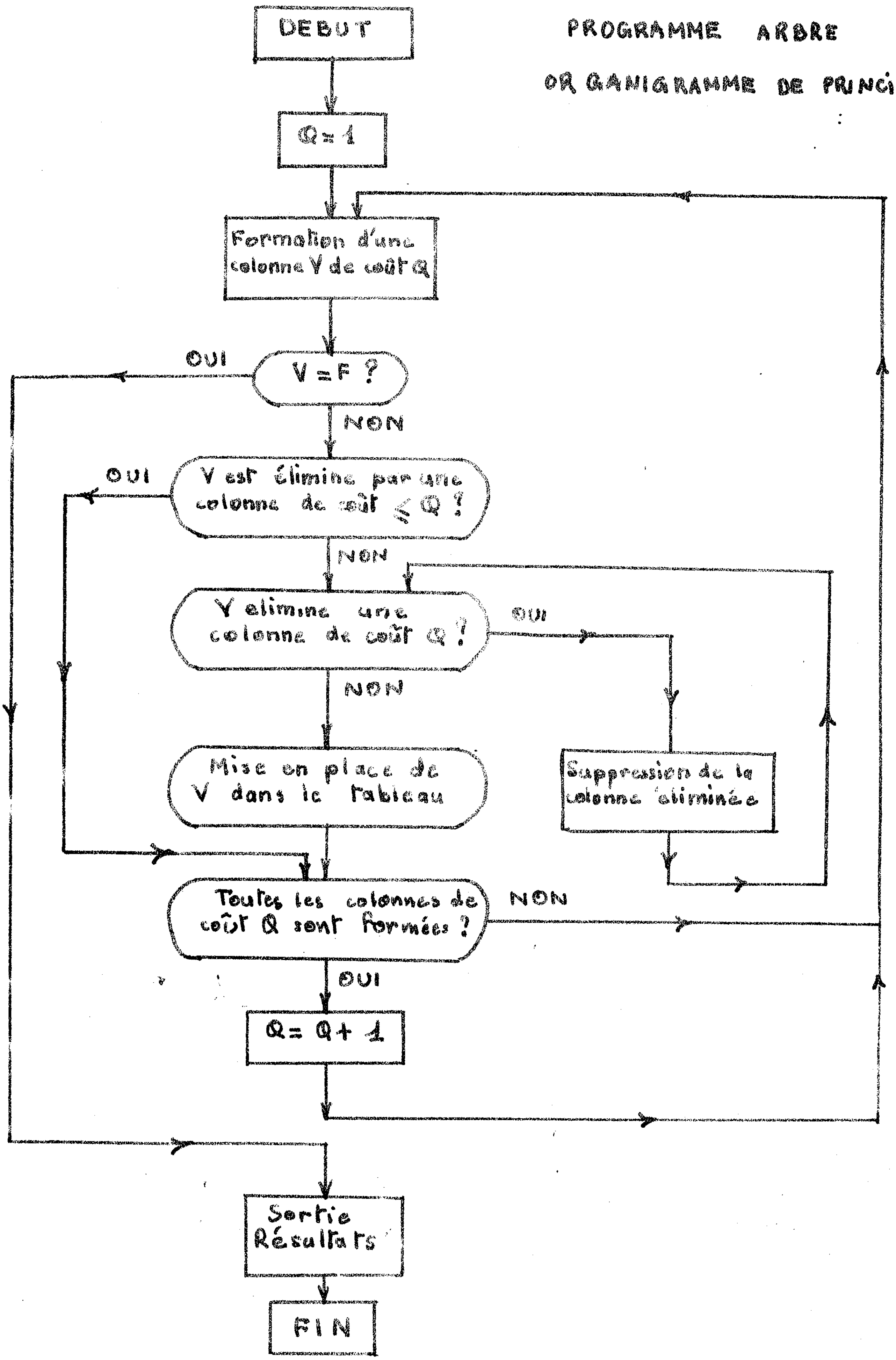
Capacité du programme.

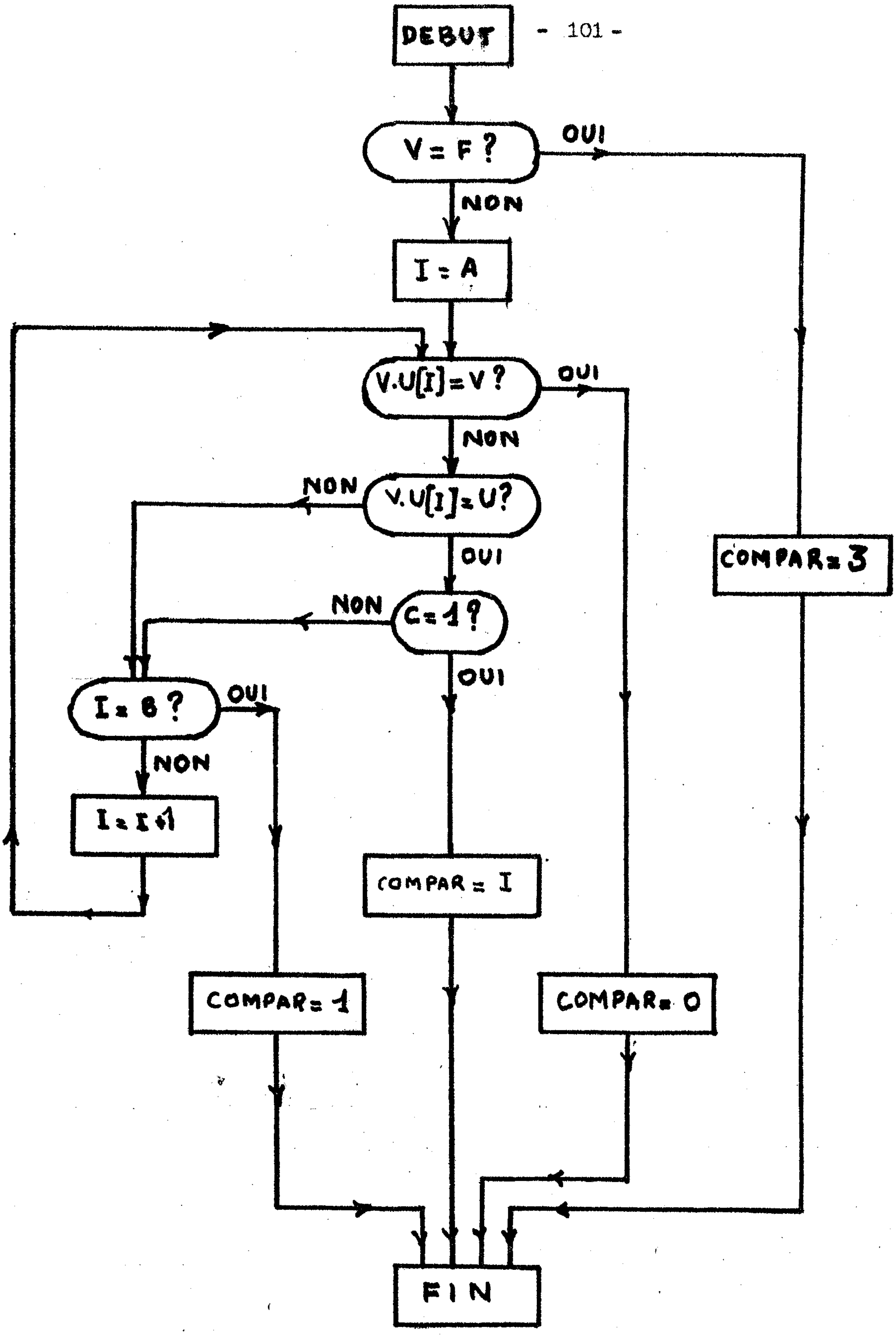
Fonctions ayant au plus 35 monômes premiers.
Le nombre de mémoires disponibles pour les colonnes à fabriquer est d'environ 15000.

On trouvera aux deux pages ci-après :
- l'organigramme de principe du programme ARBRE
- l'organigramme de la procédure COMPAR.

PROGRAMME ARBRE

ORGANIGRAMME DE PRINCIPE





Organigramme de procédure COMPAR (U, V, A, B, C, F)

II - 2 - PROGRAMME MAJOCO.

Ce programme fournit, pour une fonction réalisable une solution à l'aide de majorités composées.

Principe de la méthode.

Si x_i est une colonne du tableau on cherche une colonne M_i = Majorité composée de x_i et de x_{k_1}, x_{k_2}, ... x_{k_q} telle que :

M_i élimine x_i

M_i de coût minimal

la colonne M_i ($1 \leq i \leq n$ = nombre de colonnes du tableau) pour laquelle la quantité $C_i = \dfrac{\text{Coût}}{\text{Nombre de colonnes éliminées}}$ est minimum

est adjointe au tableau (Voir Chapitre 2 - II - A - 2.)

Caractéristiques du programme.

Langages utilisés.

Corps du programme en ALGOL

Procédures en ALGOL et en langage MACHINE IBM 7044

Entrées sorties FORTRAN

Dimensions.

5241 unités syntaxiques

Environ 5500 instructions pour le programme compilé.

Durée de compilation : 2 mn 20 s.

Liste des procédures en code.

DEFMASQUE(A); Booléen A;

Identique à TROU(A): fabrique 35 masques.

DEFNOMASQUE(A); Booléen A;

Fabrique 35 mémoires compléments des précédantes.

Permet de supprimer le $i^{ème}$ digit d'une mémoire A par l'instruction :

A := A ∩ NONMASQUE [I];

UNILIG(U,V); Booléens U, V;

Compare les 2 monômes U et V

Si U ⩾ V alors U = FAUX;

Si U < V alors V = FAUX;

Un ensemble de procédures regroupées dans la procédure PRT permet de faire sur un tableau donné toutes les réductions possibles (lignes et colonnes) jusqu'à obtenir un tableau irréductible.

Une procédure ALGOL appeler MAJCOMP(M) permet pour la Mème colonne du tableau:

De calculer la majorité composée de coût minimum éliminant cette colonne.

De calculer combien de colonnes du tableau sont éliminées par cette majorité composée.

Cette procédure laisse intact le tableau de travail.

Possibilités du programme.

Fonctions complètes : jusqu'à 11 variables.

Fonctions incomplètes : jusqu'à 18 variables à condition que le nombre de points de définition soit inférieur ou égal à 3500.

Présentation des données.

A l'entrée : les données entrent sous forme du tableau d'Ackers codé; un monôme est frappé sur 6 colonnes et est codé en octal.

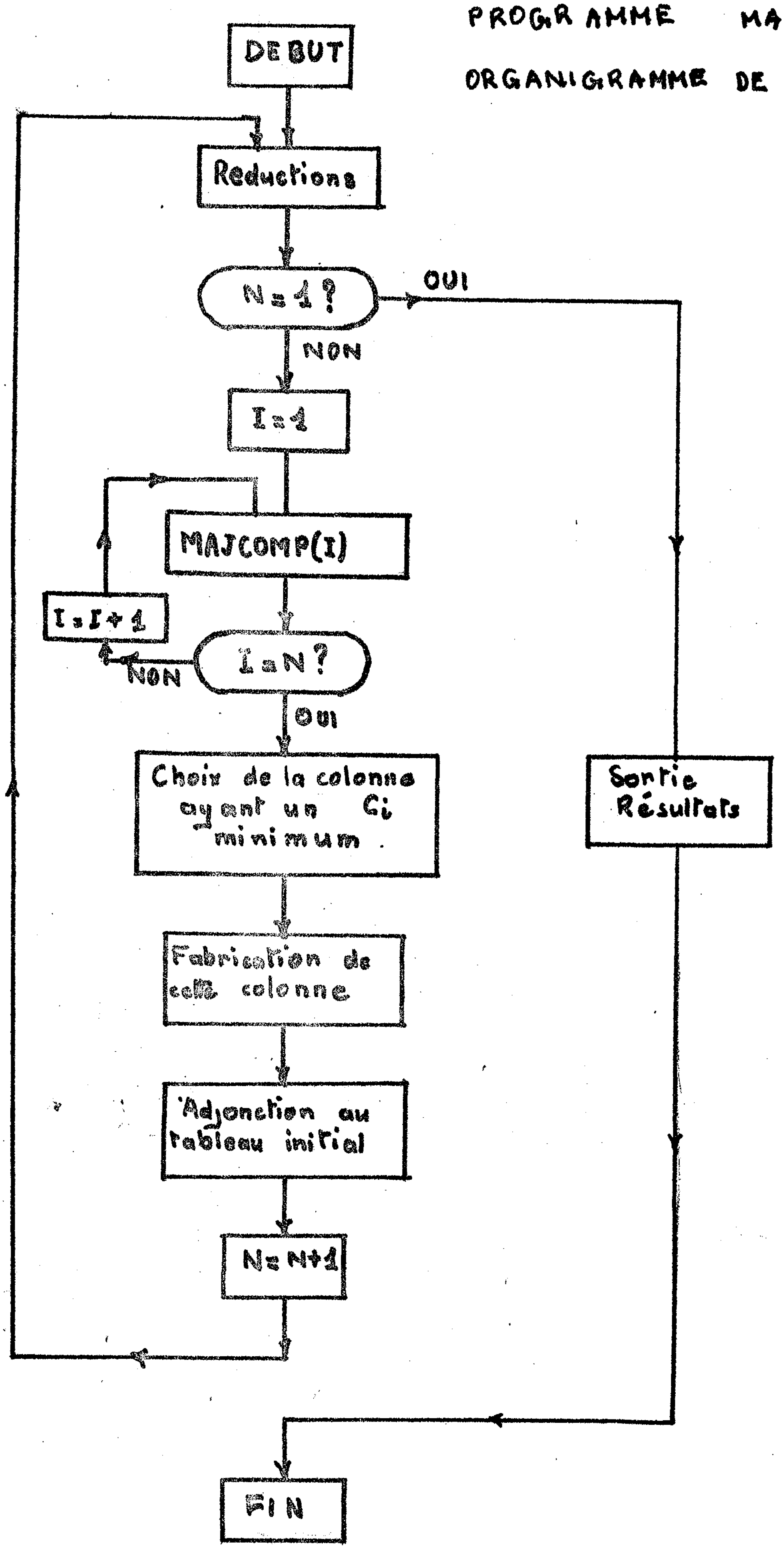
En machine : un monôme caractéristique par mémoire.

Nous donnons aux 2 pages ci-après :

- un organigramme de principe de MAJOCO
- un organigramme de la procédure MAJCOMP(M).

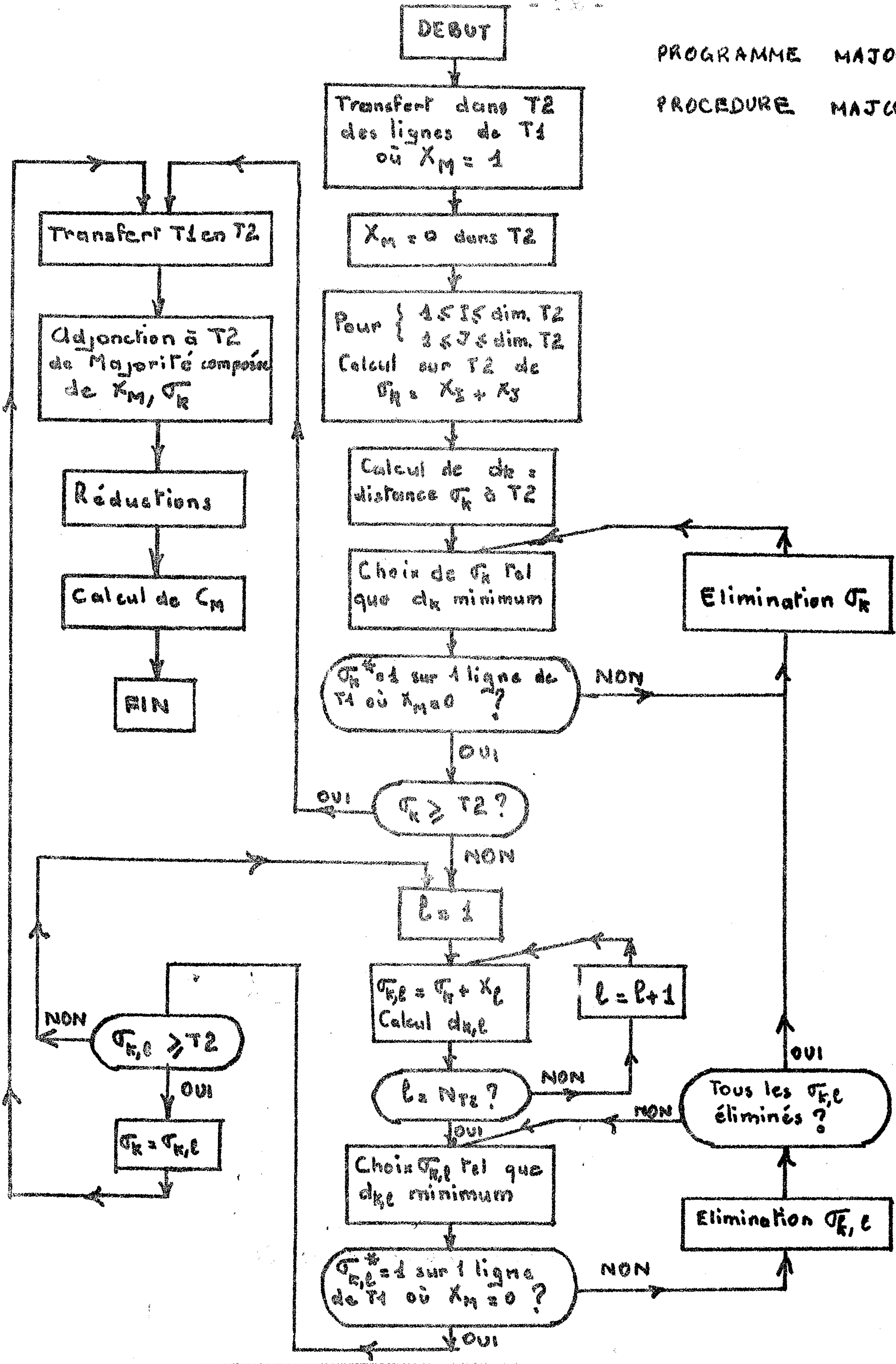
PROGRAMME MAJOCO

ORGANIGRAMME DE PRINCIPE



PROGRAMME MAJOCD

PROCEDURE MAJCOMP(M)



III - 3 - PROGRAMME MAJMAX.

Ce programme fournit pour une fonction réalisable une solution par majorités.

Principe de la méthode.

Adjonction au tableau formé des colonnes $X_1, \dots, X_M$ de la colonne $X_{M+1} = \text{Maj}\ (X_i, X_j, X_k)$.

avec : $1 \leqslant i \leqslant M$; $1 \leqslant j \leqslant M$; $1 \leqslant k \leqslant M$; distance de X_{M+1} au tableau est minimum.

(Voir Chapitre 2 - II - B - 1).

Caractéristiques du programme.

Langages utilisés.

Corps du programme en ALGOL.

3 procédures en code machine IBM 7044

Entrées sorties FORTRAN

Dimensions.

3035 unités syntaxiques soit environs 3200 instructions pour le programme compilé.

Durée de compilation : 2 mn.

Procédures en code.

Les mêmes que dans le programme MAJOCO soit : DEFMASQUE(A), DEFNONMASQUE(A), UNILIG(U,V).

Procédures ALGOL.

De même que dans le programme MAJOCO et dans MAJORF (voir ci-après) on utilise une procédure appelée PRT pour réduire les tableaux.

Cette procédure utilise deux procédures auxiliaires :

COMPCOL : élimine les colonnes majorées par d'autres.

COMPLIG : élimine les lignes minorées par d'autres.

A noter que ces 2 procédures sont de tailles différentes; en effet, la disposition adoptée est : 1 monôme par mémoire.

La comparaison de 2 monômes est alors une opération simple : comparaison des deux mémoires correspondantes par la procédure UNILIG(U, V);

Par contre la comparaison de 2 colonnes X_i et X_j revient à comparer le $i^{ème}$ et le $j^{ème}$ digit de toutes les mémoires du tableau.

<u>Possibilités du programme</u> : les mêmes que MAJOCO

Fonctions complètes : jusqu'à 11 variables

Fonctions incomplètes : jusqu'à 18 variables à condition que le nombre de points de définition soit inférieur à 3500.

