Outils d'aide à la décision dans le test de systèmes logiques

Ali Mili

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR DE 3ème CYCLE

Génie Informatique

par

Ali MILI

OUTILS D'AIDE A LA DECISION DANS LE TEST DE SYSTEMES LOGIQUES

Thèse soutenue le 24 juin 1978 devant la Commission d'Examen :

Président : L. BOLLIET

Examinateurs : M. SAKAROVITCH
S.B. AKERS
P. CASPI

Rapporteur : G. SAUCIER

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Monsieur Philippe TRAYNARD : Président

Monsieur Pierre-Jean LAURENT : Vice Président

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSON Jean	Electrochimie
	BLOCH Daniel	Physique du solide
	BONNETAIN Lucien	Chimie minérale
	BONNIER Etienne	Electrochimie et électrométallurgie
	BOUDOURIS Georges	Radioélectricité
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	COUMES André	Radioélectricité
	DURAND Francis	Métallurgie
	FELICI Noël	Electrostatique
	FOULARD Claude	Automatique
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie-Physique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	POLOUJADOFF Michel	Electrotechnique
	SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M.	ROUXEL Roland	Automatique

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BOUVARD Maurice	Génie mécanique
	COHEN Joseph	Electrotechnique
	LACOUME Jean-Louis	Géophysique
	LANCIA Roland	Electronique
	ROBERT François	Analyse Numérique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Mathématiques appliquées
	CHARTIER Germain	Electronique
	GUYOT Pierre	Chimie minérale
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du solide
	MORET Roger	Electrotechnique nucléaire
	PIERRARD Jean-Marie	Mécanique
	SABONNADIERE Jean-Claude	Informatique fondamentale et appliquée
Mme.	SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M.	LANDAU Ioan	Automatique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

MM.	FRUCHART Robert	Directeur de Recherche
	ANSARA Ibrahim	Maître de Recherche
	CARRE René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

AVANT PROPOS

Je voudrais remercier Monsieur L. BOLLIET, professeur à l'Institut Universitaire de Technologie, pour l'honneur qu'il me fait de présider le Jury.

Je voudrais exprimer ma profonde gratitude à Madame G. SAUCIER, Maître de Conférence à l'Institut National Polytechnique de Grenoble, pour m'avoir accepté dans son équipe et m'avoir initié à la recherche scientifique.

Je remercie très vivement, pour avoir accepté de faire partie du Jury,

Mr. M.SAKAROVITCH, directeur du LA7 et Maître de Conférence à l'USMG.

Mr. S.B. AKERS, chercheur à la General Electric Sompany

Mr. CASPI, attaché de Recherche au CNRS.

Je remercie également mes camarades de l'équipe "Conception et sécurité de systèmes logiques", en particulier, P.CASPI, A.VERDILLON, Ch. ROBACH, pour l'intérêt qu'ils ont porté à mon étude et les conseils qu'ils m'ont donnés.

Je tiens à remercier Madame P.SOUILLARD et l'équipe de Monsieur D. IGLESIAS pour la rapidité et le soin apportés à la réalisation matérielle de ce document.

MILI Ali

INTRODUCTION

Il existe à l'heure actuelle des outils de test perfectionnés et performants tant pour le test déterministe que pour le test aléatoire des circuits logiques élémentaires.

Ce qui reste peu exploré, par contre, ce sont les possibilités de choix premiers qui s'imposent en face d'un grand système logique à tester : c'est au choix de politiques de test que nous nous intéressons ici.

Nous considèrerons quatre points :

1. Les chapitres 1 et 2 s'intéressent à la possibilité de partitionner un réseau R à tester en plusieurs sous réseaux R_i plus petits. Cette démarche trouve sa justification dans le fait qu'il est moins coûteux et surtout plus efficace pour la localisation de tester les R_i séparément que R en bloc. En outre, on y propose des méthodes qualitatives d'élaboration de politiques de test et on montre leur insuffisance. On justifie ainsi l'intérêt des paramètres de test introduits dans la suite.

2. Dans les chapitres 3 et 4, on introduit des paramètres de test, i.e. des quantités numériques qui mesurent, pour chaque partie du système, la qualité du test qu'on peut lui faire subir ainsi que son coût.

3. Dans le chapitre 5 on utilise les paramètres de test afin de modéliser les problèmes de test sous forme de problèmes mathématiques de prise de décision.

4. Enfin en annexe on trouvera une description détaillée de l'algorithme séparant un réseau R en sous réseaux R_i, ainsi que les résultats que cet algorithme a donné sur un réseau R de grande taille.

CHAPITRE 1

RESEAUX DE PETRI

Les réseaux de Pétri sont utilisés en Informatique et automatisme pour la description de processus. L'usage qu'on en fera ici est tout autre : on les utilisera pour représenter la structure matérielle de systèmes logiques et les chemins de données qui existent sur ces systèmes.

A. Eléments de la Théorie des Graphes

Ce paragraphe a pour but d'exposer la terminologie en vigueur en Théorie des graphes, ainsi que les principales définitions, afin de faciliter l'étude ultérieure sur les réseaux de Pétri. Pour plus d'information on pourra se référer à [18].

+ Un graphe orienté est la donnée d'un ensemble de sommets X et d'un ensemble d'arcs orientés u inclus dans $(X \times X)$. Ainsi un arc de U peut être noté par deux sommets.

+ Pour tout arc u, on appellera sommet initial de u (noté $\dot{I}(u)$) et sommet terminal de u (noté $\dot{T}(u)$) respectivement les sommets dont u part et vers lequel il arrive. Si u se note (a,b), alors $J(u) = a$; $T(u) = b$; On dit que u est adjacent à $J(u)$ et à $T(u)$, que $J(u)$ est adjacent à $T(u)$ et que u est adjacent à tout arc v vérifiant $J(u) = T(v)$ ou $J(v) = T(u)$.

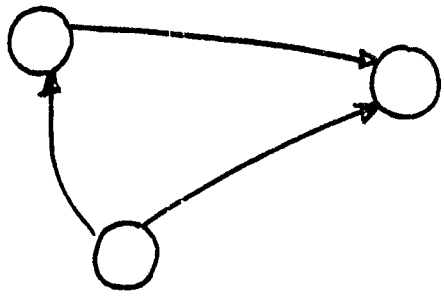
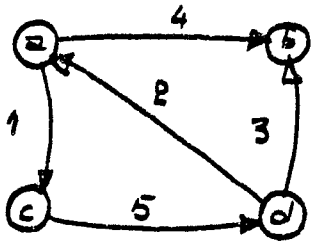


figure 1.



$G = (X, U)$

$X = a, b, c, d$ $\quad X' = a, b, d$

$U = 1, 2, 3, 4, 5$ $\quad U' = 2, 4$

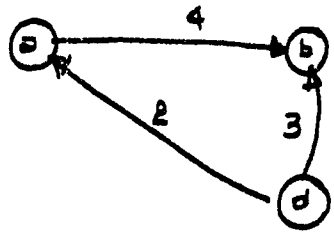


figure 2

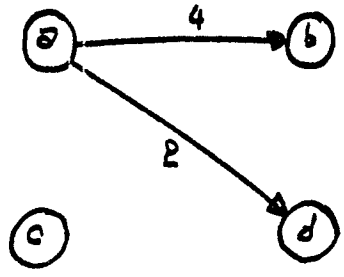


figure 3.

+ Un arc est dit simple si deux sommets ne sont jamais reliés par plus d'un arc ; sans boucle si aucun arc u ne vérifie J(u) = T(u).

+ Un chemin de x_0 à x_k est une séquence alternée de sommets et d'arcs $(x_0\ u_1\ x_1 \ldots u_{k1}\ x_k)$ tq $\forall$ i $J(u_i) = x_{i-1}$ et $T(u_i) = x_i$.

+ Un circuit est un chemin de x à x.

+ Une chaîne est une suite alternée de sommets et d'arcs $(x_0\ u_1\ x_1\ \ u_k\ x_k)$ tel que chaque arc u_i est adjacent à x_{i-1} et x_i, ainsi tout chemin est une chaîne, mais pas nécessairement l'inverse.

De façon analogue, un cycle est l'extension d'un circuit dans le cas où on ne s'intéresse pas à l'orientation des arcs. Tout circuit est un cycle : le cycle représenté à la figure 1 n'est pas un circuit.

On définit sur X deux relations binaires C (connexité forte) et c (connexité faible) par :

x C x' <==> il existe un chemin de x à x' ou un chemin de x' à x ou x = x'

x c x' <==> il existe une chaîne joignant x' et x ou x = x'

Une composante connexe forte (faible est une classe d'équivalence dans X modulo C (c).

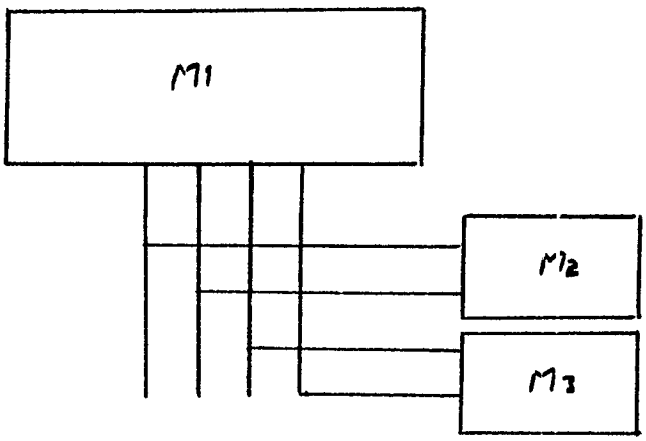
Soit G = (X,U) un graphe et soient X' $\subset$ X et U' $\subset$ U. On pose :

$U_{X'} = \{(a,b) \in U \text{ tq } a \in X', b \in X'\}$

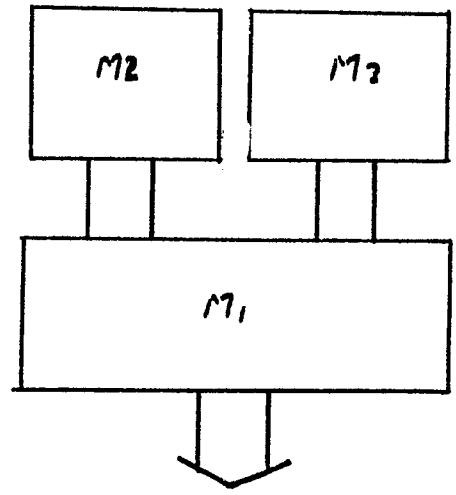
$U'_{X'} = \{(a,b) \in U' \text{ tq } a \in X', b \in X'\}$

On appelle :

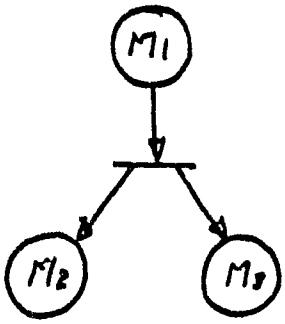
$G_{X'} = (X', U_{X'})$	sous graphe de G construit sur X'	- voir figure 2 -
$G' = (X , U')$	graphe partiel de G défini par U'	- voir figure 3 -
$G'_{X'} = (X', U'_{X'})$	sous graphe partiel de G	- voir figure 3 -



a

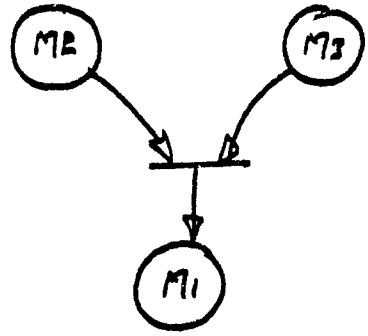


a'



α

distribution



α'

Jonction

noeuds ET.

figure 4

Soit A un ensemble inclus dans X ; On note $\Omega^+(A)$ l'ensemble des arcs u vérifiant $J(u) \in A$, $T(u) \notin A$ ($\Omega^+(A)$ est l'ensemble des arcs sortant de A) et $\Omega^-(A)$ l'ensemble des arcs vérifiant $J(u) \notin A$ et $T(u) \in A$. La paire $(\Omega^+(A), \Omega^-(A))$ est dite cocycle de A dans G. On définit $\Omega(A) = \Omega^-(A) \cup \Omega^+(A)$.