<u>Présentation des données</u> : comme pour MAJOCO

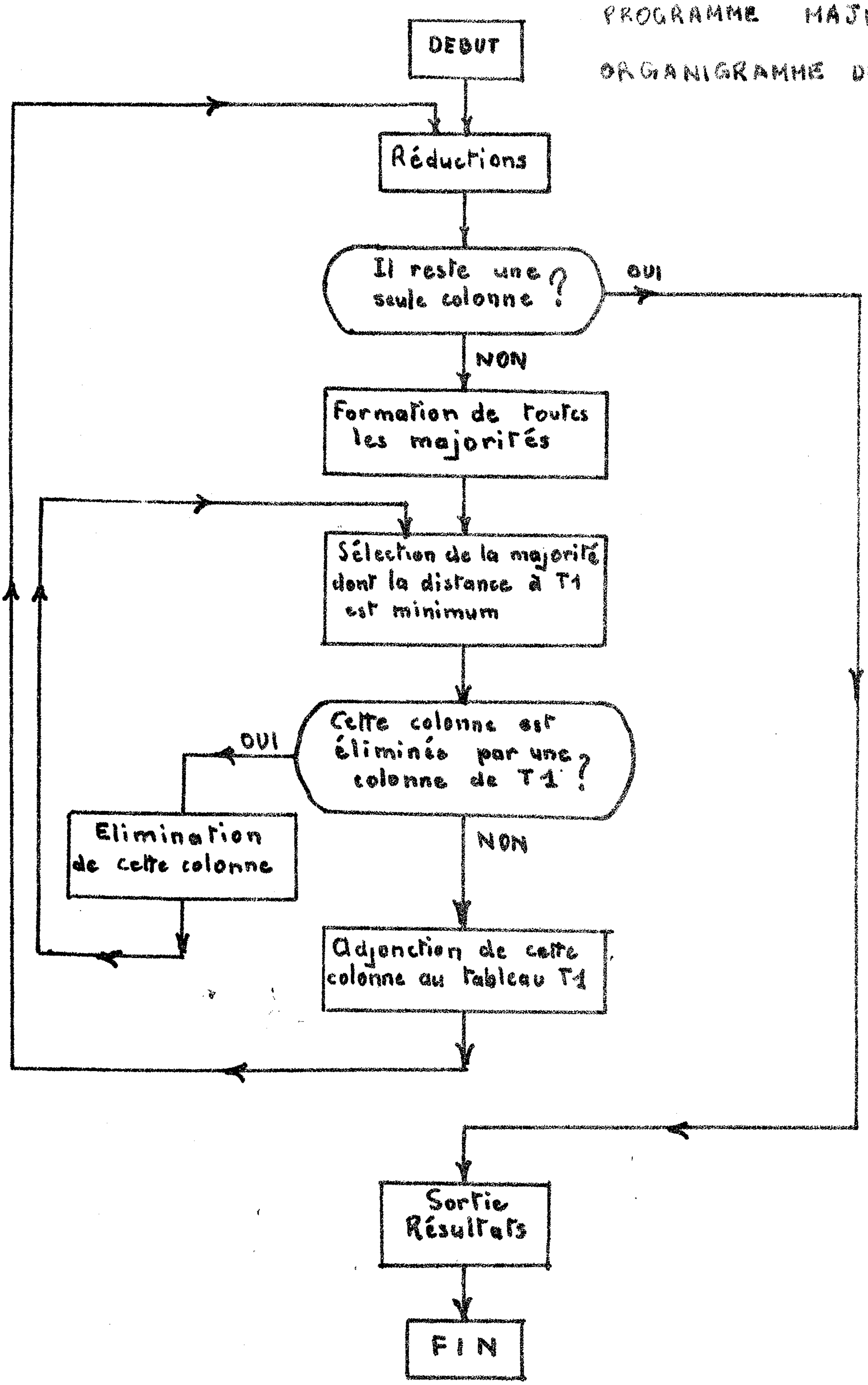
Sous forme du tableau d'Ackers

Nous donnons aux 2 pages ci- après

- un organigramme de principe du programme
- un organigramme de la procédure COMPCOL.

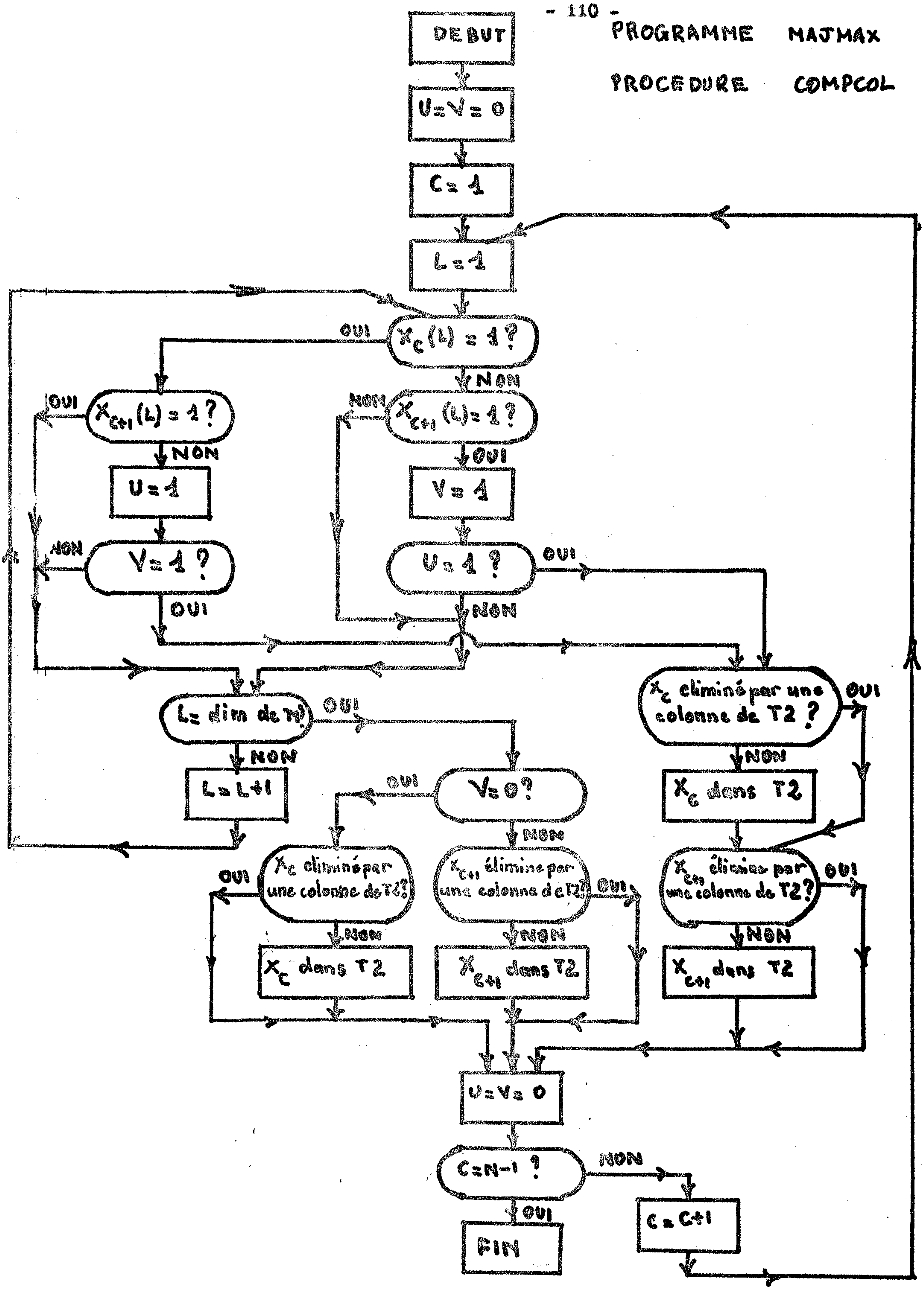
PROGRAMME MAJMAX

ORGANIGRAMME DE PRINCIPE



PROGRAMME MAJMAX

PROCEDURE COMPCOL



II - 4 - PROGRAMME MAJORF.

Fournit pour une fonction réalisable une solution majoritaire.

Principe de la méthode.

A l'aide de deux critères de distance on détermine une paire de colonnes X_i, X_j de la manière suivante :

- $X_i + X_j$ à distance minimum du tableau T1 de f.
- $X_i \cdot X_j$ à distance minimum de f.

X_i, X_j étant ainsi choisis on forme le tableau T2;
T2 est formé des lignes de T1 ou $X_i \oplus X_j = 1$.

Sur T2 on calcule la majorité à distance minimum que l'on adjoint à T1.
(voir chapitre 2 - II - B - 2).

Caractéristiques du programme.

Langages utilisés.

Corps du programme en ALGOL
4 Procédures en code machine IBM 7044
Entrées sorties FORTRAN.

Dimensions.

4641 unités syntaxiques soit environ 4600 instructions pour le programme compilé.

Durée de compilation : 2 mn.

Procédures utilisées.

DEFMASQUE(A), DEFNONMASQUE(A), UNILIG(U,V).
DETECTPAIRES(U,V,W) Booléens U,V; entier W.

Si une colonne X_i du tableau est telle qu'il n'existe aucun couple de lignes pour lequel X_i soit la seule variable à valoir 1 cette procédure fournit le nom de la variable ayant cette propriété.

Utilisation : D'après une propriété due à Ackers une telle colonne peut être éliminée; toutefois, afin de ne pas perdre de solutions qui pourraient être intéressantes, cette procédure n'est utilisée que lorsque le nombre de colonnes devient trop important.

Procédures Algol : PRT déjà vue.

MAJMAX : calcule la meilleure majorité sur T2

PAIREMAX : détermine la meilleure paire sur T1.

Nous donnons aux deux pages ci-après :

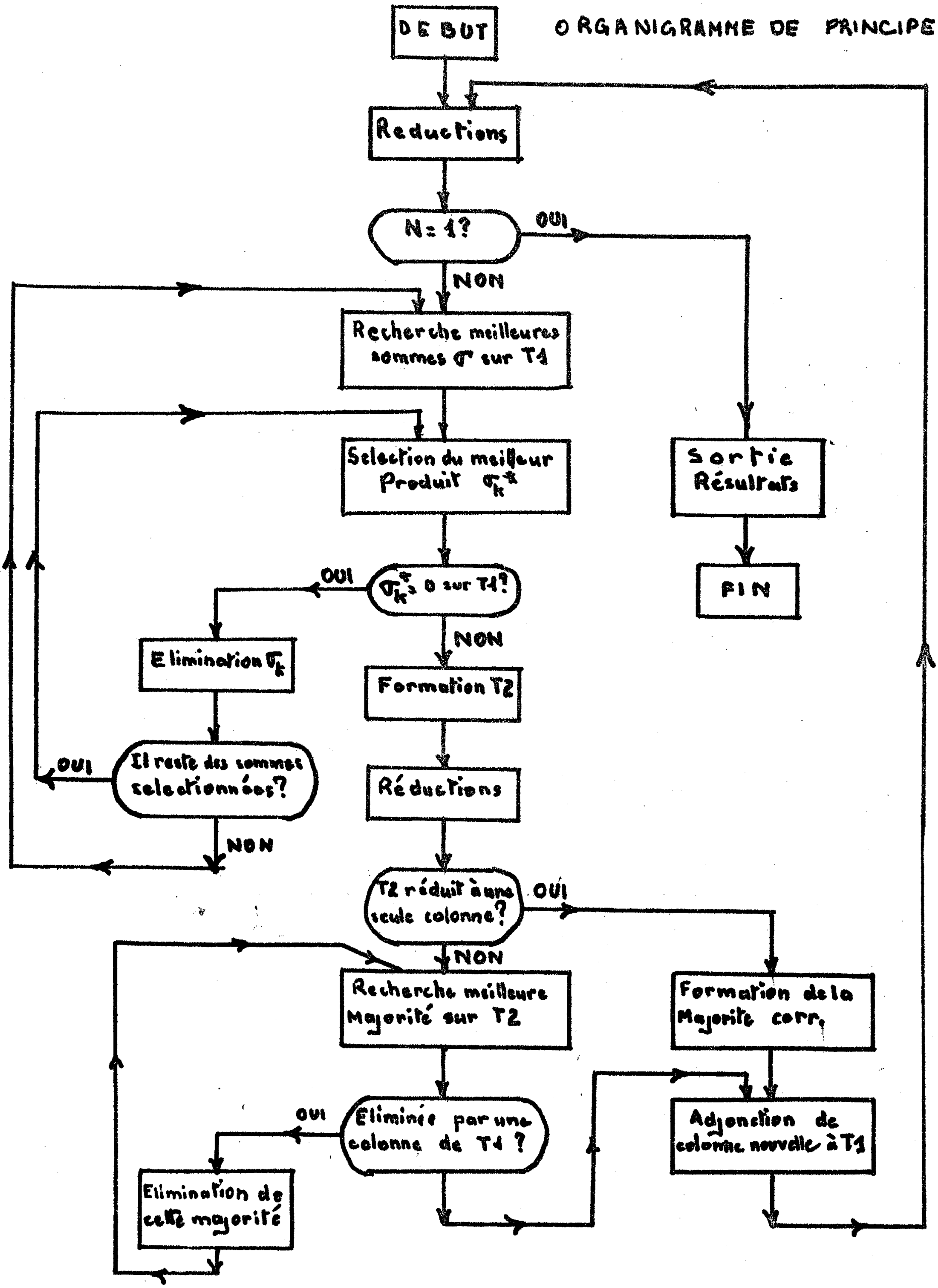
- un organigramme de principe du programme
- un organigramme de PAIREMAX.

Capacité du programme : identique à celle de MAJOCO et MAJMAX.

Fonctions complètes : jusqu'à 11 variables

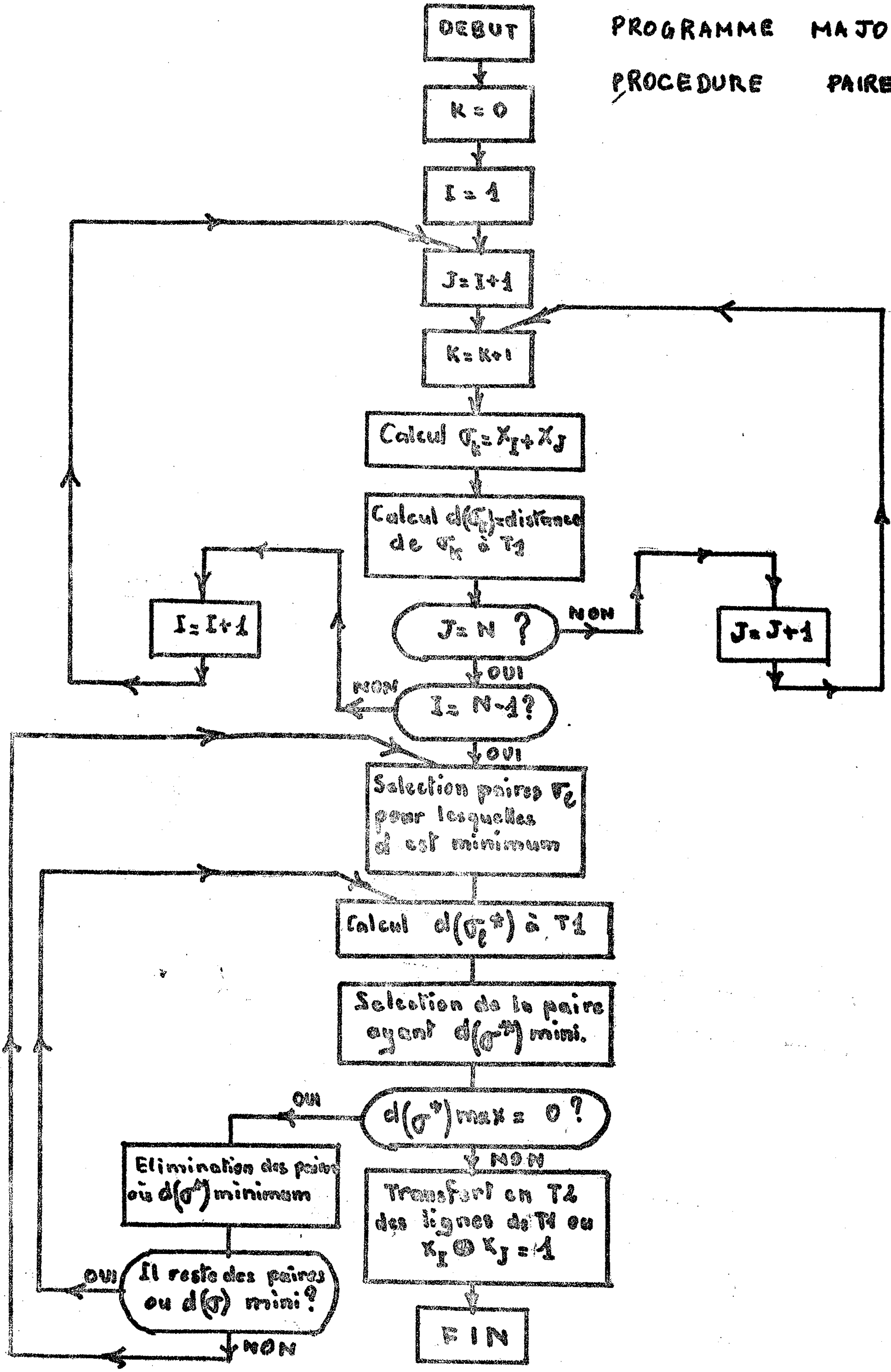
Fonctions incomplètes : jusqu'à 18 variables à condition que le nombre de monômes caractéristiques soit inférieur à 3500.

ORGANIGRAMME DE PRINCIPE



PROGRAMME MAJORF

PROCEDURE PAIREMAX



II - 5 - PROGRAMME MAJORE.

Principe de la méthode.

Partant d'une fonction impaire croissante de n variables, on obtient une fonction de n variables ayant un monôme à 2 lettres et une fonction de n-1 variables. Partant d'une fonction de p variables ayant un monôme à 2 lettres, on obtient 2 fonctions de p-1 variables ayant un monôme à 2 lettres.(voir chapitre 2-III-3).

Caractéristiques du programme.

Langages utilisés :

Corps du programme en ALGOL
7 procédures en code machine IBM 7044
Entrées sorties FORTRAN.

Dimensions :

4370 unités syntaxiques
(environ 4400 instructions pour le programme compilé).

Durée de compilation : 1 mn 40 s.

Procédures utilisées :

7 procédures en code :

- TROU(A) Booléen A
  identique à celle de ARBRE: fabrique 35 masques.
- PASTROU(A) : fabrique 35 mémoires compléments des précédentes.
- ENTREECARAC (U,V,W) entiers U, V,W;
  Place le 5ème caractère de U dans le W$^{\text{ème}}$ caractère de V.

SORTIECARAC(U,V,W) entiers U,V,W.

Place le Vème caractère de U dans le 5ème caractère de W.

COMPTELETTRES (TAB, P,U,V,W). Booléen TAB;
Entiers P,U,V,W.

Recherche parmi les mémoires TAB, TAB+1, ..., TAB+P celle ayant le moins de 1.
Place son adresse par rapport à TAB en W
Place en U le nombre de 1;
Si 2 mémoires ont deux 1 (2étant le minimum possible) place en V l'adresse par rapport à TAB de la 2ème mémoire ayant deux 1.

Utilisation. Recherche du monôme ayant le moins de lettres; en particulier détecte le cas ou une fonction possède au moins deux monômes à 2 lettres.

ELIMIN(U,V,S) Booléens U,V; entier S;

Cette entière procédure permet de savoir si le monôme contenu dans la mémoire V est éliminé par l'une des mémoires U, U+1, ..., U + S.

TRANS(A) entier A.

Même procédure que dans le programme arbre. A étant un nombre entier cette entière procédure place en TRANS le code DCB du nombre A.

Une procédure ALGOL : DUALES.

Calcule la duale d'une fonction croissante.

Capacité du programme.

Fonctions ayant jusqu'à 500 monômes et au plus 34 variables.

Présentation des données.

Entrée en machine sous forme d'expression littérale.
Traitement : sous forme de tableau d'Ackers.

Nous donnons aux deux pages ci-après :

- un organigramme de principe du programme.

- un organigramme de la détermination des deux variables x et y intervenant de la manière suivante :

$$f = xh_1 + yh_2 + h_3$$

ou h_2 est indépendant de x

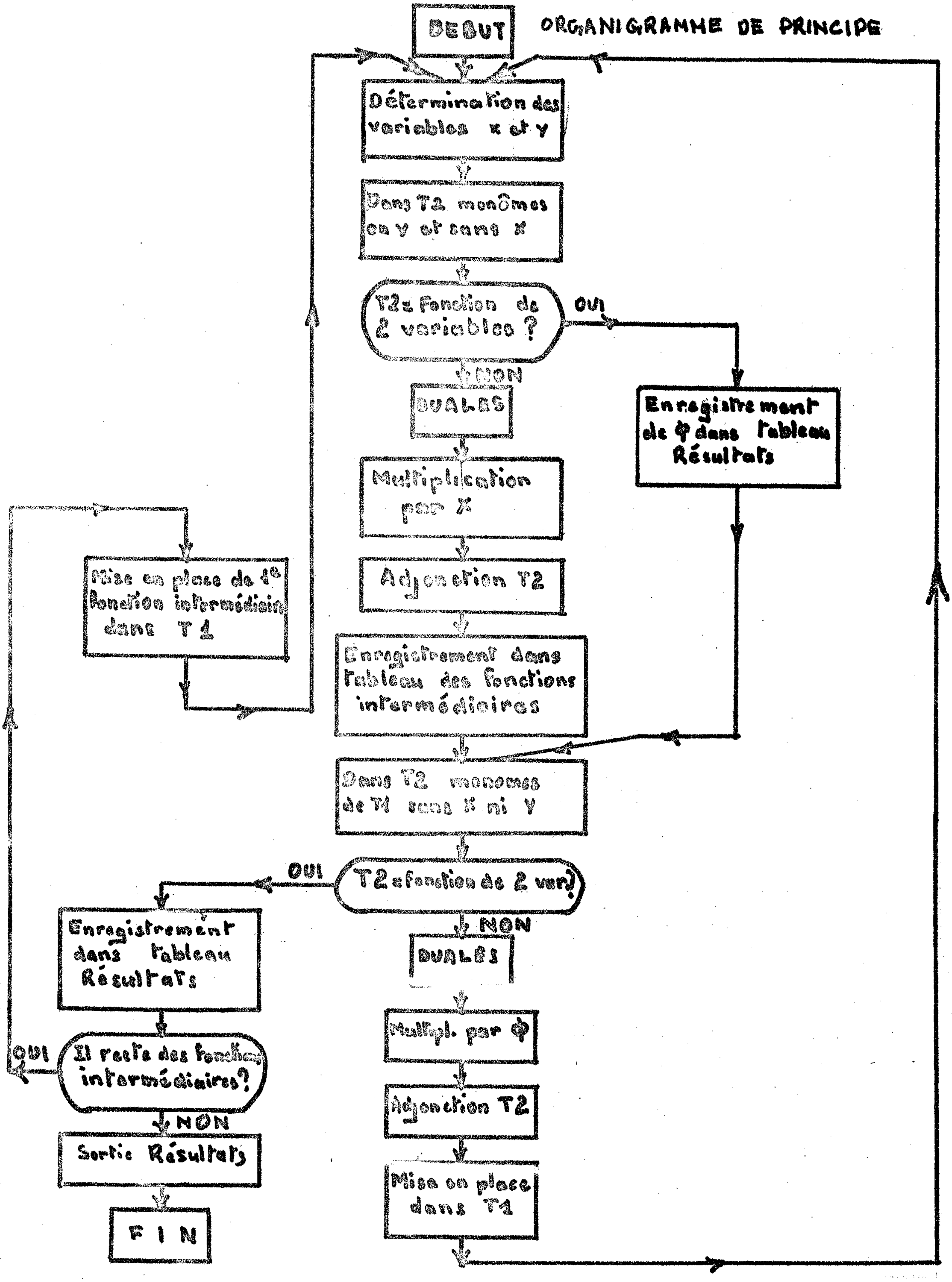
h_3 est indépendant de x et y.