On définit deux fonctions de X dans $\mathbb{N}$, d_G^+ et d_G^-, dites demi degré sortant et demi degré entrant du sommet x :

$d_G^+(x) = |\Omega^+(\{x\})|$ $\qquad$ $d_G^-(x) = |\Omega^-(\{x\})|$.

le degré de x se note $d_G(x)$ et vaut $d_G^+(x) + d_G^-(x)$.

Une source dans G est un sommet S vérifiant $d_G^-(x) = 0$.

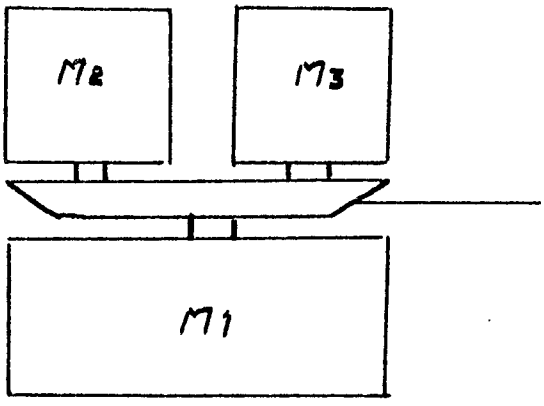
Un puits dans G est un sommet P vérifiant $d_G^+(x) = 0$.

On appelle flot sur un graphe (X,U) une fonction $f : U \to \mathbb{R}$ telle que, $\forall x$, $\sum_{u \in \Omega^+(\{x\})} f(u) - \sum_{u \in \Omega^-(\{x\})} f(u) = 0$

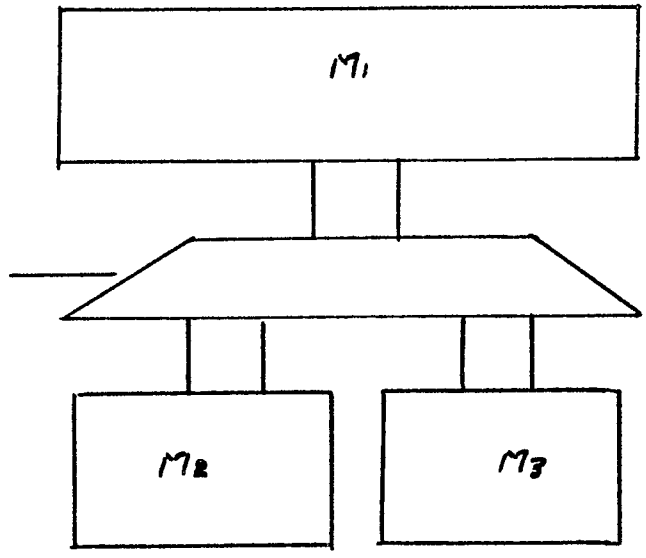
B. Représentation des systèmes logiques par des Réseaux de Pétri.

On propose dans la suite un moyen de représenter un système logique par un modèle fonctionnel. Cette représentation a été introduite par Rohach dans [8] et reprise dans [1] [2] [3] [4] [5] et [7] :
On représente le système par un graphe biparti où les deux ensembles de sommets sont appelés places et transitions.

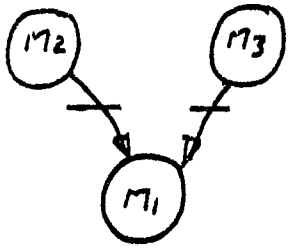
+ A chaque module du système on associe une place du réseau de Pétri : de même on associe une place à toute entrée et toute sortie du système. Les places associées aux entrées sont dites sources du système et celles associées aux sorties sont dites puits du système.



b'

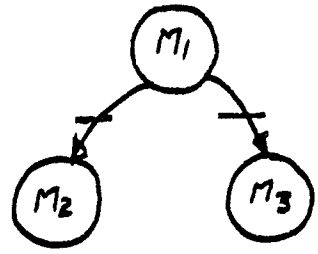


b



β'

Attribution

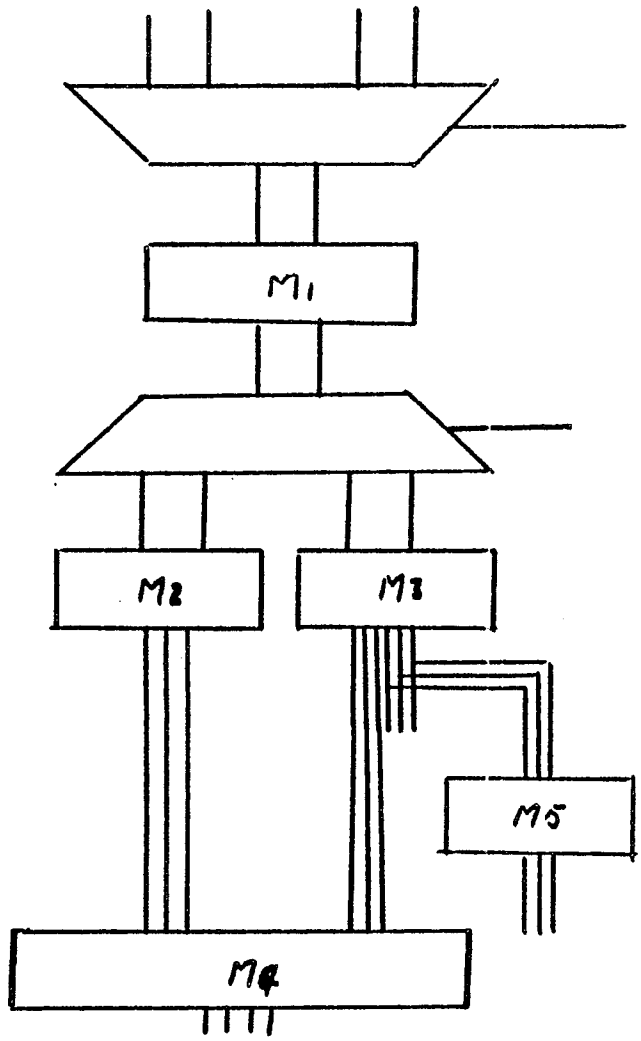


β.

Sélection.

noeuds ou.

figure 5.



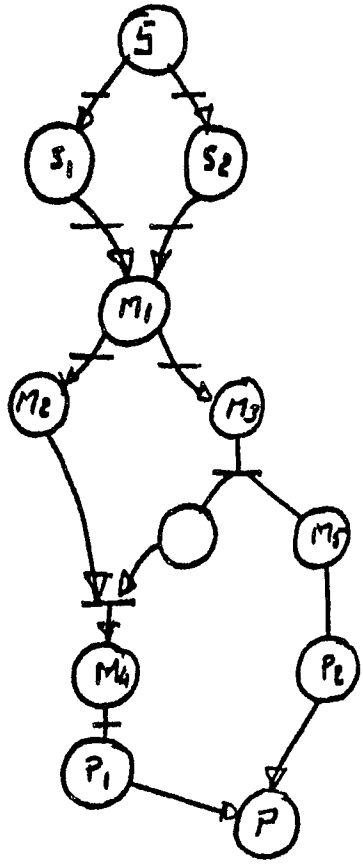


figure 6.

+ A chaque connexion physique on associe un arc orienté dans le sens du transfert d'information.

Tout arc va d'une transition à une place ou d'une place à une transition. On dira que l'extrémité initiale d'un arc u se trouve en amont de cet arc et que l'extrémité terminale se trouve en aval de u. De même on dit que J(u) est en amont de T(u) et que T(u) est en aval de J(u).

+ Les transitions expriment les conditions du contrôle autorisant le transfert de l'information depuis les modules se trouvant en amont de cette transition jusqu'à ceux se trouvant en aval.

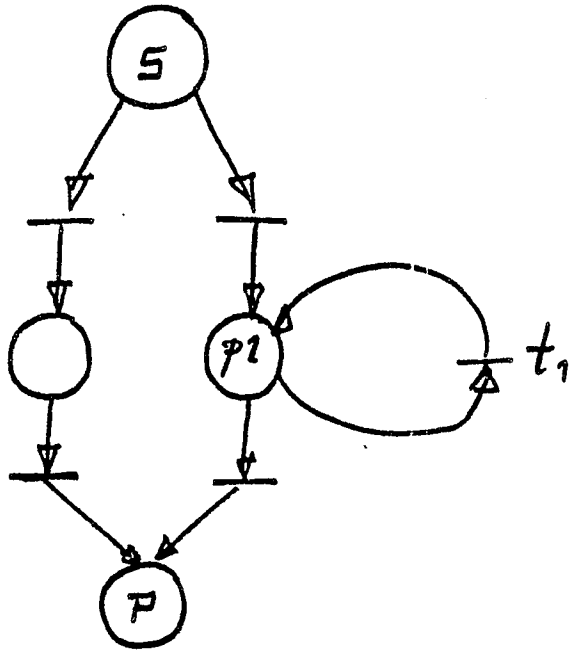
Les quatre configurations de base de cette représentation sont :

- La Jonction - voir figure 4 - Le module M_1 traite l'information qu'il reçoit de M_2 et l'information qu'il reçoit de M_3.

- La Distribution - voir figure 4b - Le module M_1 envoie simultanément une information partielle dans M_2 et une information partielle dans M_3.

- L'Attribution - Considérons un multiplexeur qui reçoit, par exemple, les sorties de deux modules M_2 et M_3 pour en sélectionner une et l'envoyer dans le module M_1. Le module M_1 peut traiter de façon exclusive l'information venant de M_2 ou l'information venant de M_3. Pour exprimer cette idée de choix, on représente ce montage par un noeud attribution comme il est indiqué à la figure 5.

- La Sélection - Une démultiplexeur reçoit la sortie d'un module M_1 et l'aiguille, en fonction de la valeur de S générée par le contrôle vers l'entrée de M_2 ou l'entrée de M_3. M_2 ou M_3 peuvent traiter toute l'information fournie par M_1. On ne représente pas le démultiplexeur sur le réseau de Pétri mais on exprimera le choix par un noeud Sélection tel que c'est indiqué à la figure 5.



le sous graphe construit sur $[p_1, t_1]$ vérifie les conditions de DEF2 mais ne vérifie pas celles de DEF1: Il y a un circuit dans le reseau considéré

figure 7.

Ces noeuds jonction et distribution sont appelés noeuds ET alors que les noeuds attribution et sélection sont appelés noeuds OU.

On adjoint une place S dite source principale dont partent des arcs vers les sources ordinaires et une place P dite puits principal vers laquelle arrivent des arcs partant de tous les puits ordinaires du réseau. Sur chacun de ces arcs on porte une transition afin que le réseau reste biparti.

La figure 6 représente un exemple simple de système logique et son réseau de Pétri complet.

On limitera notre étude à des réseaux de Pétri sans circuits et on admet sans les démontrer les quatres propositions suivantes :

P1 : Toute place admet des transitions en amont (sauf S)

P2 : Toute place autre que P admet une transition en aval.

P3 : Toute transition admet des places en amont et des places en aval.

P4 : Soit x un sommet du réseau (x pouvant être une place ou une transition), il existe un chemin de S à x et un chemin de x à P.

C. Ecoulements

DEF 1 : Un écoulement est un sous graphe construit sur un ensemble de sommets X' inclus dans X et vérifiant :

1. Si une transition appartient à X', alors toute place qui lui est adjacente appartient à X'.

2. Quelle que soit la place x appartenant à X', il existe dans X' un chemin de S à x et un chemin de x à P.

Nous allons présenter une seconde définition des écoulements dont nous montrerons qu'elle est équivalente à DEF1 dans les réseaux sans circuits.