On pose $\varphi = x\ (y+h_2^*) + yh_2$

$$f = \varphi h_3^* + h_3$$

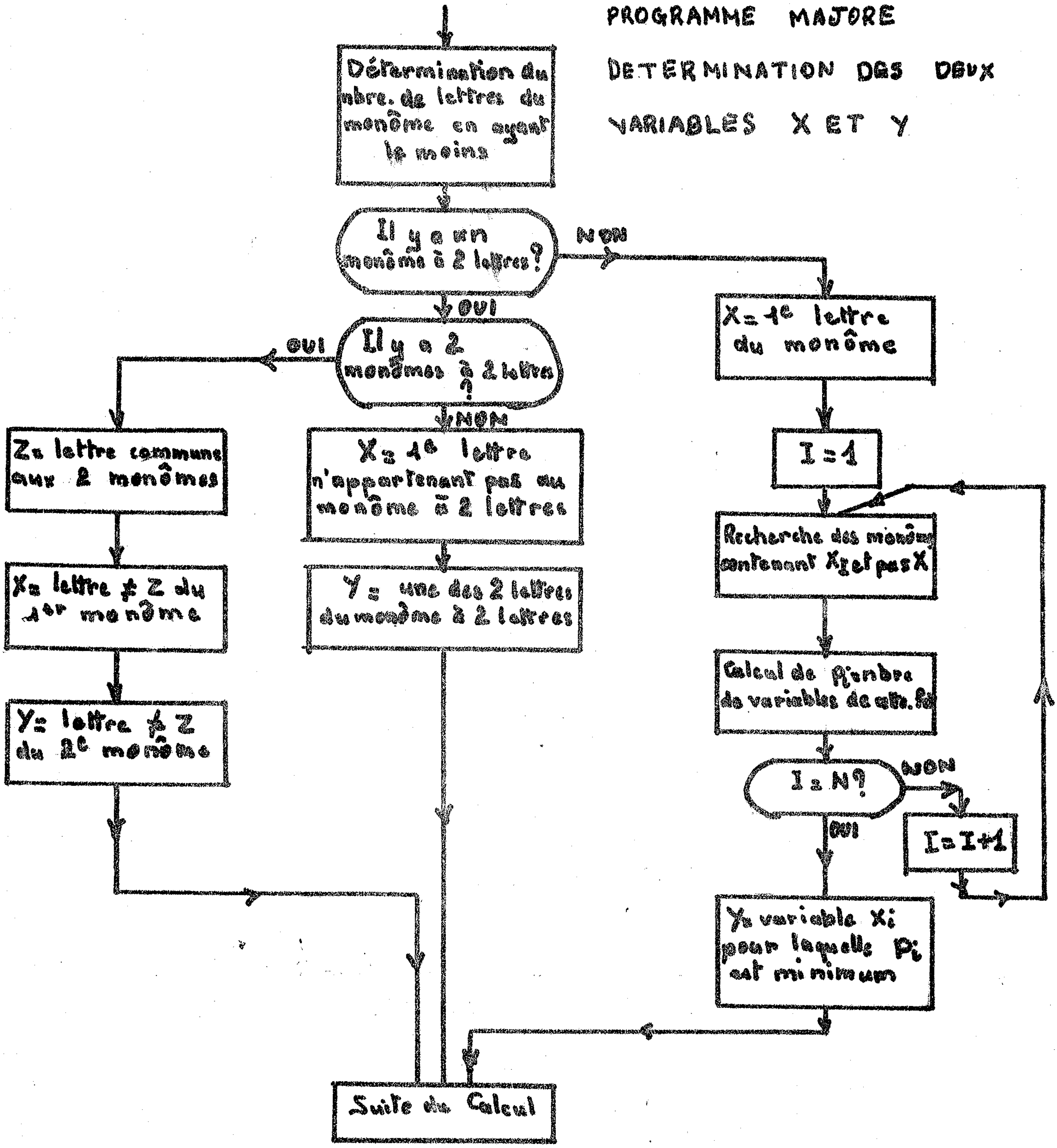
PROGRAMME MAJORE

ORGANIGRAMME DE PRINCIPE



PROGRAMME MAJORE

DETERMINATION DES DEUX VARIABLES X ET Y



III - RESULTATS.

III - 1 - FONCTIONS ETUDIEES.

Il n'était pas nécessaire d'étudier comme exemples des fonctions de moins de 6 variables; en effet, si $n \leqslant 5$ on connait des représentations minimales pour toute fonction impaire croissante.

9 exemples ont été étudiées, représentant des fonctions complètes ou incomplètes, réalisables, d'un nombre de variables allant de 6 à 15.

III - 2 - RESULTATS.

En ce qui concerne le programme ARBRE, programme donnant une solution arborescente de coût minimal, aucun des exemples de 12 et 15 variables n'a pu être traité durant un temps de 20 mn.

Nous donnons ci-dessous à titre d'exemple la solution fournie par chaque programme pour une même fonction.

Il s'agit d'une fonction incomplète de 8 variables dont nous donnons la borne inférieure; rappelons que la borne supérieure étant duale de la borne inférieure est inutile ici.

$$F = cdefg + bdeg + abcdg + bceg + acfg + abch + bceh + cefh + abdh + acdfh + defh.$$

ARBRE (programme de minimisation arborescente).

F = Maj (c, 38, 49)
38 = Maj (e, b, h)
49 = Maj (d, f, 42)
42 = Maj (a, b, g)

MAJOCO

F = Maj. comp. (14, 13, e)
14 = Maj. comp. (d, 12, 13)
13 = Maj. comp. (10, e, d, 11)
12 = Maj. comp. (f, h, 11)
10 = Maj. comp. (c, d, 9, b)
11 = Maj. comp. (b, g, 10)
9 = Maj. comp. (a, c, h)

MAJMAX

F = Maj (11, g, 13)
11 = Maj (10, c, d)
13 = Maj (11, f, 12)
10 = Maj (e, h, 9)
12 = Maj (11, a, c)
9 = Maj (b, c, d)

MAJORF

F = Maj (9, 11, c)
9 = Maj (b, e, h)
11 = Maj (d, 10, a)
10 = Maj (d, f, g)

MAJORE

F = Maj (10, 11, 12)

9 = Maj (17, 18, 19)

10 = Maj (a, c, 9)

11 = Maj (13, 14, 15)

12 = Maj (f, g, 11)

13 = Maj (b, 9, 10)

14 = Maj (h, 9, 13)

15 = Maj (f, g, 14)

16 = Maj (c, d, e)

17 = Maj (20, 21, 22)

18 = Maj (h, 16, 17)

19 = Maj (f, b, 18)

20 = Maj (d, g, 16)

21 = Maj (d, b, 20)

22 = Maj (f, h, 21).

Tableau des résultats.

Nous donnons ci-dessous, pour chaque programme, le coût de la solution obtenue et le temps d'exécution, ceci pour chacun des exemples étudiés.

A R B R E		M A J O C O		M A J M A X		M A J O R F		M A J O R E	
Coût en pérateurs	temps en s.	Coût en opérat.	temps en sec.	Coût en Opérat.	temps en sec.	Coût en Opérat.	temps en sec.	Coût en Opérat.	temps en sec.
5	61	9	16	5	8	5	4	9	4
4	52	9	28	6	12	4	6	15	8
3	13	5	14	4	7	4	3	26	13
		12	240	14	74	12	33	56	34
		15	200	15	68	15	40	48	29
		15	180	14	81	15	40	61	36
		9	160	9	44	7	35	53	31
		13	210	10	54	13	80	64	38
		19	660	21	420	22	302	230	193

III - 3 - EXPLOITATION DES RESULTATS.

III - 3 - a - Etude des coûts.

Afin de comparer les coûts des solutions fournies par les différents algorithmes il était nécessaire d'avoir un élément de mesure; or, pour un coût réel trouvé, supérieur ou égal à 7, le programme de minimisation arborescente n'a pu fournir de solution en 20 mn.

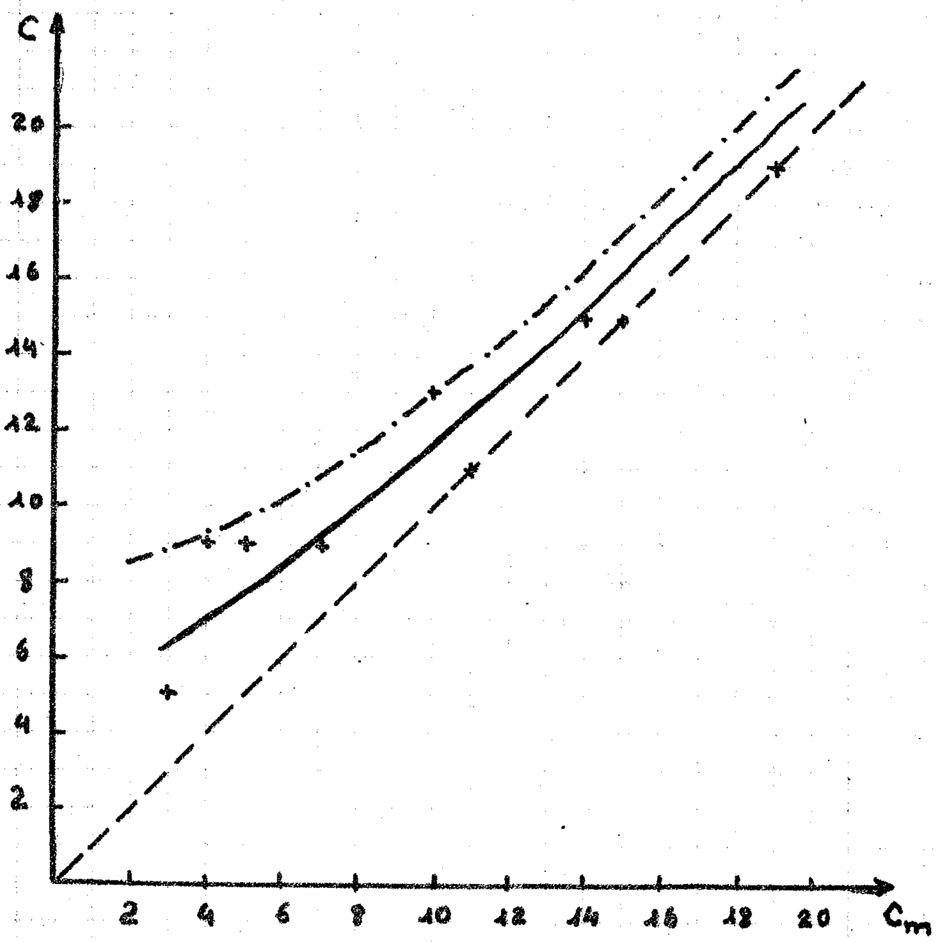
Les courbes ci-dessous ont donc été tracées de la manière suivante : on a porté en abcisse un "coût minimum" expérimental qui est :

- celui fourni par le programme de minimisation arborescente quand celui-ci fournit une solution.
- le coût de la solution de coût minimal obtenue par l'un quelconque des 4 autres programmes dans les autres cas.

En ordonnée on a porté le coût de la solution fournie par le programme.

L'ensemble des points obtenus n'est pas sur une courbe, mais couvre une zone; cette zone est évidemment limitée en bas par la 1ère bissectrice (valeurs minimales).

On a tracé pour chaque programme la courbe correspondant à la limite supérieure du coût expérimental et une courbe de valeur moyenne du coût.

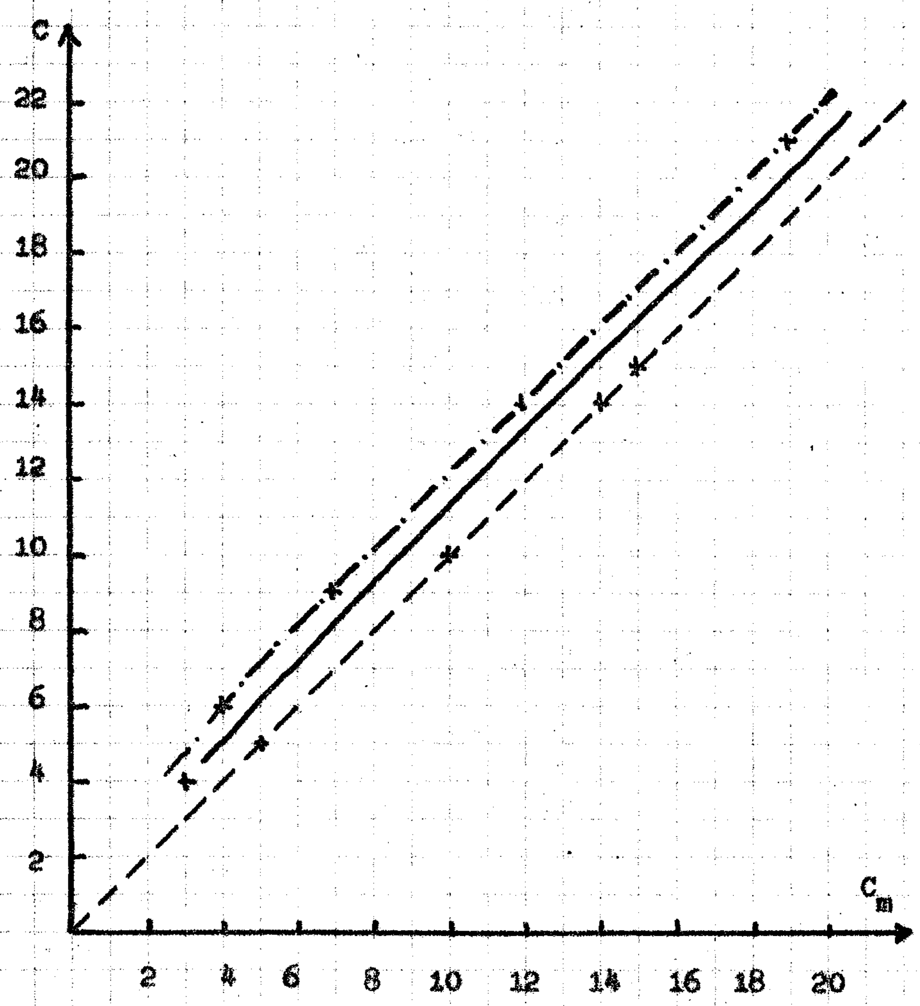


Programme MAJOCO

\+ : Points fournis par l'expérience.

—·— : Borne supérieure du coût.

—— : Valeur moyenne du coût.



Programme MAJMAX

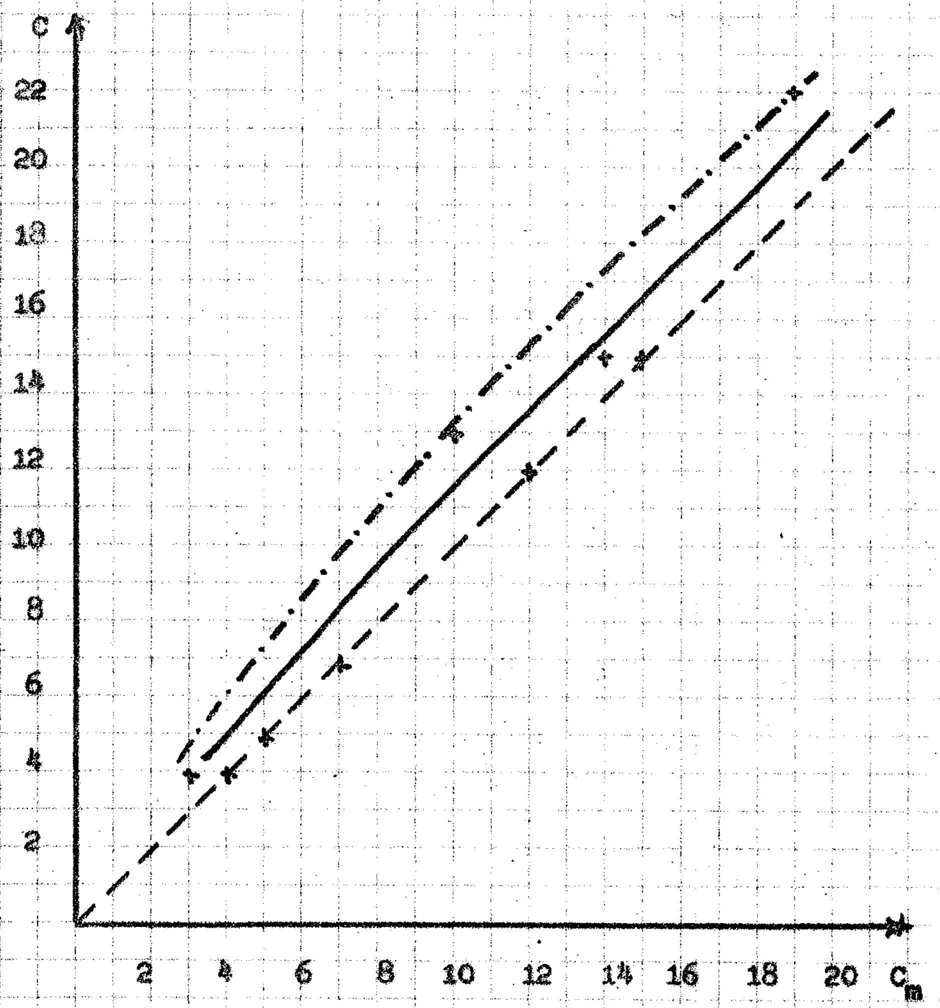
C_m = Coût minimal trouvé

C = Coût fourni par le programme

\+ : points fournis par l'expérience

—·— : borne supérieure du coût

—— : coût moyen

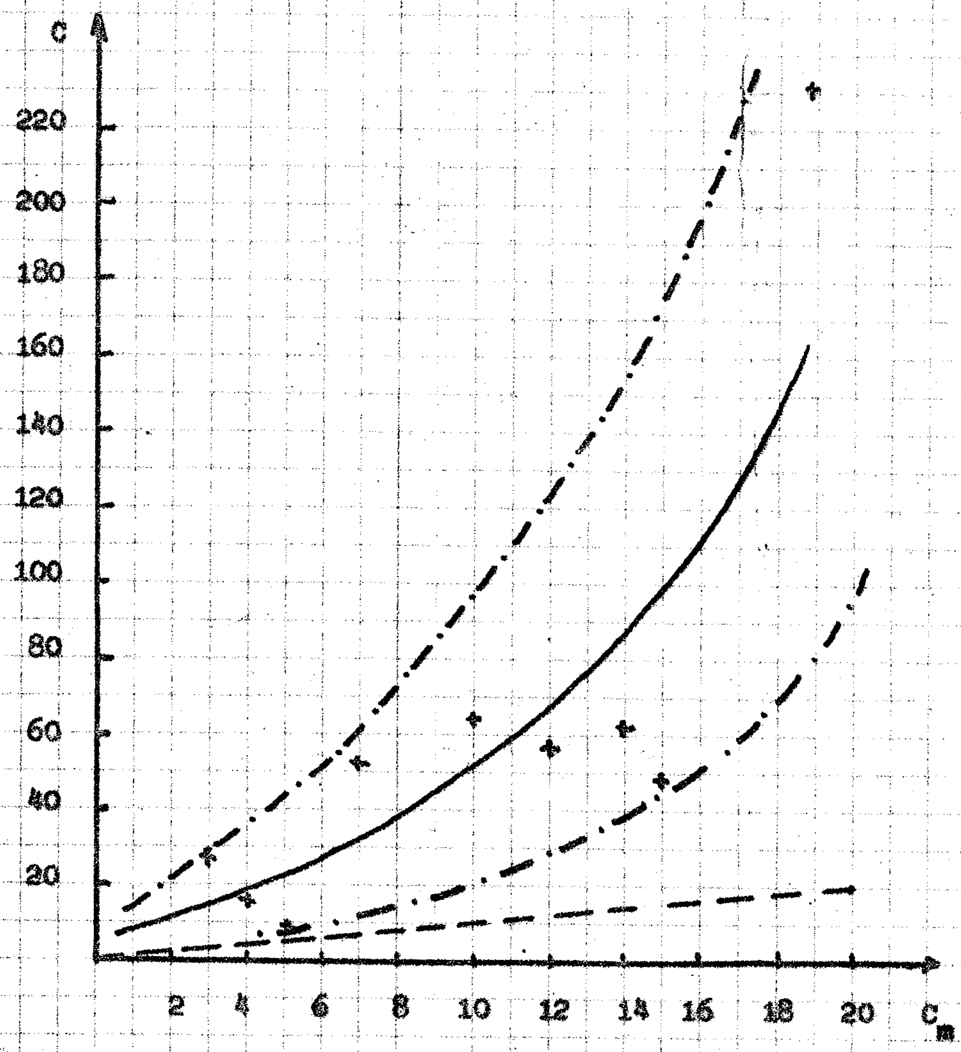


Programme MAJORF

+ : Points fournis par l'expérience

—·— : Borne supérieure du coût

—— : Valeur moyenne du coût



Programme MAJORE

+ : Points expérimentaux

— · — : Borne supérieure du coût

—— : Valeur moyenne du coût

Premières conclusions.