DEF 2 : Un écoulement E est un sous graphe construit sur un ensemble de sommets X' non vide, inclus dans X et vérifiant :

i - Si une place appartient à X', alors elle admet au moins une transition en amont et une transition en aval dans X'.

ii - Si une transition appartient à X', alors toute place qui lui est adjacente appartient à X'.

PROP 1 : Les définitions DEF1 et DEF2 sont équivalentes sur les réseaux sans circuits.

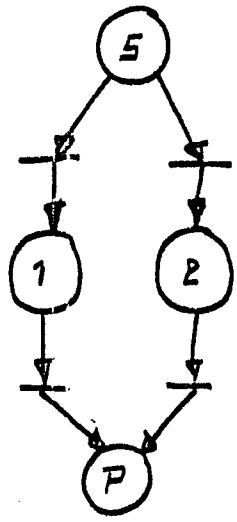
DEM 1 : Soit un graphe E vérifiant les conditions 1 et 2. Il vérifie 1 => il vérifie ii. Il vérifie 2 => Toute place x de E admet au moins une transition amont (l'avant dernier sommet du chemin de S à x) et au moins une transition aval (le deuxième sommet du chemin de x à P) => il vérifie i.

Considérons un sous graphe E dans un réseau R sans circuit. Supposons que E vérifie i et ii. E vérifie ii => il vérifie 1.

Soit x une place de E. On marque x et on propage le marquage d'un sommet à l'autre dans le sens d'orientation des arcs (dès que J(u) est marqué, on marque T(u)).

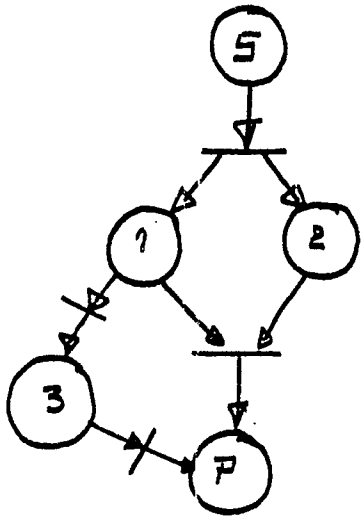
Les conditions i et ii assurent que cette propagation est possible. L'hypothèse d'absence de circuit permet d'assurer que l'on ne marque pas le même sommet plus d'une fois => le marquage s'arrête en P. Il existe un chemin de x à P. En propagant le marquage depuis x dans le sens opposé à l'orientation des arcs, on met en évidence un chemin de S à x . c.q.f.d. voir figure 7.

Vu que l'on s'intéresse, dans le cadre de cette thèse , à des réseaux de Pétri sans circuits, on adoptera la définition DEF 2.



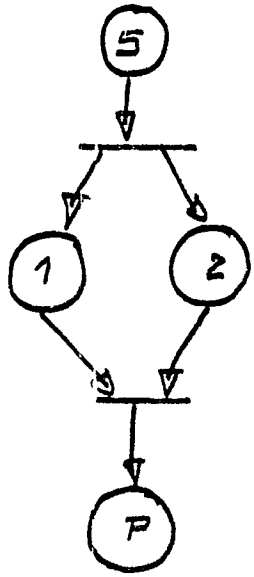
(S,1,P) et (S,2,P) sont 2 écoulements distincts de (S,1,2,P). donc (S,1,2,P) n'est pas élémentaire.

figure 8.



(S,1,2,P) est minimal
(S,1,2,3,P) est elementaire
(S,2,P) est le seul minimal ⇒ l'union des minimaux n'est pas égale au réseau.

figure 9.



Cet écoulement est minimal.

figure 10.

Propriétés des écoulements

Prop 2 : Etant donnée une place x dans un réseau de Pétri R. Il existe un écoulement auquel x appartient.

Dem 2 : R vérifie les conditions i et ii de def2 ==> R est un écoulement cqfd. Toutefois on s'intéressera à des écoulements plus fins pour avoir une place donnée x.

Prop 3 : L'union d'écoulements est un écoulement.

Dem 3 : Résulte de DEF2

Prop 4 : Un écoulement est faiblement connexe.

Dem 4 : Soient x et y deux sommets d'un écoulement. D'après DEF1, il existe un chemin de S à x et un chemin de S à y, donc une chaîne de x à y.

DEF3 : Un écoulement est dit minimal s'il ne contient pas d'écoulements différents de lui-même.

DEF4 : Un écoulement est dit élémentaire s'il n'est pas union de deux écoulements distincts de lui-même.

Remarque 1 : L'intérêt des écoulements minimaux est trivial : ce sont les plus petits. Toutefois, leur union n'est pas égale au réseau - voir figures 9 et 10 ; alors que, l'union des élémentaires d'un réseau est égale à ce réseau (ce sera montré au chapitre suivant) : d'où l'intérêt des écoulements élémentaires. Il est bien entendu qu'un écoulement minimal est élémentaire. On trouvera aux figures 8, 9 et 10 des exemples d'écoulements non élémentaires, élémentaires non minimal et minimal respectivement.

Interprétation des Ecoulements : On admet l'hypothèse suivante :

- Tout arc arrivant à une place permet d'acheminer toutes les données nécessaires au test de cette place.

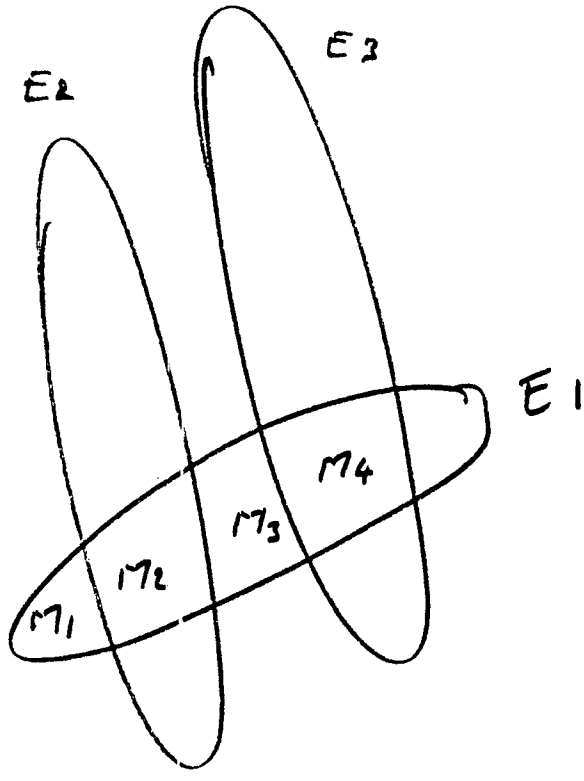


figura 11.

- Tout arc partant d'une place permet d'observer tous les résultats de test de cette place.

Alors on peut dire que le test d'une place dans un écoulement peut se faire en ne faisant intervenir que le matériel représenté par cet écoulement : de la sorte on peut définir une politique de test et de diagnostic en termes d'écoulements.

Exemple : voir figure 11. On se place sous l'hypothèse de panne simple.

Si E_1 et E_2 sont en panne, on accuse M_4

Si E_1 et E_3 sont en panne, on accuse M_2

Si E_1 est seul en panne, on accuse M_1 ou M_3 ; dans le cas de cette figure, aucun moyen ne permet de savoir si c'est M_1 ou M_3 qui est en panne sachant que E_1 est seul en panne. Ceci nous mène à poser la définition :

DEF 5 : Deux modules M et M' sont indiscernables si dès que l'un deux appartient à un écoulement, l'autre aussi.

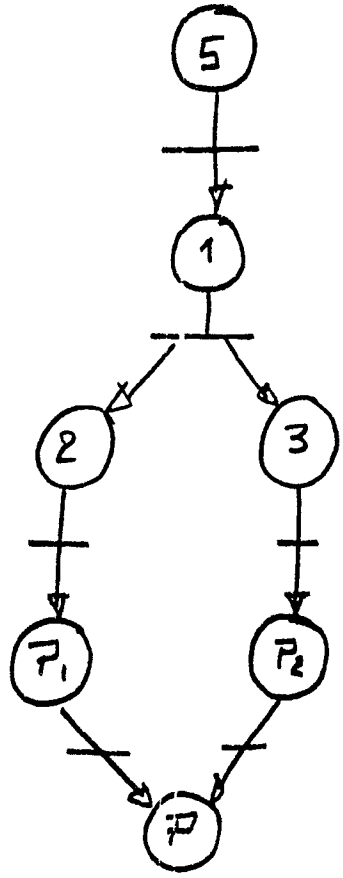
D. Sous écoulement

Considérons l'écoulement représenté à la figure 12. Supposons que l'on émette une information de test destinée à tester cet écoulement.

Si l'on observe les réponses de l'écoulement au puits P, deux cas sont possibles :

- Les réponses fournies en P sont correctes, alors 1 2 et 3 sont sains.
- Une au moins des réponses est incorrecte, alors 1,2 ou 3 est en panne.

Si l'on observe séparément les puits P1 et P2, on peut établir sous l'hypothèse de panne simple, un diagnostic bien plus fin, comme l'indique le tableau de la figure 9.



P_1 P_2	1	2	3
0 0	0	0	0
0 1	φ	0	$\bar{\varphi}$
1 0	φ	$\bar{\varphi}$	0
1 1	1	0	0

$P_i = 0$ ⇔ la valeur sortie en P_i est correcte.
mettre 0 dans la case i ⇔ le module i est nain
mettre φ dans la case i et $\bar{\varphi}$ dans la case j ⇔ i ou (exclusif) j est en panne.

figure 12.

Ceci prouve qu'il peut parfois être intéressant pour la finesse de la localisation de s'intéresser à des parties d'écoulements, les sous écoulements, que l'on définira par :

DEF3 : Un sous écoulement SE est un sous graphe construit sur un sous ensemble de sommets X' tel que :

- $\forall x \in X'$ il existe dans X' un chemin de S à P passant par x.

- $\forall$ transition $t \in X'$, toutes les places en amont de t sont dans X'.

Construction des sous écoulements : Pour des raisons purement pratiques que l'on verra dans le chapitre suivant, il se trouve que, pour mettre en évidence des sous écoulements, il suffira de remplacer dans le réseau de Pétri initial certains noeuds "distribution" par des noeuds "sélection". Toutefois il faudra toujours garder en mémoire que ces noeuds "sélection" résultent de l'éclatement d'un noeud "distribution".

E. Exemple d'application

On se propose de représenter par son réseau de Pétri un interface entre processeurs. Nous allons d'abord expliciter les fonctions que réalisent les divers modules de ce système, voir figure 12 :

+ le 7430 réalise le ET logique entre ses huit bits d'entrée

+ le 74 125 : Porte à 3 états. La sortie vaut, sur commande, soit l'entrée, soit le troisième état - haute indépendance -

+ le 74 154 est un décodeur 4 entrées / 16 sorties dans le cas qui nous occupe, 10 fils sur 16 sont utilisés.

+ le 193 est un compteur d'impulsions modulo 16. En 15.1.10.9 on entre la valeur initiale : le débordement est livré sur le bit 12.