Le programme MAJORE, correspondant à une méthode de fractionnement donne des résultats très mauvais.

D'abord en comparaison des autres algorithmes : les coûts sont de 5 à 10 fois plus élevés.

D'autre part, si C_m augmente les résultats s'écartent de plus en plus de la première bissectrice.

Les 3 programmes MAJOCO, MAJMAX, MAJORF semblent donner de très bons résultats.

De plus et ce résultat est intéressant, les solutions fournies sont très voisines l'une de l'autre au point de vue coût; les méthodes de composition seraient donc assez voisines au point de vue coût de la solution.

Nous laisserons donc MAJORE de côté et étudierons pour les trois algorithmes correspondant à MAJOCO, MAJMAX, MAJORF l'écart relatif du coût $\frac{\Delta C_m}{C_m}$

C_m est le coût minimum obtenu.

Pour ΔC on a évalué l'écart de la manière suivante :

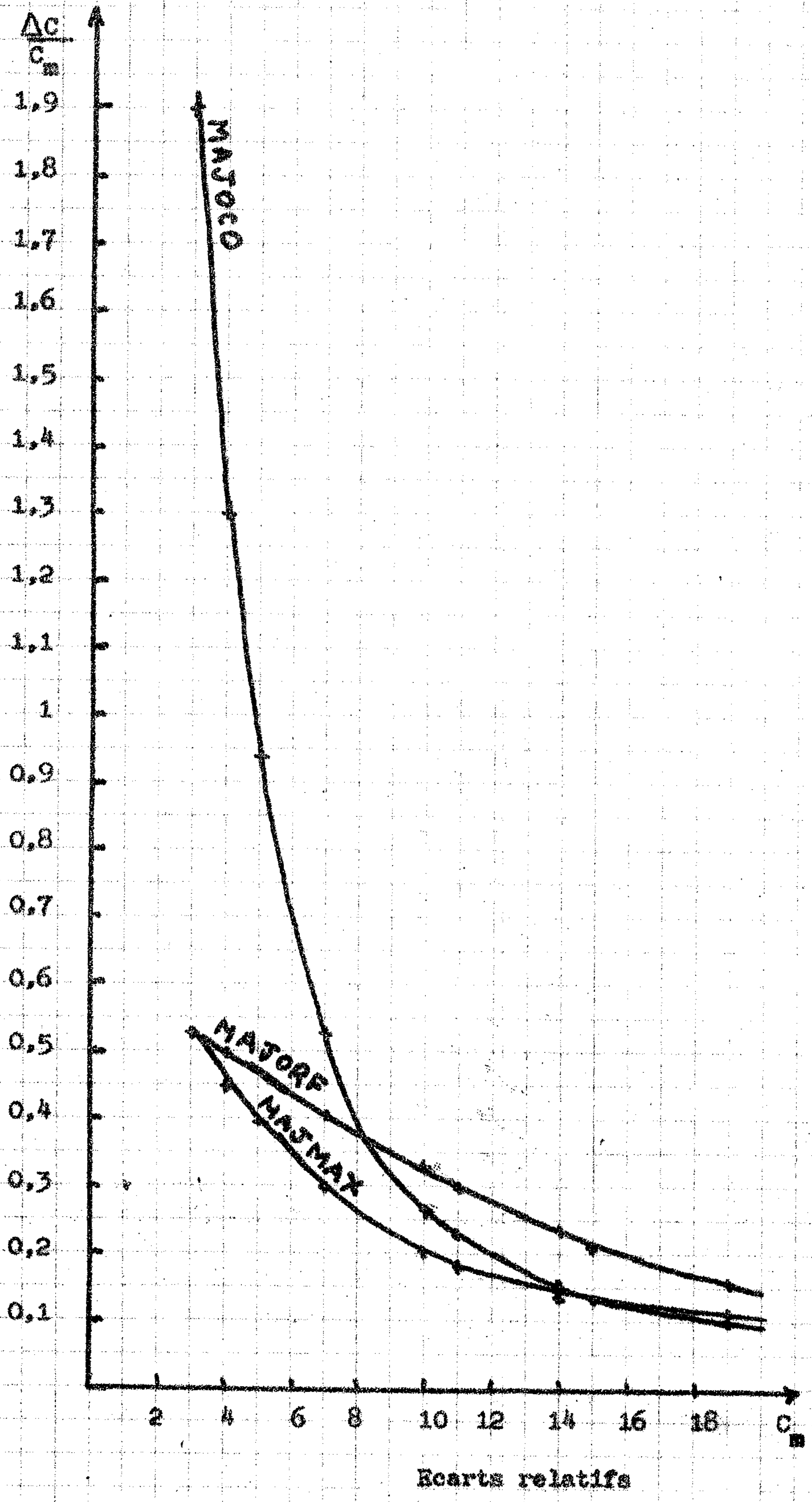
- pour un point P d'abcisse C_m et d'ordonnée C (il s'agit des points réellement obtenus par l'expérience) on a posé :

$$\Delta C = C_M - C_m$$

C_M étant l'ordonnée du point situé sur la courbe de borne supérieure du coût, à l'abcisse C_m.

$\frac{\Delta C}{C}$ ainsi évalué correspond donc à un écart relatif expérimental maximal.

C_m	MAJOCO		MAJMAX		MAJORF	
	ΔC	$\frac{\Delta C}{C_m}$	ΔC	$\frac{\Delta C}{C_m}$	ΔC	$\frac{\Delta C}{C_m}$
3	5,8	1,9	1,6	0,53	1,6	0,53
4	5,2	1,3	1,8	0,45	2	0,5
5	4,7	0,94	2	0,4	2,4	0,48
7	3,7	0,53	2	0,3	2,8	0,4
10	2,6	0,26	2	0,20	3,3	0,33
11	2,6	0,24	2	0,18	3,2	0,3
14	2,1	0,15	2	0,14	3,3	0,24
15	2	0,13	2	0,13	3,2	0,21
19	1,9	0,1	2	0,11	3	0,16



$\frac{\Delta c}{C_m}$
1,9
1,8
1,7
1,6
1,5
1,4
1,3
1,2
1,1
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
MAJOCO
MAJORF
MAJMAX
2
4
6
8
10
12
14
16
18
C_m
Ecarts relatifs

Conclusions tirées de l'étude des coûts comparés.

Les 3 programmes MAJOCO, MAJMAX, MAJORF correspondent à 3 méthodes de composition. MAJORE correspond à une méthode de fractionnement.

Nous avons vu que MAJORE donnait au point de vue coût des résultats très mauvais; par contre les 3 méthodes de composition semblent donner des résultats très acceptables d'une part, et très voisins d'autre part.

De l'étude des coûts relatifs comparés des 3 programmes par composition nous tirerons les conclusions suivantes :

Pour $C_m \geqslant 15$ les 3 programmes donneront des solutions différant entre elles d'au plus 20 °/₀; de plus cet écart tendrait à décroitre.

On remarque que MAJOCO étant le plus près de la première bissectrice et s'en rapprochant le plus rapidement, il semblerait qu'il donne les meilleurs résultats quand C_m croit.

Il est peut être possible d'expliquer pourquoi MAJOCO donne les meilleurs résultats.

En premier lieu des 3 algorithmes, c'est celui où l'on examine le plus grand nombre de possibilités; nous verrons d'ailleurs comme conséquence que ce programme est le plus lent des trois.

D'autre part, il y a dans cet algorithme un critère de choix fondé sur le coût; lorsque les critères de distance laissent la possibilité de choisir entre plusieurs colonnes, on prend la première venue dans le cas de MAJMAX et MAJORF tandis qu'on utilise dans MAJOCO un critère de qualité faisant intervenir le coût (voir Ch. 2).

Les conclusions tirées de l'étude des coûts seront les suivantes.

- Primauté des méthodes par composition sur les méthodes de fractionnement; ces dernières fournissent des solutions bien plus mauvaises que les premières et cet écart ne fait que croitre avec le coût.

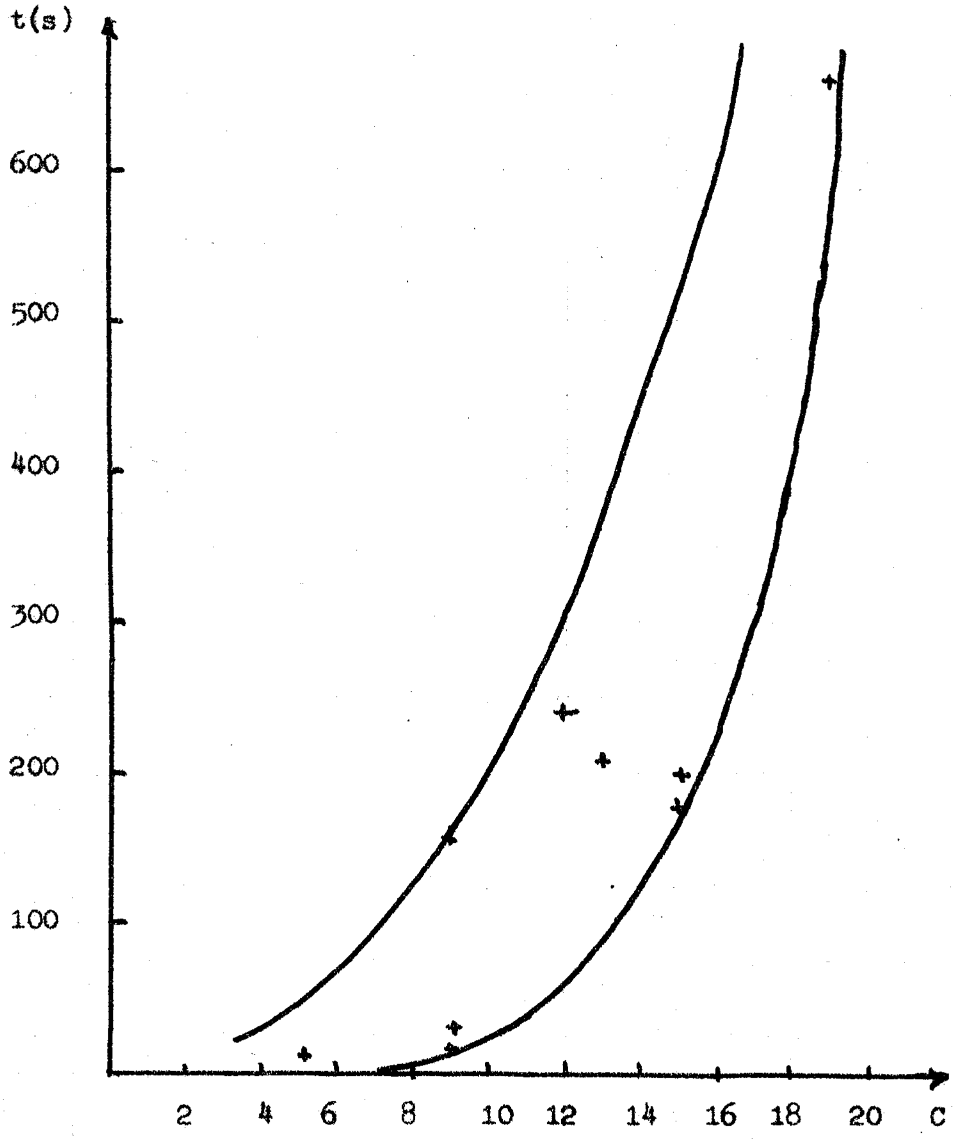
- Les méthodes par composition fournissent des solutions :

- qui semblent très acceptables en valeur absolue
- très voisines l'une de l'autre : leurs résultats tendent à se rapprocher l'un de l'autre si le coût croit.

III - 3 - b - Comparaison au point de vue rapidité.

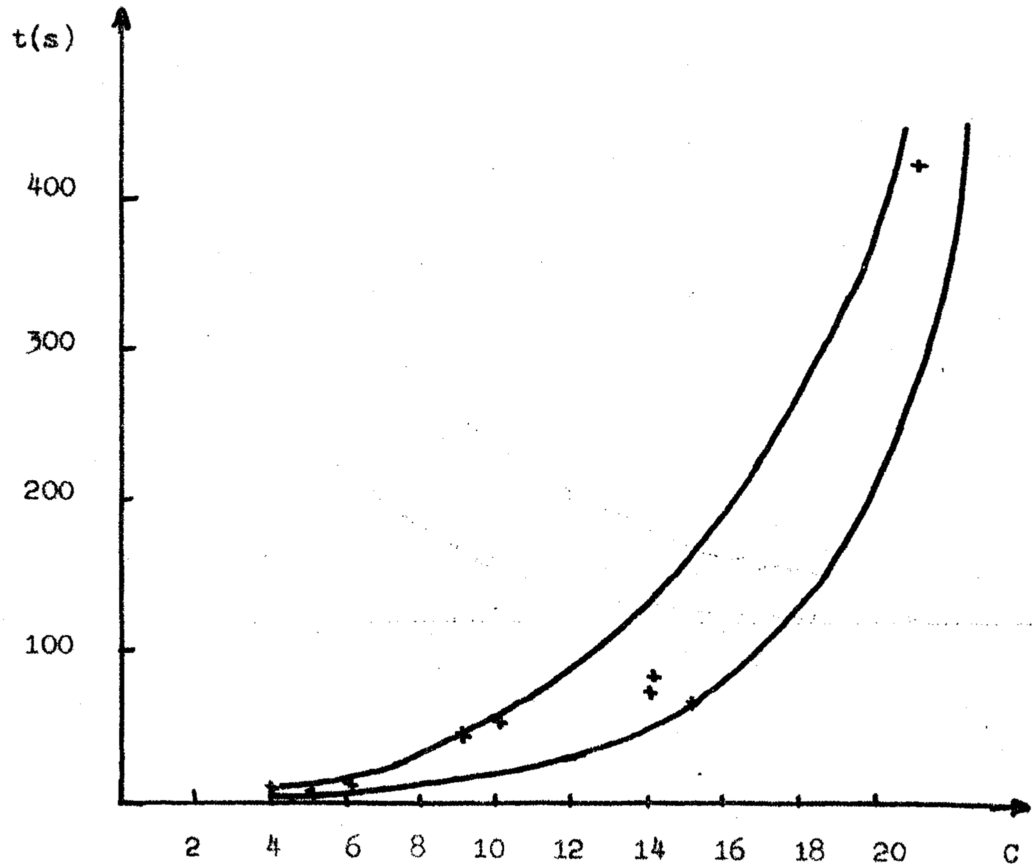
Nous avons étudié deux séries de courbes.
D'une part, pour chaque programme le temps d'exécution en secondes, en fonction du coût de la solution qu'il fournit.
D'autre part, des courbes comparées : temps d'exécution pour une fonction donnée, en fonction du coût de la solution de coût minimal obtenue pour cet exemple par l'un quelconque des 5 programmes.

Nous n'avons pas tracé la courbe correspondant au programme de minimisation ARBRE : en effet, on ne dispose que de trois points ce qui est insuffisant. D'autre part, les essais infructueux montrent que le temps d'exécution est nettement prohibitif.



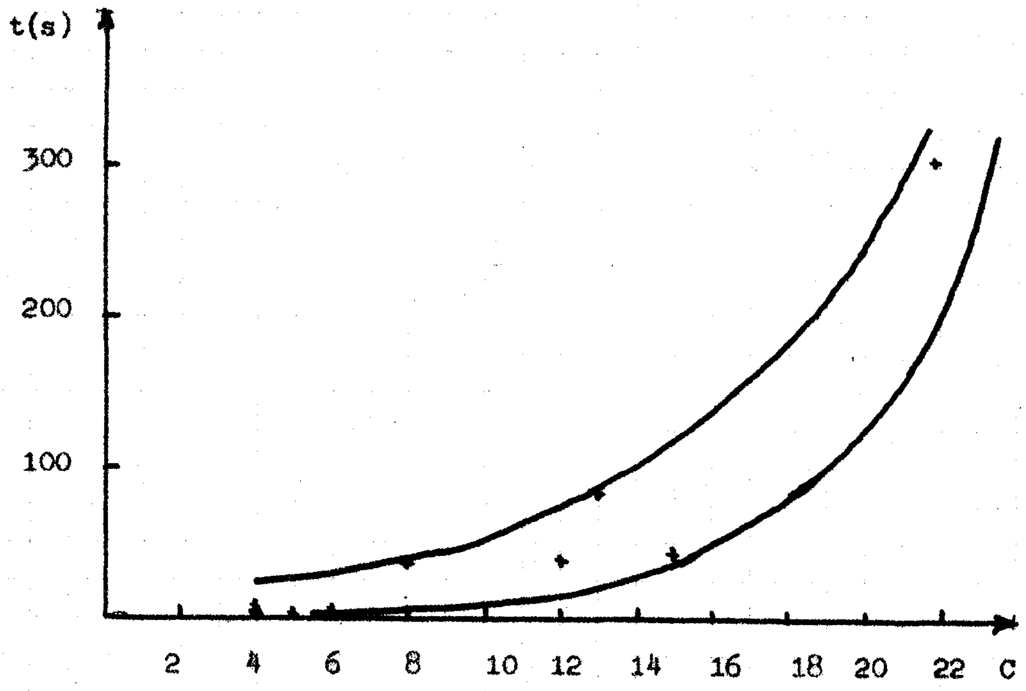
Programme MAJOCO

Temps d'exécution = fonction du coût



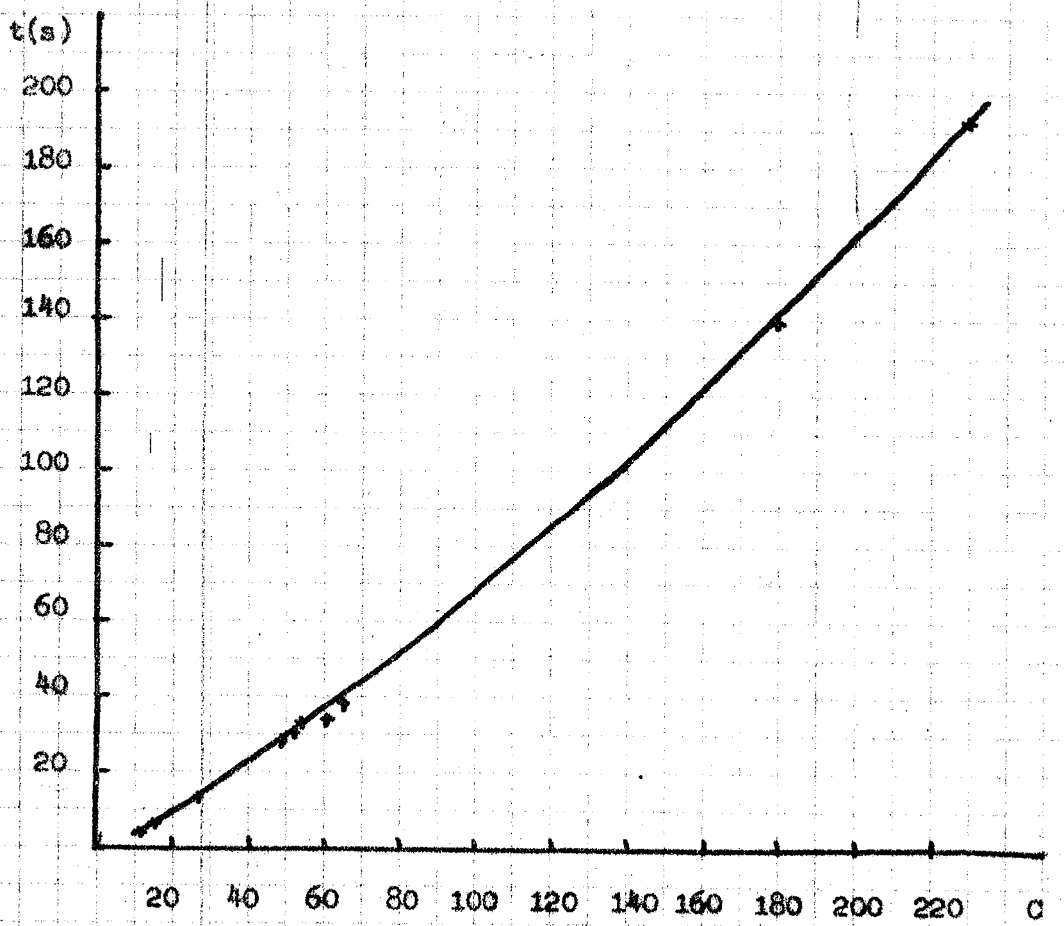
Programme MAJMAX

Temps d'exécution = fonction du coût



Programme MAJORF

Temps d'exécution = fonction du coût



Programme MAJORE

Temps d'exécution = fonction du coût

Premières conclusions.

Si l'on s'intéresse uniquement à l'aspect graphique des résultats une première constation s'impose :

- la dispersion des points expérimentaux ne permet pas théoriquement de tracer une courbe mais simplement de délimiter une zone.

Dans le cas du programme MAJORE cette zone s'avère extrêmement étroite au point que sur le graphique nous avons tracé une courbe.

Nous pouvons donc supposer que le programme MAJORE permet, avec une assez bonne précision de déterminer la relation temps d'exécution - coût de la solution fournie.

Une explication est possible :

- Dans les 3 autres algorithmes le nombre de pas du programme correspond exactement au nombre de colonnes fabriquées; mais ce nombre ne correspond pas au coût de la solution fournie : on peut très bien fabriquer des colonnes n'intervenant pas dans la solution; c'est ce qui explique la relative dispersion des points.

Par contre, dans le cas du programme MAJORE chaque pas correspond à une fonction apparaissant effectivement dans la solution.

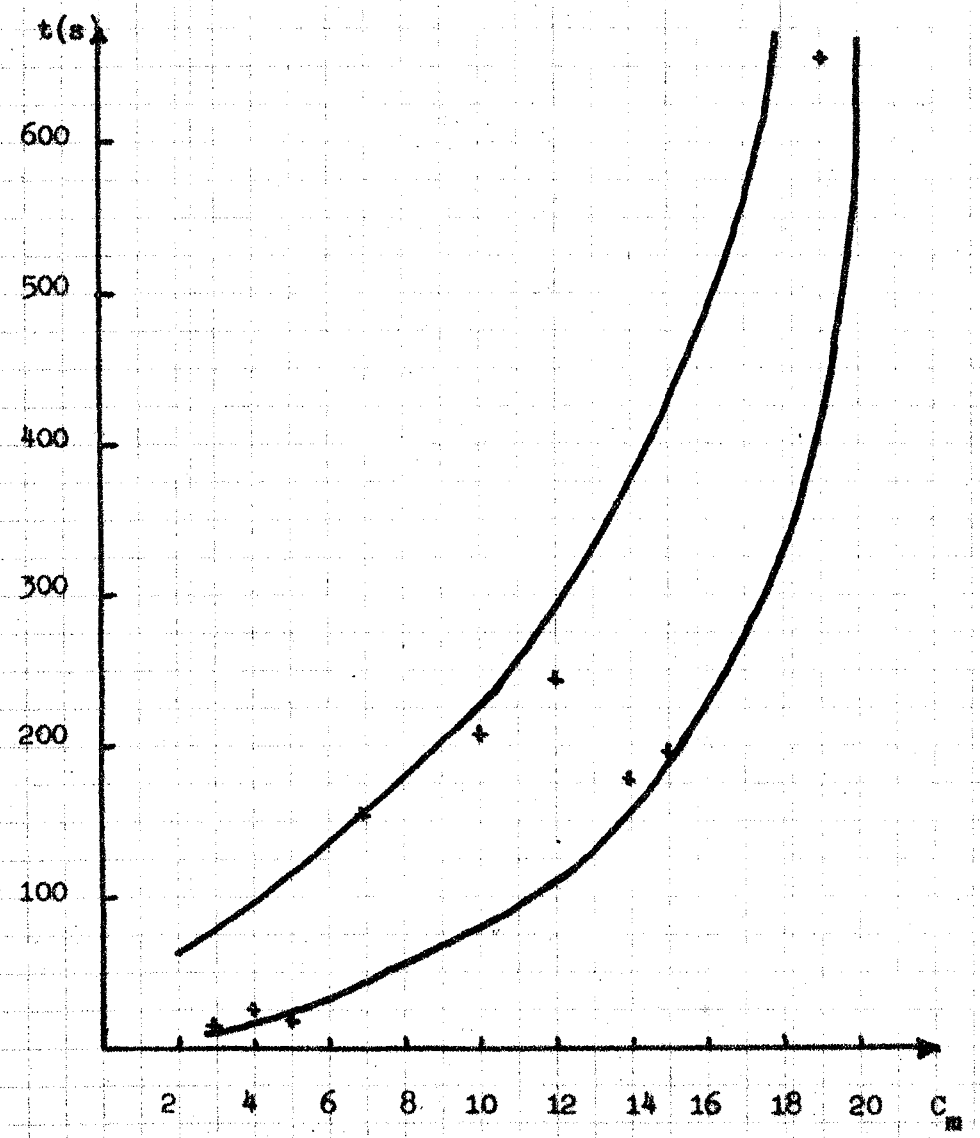
Ceci est d'ailleurs commun à toutes les méthodes de fractionnement : quelle que soit la méthode employée tout résultat calculé est effectivement utilisé dans la solution; il sera donc possible dans le cas d'une méthode par fractionnement de relier de manière précise le coût et le temps, ou le coût et le nombre de variables.

<u>Temps rapportés.</u>

On a tracé cette fois des courbes donnant le temps d'exécution, fonction du coût minimal.

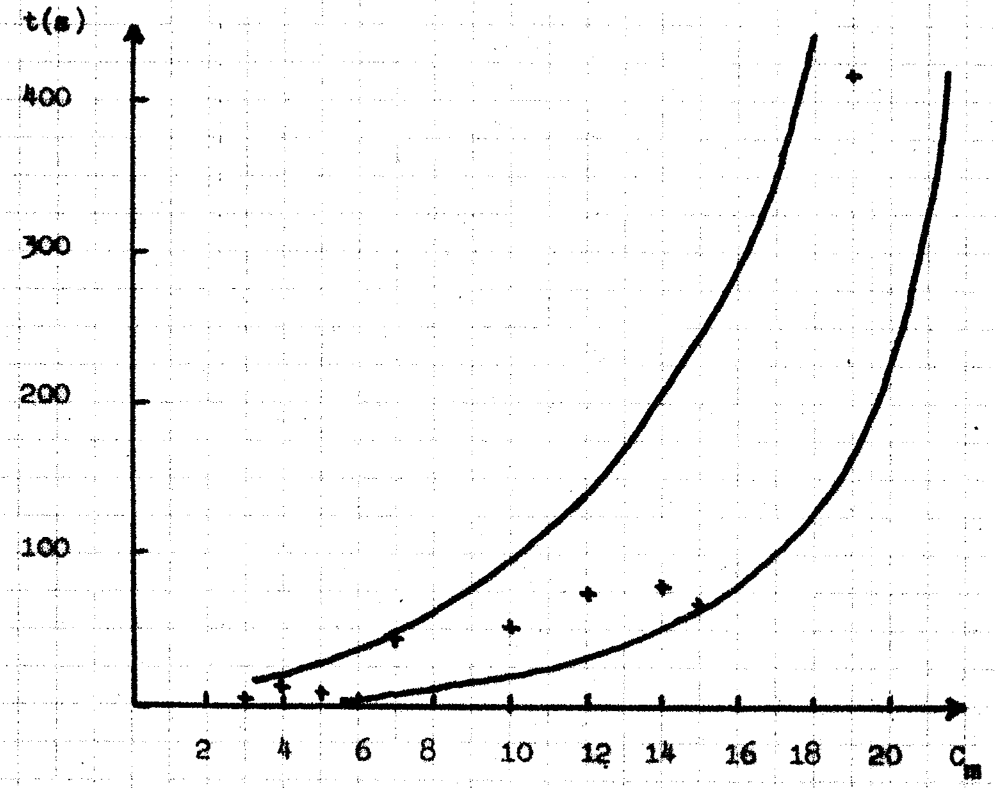
Il faut s'attendre cette fois à ce que les résultats de MAJORE ne donnent plus une courbe mais une zone comme les autres programmes.

Comme précédemment nous n'avons tracé aucune courbe en ce qui concerne le programme de minimisation.



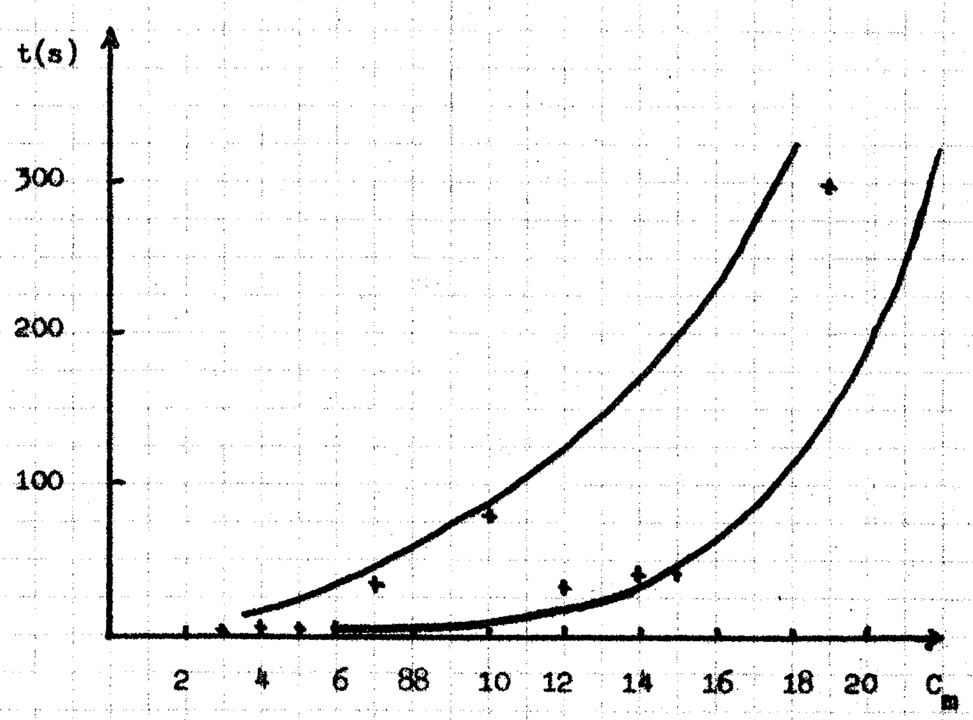
Programme MAJOCO

Temps d'exécution = fonction du coût minimal



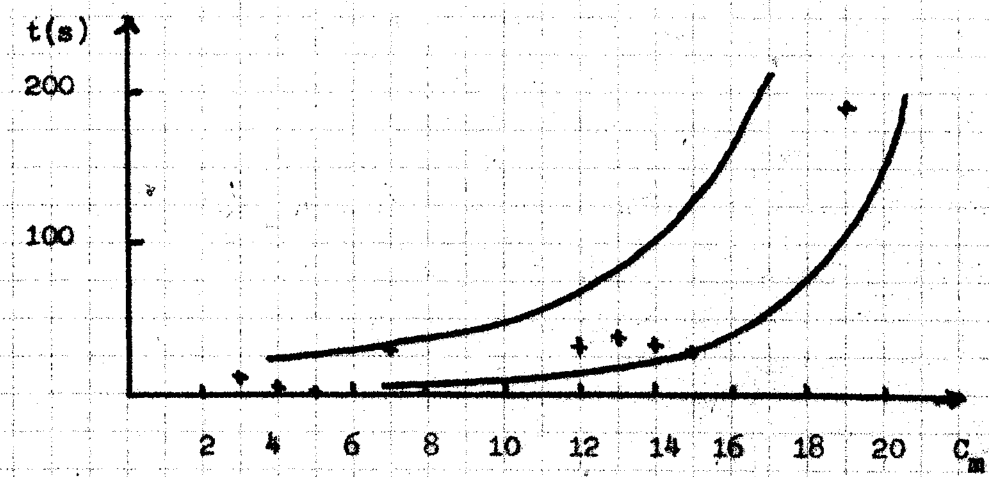
Programme MAJMAX

Temps d'exécution = fonction du coût minimal



Programme MAJORF

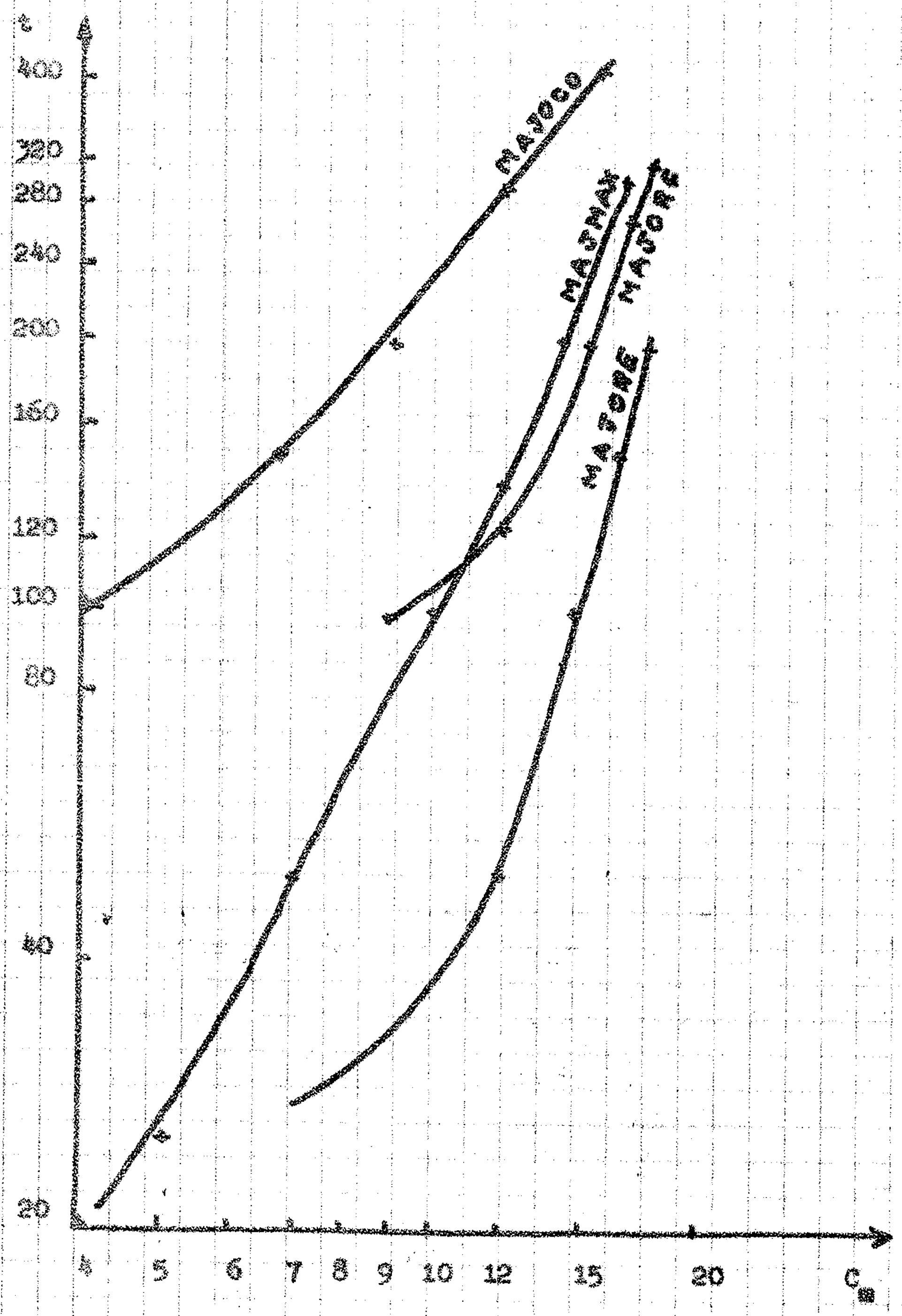
Temps d'exécution = fonction du coût minimum



Programme MAJORE

Temps d'exécution = fonction du coût minimum

Les 4 zones obtenues sont encadrées par des courbes.
Pour essayer de connaître leur allure nous avons tracé les courbes correspondant à la borne supérieure sur papier logarithmique.



Conclusions tirées de l'étude des temps d'exécution.

- Classification - Le programme MAJORE est le plus rapide. De plus les 4 programmes se classent au point de vue rapidité d'exécution dans l'ordre inverse de celui où ils se classent au point de vue coût (graphique des coûts relatifs).

L'explication correspond à celle donnée précédemment :

- MAJORE est le plus rapide :
  - parce que tout résultat intermédiaire qu'il fabrique intervient dans la solution.
  - un résultat intermédiaire n'est pas choisi parmi toutes les combinaisons possibles mais déterminé par un choix très rapide entre un très petit nombre de possibilités.

Il est normal que MAJOCO soit le plus lent car il examine le plus grand nombre de possibilités.

L'étude du graphique en échelle doublement logarithmique fournit des conclusions provisoires intéressantes.

Dans les limites du nombre de points expérimentaux et des valeurs les plus élevées que l'on a pour C_m, il semblerait que pour $C_m > 12$ les 4 courbes tendent vers des droites ce qui correspondrait à des courbes temps = fonction puissance du coût.

Si l'on admet cette hypothèse on constate que la pente de MAJOCO est nettement plus faible (environ 1,3) que celle des 3 autres (supérieure à 3).

Approximativement, dans l'hypothèse de courbes puissances MAJOCO donnerait des temps inférieurs à ceux des autres programmes pour $C_m > 25$.

Si l'on se place au point de vue temps d'exécution MAJORE, dans les limites ci-dessus ($C_m \leqslant 25$) est le plus rapide des 4 programmes.

Toutefois, si l'on voulait obtenir des solutions de coût avantageux en se contentant d'un temps d'exécution raisonnable les 2 programmes MAJMAX et MAJORF semblent le mieux répondre à la question.

III - 4 - CONCLUSIONS

III - 4 - a - Intérêt du programme de minimisation.

Le programme de minimisation ARBRE a des performances très mauvaises au point de vue temps. Toutefois, son intérêt provient de ce que :

- il prouve que la minimisation est possible; on pourrait en effet imaginer une machine suffisamment rapide pour obtenir des résultats acceptables.

- D'autre part, il a révélé un facteur intéressant. Contrairement à ce qu'on aurait pu imaginer ce n'est pas la place disponible en machine qui est restrictive, c'est le temps; on aboutit à des temps considérables bien avant que toute la machine ne soit pleine.

III - 4 - b - Intérêt pratique des méthodes par composition.

Les méthodes heuristiques de synthèse par composition offraient un inconvénient.

Rien ne permettait de dire à priori quelle était leur efficacité au point de vue coût; et surtout, si l'on se place à un point de vue théorique on pouvait uniquement affirmer qu'une telle méthode était convergente : on ne pouvait prévoir au bout de combien d'essais. L'étude de la programmation de trois méthodes par composition permet de voir :

- que ces méthodes sont très intéressantes au point de vue coût, et très voisines l'une de l'autre.
- que pour 2 d'entre elles au moins les temps d'exécution sont tout à fait raisonnables.

Si l'on se place au point de vue pratique de l'exploitation on peut considérer que le temps d'exécution est un facteur d'importance négligeable par rapport au coût de la solution fournie, à condition toutefois que ce temps d'exécution reste dans des limites raisonnables.

De ce point de vue, les méthodes heuristiques par composition peuvent être considérées comme d'excellentes méthodes, rapides et donnant des solutions de coût très raisonnable.

III - 4 - c - <u>Intérêt théorique des méthodes par fractionnement.</u>

Les méthodes par fractionnement ont toutes un point commun : on ne fabrique que le strict nécessaire pour l'obtention d'une solution. On ne fait que très peu de choix et le temps pris en machine par ces choix est insignifiant par rapport au reste.

<u>Conséquence</u> : Comme nous l'avons vu dans le cas de MAJORE le temps sera une fonction du coût, que l'on pourra déterminer avec précision.

D'autre part, la plupart des méthodes de frationnement permettent de calculer mathématiquement une borne supérieure du coût en fonction d'un paramètre de la fonction.

Dans le cas du programme MAJORE, comme nous le verrons au chapitre 4 on peut obtenir une borne supérieure du coût en fonction du nombre de variables.

Nous verrons que cette borne obtenue est déjà assez faible (moins du quart de celle obtenue par la formule de Lagrange).

Le fait que des programmes heuristiques par composition donnent des résultats de coût nettement moindre tendrait à monter que l'opérateur majorité est un opérateur "économique": il permet de réaliser la synthèse d'une fonction avec très peu d'opérateurs.

CHAPITRE - 4 -

ETUDE DU COUT

I - PROPRIETES DES RESEAUX DE COUT MINIMAL

II - ETUDE DE FONCTIONS PARTICULIERES

III - BORNES SUPERIEURES DU MINIMUM DU NOMBRE DE COUCHES

IV - BORNES INFERIEURES DU COUT

V - BORNES SUPERIFURES DU COUT

INTRODUCTION

Nous avons vu au chapitre II un certain nombre de méthodes de synthèse. Le présent chapitre est consacré à l'étude du coût des fonctions représentées par un réseau majoritaire. On dégagera dans ce chapitre des propriétés indépendantes des algorithmes de synthèse et on calculera des valeurs numériques limites, tant pour le coût que pour le nombre de couches.

Nous donnons d'abord un ensemble de propriétés caractéristiques des réseaux minimaux de fonctions impaires croissantes, puis nous étudions certaines fonctions très particulières et dont l'utilisation est fréquente.

Les résultats de ces deux études seront alors utiles à l'établissement de valeurs limites tant pour le coût que pour le nombre de couches.

I - PROPRIETES DES RESEAUX DE COUT MINIMAL.

I - 1 - FONCTIONS N'INTERVENANT JAMAIS DANS UNE REPRESENTATION DE COUT MINIMAL.

Théorème 1. Soit $f(X)$ une fonction impaire croissante de n variables, $(n > 3)$. Si une fonction impaire croissante couvre au plus deux monômes premiers de $f(X)$, elle ne peut intervenir dans une représentation minimale de $f(X)$.

En effet, il y a pour ces deux monômes, au moins une variable qui les couvre; on peut donc remplacer dans le réseau, le sous-réseau réalisant la fonction par cette variable qui est de coût 0.

Théorème 2. Soit $f(x, y, z, X)$ une fonction impaire croissante de n variables $(n > 3)$. S'il est impossible d'obtenir, par réduction à partir de $f(x, y, z, X)$ la fonction $xy + xz + yz$, alors $\mathrm{Maj}\,(x, y, z)$ ne peut figurer dans aucune représentation de coût minimal de $f(x, y, z, X)$.

Démonstration.

$$f(x, y, z, X) = xyz\, h_1(X) + xy\, h_2(X) + xz\, h_3(X) + yz\, h_4(X)$$
$$+ x\, h_5(X) + y\, h_6(X) + z\, h_7(X) + h_8(X).$$

$h_1(X), h_2(X), \dots , h_8(X)$ sont des sommes de monômes dépendant de X et indépendantes de x, y, z.

Considérons un monôme m de $h_2(X)$: $m = m_1 \, . \, m_2$

Réduisons dans f toutes les lettres de m_1 à x, toutes celles de m_2 à y. Soit f_r la fonction obtenue.

f_r dépend effectivement de x et y; en effet, $xy\ h_2(X)$ est alors devenu xy; ce monôme ne peut être éliminé que par un monôme provenant de $x\ h_5(X)$, de $y\ h_6(X)$ ou de $h_8(X)$.

Si xy est éliminé par un monôme de $h_8(X)$, le monôme $xy\ m$ n'aurait pas été premier.