+ Le 8212 est un latch à 3 états ayant une entrée (11)

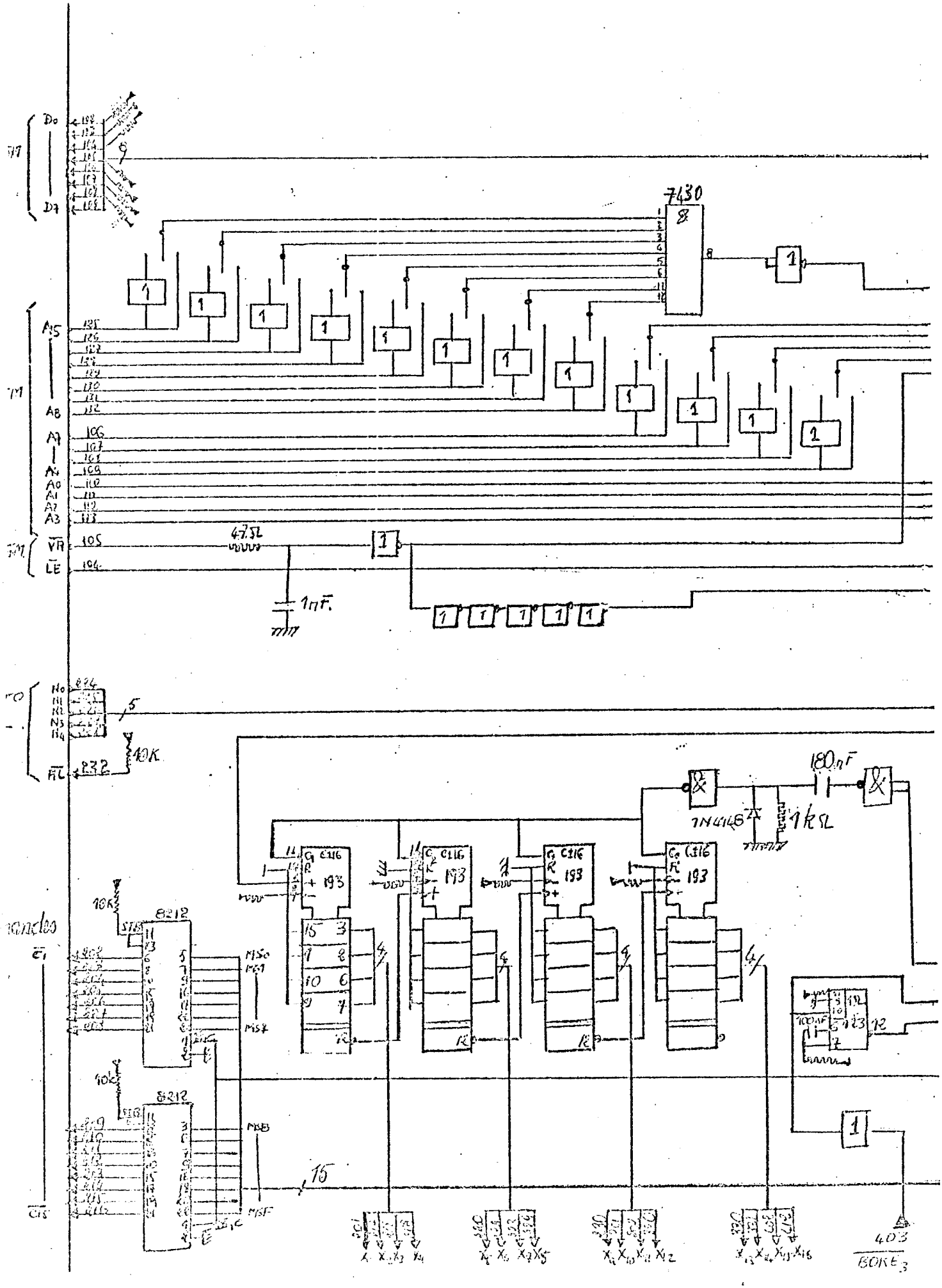
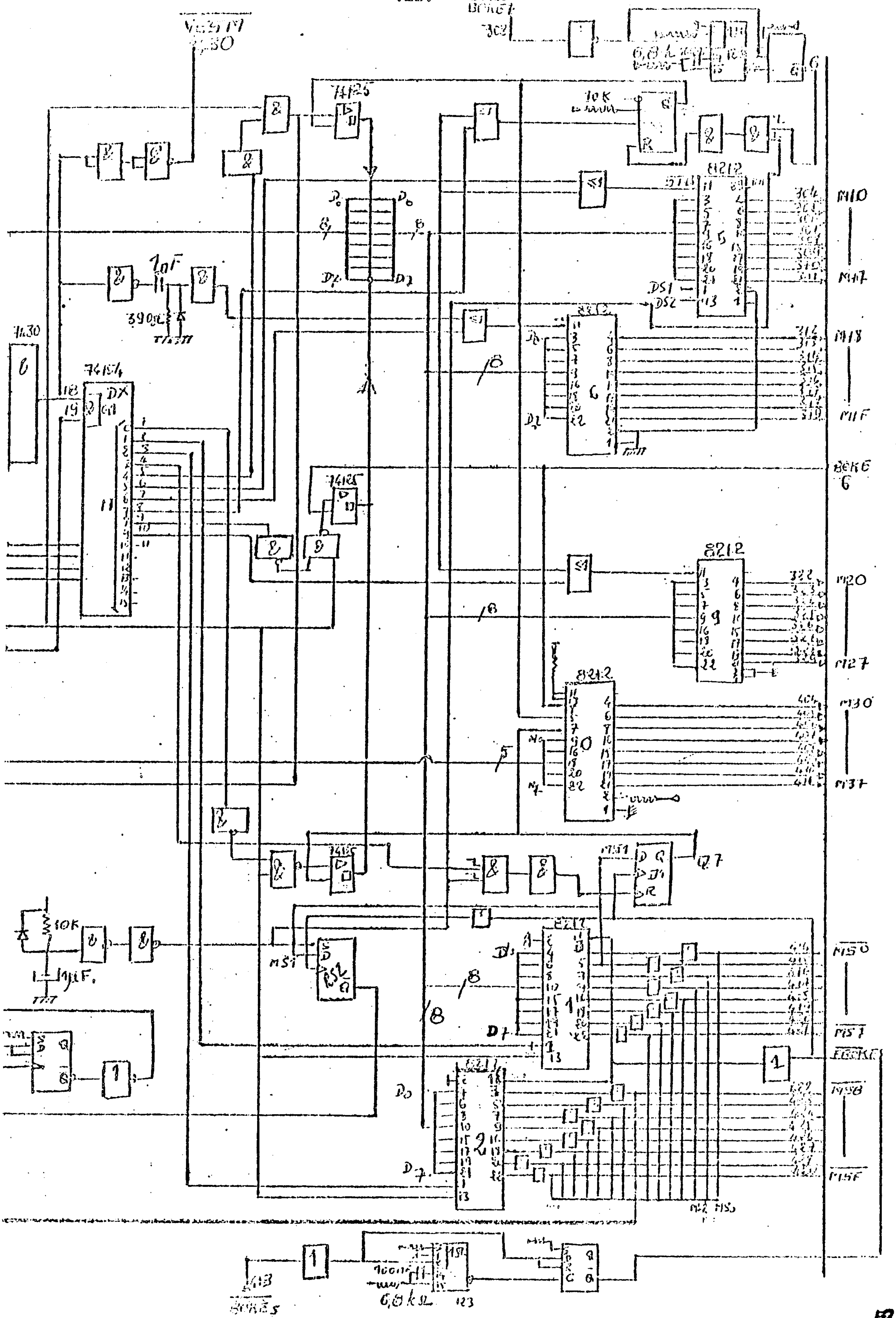

7430
8
1
1nF.
47Ω
10K
180nF
1kΩ
1N4148
8212
8212
193
MSO
MS1
MSB
MSF
403
BOKE3

figure 12


74125
8212
74154
7430
10K
M10
M17
M18
M1F
M20
M27
M30
M37
M50
M57
M58
M5F
6,8 kΩ
123

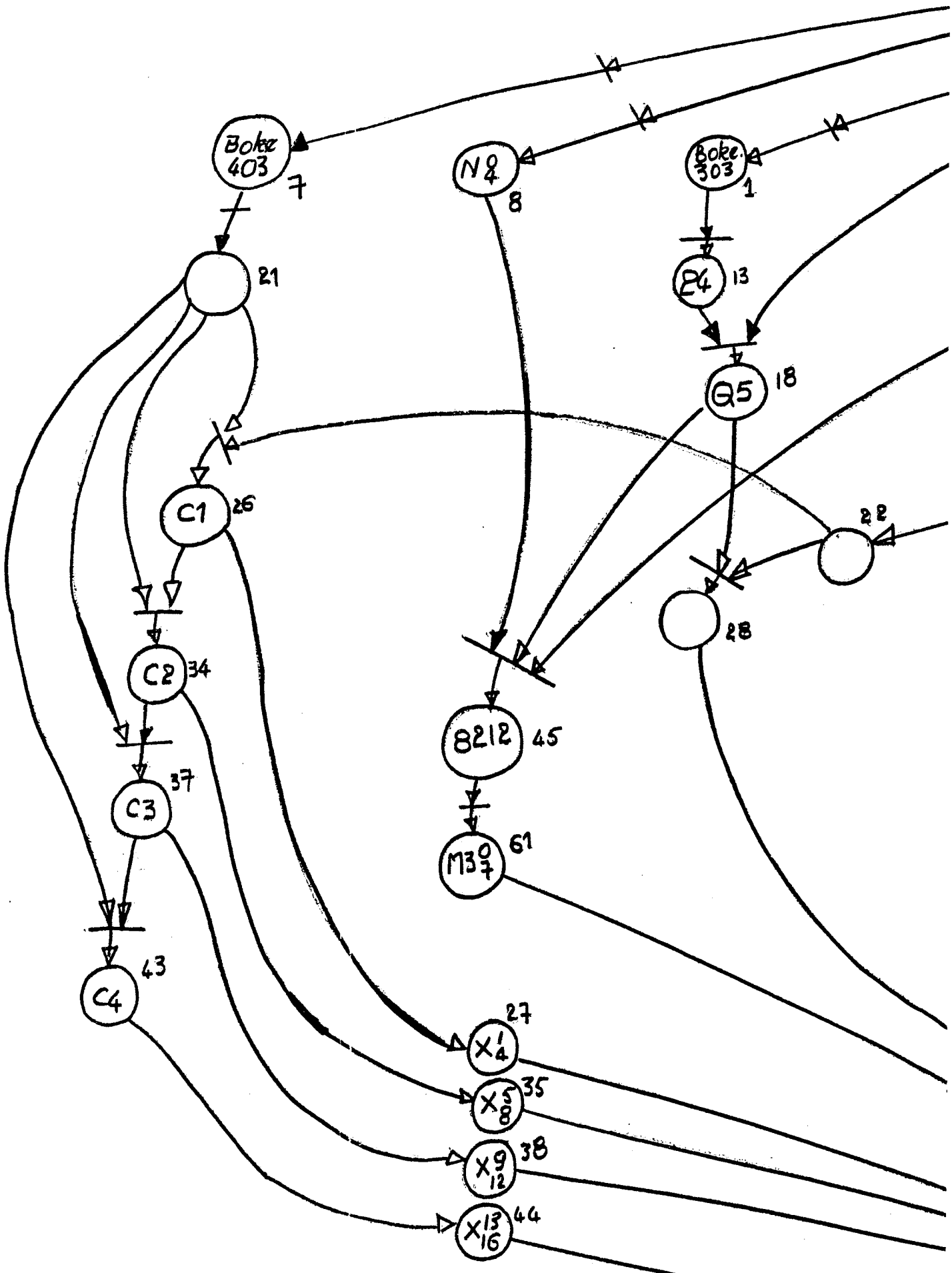
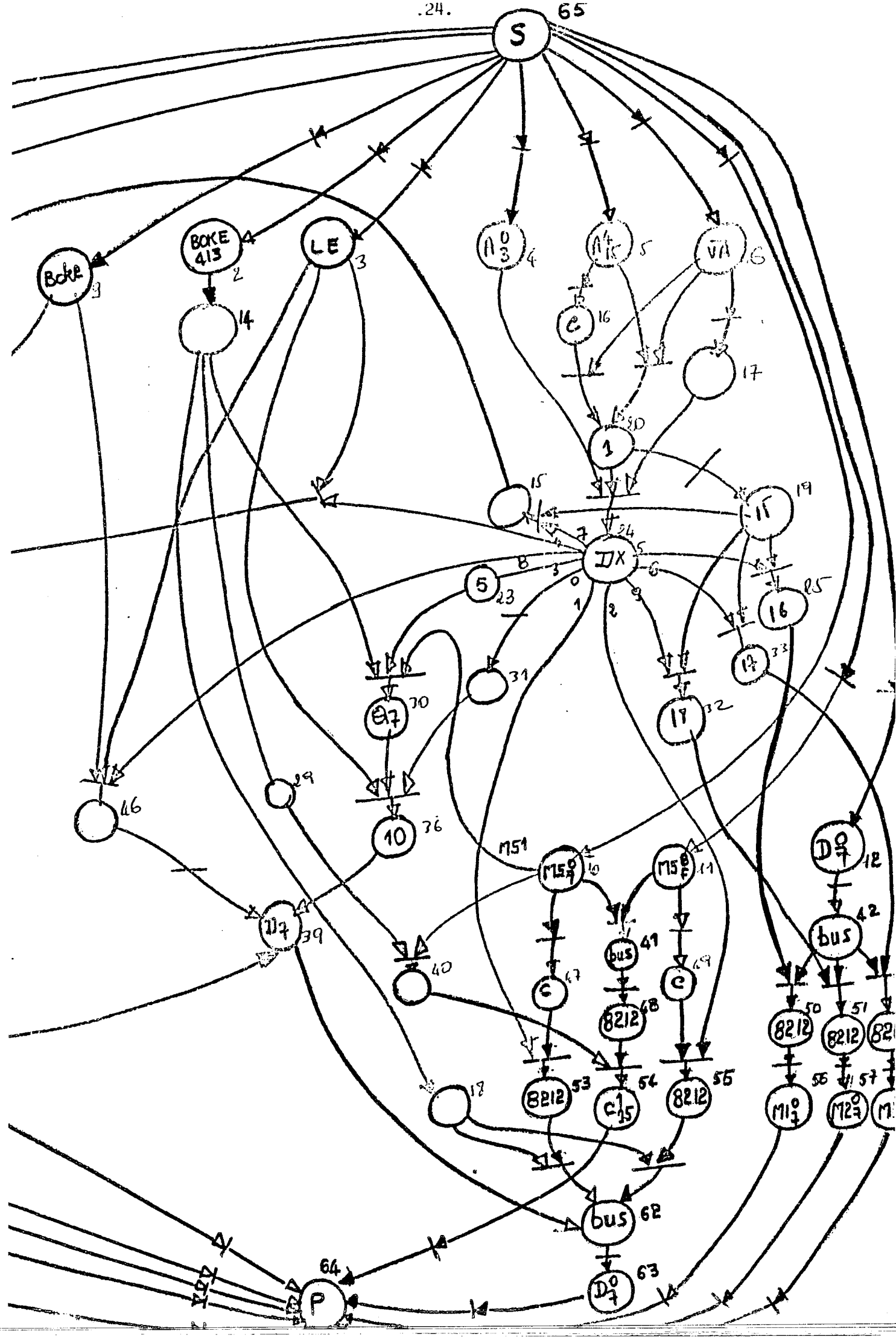