De même si xy était éliminé par un monôme de $x\ h_5(X)$ ou de $y\ h_6(X)$; on peut donc écrire :

$$f_r = xy + \varphi$$

f_r dépend effectivement de 3 variables au moins; φ contient donc au moins une variable différente de x et y et ne contient pas de monôme à une lettre.

Réduisons à z toutes les variables de φ autres que x et y; on obtient une fonction f_{r_1} :

$$f_{r_1} = xy + \psi\ (x, y, z).$$

f_{r_1} dépend effectivement de 3 variables; comme elle est impaire croissante ce ne peut être que :

$$f_{r_1} = xy + xz + yz$$

ce qui est contraire à l'hypothèse; conséquence : le monôme $xy\ m$ n'existe pas d'où :

$$h_2(X) = 0$$

On aurait de même : $h_3(X) = h_4(X) = 0$ et $f(x, y, z, X)$ s'écrit =

$$f(x, y, z, X) = xyz\ h_1(X) + x\ h_5(X) + y\ h_6(X) + z\ h_7(X) + h_8(X).$$

d'où :

Pour la fonction Maj (x, y, z) on est dans le cas du théorème d'élimination : x (ou y ou z) couvre tous les monômes couverts par Maj (x, y, z) et coûte moins cher.

I - 2 - FREQUENCE MINIMUM D'UNE VARIABLE DANS UN RESEAU.

I - 2 - a - Position du problème.[10]

Considérons le problème de la synthèse d'une fonction impaire, (pas nécessairement croissante) à l'aide de l'opérateur majorité.

Si la fonction $f(x, X)$, impaire, est monotone en x on a :

$$f(x, X) = \tilde{x}\, \varphi(X) + \varphi^{*}(X)$$

ou $\varphi(X)$ est sur impaire et indépendante de x.

Or une fonction sur impaire peut toujours être considérée, et cela de plusieurs manières, comme une somme de deux fonctions impaires: soit par exemple :

$$\varphi(X) = \varphi_1(X) + \varphi_2(X)$$

φ_1 et φ_2 étant deux fonctions impaires; d'où :

$$f(x, X) = \tilde{x}\,[\varphi_1(X) + \varphi_2(X)] + \varphi_1(X) \cdot \varphi_2(X)$$

soit :

$$f(x, X) = \text{Maj}\,[\tilde{x}, \varphi_1(X), \varphi_2(X)].$$

Il existe donc un réseau représentant $f(x, X)$ et tel que la variable x apparait une seule fois dans ce réseau.

Mais, même si $f(x, X)$ est croissante par rapport à toutes ses variables, les fonctions $\varphi_1(X)$ et $\varphi_2(X)$ ne sont pas en général croissantes; or, si l'on se propose de représenter une fonction impaire croissante, on n'a à sa disposition que les variables directes.

Nous allons préciser dans ce cas une borne inférieure du nombre d'occurences d'une variable x dans le réseau d'une fonction impaire croissante $f(x, X)$.

I - 2 - b - Cas d'un réseau quelconque.

Théorème 1. Une condition nécessaire et suffisante pour que l'on puisse représenter une fonction impaire croissante $f(x, X)$ par un réseau majoritaire dont une seule entrée est égale à x est que le coefficient de x dans le développement $f(x, X) = x\psi(X) + \psi^*(X)$ puisse être considéré comme une somme de deux fonctions impaires croissantes.

Condition nécessaire.

Supposons que x apparaisse une seule fois comme entrée dans un réseau majoritaire représentant $f(x, X)$.

Soit φ la fonction sortie de l'opérateur dont une des entrées est x.

$$\varphi = x(\varphi_1 + \varphi_2) + \varphi_1 \varphi_2 \qquad \varphi_1 \text{ et } \varphi_2 \text{ impaires croissantes.}$$

Quel que soit le nombre d'occurences de φ dans le réseau on peut toujours écrire :

$$f(x, X) = \varphi g + g^*$$

ou g est sur impaire croissante indépendante de x puisque φ est le seul sous réseau contenant x.

$$f(x, X) = [x(\varphi_1 + \varphi_2) + \varphi_1 \varphi_2].g + g^* = x[\varphi_1 g + \varphi_2 g + g^*] +$$
$$\varphi_1 \varphi_2 g + g^* = x[(\varphi_1 g + g^*) + (\varphi_2 g + g^*)] +$$
$$(\varphi_1 g + g^*)(\varphi_2 g + g^*)$$

Le coefficient de x est une somme de 2 fonctions impaires croissantes ce qui est contraire à l'hypothèse.

<u>Condition suffisante.</u> Elle est évidente :

si $$f(x, X) = x\,[\psi_1(X) + \psi_2(X)] + \psi_1(X)\,.\,\psi_2(X)$$

on a :

$$f(x, X) = \mathrm{Maj}\,[x, \psi_1(X), \psi_2(X)]$$

Exemple :

$$f = xy + xz + xt + yzt.$$

Développons en x :

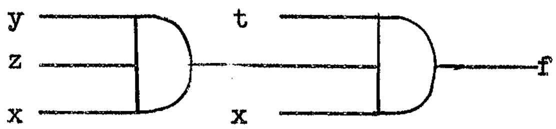
$$f = x(y + z + t) + yzt$$

Le coefficient de x est une somme de 3 fonctions impaires croissantes au moins; il faudra 2 entrées au moins égales à x.

Développons en y :

$$f = y\,(x + zt) + xz + xt$$

Le coefficient de y peut être considéré comme une somme de 2 fonctions impaires croissantes : x et $xz + xt + zt$, il est possible de trouver un réseau ou y apparait une seule fois.



I - 2 - c - <u>Cas d'un réseau arborescent.</u>

<u>Théorème 2.</u> Soit $f(x, X)$ une fonction impaire croissante de n variables.

$$f(x, X) = x\,\psi(X) + \psi^*(X)$$

où $\psi(X)$ est une fonction sur impaire croissante de $n-1$ variables.

Si $\psi(X)$ est, considéré comme somme de fonctions impaires croissantes de toutes les façons possibles, une somme d'au moins p fonctions impaires croissantes, alors dans tout réseau majoritaire arborescent représentant $f(x, X)$ $p-1$ entrées au moins sont égales à x.

<u>Démonstration</u>. Ceci est évidemment vrai si $p = 2$; d'autre part, le théorème précédant le démontre si $p = 3$.

Supposons le vrai pour les fonctions ou on a $p-1$ fonctions impaires au moins dans le coefficient de x; soit $f(x, X)$ une fonction impaire croissante telle que le coefficient de x soit une somme d'au moins p fonctions impaires croissantes.

Supposons un réseau arborescent représentant $f(x, X)$; Soient g_1, g_2, g_3 les 3 fonctions du dernier étage telles que :

$$f(x, X) = \text{Maj } (g_1, g_2, g_3)$$

x apparait q_1 fois dans le réseau de g_1, q_2 fois dans le réseau de g_2, q_3 fois dans le réseau de g_3 et supposons :

$$q_1 + q_2 + q_3 < p - 1$$

$f(x, X)$ dépendant effectivement de x on peut toujours supposer $q_3 \geqslant 1$

$$g_1 = x(h_1 + h_2 + \dots h_{t_1}) + h_1 . h_2 \dots h_{t_1}$$

Rappelons qu'il existe une représentation de g_1 dans laquelle x apparait au plus $t_1 - 1$ fois : c'est la majorité composée de x; $h_1, \dots h_{t_1}$.

On peut donc toujours supposer que :

$$t_1 - 1 \leqslant q_1$$

De même $g_2 = x(k_1 + \ldots + k_{t_2}) + k_1 \ldots k_{t_2}$

$$t_2 - 1 \leqslant q_2$$

$$g_3 = x\ (l_1 + l_2 + \ldots + l_{t_3}) + l_1\ \ldots\ l_{t_2}$$

$$t_3 - 1 \leqslant q_3$$

Soit :

$$f(x,\ X) = [x(l_1 + l_2 + \ldots + l_{t_3}) + l_1 \ldots\ l_{t_3}][x(h_1 + \ldots + h_{t_1}) + h_1 \ldots h_{t_1}$$

$$+ x(k_1 + \ldots + k_{t_2}) + k_1\ \ldots k_{t_2}] + [x(h_1 + \ldots + h_{t_1}) + h_1 \ldots h_{t_1}]$$

$$[x(k_1 + \ldots\ k_{t_2}) + k_1\ \ldots\ k_{t_2}] = x\ [(l_1 + \ldots + l_{t_3})(h_1 + \ldots + h_{t_1}$$

$$+ k_1 + \ldots + k_{t_2}) + (h_1 + \ldots + h_{t_1})\ (k_1 + \ldots + k_{t_2})] + l_1\ \ldots\ l_{t_3}$$

$$(h_1 + \ldots + h_{t_1} + k_1 + \ldots + k_{t_2}) + h_1\ \ldots\ h_{t_1} \ldots k_1 \ldots k_{t_2}$$

Il est possible de mettre le coefficient de x sous la forme d'une somme de fonctions impaires croissantes du type :

$$= h_i(l_1 + \ldots + l_{t_3} + k_1 + \ldots + k_{t_2}) + l_1 \ldots l_{t_3} \cdot k_1 \ldots k_{t_2} \quad (1)$$

ou

$$= k_j(l_1 + \ldots + l_{t_3} + h_1 + \ldots + h_{t_1}) + l_1 \ldots l_{t_3} \cdot h_1\ \ldots h_{t_1} \quad (2)$$

Il faut au plus t_1 fonctions du type (1) , t_2 fonctions du type (2); Le coefficient de x dans $f(x,\ X)$ pourrait se mettre sous la forme d'une somme d'au plus $t_1 + t_2$ fonctions impaires croissantes.

Soit :

$$t_1 + t_2 \leqslant q_1 + q_2 + 2 \leqslant q_1 + q_2 + q_3 + 1 < p$$

Soit $t_1 + t_2 < p$

ce qui est contraire à l'hypothèse.

Remarques. On a supposé $q_3 \geqslant 1$; il est très possible que $q_1 = q_2 = 0$ ce qui revient à dire $t_1 = t_2 = 1$ et ne change rien à la démonstration.

D'autre part, le cas où $g_i = x$ est possible; ceci ne serait valable que pour un des g_i; il suffit de supposer que c'est g_3.

I - 3 - RELATIONS ENTRE LES COUTS.

Rappel d'une définition : Soit f(X) une fonction impaire croissante de n variables ($n \geqslant 3$). On appellera réduite d'ordre p de f la fonction obtenue en réduisant p + 1 variables dans f.

Exemple :

$$f = xy + xz + xt + yzt$$

faisons y = x on a $f_{r_1} = x$ qui est une réduite d'ordre 1

y = z $f_{r_2} = xz + xt + zt$

I - 3 - a - Coût des réduites.

Théorème 1. Soit f(X) une fonction impaire croissante de n variables ($n \geqslant 3$). Il existe au moins 1 fonction, réduite d'ordre 1 de f, et de coût strictement inférieur à celui de f.

<u>Démonstration</u>. Soit une représentation minimale de $f(X)$; soit C_f son coût.

Il y a au moins un opérateur du réseau dont les trois entrées sont des variables; soit Maj (x_i, x_j, x_k); $\{x_i, x_j, x_k\} \subset X$.

Considérons la fonction f_r obtenue en faisant $x_j = x_i$ dans f.

On obtient un réseau représentant f_r en remplaçant dans le réseau de f chaque occurence de x_j par x_i. Dans ce réseau l'opérateur Maj (x_i, x_j, x_k) a maintenant pour entrées x_i, x_i, x_k; sa sortie vaut x_i et on peut le supprimer en le remplaçant par la variable x_i.

On a donc pour f_r un réseau de coût $C_f - 1$.

Donc :

$$\text{Coût } [f_r] \leqslant C_f - 1 \text{ d'où :}$$

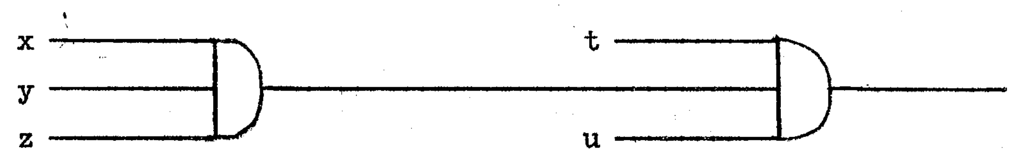
$$\text{Coût } f_r < \text{Coût } f.$$

On pourrait faire de même avec la fonction obtenue en faisant $x_i = x_k$ ou $x_j = x_k$.

<u>Remarque 1</u>. Il est évident que toute réduite de f est de coût au plus égal à f: le réseau de f, sur lequel on fait les remplacements de variables correspondant aux réductions donne une représentation de la réduite.

<u>Remarque 2</u>. Il y a des fonctions pour lesquelles certaines réduites sont de même coût que la fonction elle-même.

<u>Exemple</u> : $f = (xy + xz + yz)\,(t + u) + t\,u$ est représentée par le réseau ci-dessous qui est minimum.



faisons la réduction $t = x$; on obtient $xy + xz + xu + yzu$ qui nécessite également 2 opérateurs.

Applications. Ce théorème permettra dans certains cas d'obtenir des bornes inférieures de coût pour des fonctions et par suite des coûts minimum rigoureux. On en trouvera des exemples dans l'étude de fonctions particulières.

I - 3 - b - Coûts comparés des fonctions de n et $n-1$ variables.

Théorème 2. Toute fonction de n variables peut être considérée comme une réduite d'une fonction de $n+1$ variables.

Démonstration.

$$f(x, X) = x\,\varphi(X) + \varphi^*(X)$$

$f(x, X)$ est une fonction de n variables

$\varphi(X)$ étant sur impaire croissante on peut toujours écrire :

$$\varphi(X) = h(X) + \psi(X)$$

où $\psi(X)$ est impaire croissante de $n-1$ variables.

Considérons alors $g(x, y, X) = xyh(X) + (x+y)\,\psi(X) + \varphi^*(X)$

Si l'on fait $y = x$ on a $g(x, x, X) = f(x, X)$

$g(1, y, X) = yh(X) + \psi(X) + \varphi^*(X)$

$g^*(1, y, X) = [y + h^*(X)]\,\psi^*(X)\,.\,\varphi(X) = y\,\psi(X)\,.\,\varphi(X) + \varphi^*(X).\,\varphi(X)$

$= y\,\psi(X) + \varphi^*(X) = g(0, y, X)$

g est impaire croissante.

Corollaire. Si C_n est le coût de la fonction de n variables la plus chère on a :

$$C_n \geqslant C_{n-1}$$

En effet, si $f(X)$ est la fonction de $n-1$ variables de coût le plus élevé elle est réduite d'une fonction de n variables.

II - ETUDE DE FONCTIONS PARTICULIERES.

II - 1 - DEUX REPRESENTATIONS DE COUT MINIMAL.

II - 1 - a - Représentation de la majorité composée.

On appelle majorité composée à p + 1 variables la fonction impaire croissante :

$$f = x\,S_p^1 + S_p^p = x(y_1 + y_2 + \dots + y_p) + y_1\,y_2 \dots y_p$$

Théorème 1. Toute réduite de f est de coût strictement inférieur à celui de f.

Démonstration. f étant symétrique par rapport aux variables y_1, y_2, ..., y_p les réduites de f se rangent en deux classes d'équivalence :

- celles obtenues en faisant $y_i = y_j$: on obtient $x\,S_{p-1}^1 + S_{p-1}^{p-1}$
- celles obtenues en faisant $y_k = x$; on obtient la variable x qui est de coût 0.

Soit une représentation minimale de f: un opérateur au moins du réseau est tel que ses 3 entrées sont des variables; l'une de ces variables peut être x mais deux d'entre elles au moins seront prises dans l'ensemble $\{y_1, y_2, \dots, y_p\}$; soit y_i et y_j.

En faisant la réduction $y_i = y_j$, la sortie de l'opérateur valant y_j, celui ci peut être remplacé par y_j: la réduite possède une représentation de coût strictement inférieur à celui de f.

Théorème 2. $f(x,Y) = xS_p^1 + S_p^p = x(y_1+y_2+\dots y_p) + y_1\,y_2 \dots y_p$ est de coût p-1.

Nous avons $xS_2^1 + S_2^2 = x(y_1 + y_2) + y_1 y_2 = \mathrm{Maj}(x, y_1, y_2)$

Supposons que ce soit vrai pour les majorités composées à p variables :

$f(x, X) = x\,(y_1 + y_2 + \dots + y_{p-1} + y_p) + y_1 \dots y_p$

$= x\,[x(y_1 + y_2 + \dots + y_{p-1}) + y_1 y_2 \dots y_{p-1}] + y_p\,[x(y_1 + \dots + y_{p-1})$

$\quad + y_1 \dots y_{p-1}]\, y_p$

$x(y_1 + y_2 + \dots + y_{p-1}) + y_1 y_2 \dots y_{p-1} = x\,S_{p-1}^1 + S_{p-1}^{p-1}$

possède une représentation de coût p-2.

D'où $f = \mathrm{Maj}\,[x, y_p, x(y_1 + \dots + y_{p-1}) + y_1 \dots y_{p-1}]$ est de coût p-2 + 1 = p-1.

Cette représentation est minimale : par hypothèse la représentation de $xS_{p-1}^1 + S_{p-1}^{p-1}$ de coût p-2 est minimale et d'après le théorème précédant $xS_p^1 + S_p^p$ est de coût strictement supérieur à celui de $x\,S_{p-1}^1 + S_{p-1}^{p-1}$.

II - 1 - b - Représentation minimale de S_5^3.

Borne inférieure du coût pour S_5^3.

$$S_5^3 = x\,S_4^2 + S_4^3$$

$= x(yz + yt + yu + zt + zu + tu) + yzt + yzu + ytu + ztu.$

S_5^3 est une fonction totalement symétrique; toutes les réduites sont du même type, donc du même coût; étant donné qu'une réduite au moins, est de coût strictement inférieur à celui de f, toutes les réduites sont de coût strictement inférieur à celui de f.

Faisons par exemple la réduction $z = y$

On obtient la fonction g

$$g = x\ (y + tu) + yt + yu$$

$$g = y\ (x + t + u) + xtu$$

Nous avons vu que le coût minimum de cette fonction était de 2.

D'où :

$$\boxed{\text{Coût } (S_5^3) \geqslant 3}$$

<u>Une représentation minimale de S_5^3.</u>

$S_5^3 = x(yz + yt + yu + zt + zu + tu) + yzt + yzu + ytu + ztu$

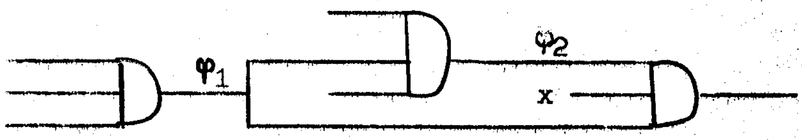
Nous allons montrer qu'il n'y a pas de représentation de coût 3 possible.

Si une représentation est de coût 3 l'une au moins des 3 fonctions entrant dans l'opérateur final est une variable; S_5^3 étant totalement symétrique on peut toujours supposer que c'est x.

Il est impossible qu'une autre entrée soit une variable : la réalisation serait de la forme $\mathrm{Maj}(x,y,\varphi) = xy + (x+y)\ \varphi$ et S_5^3 ne comporte aucun monôme à 2 lettres.

Reste donc le cas où les 2 autres entrées sont des fonctions. Si ce sont 2 opérateurs disjoints, il y a 6 entrées disponibles pour 5 variables : 1 variable au moins est répétée : ce ne peut être x car on aurait un monôme à 2 lettres; reste 4 variables pour 6 entrées : 2 variables y et z seraient répétées et on aurait un monôme à 2 lettres.

Reste le cas du réseau ci-dessous :



Il y a 5 entrées disponibles pour 5 variables; 1 variable doit être répétée; ce ne peut être x pour les mêmes raisons que ci-dessus; ce ne peut être que y (ou z, t, u).
La fonction s'écrirait :

$$x(\varphi_1 + \varphi_2) + \varphi_1 \varphi_2$$

avec $\varphi_2 = \mathrm{Maj}(y, z, \varphi_1)$; $\varphi_1 \varphi_2 = (yz + y\varphi_1 + z\varphi_1)\, \varphi_1$

Il y a un terme en $y\varphi_1$ et $\varphi_1 = y\alpha + \beta$

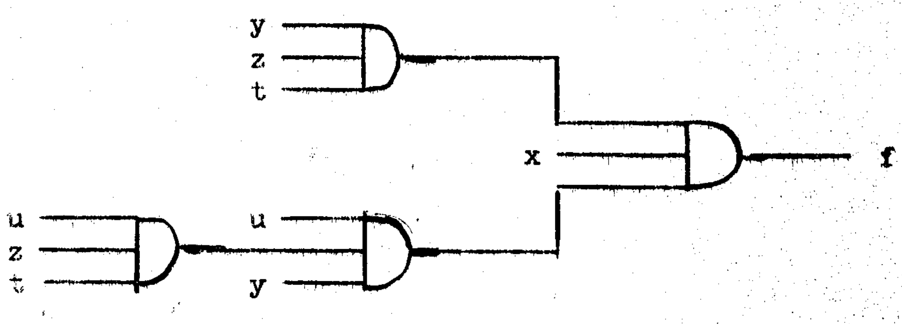
ou α désigne t ou u.
On aurait un monôme à 2 lettres : c'est impossible;
d'où :

$$\text{Coût } [S_5^3] \geqslant 4$$

On a par contre :

$$S_5^3 = \mathrm{Maj}\left\{x,\ \mathrm{Maj}(y, z, t),\ \mathrm{Maj}\,[u, y\mathrm{Maj}(u,z,t)]\right\}$$

qui est à 4 opérateurs
Une telle représentation est donc minimale.



Remarque : Nous verrons en III que cette représentation est minimale en nombre de couches également.