Boke 403
7
N 8 4
8
Boke 303
1
21
E4
13
Q5
18
C1
26
22
28
C2
34
8212
45
C3
37
M30 7
61
C4
43
X 1 4
27
X 5 8
35
X 9 12
38
X 13 16
44

figure 14


S
65
BOKE
BOKE 413
LE
A 0 3
A 4 15
VA
C
1
IT
DX
5
16
17
M15 9 7
M15 8 F
D 0 7
bus
8212
B212
C 1 15
M11 0 7
M12 0 7
P
D 0 7

On trouvera ci-après certains commentaires destinés à faire voir le lien entre le système à représenter et son réseau de Pétri.

+ Les places numérotées 1 à 12 sont les sources ordinaires du réseau. La place numéro 65 est la source principale.

+ De même, les places 27, 35, 38, 44, 54, 56, 57, 60, 61, 63 sont des puits ordinaires alors que 64 est le puits principal.

+ On a opté pour l'aternative β au niveau du décodeur DX voir place 24, car, sinon toute chance de séparation serait compromise. On retiendra que ce noeud "sélection" est l'éclatement d'une "distribution".

+ Une place peut représenter plus d'un module physique, dans le cas, par exemple, où certains modules sont indiscernables, ou bien dans le cas où leur séparation ne serait d'aucun intérêt ; ainsi la place n°46 du réseau représente les deux portes Nand branchees en série à la sortie du fil sortant du décodeur (place 24) sous le numéro 8-9. De même, la place 2 représente l'entrée $\overline{\text{BOKE}}$ 413 et toute la circuiterie qui vient immédiatement après - cf. en bas de la deuxième page du schéma.

+ Quand certains composants du circuit sont prévus pour générer des constantes, on supposera qu'ils font partie des modules qui se trouvent en aval : quand un module est prévu pour exécuter la fonction f(XY) et que l'un des arguments est fixé à une constante, on supposera qu'il exécute en fait la fonction à un seul argument $\phi(Y) = f(X,Y)$. Si bien que, formellement, on peut ignorer la circuiterie qui génère la constante. Ceci est en particulier le cas de l'assemblage des deux portes Nand, de la résistance, du transistor et de la capacité qui se trouvent à gauche de la page 18, vers le bas. De même, sans entrer dans le détail du fonctionnement, on représente par une place - en l'occurence la place 81 - le circuit composé de deux Nands, une capacité, une résistance et un transistor qui attaque les compteurs représentés page 18. (En supposant que les modules qui sont en amont font partie de la place - source $\overline{\text{Boke}}$ 403 -).

+ En général, on a représenté la configuration a par α, b par β, a' par α' et b' par β'. Entre autres exceptions à cette règle, on cite le cas fort intéressant du décodeur dont les bits de sortie sont dispersés en noeud DU alors que c'est une configuration a. Les raisons de ce choix ont été évoquées plus haut. Ce choix ne va pas altérer la facilité d'observer la sortie du décodeur. Ainsi, dans un écoulement contenant, par exemple le décodeur et le fil numéro i, tout ce qu'on peut tester est de savoir si, en émettant le code binaire de i en entrée du décodeur on a bien 1 sur le fil i et si en émettant les 15 autres valeurs en entrée du décodeur on obtient 0 sur le fil i. Ceci n'est évidemment pas un test complet du décodeur. On propose quand même de mener ce test pour les dix sous écoulements contenant le décodeur et le fil i et de faire ensuite un test mettant en oeuvre les dix fils qui sortent du décodeur, mais en sachant alors que seul le décodeur est en cause car toute la circuiterie autour a déjà été testée.(Il y a au maximum dix sous systèmes contenant le décodeur et le fil i $0 \leq i \leq 9$, car il peut arriver que deux fils i' et i" soient indiscernables); dans le cas qui nous occupe, le décodeur n'est complètement testable dans aucun des sous systèmes évoqués.

. Les numéros ou les lettres mises dans les places permettent de mieux faire le lien avec le schéma logique. Quant aux numéros placés à côté des places, ce sont les numéros de ces places dans le réseau.

F. Ebauche d'une politique de test

DEF1 : On dit qu'un ensemble ε d'écoulements couvre le réseau R ou qu'il réalise une couverture du réseau si quelque soit la place P qu'on considère dans R, il existe un élément de ε qui la couvre.

Le problème que l'on se pose dans le présent paragraphe est de trouver, à partir des écoulements trouvés par AT, voir chapitre suivant, une couverture satisfaisante dans un réseau qui reste à définir.

DEF2 : La couverture U_1 d'un réseau est dite une base si tout écoulement de ce réseau est union d'écoulements de U_1.

PROP1 : L'ensemble des élémentaires d'un réseau R est une base de ce réseau.

DEM1 : Soit un réseau R et soient U l'ensemble de tous ses écoulements, U_1 l'ensemble de ses écoulements élémentaires et U_2 l'ensemble de ses écoulements qui sont union de deux éléments distincts de U_1. Alors $U = U_1 \oplus U_2$. En effet $U_1 \cap U_2 = \emptyset$ par définition d'un écoulement élémentaire. Par ailleurs U_2 = le complémentaire dans U de U_2, par définition de U_1 et $U_2 \Rightarrow U = U_1 \cup U_2$, donc tout élément de U est union d'éléments de U_1 : U_1 est une base.

PROP2 : Soit une place P n'appartenant qu'à un écoulement E de U_1 et soit E' un élément de U couvrant P_1 alors $\forall$ E', P n'est pas plus facile à tester dans E' que dans E.

DEM2 : Soit U_1 est une base, donc E' est union d'écoulements de U_1 $E' = \bigcup_{i=1}^{k} E^i$. Parmi les E^i figure certainement E car il est le seul élément de U_1 à couvrir P et que $P \in E'$. On écrit alors $E' = E \cup \bigcup_{i=1}^{k-1} E^i = E \cup E''$. Tout le matériel nécessaire au test de P se trouve - par définition d'un écoulement - inclus dans $E \Rightarrow$ P est aussi testable dans E que dans E'.

Ainsi dès qu'une place appartient à un seul écoulement de U_1, il suffit de la tester dans cet écoulement même si on ne peut pas lui envoyer tous les vecteurs de test qu'on veut et discerner tous les résultats qu'on veut. D'ailleurs à quoi cela servirait-il de savoir si P réagit correctement ou non à un vecteur qui ne peut jamais lui être forcé en fonctionnement normal ?

Quand une place appartient à deux écoulements de E_1, elle peut être entièrement testable dans l'un ou l'autre. Vu que dans le cadre de ce chapitre, on n'a pas les moyens de savoir lequel des deux écoulements permet de la tester le plus facilement et si, éventuellement elle est entièrement testable dans le meilleur d'entre eux, on se contentera de la couvrir par l'un d'entre eux en mettant l'accent sur des critères de choix beaucoup plus empiriques. Il s'agit donc pour nous de chercher une couverture incluse dans U et "optimale" dans un certain sens. Comme on a vu que les écoulements non élémentaires ne sont d'aucun intérêt, on doit, théoriquement, chercher la couverture dans U_1. Par approximation (en supposant que AT trouve tous les élémentaires du réseau considéré), on la cherchera dans $\tilde{U}_1$, l'ensemble des écoulements trouvés par AT.

Nous allons proposer dans la suite deux exemples de recherche de couverture "optimale".

On définit la bijection β : $(\tilde{U}_1) \rightarrow [0,1]^m$, qui, à tout sous ensemble de $\tilde{U}_1$ associe le m-vecteur booléen N ayant "1" aux indices des écoulements du sous ensemble (en supposant que les éléments de $\tilde{U}_1$ sont numérotés de 1 à m) et "0" ailleurs, et soit M la matrice de couverture définie par $\tilde{U}_1$. Alors la condition de réalisabilité d'un vecteur N (i.e. la condition pour qu'il soit en bijection avec une couverture du réseau) s'écrit $MN \geq e$ où e est un vecteur colonne ayant autant de lignes qu'il n'y a de places et valant 1 à toutes ses composantes.

Exemple 1 : Soit un écoulement $E = (X', U_{X'})$. On appelle arcs d'isolation entrants les éléments de $\Omega^-(X')$. Le test de E se passe dans des conditions d'autant meilleures que le nombre de ces arcs est plus réduit. Ainsi, si on associe à chaque écoulement une quantité numérique c(E) qui est fonction croissante du nombre de ses arcs d'isolation, on peut caractériser la qualité d'une couverture du réseau par la somme de ces quantités numériques.

Si on appele C le m-vecteur ligne tel que $C_i = c(E_i)$;

$E_i \in \tilde{U}_1$, on peut évaluer la qualité d'une couverture N par C.N.

La recherche d'une couverture qui minimise le nombre des arcs d'isolation d'une couverture s'écrit :

$$\begin{cases} MN \geq e & N_i = 0 \text{ ou } 1. \\ CN \text{ à minimiser.} \end{cases} \qquad (SB) ;$$

dans [15], J.J. Dent classe les méthodes de test en trois grandes catégories : le Start Small, le Multiple Clue et Start Big. Les couvertures optimales trouvées par SB conviennent tout particulièrement à la méthode Start Big.

Exemple 2 : Supposons que, pour les besoins d'une localisation fine, on s'intéresse à couvrir le réseau avec des écoulements réduits.
On appelle norme d'une couverture γ et on note $\phi(\gamma)$ le maximum des nombres de places des écoulements de γ.
Le critère de comparaison que l'on définit sur l'ensemble des couvertures pour cet exemple s'écrit alors :

γ_1 est meilleur que $\gamma_2 \iff [\varphi(\gamma_1) < \varphi(\gamma_2)]$ ou $[\varphi(\gamma_1) = \varphi(\gamma_2)$ et $|\gamma_1| < |\gamma_2|]$

Soit encore, si on pose $w(\gamma) = m\varphi(\gamma) + |\gamma|$ où ($m = |\tilde{U}_1|$, donc, en particulier la valeur maximale que peut prendre $|\gamma|$).

γ_1 est meilleur que $\gamma_2 \iff w(\gamma_1) < w(\gamma_2)$.

Par abus de notation, on écrira w(N) pour désigner $w(\gamma)$ où γ est l'image inverse de N par β. Le programme d'optimisation discrète qui définit la couverture optimale pour le critère w s'écrit :

$$\begin{cases} MN \geq e & N_i = 0 \text{ ou } 1 \\ w(N) \text{ à minimiser} \end{cases} \qquad (W)$$

Vu qu'ils s'écrivent très simplement; les programmes SB et W sont justiciables des méthodes classiques d'optimisation discrète. Toutefois la taille de la matrice M est en général telle que l'énumération exhaustive de toutes les solutions réalisables est souvent plus raisonnable.

En outre, ils admettent tous deux une solution optimale car $N_0 = (1,1 \ldots , 1)$ est réalisable puisque $\tilde{U}_1$, couvre le réseau ; en outre, l'ensemble des solutions réalisables est fini.

Remarque 1 : Considérons la matrice de couverture donnée à la figure 8 ; la place i sera notée P_i et l'écoulement j sera noté E_j.

- P_5 n'est couvert que par E_3 => E_3 est bligatoire.
- P_6 n'est couvert que par E_2 => E_2 est obligatoire.
- P_8 n'est couvert que par E_5 => E_5 est obligatoire.
- P_9 n'est couvert que par E_6 => E_6 est obligatoire
- P_{12} n'est couvert que par E_1 => E_1 est obligatoire.
- P_{14} n'est couvert que par E_4 => E_4 est obligatoire.

On a vu que dans le cadre de ce chapitre on se contenterait de couvrir le réseau. Or $\gamma = (E_1\ E_2\ E_3\ E_4\ E_5\ E_6)$ couvre le réseau ; de plus elle est minimale pour cette propriété car tous ses éléments sont obligatoires , donc elle est optimale. Ce cas est évidemment fortuit ; en général il est conseillé d'enlever de la matrice M les colonnes correspondant aux écoulements obligatoires, les lignes correspondant aux places qu'ils couvrent et d'activer le programme d'optimisation sur la matrice ainsi trouvée.

Remarque 2 : Soit SB' le programme qui cherche une couverture optimale parmi les écoulements élémentaires du réseau, selon la fonction objective w. Vu qu'il n'est pas montré que AT trouve tous les élémentaires du réseau, on doit admettre la possibilité $U_1 \subset \tilde{U}_1$ strictement. Toutefois, vu que

la fonction w pondère beaucoup plus le terme $\phi(\gamma)$ que le terme $|\gamma|$, il est quasiment improbable que la solution de SB' ne soit pas composée d'éléments de $\tilde{U}_1$. Auquel cas SB donnerait la même solution que SB', mais en occupant moins d'espace mémoire et en consommant moins de temps (dans le cas d'une exploration exhaustive des solutions réalisables chaque écoulement supplémentaire dans SB' multiplie le temps calcul par 2).

P_3 est testable dans E_1 E_4 E_5 et E_6. Dans lequel est-elle plus facile à tester ? Supposons qu'on tente d'abord de la tester dans E_5 : son test est-il complet ? Par quel autre écoulement faut-il le compléter ? Le réseau est couvert par $\gamma = (E_1\ E_2\ E_3\ E_4\ E_5\ E_6)$. L'adjonction des écoulements E_7 ou E_8 améliorerait-elle la qualité du test ? Soient deux places P_i et $P_{i'}$ d'un même écoulement E_j. Certains vecteurs de test destinés à l'une peuvent informer sur l'autre si bien que pour tester la deuxième place, on n'activera qu'une partie des vecteurs de test qui lui sont destinés. Faut-il tester d'abord P_i ou d'abord $P_{i'}$, pour minimiser le total des vecteurs envoyés ?

C'est pour tenter de répondre à ces questions entre autres que nous allons, dans les chapitres 3, 4 et 5 , définir des paramètres de test et en proposer des applications.

Au chapitre 2, nous allons exposer et étudier un algorithme de recherche d'écoulements.

CHAPITRE 2

RECHERCHE D'ECOULEMENTS : ALGORITHMES AT

A. Introduction

On se propose de construire un algorithme de marquage destiné à trouver des écoulements. Dans le paragraphe F du chapitre 1, on a vu l'intérêt que présentent les écoulements élémentaires dans un réseau : On s'efforcera donc de trouver les écoulements les plus petits.

Il se trouve qu'une condition nécessaire pour qu'un sous graphe soit un écoulement est qu'il contienne un chemin de S à P ; donc, pour trouver des écoulements, on commencera par mettre en évidence des chemins de S à P en marquant leurs sommets. Ensuite, on marquera, si nécessaire, d'autres sommets de sorte que l'ensemble de sommets cochés vérifie les conditions i et ii de la DEF2 du chapitre 1. L'organigramme général de cet algorithme est présenté à la figure 1.

B. Présentation de AT

Nous allons détailler les divers segments de AT.

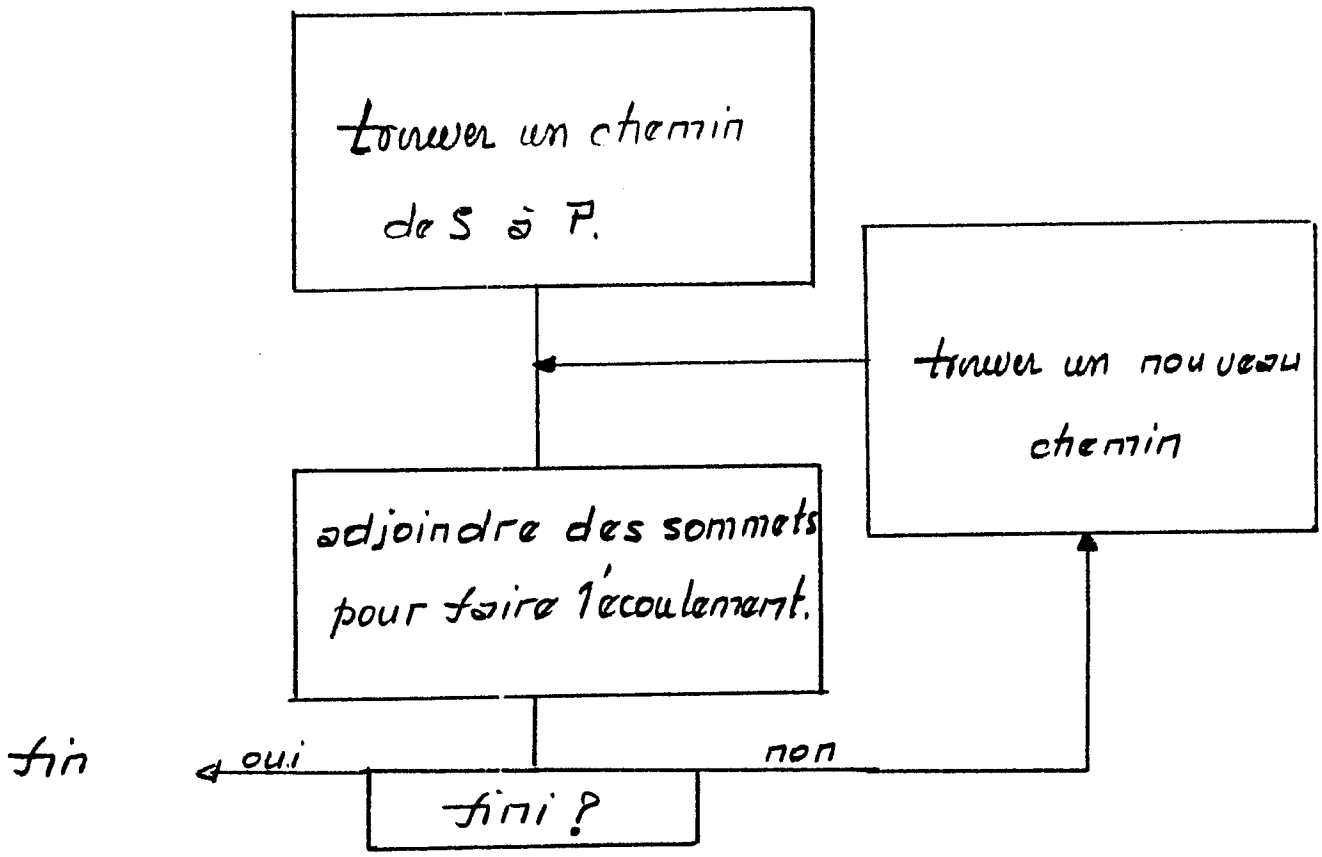


figure 1.

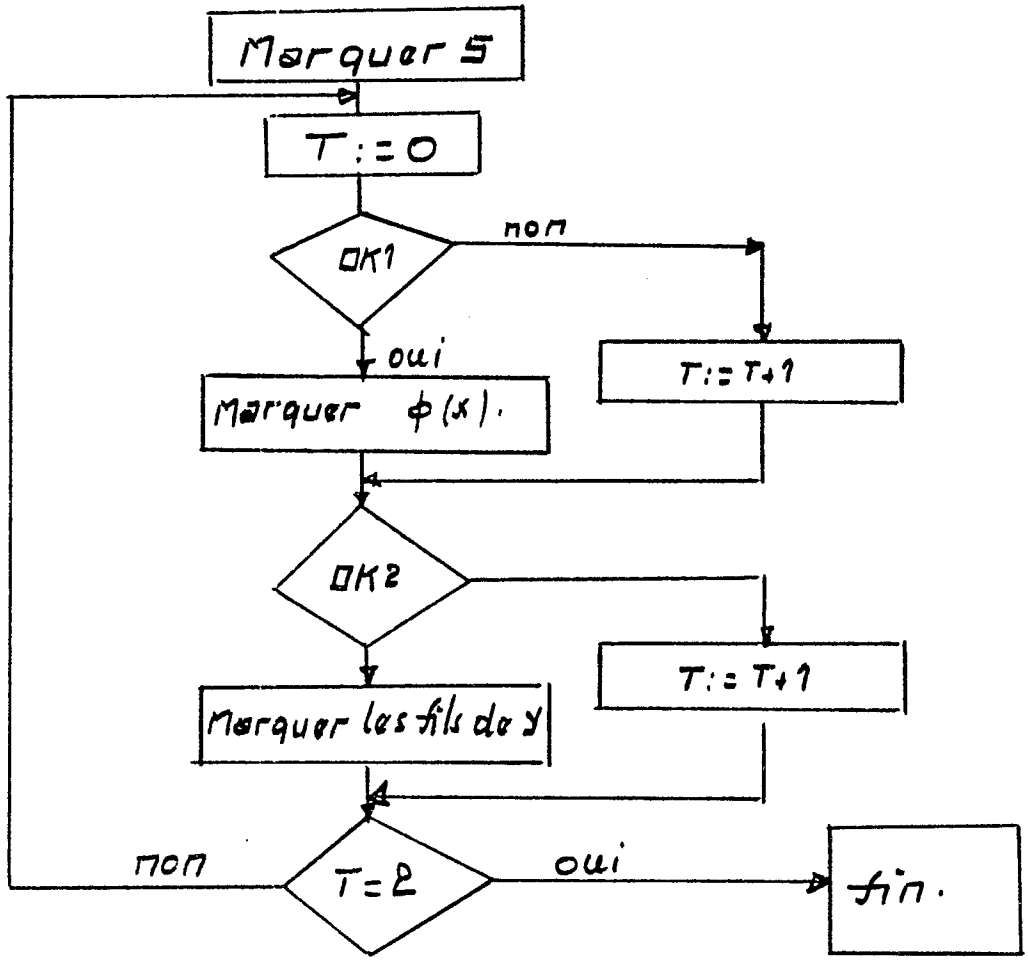


figure 2.