II - 2 - REPRESENTATION DES FONCTIONS DU TYPE $x\, S_k^1 + S_k^{k+1-1}$

S_k^1 désigne la somme des produits 1 à 1 de k variables. Pour qu'une telle fonction soit impaire croissante il faut et il suffit que S_k^1 soit strictement sur impaire croissante soit :

$$\frac{k}{2} \geqslant 1$$

Nous supposerons cette condition remplie et appellerons, par abréviation $g_x(k, 1)$ une telle fonction.

II - 2 - a - Représentation de $g_x(5, 2)$.

$g_x(5, 2) = x(yz+yt+yu+yv+zt+zu+zv+tu+tv+uv)+yztu+yztv+yzuv+ytuv+ztuv$

$= y(xz+xt+xu+xv+ztu- ztv+tuv)+xzt+xzu+xtu+xtv+xuv+xzv+ztuv$

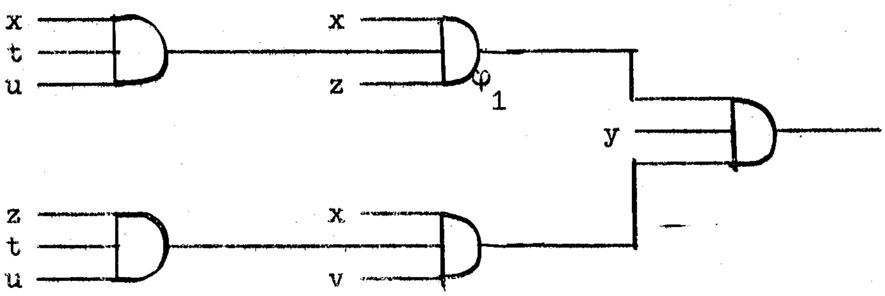
$= y[x(z+t+u)+ztu+v(x+zt+tu+zu) + xzt+xtu+xzu]+xzt+xzu+xtu+xtv+xuv+xzv$
$+ztuv$

Posons : $\varphi_1 = x(z+t+u) + ztu$ fonction de coût 2

$\varphi_2 = v(x+zt+zu+tu)+xzt+xzu+xtu = \mathrm{Maj}[v,x,\mathrm{Maj}(z,t,u)]$

de coût 2

Nous obtenons $g_x(5, 2) = \mathrm{Maj}(y, \varphi_1,\varphi_2)$ qui est de coût 5.



$$\boxed{\text{Coût } [g_x(5,2)] \leqslant 5}$$

II - 2 - b - Formes systématiques pour les fonctions $g_x(k, l)$

On a la forme systématique suivante :

$$\boxed{g_x(k, l) = \text{Maj } [y, g_x(k-1, l-1), g_x(k-1, l)]} \quad \text{I}$$

En effet, posons pour alléger l'écriture :

$$\varphi = \text{Maj}[y, g_x(k-1, l-1), g_x(k-1, l)]$$

$$\varphi = y[x\, S^{l-1}_{k-1} + S^{k-l+1}_{k-1} + x\, S^{l}_{k-1} + S^{k-l}_{k-1}] + (x\, S^{l-1}_{k-1} + S^{k-l+1}_{k-1})$$
$$\cdot\ (x\, S^{l}_{k-1} + S^{k-l+1}_{k-1})$$

$$\varphi = x[y\, S^{l-1}_{k-1} + S^{l}_{k-1} + S^{k-l}_{k-1}] + y\, S^{k-l}_{k-1} + S^{k-l+1}_{k-1}$$

or $k \geqslant 2l$ entraine $k-l > l$ soit $S^{k-l}_{k-1} < S^{l}_{k-1}$

d'où :

$$\varphi = [y\, S_{k-1}^{l-1} + S_{k-1}^{l}] + y\, S_{k-1}^{k-l} + S_{k-1}^{k-l+1}$$

$$\varphi = x\, S_{k}^{l} + S_{k}^{k-l+1} \qquad = g_x(k,\ l)$$

Remarque. Dans le cas : $k = 5$, $l = 2$ la formule ci-dessus conduit à une représentation moins avantageuse que celle trouvée en II-2-a; toutefois cette formule offre l'avantage d'être systématique.

Cas particulier. $k = 2\,l$. On est alors dans le cas des fonctions impaires croissantes totalement symétriques, d'un nombre impair de variables : S_{2l+1}^{l+1}

La formule précédente, appliquée à ce cas particulier redonne la décomposition décrite par Amarel Cooke et Winder [5].

$$\boxed{S_{2l+1}^{l+1} = \mathrm{Maj}\ [x,\ S_{2l-1}^{l},\ \ g_y(2l-1,\ l-1)\ ]} \qquad \mathrm{II}$$

En effet, si $k = 2l$:

$$g_y\ (k-1,\ l) = y\, S_{2l-1}^{l} + S_{2l-1}^{2l-1-l+1} = y\, S_{2l-1}^{l} + S_{2l-1}^{l}$$

$$= S_{2l-1}^{l}$$

II - 3 - APPLICATIONS A CERTAINES FONCTIONS S^{p+1}_{2p+1}.

Représentation de S^4_7.

$$S^4_7 = \mathrm{Maj}\ [x,\ S^3_5,\ g_y(5,\ 2)]$$

Nous avons vu pour S^3_5 une réalisation de coût 4 et pour $g_y(5,2)$ une réalisation de coût 5.

Dans la réalisation de $g_y(5,\ 2)$ intervient une majorité de 3 variables dont aucune n'est y.

Dans S^3_5, fonction indépendante de y il y a une majorité de 3 variables; il suffit de choisir celle intervenant dans $g_y(5,2)$.

De cette manière on a :

$$\text{Coût } S^4_7 \leqslant 1 + \text{Coût } (S^3_5) + \text{Coût } [g_y(5,2)] - 1$$

Soit une réalisation de coût 9.

$$\boxed{\text{Coût } (S^4_7) \leqslant 9}$$

Représentation de S^5_9.

Par application de la formule II on obtient :

$S^5_9 = \mathrm{Maj}\ [x,\ S^4_7,\ g_y(7,\ 3)\ ]$

Posons $\varphi_1 = S^4_7$; φ_1 est une fonction ne dépendant ni de x ni de y.

Coût $(S^5_9) = 1 +$ Coût $(S^4_7) +$ coût $[g_y(7,\ 3)]$

Par application de I on obtient :

$g_y(7,3) = \mathrm{Maj}\ [z,\ g_y(6,2)\ g_y(6,3)]$

$g_y(6,\ 3) = S_7^4 \ = \ \varphi_2$

φ_2 est indépendante de x et de z.
On peut donc écrire :

$$\varphi_1 = z\ S_6^3 + S_6^4$$

$$\varphi_2 = y\ S_6^3 + S_6^4$$

Si l'on fabrique φ_1 suivant le développement ci-dessus, la représentation de φ_2 nécessitera un seul opérateur supplémentaire.
D'où :

Coût (S_9^5) = 1 + Coût (S_7^4) + 1 + 1 + Coût $[g_y(6,2)]$

$g_y(6,2) = \mathrm{Maj}[t,\ g_y(5,1),\ g_y(5,2)]$

Dans la réalisation de $g_y(5,2)$ il y a une majorité indépendante de /;
celle-ci peut donc entrer aussi dans la réalisation de la partie commune de φ_1 et φ_2.

Dans $g_y(5,2)$ il entre une fonction de 4 variables de la forme $y(u+v+w) + uvw$; une telle fonction peut également servir dans la représentation de $g_y(5,1)$.

Soit 3 opérateurs à compter en moins.

Coût (S_9^5) = 3 + 9 + 1 + Coût $[g_y(5,1)]$ + Coût $[g_y(5,2)]$-3

= 3 + 9 + 1 + 4 + 5 - 3 = 19

Soit :

$$\boxed{\text{Coût } (S_9^5) \ \leqslant \ 19}$$

Réalisation de S_{11}^{6}.

S_{11}^{6} = Maj $[x,\ S_9^5,\ g_y(9,\ 4)]$ $\qquad S_9^5 = \varphi_1$

Coût $(S_{11}^6) = 1 +$ Coût $(S_9^5) +$ Coût $[g_y(9,4)]$

$g_y(9,4)$ = Maj $[z,\ g_y(8,3),\ g_y(8,4)]$ $\quad g_y(8,4) = S_9^5 = \varphi_2$

Comme précédemment le coût de φ_1 et φ_2 sera

Coût $(S_9^5) + 1$.

Coût $(S_{11}^6) = 1 +$ Coût $(S_9^5) + 1 + 1 +$ Coût $[g_y(8,3)]$

$= 22 +$ Coût $[g_y(8,3)]$

$g_y(8,3)$ = Maj $[t,\ g_y(7,2),\ g_y(7,3)]$

$$\begin{cases} g_y(7,2) = \text{Maj } [u,\ g_y(6,1),\ g_y(6,2)] \\ g_y(7,3) = \text{Maj } [u,\ g_y(6,2),\ g_y(6,3)] \end{cases}$$

$g_y(6,2)$ intervenant à la fois dans $g_y(7,2)$ et dans $g_y(7,3)$ sera réalisée une seule fois.

$$g_y(6,3) = S_7^4 = \varphi_3$$

Dans la réalisation de φ_3 intervient, nous l'avons vu plus haut, une fonction du type S_5^3 indépendante de y. Celle ci pourra donc intervenir dans la réalisation de la partie commune de φ_1 et φ_2.

D'où : Coût $g_y(8,3) = 1 + 1 +$ Coût $[g_y(6,1)] +$ Coût $g_y(6,2) +$

Coût$(S_7^4) - 4 + 1 = 8 +$ Coût $[g_y(6,1)] +$ Coût $[g_y(6,2)]$

Coût $[g_y(6,1)] = 5$

$g_y(6,2)$ = Maj $[v,\ g_y(5,1),\ g_y(5,2)]$

Comme précédemment on peut trouver pour $g_y(5, 2)$ et $g_y(5,1)$ un sous réseau commun à 2 opérateurs.

Soit Coût $[g_y(6,2)]$ = 1 + Coût $[g_y(5,1)]$ + Coût $[g_y(5,2)]$ - 2

$$= 1 + 4 + 5 - 2 = 8$$

Soit coût $[g_y(8,3)] = 8 + 5 + 8 = 21$

Coût $(S^6_{11}) = 22 + 21 = 43$

$$\boxed{\text{Coût } (S^6_{11}) \leqslant 43}$$

II - 4 - REALISATION DES FONCTIONS LINEAIRES.

II - 4 - a - Réalisation de $x \oplus y \oplus z$.

Il s'agit d'une fonction impaire non croissante. Comme elle n'est croissante par rapport à aucune variable il faut dédoubler x y et z.

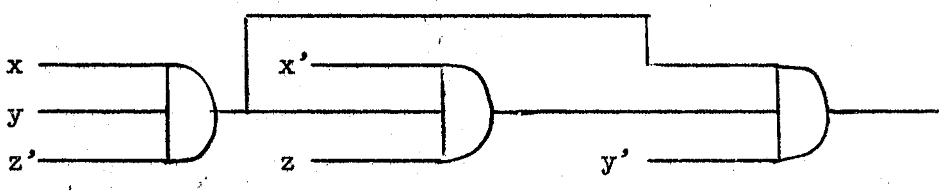
On posera : $x' = u$; $y' = v$; $z' = w$;

$$x \oplus y \oplus z = xyz + xy'z' + x'yz' + x'y'z = xyz + xvw + uyw + uvz$$

Nous obtenons une fonction sous impaire croissante de 6 variables.

Nous avons pour cette fonction une représentation de coût 3 vue au chapitre 3 - II - 2 - b -

$$x \oplus y \oplus z = \text{Maj}\,[y', \text{Maj}(x, y, z'), \text{Maj}\,[z, x', \text{Maj}(x,y,z')]\,]$$



Cette réalisation est d'ailleurs minimale : il faut 6 entrées libres; on ne peut utiliser moins de 3 opérateurs pour connecter les 3 variables et leurs compléments.

II - 4 - b - Réalisation des disjonctions d'un nombre impair de variables.

Théorème. Il existe pour la disjonction de $2p+1$ variables une réalisation de coût $3(2p-1)$

Cette représentation existe si $p = 1$

Supposons qu'il existe une telle représentation pour les fonctions où $p \leq p_o - 1$ et soit :

$f = x_1 \oplus x_2 \oplus \dots x_{2p_o+1}$

3 cas sont possibles :

a)- $2p_o+1$ est un multiple de 3. $2p_o+1 = 3k$ $k < p_o$

$2p_o + 1$ étant impair k est impair.

Groupons les variables par paquets de 3 :

$x_1 \oplus x_2 \oplus x_3 = u_1$; $x_4 \oplus x_5 \oplus x_6 = u_2$; .... $x_{2p_o-1} \oplus x_{2p_o}$

$\oplus x_{2p_o+1} = u_k$

On connait pour u_1 et u'_1 une réalisation de coût 3 pour chaque.

$g = u_1 \oplus u_2 \oplus \dots \oplus u_k$ est de coût $3(k-2)$

à cela il faut ajouter les coûts des réseaux $u_1, u'_1, \dots u_k, u'_k$

Soit $6k$.

Soit une réalisation de coût $3k - 6 + 6k = 9k - 6$

$= 3(2p_o + 1) - 6 = 6p_o - 3$

$= 3(2p_o - 1)$

b)- $2p_o + 1 = 3\,k + 1$ $\qquad$ $1 + k < p_o$

$2p_o + 1$ étant impair $3\,k$ est pair et k l'est également.
Groupons les $2p_o$ premières variables par paquets de 3 :
$x_1 \oplus x_2 \oplus x_3 = u_1 ; x_4 \oplus x_5 \oplus x_6 = u_2 ; \ldots . \; x_{2p-2} \oplus x_{2p_o-1} \oplus x_{2p_o} = u_k ;$
et réalisons la fonction :

$$g = u_1 \oplus u_2 \oplus \ldots \oplus u_k \oplus x_{2p_o+1}$$

g est une disjonction d'un nombre impair de variables, il en existe une représentation de coût :

$$3(\,k + 1 - 2) = 3\,(k-1)$$

Pour obtenir f il faut remplacer $u_1, u_1', u_2, \ldots u_k, u_k'$ par leurs réalisations en fonction des x_i ; on a un coût total de 6k.

Coût $f = 3k - 3 + 6k = 3\,(3k - 1)$

or $3k = 2p_o$

Soit Coût $= 3(2p_o - 1)$

c)- $2p_o + 1 = 3k + 2.$ $\qquad$ $2 + k < p_o$

$2p_o + 1$ étant impair $3\,k$ l'est aussi et k également.
Groupons les $2p_o-1$ premières variables par paquets de 3.

$u_1 = x_1 \oplus x_2 \oplus x_3 ; \ldots$ $\qquad$ $u_k = x_{2p_o-3} \oplus x_{2p_o-2} \oplus x_{2p_o-1}$

et considérons la fonction :

$$g = u_1 \oplus u_2 \oplus \ldots . \oplus u_k \oplus x_{2p_o} \oplus x_{2p_o+1}$$

k étant impair $k + 2$ l'est également et g est réalisable avec un coût de $3(k+2-2) = 3\,k$

A cela s'ajoute le coût des fonctions u_1, ...,u_k,u'_k soit un coût total de 6k

Soit en tout 9k $= 3\ (3k) = 3\ (2p_o-1)$

II - 4 - c - Disjonction d'un nombre pair de variables.

Il suffit de remplacer f par $f \oplus z$ en faisant ensuite $z = 0$.

D'où pour une fonction de 2p variables un coût de 3 (2p-1) également pour 2p variables effectives.

III - BORNES SUPERIEURES DU MINIMUM DU NOMBRE DE COUCHES.

III - 1 - FONCTIONS IMPAIRES OU SOUS IMPAIRES CROISSANTES.

III - 1 - a - Borne supérieure du nombre de couches d'un réseau couvrant un nombre donné de lignes.

Etant donné le tableau d'une fonction réalisable on se propose de déterminer une borne supérieure du nombre de couches d'un réseau couvrant ce tableau et ce, en fonction du nombre de lignes du tableau.

Supposons donc que tout ensemble de p_k lignes de ce tableau puisse être couvert par une fonction à k couches au plus.

On peut alors fabriquer un réseau k+1 couches avec 3 réseaux k couches qui couvriront en tout :

$$p_{k+1} = p_k + \mathrm{Min}(q, r) \qquad \text{avec} \quad q + r = p_k$$

On voit que p_{k+1} sera maximum pour $q = r$.
Si p_k est impair on peut donc généraliser en écrivant :

$$\boxed{p_{k+1} = p_k + E\left(\frac{p_k}{2}\right)}$$

Or on sait que $p_o = 2$; en effet, tout ensemble de 2 lignes est couvert par une variable.
On peut donc calculer le nombre maximum de couches permettant de couvrir un nombre donné de lignes par récurence.

Nombre de lignes	Borne supérieure du nombre minimum de couches
$p \leqslant 2$	0
$p = 3$	1
$p = 4$	2
$5 \leqslant p \leqslant 6$	3
$7 \leqslant p \leqslant 9$	4
$10 \leqslant p \leqslant 13$	5
$14 \leqslant p \leqslant 19$	6
$20 \leqslant p \leqslant 28$	7
$29 \leqslant p \leqslant 42$	8
$43 \leqslant p \leqslant 63$	9
$64 \leqslant p \leqslant 94$	10

III - 1 - b - Fonctions impaires croissantes.

Soit $f(x, y, z, X)$ une fonction impaire croissante. Nous avons vu au Chapitre 2 une forme systématique :

$$f(x, y, z, X) = \mathrm{Maj}\,[f(x,x,z,X),\ f(x,y,y,X),\ f(z,y,z,X)]$$

$f(x,x,z,X)$, $f(x,y,y,X)$, $f(z,y,z,X)$ sont des fonctions de n-1 variables au plus. En appliquant systématiquement la formule ci-dessus, au bout de n-3 couches , on aboutit à des fonctions de 3 variables au plus c'est-à-dire des fonctions 1 couche au plus d'où :

<u>Théorème</u>. Toute fonction impaire croissante de n variables possède une représentation à n-2 couches au plus.

Exemple : Reprenons l'exemple de S_5^3; d'après le théorème ci-dessus il doit exister une représentation 3 couches; celle trouvée au chapitre 4 - II - en a d'ailleurs 3 -.

Montrons que 3 est le minimum du nombre de couches.

Etant donné que le nombre minimum d'opérateurs est 4 la seule possibilité, dans le cas d'une représentation 2 couches est : Maj $(\varphi_1, \varphi_2, \varphi_3)$ où $\varphi_1, \varphi_2, \varphi_3$ sont des fonctions de 3 variables.

On peut toujours, à cause de la symétrie totale supposée $\varphi_1 = xy + xz + yz$.

Il y a 9 entrées libres; comme il y a 5 variables, l'une d'entre elles soit x n'apparait qu'une fois donc dans φ_1 d'où :

$$f = x\,(y + z)\,\Psi + h$$

Or, il est impossible de mettre le coefficient de x sous la forme $(y + z)\,\Psi$; la représentation trouvée précédemment était donc également minimale en nombre de couches.

III - 2 - FONCTIONS CROISSANTES.

Si l'on considère la forme systématique vue au chapitre 2, consistant à faire 3 réductions pour obtenir 3 fonctions de n-1 variables cette méthode est valable pour toutes les fonctions croissantes; par ce procédé, au bout de n-2 réduction, on aboutira à un ensemble de fonctions croissantes de 2 variables au plus; ces fonctions ont une représentation 1 couche au plus. d'où :

Théorème : Pour une fonction croissante de n variables, il existe une représentation à n-1 couches au plus.

III - 3 - FONCTIONS QUELCONQUES.

Nous utiliserons de préférence à la formule de Lagrange qui donnerait 2n-2 couches, une forme systématique vue au chapitre 2 - III -.

Soit $f(x, y, z, X)$ une fonction quelconque de n variables. On a vu qu'en considérant x et x' comme deux variables indépendantes on effectuait l'opération consistant à remplacer dans l'écriture de f, chaque occurence de x par y sans toucher à x' et chaque occurence de y' par x'; symboliquement on avait noté cette opération :

$\begin{cases} x \to y \\ y' \to x' \end{cases}$ On appelle φ_1 la fonction obtenue.

$\begin{cases} y \to z \\ z' \to y' \end{cases}$ φ_2 est la fonction obtenue

$\begin{cases} z \to x \\ x' \to z' \end{cases}$ φ_3 est la fonction obtenue

φ_1 est croissante en y, décroissante en x.
φ_2 - - z, - y
φ_3 - - x, - z.

φ_1, φ_2, φ_3 sont donc des fonctions monotones par rapport à au moins deux de leurs variables et on avait vu que :

$$\mathrm{Maj}(\varphi_1, \varphi_2, \varphi_3) = f.$$

Au bout d'un nombre d'opérations de ce type égal à $E(\frac{n}{2})$ où $E(\frac{n}{2})$ est le plus grand entier inférieur ou égal à $\frac{n}{2}$, on obtient un ensemble de fonctions monotones par rapport à toutes les variables sauf une.

En dédoublant cette dernière variable comme il est indiqué au chapitre 1 - II, on obtient des fonctions monotones d'au plus n+1 variables.

En leur applicant la méthode de réduction des fonctions croissantes, on aboutit aux variables au bout de n couches au plus. D'où :

Théorème. Pour une fonction quelconque de n variables, il existe une représentation majoritaire à $n + E\left(\frac{n}{2}\right)$ couches au plus.

Remarque 1. Il est possible de rendre la formule plus précise. Soit $f(X)$ une fonction de n variables, qui est non monotone par rapport à p d'entre elles . $p \leqslant n$

Il suffit d'appliquer la formule des fonctions quelconques aux p variables par rapport auxquelles la fonction n'est pas monotone.