B1 . Trouver un chemin de S à P. On marque le sommet S puis on propage le marquage dans le sens des arcs selon les consignes suivantes :

- dès qu'une transition est marquée, on marque tous ses fils (car tout écoulement contenant une transition contient tous ses fils).

- dès qu'une place est marquée, on marque un de ses fils (un seul car on s'efforce de trouver des écoulements petits). Ceci suppose que l'on a défini un fils privilégié pour chaque place - quand celle-ci en a plus d'un. On appellera ϕ l'application qui à toute place associe sa transition privilégiée $\phi(X)$.

Considérons l'algorithme de marquage $\Delta^+(\phi)$ représenté figure 8.
OK1 <==> il existe une place X marquée différente de P n'ayant pas de fils marqué.

OK2 <==> il existe une transition Y marquée, dont les fils ne sont pas tous marqués.

Voir la figure 3. Sur chaque X, on indique $\phi(X)$ par une croix sur l'arc partant de X vers $\phi(X)$. Alors l'ensemble des sommets cochés par $\Delta^+(\phi)$ est marqué par des astérisques.

Prop1 : L'algorithme Δ^+ se termine en un nombre fini de pas.

Dem 1 : Car chaque itération consacre le marquage d'un nouveau sommet au moins et que, quand l'ensemble des sommets marqués sera R tout entier, la condition T = 2 sera vraie puisqu'un réseau de Pétri vérifie la DEF2.

Prop2 : Si, après application de Δ^+, P est coché, alors on a mis en évidence un chemin de S à P (il existe un chemin de S à P dans le sous graphe coché).

Dem2 : Considérons un promeneur qui se déplace sur le réseau, d'un sommet à l'autre, avec les consignes suivantes :

- Pars de P

- Si tu arrives en S alors Stop

- Sinon, si tu es à un sommet, alors va à un sommet coché en amont.

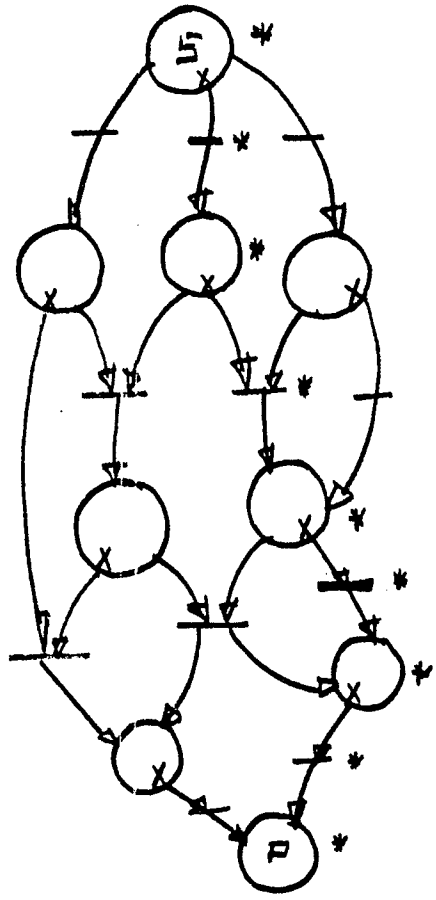
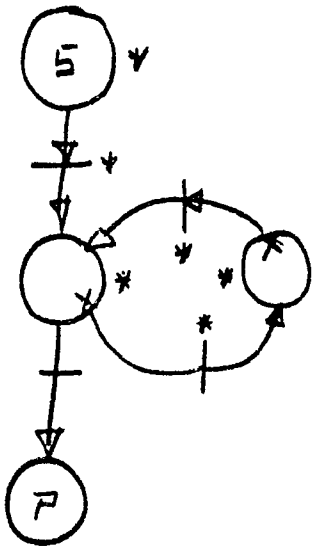
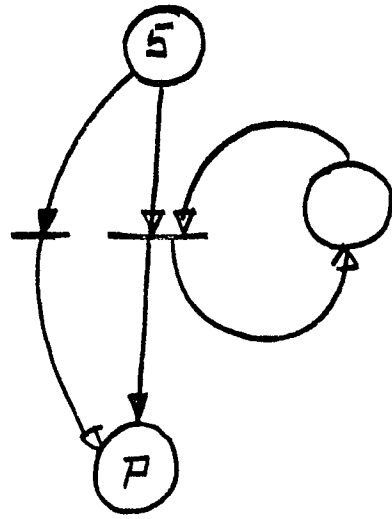


figure 3



a



b.

Si Φ a la valeur indiquée, Δ^+ ne coche pas P. On détecte le circuit.

Pour ce réseau, par contre, quelle que soit Φ, $\Delta^+(\Phi)$ coche P. On ne détectera pas le circuit.

figure 4

+ Tous les sommets visités sont cochés puiqu'il part de P(coché) et qu'il se déplace toujours vers des sommets cochés.

+ Les consignes sont applicables car tout sommet coché admet au moins un sommet coché en amont (celui qui a permis de le cocher).

+ Vu que le réseau est sans circuit, le promeneur ne passe jamais plus d'une fois sur le même sommet => il finira par s'arrêter car le réseau est fini ; donc il arrivera en S, seul sommet où il est autorisé à s'arrêter. Comme il a parcouru tous les arcs en contresens, on a prouvé l'existence d'un chemin de S à P.

Prop8 : Si, après application de Δ^+, P n'est pas coché, alors il existe un circuit dans R.

Dem8 : On considère à nouveau le promeneur avec les consignes suivantes :

- Pars de S
- Si tu est en P alors arrête.
- Sinon, si tu es à une place X alors va à $\phi(X)$.
  Sinon (tu es à une transition Y) alors passe à un fils quelconque de Y

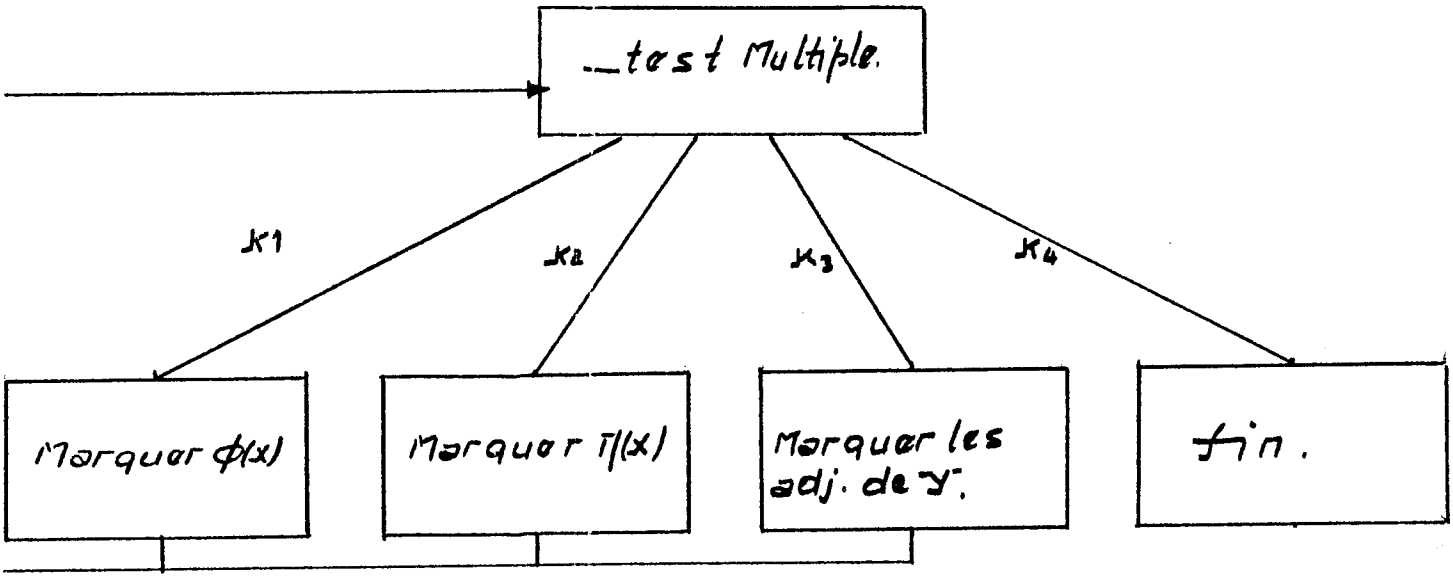
+ Les instructions sont applicables : trivial.

+ Tous les sommets visités sont cochés car S est coché, que si une place X est cochée, alors $\phi(X)$ est cochée et que, si une transition Y est cochée alors tout fils de Y est coché.

+ Le voyageur ne s'arrête pas car il ne croise que des sommets cochés et que P (le seul sommet où il doit s'arrêter n'est pas coché.

+ R est fini donc si le promeneur s'y déplace indéfiniment , c'est qu'il existe des sommets qu'il visite plus d'une fois : il existe un circuit dans R. (Il s'agit d'un circuit et non d'un cycle car le promeneur se déplace toujours dans le sens des arcs).

Remarque : la figure 4 présente en a un cas où Δ^+ détecte un circuit et en b un cas où il ne le détecte pas. Le fait que P ne soit pas coché par $\Delta^+(\phi_0)$ est dû à ce que la configuration particulière de ϕ_0 n'autorise pas



ferme (Π, φ)

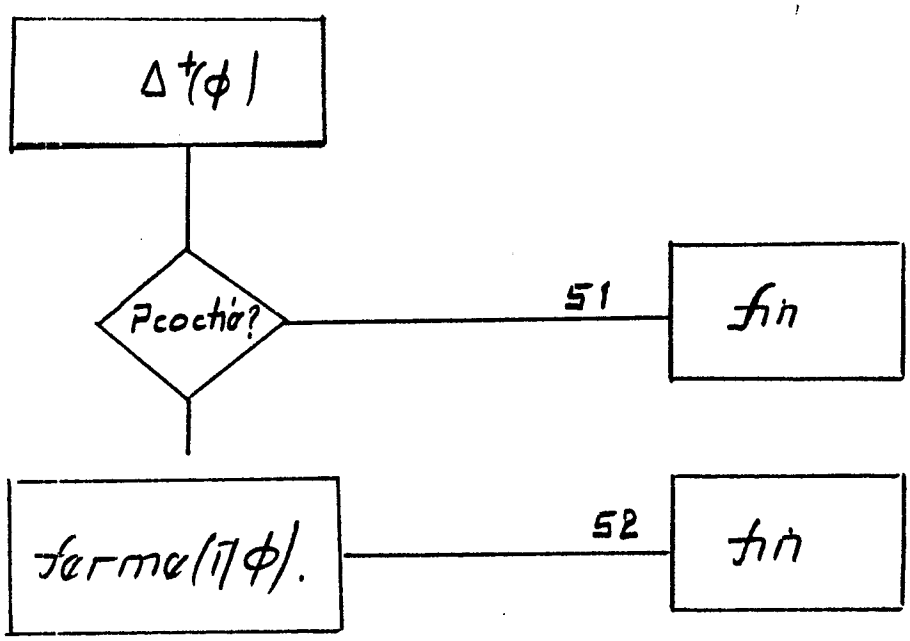
r1 - Il existe une place cochée x non égale à P qui n'a pas de fils marqué.

r2 - Il existe une place cochée x non égale à S qui n'a pas de père marqué

r3 - Il existe une transition y cochée dont les adjacents ne sont pas tous cochés

r4 - ¬(K1 ∨ K2 ∨ K3) - l'ensemble des sommets cochés vérifie i et ii.

figure 5.