La borne supérieure du nombre de couches peut alors s'écrire :

$$n - 1 + E\left(\frac{p}{2}\right) + i$$

$i = 1$ si $p \neq 0$ $\qquad$ $i = 0$ si $p = 0$

On voit que si $p = 0$ (fonction monotone par rapport à toutes les variables) ou retrouve $n-1$.

Remarque 2. En général il y a une certaine incompatibilité entre un faible nombre de couches et un faible coût.

En particulier les méthodes de réalisation conduisant à un nombre de couches faible conduisent à un nombre d'opérateurs élevé.

IV - BORNES INFERIEURES DU COUT D'UNE FONCTION IMPAIRE CROISSANTE.

IV - 1 - BORNE INFERIEURE STRICTE DU COUT D'UNE FONCTION DE N VARIABLES.

Définition d'une entrée libre d'opérateur.

Dans un réseau on appelle entrée libre d'un opérateur une entrée qui n'est pas connectée à la sortie d'un autre opérateur.

Théorème. Dans un réseau majoritaire arborescent le nombre d'entrées libres du réseau est indépendant de la structure de celui-ci et vaut $2p + 1$ s'il y a p opérateurs.

Démonstration. La propriété est vérifiée si $p = 1$.

Supposons la vérifiée par les réseaux ayant au plus $p-1$ opérateurs; soit un réseau arborescent ayant p opérateurs.

Un opérateur du réseau au moins, possède 3 entrées libres. Le réseau étant arborescent la sortie de cet opérateur est connectée à l'entrée d'un seul autre opérateur.

Supprimons l'opérateur (ou l'un des opérateurs) dont les 3 entrées sont libres; nous avons par rapport au réseau initial :

- supprimé 3 entrées libres
- crée une entrée libre.

Soit supprimé 2 entrées libres; le réseau restant est à $p - 1$ opérateurs, donc il possède $2(p-1) + 1$ entrées libres. Le réseau initial avait $2(p-1) + 1 + 2 = 2p + 1$ entrées libres.

Remarque. Si un réseau majoritaire comporte n entrées libres il est de coût minimal lorsque'il est arborescent.

En effet, supposons le non arborescent : on peut le rendre arborescent de deux manières :

- en dédoublant un opérateur utilisé plusieurs fois : coût et nombre d'entrées libres augmentent.

- en coupant les liaisons pour rétablir l'arborescence : le nombre d'entrées libres augmente.

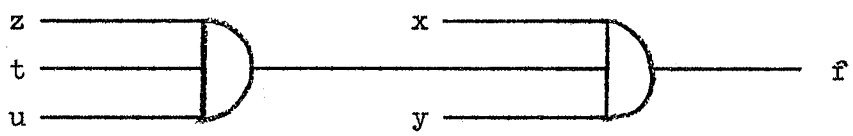
Application 1. Borne inférieure du coût d'une fonction impaire croissante de n variables.

Si une fonction impaire croissante dépend effectivement de n variables, il faut pouvoir disposer, dans tout réseau la représentant, d'au moins n entrées libres; or le réseau ayant n entrées libres et de coût minimal est arborescent et de coût $< \frac{n}{2}$.

Soit : la borne inférieure du coût d'une fonction impaire croissante de 2p variables est p; celui d'une fonction impaire croissante de 2p + 1 variables p également.

Remarque : Cette borne peut être effectivement atteinte comme le montre l'exemple de cette fonction de 5 variables ayant un réseau de coût 2:

$$f = xy + (x+y)\ (zt+zu+tu)$$



Application 2.

Borne inférieur du coût arborescent d'une fonction impaire croissante.

Soit f(X) une fonction impaire croissante de n variables.

On peut écrire :

$$f(X) = x \quad (X_1) + \psi^{*}(X_1)$$

$$x \cup X_1 = X \qquad\qquad x \cap X_1 = \emptyset$$

(X_1) est une fonction sur impaire croissante de n-1 variables.

Il est alors possible de trouver p fonctions impaires croissantes de n-1 variables au plus $_1(X_1)$, $_2(X_1)$....... $_p(X_1)$ telles que :

- $\psi_1(X) + \ldots\ldots + \psi_p(X) = \psi\ (X)$
- p est minimum.

D'après un théorème vu au chapitre 4 tout réseau majoritaire arborescent représentant f est tel que p-1 entrées au moins sont égales à x.

Soit p_x le nombre ainsi trouvé.

On peut de même chercher p_y, p_z,....., p_1; $\{y, z, \ldots, 1\} = X_1$

La somme $p_x + p_y + \ldots\ldots + p_1$ représente alors un minimum strict du nombre d'entrées libres pour tout réseau arborescent représentant f au moyen de majorités et le plus grand entier inférieur ou égal à $\frac{1}{2}(p_x + p_y + \ldots + p_1)$ un minimum strict du coût majoritaire arborescent de f(x, X).

<u>Exemple d'application</u>.

$$f(x, y, z, t, u) = xy + xz + xtu + yzt + yzu$$

Il s'agit d'une fonction impaire croissante de 5 variables; le minimum absolu des fonctions de 5 variables est :

$$\frac{1}{2}(5-1) = 2 \text{ opérateurs.}$$

Cherchons le minimum arborescent du coût pour cette fonction.

Développons en x : $f = x(y + z + t\,u) + yzt + yzu$

Le coefficient de x est une somme de 3 fonctions impaires croissantes au moins; d'où :

$$p_x = 3 - 1 = 2$$

f étant symétrique par rapport à y et z il suffit de calculer p_y.

$$f = y\,(x + zt + zu) + zx + xtu$$

Le coefficient de y peut être considéré comme somme de 2 fonctions impaires croissantes : x et par exemple $z(x+t+u)+xtu$ d'où :

$$p_y = p_z = 2 - 1 = 1$$

f étant symétrique par rapport à t et u il suffit de calculer p_t

$$f = t(xy + yz) + xy + xz + yzu$$

$$f = t(xy + xz + xu + yz) + xy + xz + yzu$$

là encore 2 fonctions impaires croissantes suffisent : par exemple :

$$x(y + z + u) + yzu$$

et $$xy + xz + yz$$

d'où : $$p_t = p_u = 1$$

Soit :

$$p_x + p_y + p_z + p_t + p_u = 2 + 1 + 1 + 1 + 1 = 6$$

D'où le coût arborescent minimum : $\frac{1}{2}\ 6 = 3$

Ce coût est d'ailleurs effectivement atteint par les réseaux de coût minimum représentant f.

IV - 2 - UTILISATION D'UNE PROPRIETE DES REDUITES.

Si $f(X)$ est une fonction impaire croissante de n variables on a vu en I - 3 - a - que :

- toute réduite de f (fonction obtenue en réduisant 2 ou plusieurs variables dans f) est de coût au plus égal à celui de f.
- 1 fonction réduite au moins, est de coût strictement inférieur à celui de f.

Cette propriété permet dans certains cas d'obtenir des bornes inférieures de coût.

Exemple. $$f(x,y,z,t,u) = S_5^3 = xyz + xyt + xyu + xzt + xzu + yzt + yzu + ytu + ztu.$$

La fonction étant de 5 variables nous avons vu que le minimum absolu est $\frac{1}{2}(5-2) = 2$.

Cherchons le minimum arborescent :

$$f = x\,(yz + yt + yu + zt + zu + tu) + yzt + yzu + ytu + ztu$$

2 fonctions impaires croissantes suffisent : par exemple :

$$y(z + t + u) + ztu$$

et

$$zt + zu + tu$$

d'où :

$$p_x = 1$$

La fonction étant totalement symétrique $p_x = p_y = p_z = p_t = p_u$ et

$$p_x + p_y + p_z + p_t + p_u = 5$$

Le minimum arborescent trouvé est également 2.

On a vu par ailleurs que pour cette fonction le coût minimum était de 4.

Utilisons le théorème sur les réduites. f étant totalement symétrique, toutes les fonctions obtenues en réduisant 2 variables sont équivalentes, donc de même coût; donc elles sont toutes de coût strictement inférieur à celui de f.

Faisons par exemple la réduction $y = x$

$$f_r = x(z + t + u) + ztu$$

Cette fonction a pour coût minimum 2. (fonction de 4 variables) d'où :

$$\text{Coût } f > 2$$

Soit une borne inférieure du coût égale à 3.

On a vu en II - 1 - b - que le coût minimum de cette fonction est de 4.

V - BORNES SUPERIEURES DU COUT POUR LES FONCTIONS IMPAIRES CROISSANTES.

V - 1 - RAPPELS ET REMARQUES.

V - 1 - a - Utilisation de la formule de Lagrange pour les fonctions quelconques.

Soit $f(x, X)$ une fonction quelconque de n variables; on a :

$$f(x, X) = xf(1, X) + x'f(0, X)$$

$$= \text{Maj}\ \{1, \text{Maj}[x, 0, f(1,X)], \text{Maj}\ [x', 0, f(0,X)]\ \}$$

Si l'on appelle C_n une borne supérieure du coût pour une fonction de n variables on peut écrire :

$$C_n \leqslant 2\ C_{n-1} + 3$$

$$C_{n-1} \leqslant 2\ C_{n-2} + 3$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$C_3 \leqslant 2\ C_2 + 3$$

$$C_2 \leqslant 3 \qquad \text{Car } C_1 = 0$$

Soit en multipliant par 2 les termes de l'inégalité de la 2ème ligne, par 2^2 pour la 3ème ligne etc... par 2^{n-2} ceux de la dernière ligne il vient :

$$C_n \leqslant 3 + 3 \cdot 2 + 3 \cdot 2^2 + \ldots + 3 \cdot 2^{n-2}$$

$$C_n \leqslant 3(2^{n-1} - 1)$$

$$\boxed{C_n \leqslant \frac{3}{2}(2^n - 1)}$$

V - 1 - b - Cas particulier des fonctions croissantes. (ou monotones)

$$f(x, X) = xf(1, X) + f(0, X)$$

$$= \mathrm{Maj}\ \{1, f(0,X), \mathrm{Maj}[x, 0, f(1, X)]\}$$

Si K_n est une borne supérieure du coût pour une fonction croissante de n variables, on peut écrire :

$$K_n \leqslant 2 + 2\ K_{n-1}$$

$$K_{n-1} \leqslant 2 + 2\ K_{n-2}$$

$$\dots\dots\dots\dots\dots\dots$$

$$K_2 \leqslant 2 \qquad \text{car } K_1 = 0$$

Les seules fonctions croissantes de 2 variables sont xy et x + y toutes deux de coût 1.

Les seules fonctions croissantes dépendant effectivement de 3 variables sont :

xy + z	et x + yz	Coût 2
xyz	et x + y + z	Coût 2
xy + xz + yz		Coût 1

On peut donc admettre que : $K_3 \leqslant 2$

Soit :

$$K_n \leqslant 2 + 2\ K_{n-1}$$

$$K_{n-1} \leqslant 2 + 2\ K_{n-2}$$

$$\dots\dots\dots\dots\dots\dots$$

$$K_3 \leqslant 2$$

$$K_n \leqslant 2(1 + 2 + \dots + 2^{n-3})$$

$$K_n \leqslant 2(2^{n-2} - 1)$$

$$\boxed{K_n \leqslant 2^{n-1} - 2} \qquad (n \geqslant 3)$$

V - 2 - CAS DES FONCTIONS IMPAIRES CROISSANTES.

Les fonctions impaires croissantes sont un cas particulier de fonctions croissantes.

Si on s'autorise l'utilisation des deux constantes 0 et 1, nous obtenons le résultat vu ci-dessus.

Par contre si l'on s'interdit leur utilisation ces formules ne sont plus valables.

Toutefois, la formule des fonctions croissantes pourra servir de référence pour juger de la valeur des formules obtenues ci-dessous.

V - 2 - a - Cas général d'une fonction impaire croissante de n variables.

Nous allons utiliser une méthode de synthèse vue au chapitre 2.

Soit $f(x, y, X)$ une fonction impaire croissante de n variables.

$$f(x, y, X) = x\,[y\,\Psi_1(X) + \Psi_2(X)] + y\,\Psi_2^*(X) + \Psi_1^*(X)$$

où :

$$y\,\Psi_1(X) + \Psi_2(X) \text{ est sur impaire croissante.}$$

en posant :

$$g(x, y, X) = x\,[y + \Psi_2(X)] + y\,\Psi_2^*(X).$$

On a vu que :

$$f(x, y, X) = g(x, y, X)\,\Psi_1(X) + \Psi_1^*(X)$$

Soit en considérant g comme une variable nouvelle.

$$h(g, X) = g\,\Psi_1(X) + \Psi_1^*(X)$$

On a : $\quad$ Coût [f] = Coût [h] + coût [g]

Dans cette formule, on remarque :

- que $h(g, X)$ est une fonction impaire croissante de n-1 variables au plus.
- que $g(x, y, X)$ est une fonction impaire croissante ayant la particularité d'avoir un monôme à deux lettres.

- Si l'on appelle C_n une borne supérieure du coût pour une fonction de n variables.

- Si l'on appelle Q_n une borne supérieure du coût pour une fonction de n variables ayant un monôme à 2 lettres on a :

$$\boxed{C_n \leqslant C_{n-1} + Q_n} \qquad (1)$$

V - 2 - b - <u>Etude particulière des fonctions impaires croissantes ayant un monôme à deux lettres.</u>

Soit $g(x, y, z, X)$ une fonction impaire croissante, ayant un monôme à deux lettres, et de la forme :

$$g(x, y, z, X) = x\,[y + z\,\varphi_1(X) + \varphi_2(X)] + y\,[z\,\varphi_2^*(X) + \varphi_1^*(X)\,]$$

On sait que l'on peut écrire :

$$g(x, y, z, X) = \text{Maj}\,[g(x, x, z, X),\ g(x, y, y, X),\ g(z, y, z, X)]$$

$$g(x, x, z, X) = x$$

$$g(x, y, y, X) = x[y + \varphi_2(X)] + y\,\varphi_2^*(X) = g_1(x, y, X)$$

$$g(z, y, z, X) = z[y + \varphi_1(X)] + y\,\varphi_1^*(X) = g_2(y, z, X)$$

$g_1(x, y, X)$, $g_2(y, z, X)$ sont deux fonctions impaires croissantes d'au plus n-1 variables et ayant un monôme à 2 lettres.

D'où :

$$\boxed{Q_n \leqslant 2\,Q_{n-1} + 1} \qquad (2)$$

<u>Application</u>. L'utilisation des 2 formules 1 et 2 va nous permettre de calculer une borne supérieure du coût en fonction du nombre de variables.

$$Q_4 \leqslant 2$$
$$Q_5 \leqslant 2\ Q_4 + 1$$
....
$$Q_{n-1} \leqslant 2\ Q_{n-2} + 1$$
$$Q_n \leqslant 2\ Q_{n-1} + 1$$

En multipliant l'avant dernière inégalité par 2, la précédante par 2^2 etc ....., on a :

$$Q_n \leqslant 1 + 2 + \ldots + 2^{n-5} + 2\ .\ 2^{n-4}$$
$$Q_n \leqslant 1 + 2 + \ldots + 2^{n-5} + 2^{n-4} + 2^{n-4}$$
$$Q_n \leqslant 2^{n-3} - 1 + 2^{n-4} = 2\ .\ 2^{n-4} + 2^{n-4} - 1$$

$$\boxed{Q_n \leqslant 3\ .\ 2^{n-4} - 1}$$

$$C_4 = 2$$
$$C_5 = C_4 + Q_4$$
....
$$C_n = C_{n-1} + Q_n$$

En remplaçant Q_n par sa valeur on a :

$$C_4 \leqslant 2$$
$$C_5 \leqslant C_4 + 3\ .\ 2^1 - 1$$

En additionnant terme à terme on a :

$$C_n \leqslant 3\ (2^0 + 2^1 + \ldots + 2^{n-4}) - (n - 3)$$
$$C_n \leqslant 3\ (2^{n-3} - 1) - n + 3$$

$$\boxed{C_n \leqslant 3\ .\ 2^{n-3} - n}$$

Ce résultat représente, dans le cas particulier des fonctions impaires croissantes une amélioration de 25°/。 par rapport à celui obtenu en utilisant la formule de Lagrange appliquée aux fonctions croissantes.

Cependant une amélioration du résultat ci-dessus a encore été possible par l'étude systématique du coût pour des valeurs faibles de n.

V - 2 - c - Etude des fonctions de n variables pour $n \leqslant 9$.

Fonctions de 5 variables.

Toute fonction impaire croissante de 5 variables ayant un monôme à 2 lettres possède une réalisation de coût $\leqslant 3$.

S_5^3 seule fonction impaire croissante de 5 variables n'ayant pas de monômes à 2 lettres possède une représentation de coût 4.

Fonctions de 6 variables.

Une étude systématique en machine de tous les types de fonctions impaires croissantes de 6 variables ayant un monôme à 2 lettres a donné un coût au plus égal à 5.

Une étude des fonctions n'ayant pas de monôme à 2 lettres a permis d'affirmer que :

$$c_6 \leqslant 7$$

Dans le cas de 7 variables, on a obtenu :

$$c_7 \leqslant 9 \qquad C_7 \leqslant 15$$

Dans le cas de 8 variables, il a fallu se contenter de la formule, ce qui donne :

$$c_8 \leqslant 19 \qquad C_8 \leqslant 34$$

Par contre dans le cas de 9 variables, il a été possible d'obtenir :

$$Q_9 \leqslant 36 \qquad \text{d'où :} \qquad C_9 \leqslant 70$$

Partant de ces 2 valeurs Q_9 et C_9, on aboutit pour $n > 9$ à des formules qui sont :

$$\boxed{Q_n \leqslant 37 \cdot 2^{n-9} - 1}$$

$$\boxed{C_n \leqslant 5 - n + 37 \cdot 2^{n-8}}$$

Par rapport à la formule précédente, le gain est de 40°/∘ pour $n=20$.

V - 3 - RECAPITULATION DES RESULTATS.

V - 3 - a - Fonctions quelconques.

Nombre de variables	Nombre maximum de couches	Coût minimum	Borne supérieure du coût
2	2	3	3
3	4	4	9
4	6	5	21
5	7	6	45
6	9	7	93
7	10	8	189
8	12	9	381
9	13	10	765
10	15	11	1533
11	16	12	3069
12	18	13	6141
13	19	14	12285
14	21	15	24573
15	22	16	49149
16	24	17	98301
17	25	18	196605
18	27	19	393213
19	28	20	786429
20	30	21	1572861

Les valeurs sont calculées pour des fonctions ne possédant aucun caractère de monotonie, d'imparité de sous ou sur imparité.

V - 3 - b - Fonctions croissantes.

Nombre de variables	Nombre maximum de couches	Coût minimal	Borne supérieure du coût
3	2	2	2
4	3	3	6
5	4	3	14
6	5	4	30
7	6	4	62
8	7	5	126
9	8	5	254
10	9	6	510
11	10	6	1022
12	11	7	2046
13	12	7	4094
14	13	8	8190
15	14	8	16382
16	15	9	32766
17	16	9	65534
18	17	10	131070
19	18	10	262142
20	19	11	524286

V - 3 - c - Fonctions impaires croissantes.

Nombre de variables	Nombre maximum de couches	Coût minimum	Borne supérieure du coût
3	1	1	1
4	2	2	2
5	3	2	4
6	4	3	7
7	5	3	15
8	6	4	34
9	7	4	70
10	8	5	143
11	9	5	290
12	10	6	585
13	11	6	1176
14	12	7	2259
15	13	7	4626
16	14	8	9361
17	15	8	18832
18	16	9	37775
19	17	9	75662
20	18	10	151437

B I B L I O G R A P H I E

[1] MILLER

Majority logic analysis.
Herres electr. Comp. Pub. MP 95
Août 1960 -

[2] COHN M. et LINDAMAN R.

Axiomatic Majority decision logic.
Proc. IRE E.C. - Tome 10 - pp 17-21.
1961 -

[3] ACKERS S.B. et ROBBINS T.C.

Synthesis of combinational logic using 3 input majority gates. Computing review vol. 5 n° 5- Sept. Oct. 1964 -

[4] MYATA F.

Realization of arbitrary logical function using majority elements.
IEEE EC, 12 n° 3 - pp 183-191 - 1963 -

[5] AMAREL COOKE WINDER

Majority gate Networks.
IEEE EC, USA, 13 n° 1 - pp 4-13 - 1964 -

[6] NEGRIN

Synthesis of pratical 3 input majority logic networks.
IEEE, EC, 13 n° 3 - pp 296-299 - 1964 -

[7] MUROGA S.

Functional forms of dual comparable functions and a necessary and suffisant condition for a realization of a majority function.
IEE Trans. Comm. and Electr. 83 n° 74
pp 474-486 - Sept. 1964 -

[8] ACKERS S.B.

Synthesis of combinational logic using 3 input majority gates.
Computing review vol. 5 n° 5 -
Sept. Oct. 1964 -

[9] MYATA F
An extension of method of Cohn and Lindaman -
IEEE Trans EC 13 - pp 625-629
Oct. 1964 -

[10] TOHMA Y.
Decomposition of logical function by majority decision elements.
IEEE Trans EC 13 n° 6 - pp 698-705
Dec. 1964 -

[11] KUNTZMANN J. LUSTMAN F.
Méthodes de synthèse -
Colloque Algèbre de Boole
Grenoble - Janvier 1965 -

[12] KUNTZMANN J.
Algèbre de Boole
Dunod - Paris Avril 1965 -

[13] KUNTZMANN J.
Méthodes générales de réalisation des fonctions Booléennes par des organes technologiques donnés.
Automatisme - Tome X - pp 58-60
Février 1965 -

[14] PICHAT E.
Décomposition des fonctions Booléennes - Thèse 3ème cycle
Janvier 1966 - GRENOBLE

TABLE des MATIERES

Pages

-:-:-:-:-:-:-

VU

Grenoble, le

Le Président de la thèse

VU

Grenoble, le

Le Doyen de la Faculté des Sciences

VU, et permis d'imprimer,

Le Recteur de l'Académie de Grenoble