New (Π, φ)

figure 6.

à cocher un chemin de S à P. Par contre, pour tout chemin C de S à P il existe une configuration ϕ telle que $\Delta^+(\phi)$ coche tous les sommets de C. Pour cela, on définira, pour chaque place X appartenant à C $\phi(X)$ = la transition suivant X dans C.

Remarque : dans le chapitre 1, nous avons proposé de relier S aux sources ordinaires par un noeud (multiple) OU. L'intérêt pratique de cette proposition apparaît ici : Si on opte pour le noeud OU, Δ^+ ne coche qu'une seule source ordinaire alors que si on opte pour le noeud ET, Δ^+ coche toutes les sources ordinaires donc tout le réseau.

B2 : Transformer le chemin en écoulement

Après application de $\Delta^+(\phi)$ deux cas se présentent :

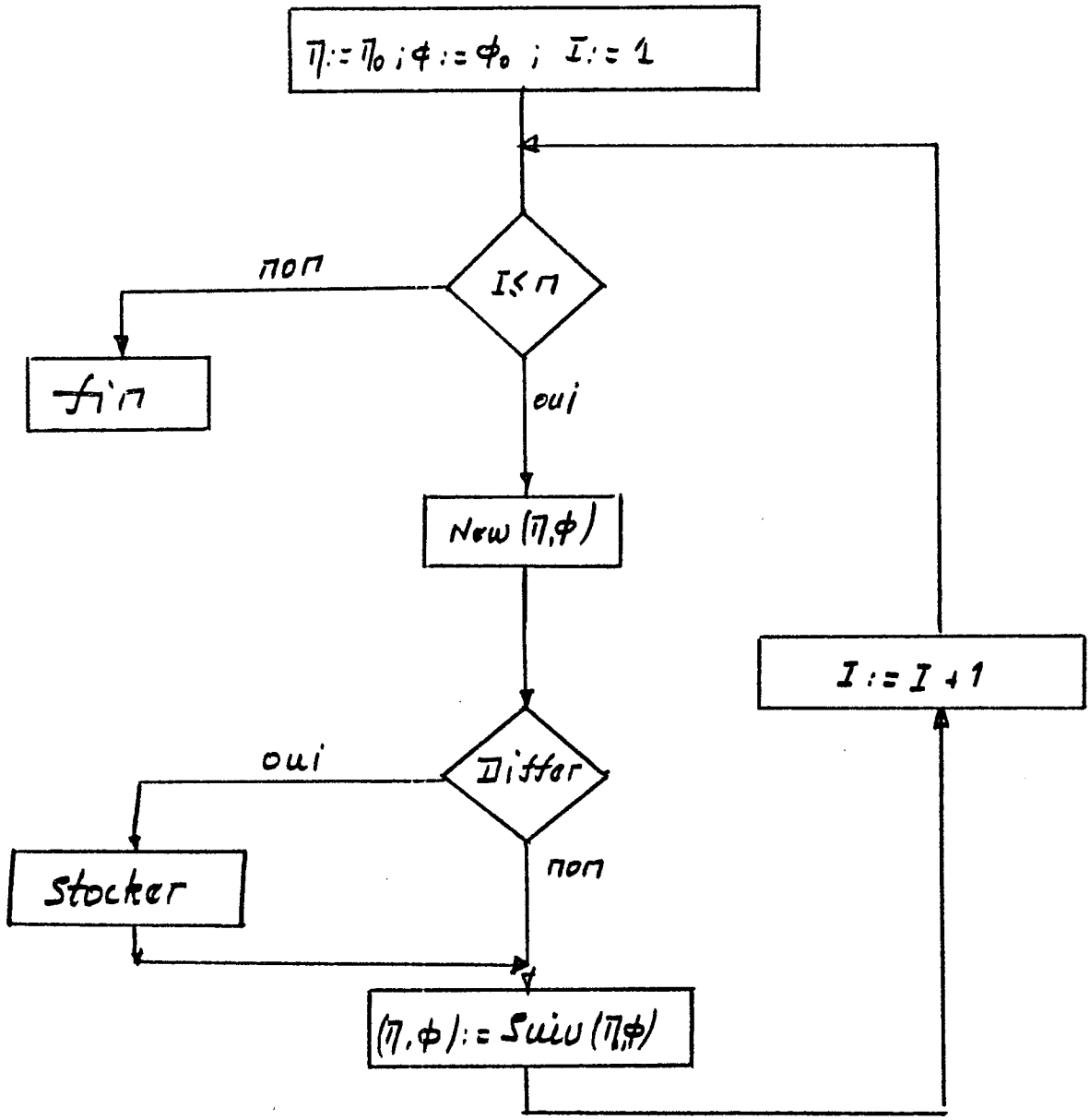
1. L'ensemble coché vérifie les conditions i et ii auquel cas il représente un écoulement.

2. L'ensemble coché ne vérifie pas les conditions i et ii, alors il faut lui adjoindre des sommets supplémentaires de sorte qu'il les vérifie. Cette adjonction se fait de la façon suivante :

Dès qu'une transition est cochée, on coche tous ses adjacents (car tout écoulement contenant une transition contient ses adjacents): dès qu'une place est cochée et qu'il lui manque une transition père, ou une transition fils, on coche un père (et un seul) ou un fils de cette place (car on s'intéresse aux écoulements petits). Ceci suppose que l'on a défini pour chaque place X, en plus de son fils privilégié $\phi(X)$, un père privilégié qu'on notera $\Pi(X)$. Considérons l'algorithme de marquage ferme (Π,ϕ) présenté à la figure 5.

Prop4 : L'algorithme ferme se termine

Dem4 : Car toute itération consacre le marquage d'au moins un sommet supplémentaire et que le réseau vérifie K4.



Organigramme de AT.

figure 7.

Soit New (Π,ϕ) l'algorithme dont l'organigramme est présenté à la figure 6.

Prop5 : L'ensemble de sommets cochés à la sortie s_2 de New vérifie i et ii (donc le sous graphe construit dessus est un écoulement).

Dem5 : Ceci résulte immédiatement de la définition des conditions K1, K2 K3 et K4.

Remarque : On a proposé, au chapitre 1, de lier le puits principal P aux puits ordinaires par un noeud OU (multiple). L'intérêt pratique de cette proposition apparaît ici : Si on opte pour le noeud ET, alors dès que P est cochée, ferme coche tous les puits ordinaires du réseau, donc tout le réseau.

Remarque : Il peut arriver qu'après passage de New, une place ait plus d'un père ou plus d'un fils ; c'est le cas, par exemple, des places 1 et 6 de la figure 8. Les sommets cochés par Δ^+ sont marqués d'une astérisque et ceux cochés par ferme sont marqués de deux astérisques. Π et ϕ sont indiqués par les croix placées sur chaque place.

B3. Algorithme Global

Π et ϕ étant fixés, New $(\Pi\phi)$ peut, soit détecter un circuit dans le réseau soit trouver un écoulement. L'idée de base de AT consiste à dire que l'on appliquera New $(\Pi\phi)$ en donnant à $\Pi\phi$ toutes les valeurs qu'elles peuvent avoir : Soit N le nombre de configurations possibles que peuvent prendre les applications Π et ϕ (définies de l'ensemble des places vers l'ensemble des transitions). Soit SUIV une application qui, à toute configuration de $\Pi\phi$ en associe une autre distincte de sorte que $SUIV^N(\Pi\phi) = (\Pi\phi)$ et que les N applications $SUIV^i(\Pi\phi)$ obtenues pour $0 \leq i \leq N-1$ sont toutes distinctes.

L'algorithme AT a son organigramme présenté à la figure 7.

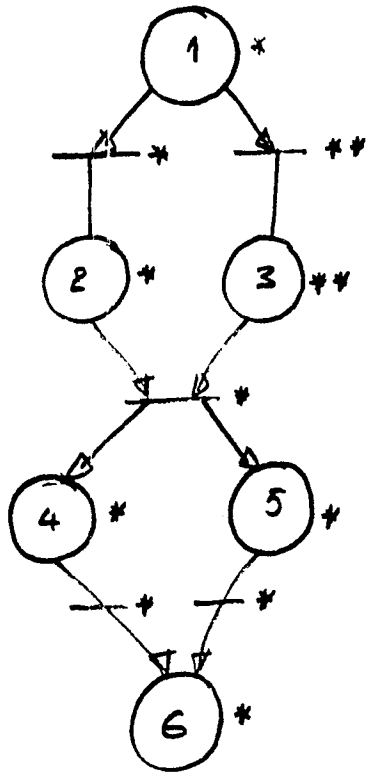


Figure 8

La logique procédure Differ délivre vrai si l'écoulement qui vient d'exhiber New (Π,ϕ) est distinct de ceux qui ont déjà été trouvés et stockés en mémoire. Si tel est le cas, il est stocké.

Π_0 et ϕ_0 sont des valeurs initiales arbitraires pour Π et ϕ.

C. Propriétés de l'algorithme AT.

Prop 11 : Les écoulements qu'exhibe l'algorithme AT réalisent une couverture du réseau R.

Dem 11 : Soit une place x dans R. Il existe un chemin de S à x et un chemin de x à P donc un chemin C de S à P passant par x. Soit ϕ_0 une application qui, à toute place de C associe la transition suivant cette place dans le chemin (et à toute place n'appartenant pas à C associe une transition fils quelconque). Alors $\Delta^+(\phi_0)$ coche un ensemble de sommets contenant C. P étant coché puisque appartenant à C ferme $(\Pi\ \phi_0)$ sera appliqué pour donner un écoulement qui couvre x.

Prop 12 : Les écoulements trouvés par AT sont élémentaires.

Dem 12 : La démonstration de cette proposition a été faite dans [7] . Elle est très longue, on n'en donnera ici que l'idée générale : elle consiste à considérer un écoulement Δ trouvé par New, supposer qu'il est union de deux écoulements Δ_1 et Δ_2 distincts de lui-même et montrer qu'il existe une transition appartenant, par exemple à Δ_1 et adjacente à une place appartenant à $\Delta_2 - \Delta_1$ ce qui est impossible.

Prop 13 : AT exhibe tous les éléments minimaux de R.

Dem 13 : Soit E un écoulement minimal de R. On définit deux applications de l'ensemble des places de R vers l'ensemble de ses transitions par :

$\bar{\Pi}(X)$ = Si X est dans E alors un père de X dans E sinon un père quelconque de X dans R.

$\bar{\phi}(X)$ = Si X est dans E alors un fils de X dans E sinon un fils quelconque de X dans R.

Soit Δ l'écoulement détecté par New $(\bar{\Pi},\bar{\phi})$. (On suppose que $\bar{\phi}$ est tel que

Attak($\bar{\Pi}$ $\bar{\phi}$) coche P. Pour cela on peut construire $\bar{\phi}$ à partir d'un chemin de S à P inclus dans E).
On se propose de montrer en deux étapes que $\Delta = E$.

$\Delta \subset E$. Si nous considérons l'algorithme de marquage décrit par New, nous voyons qu'à chaque fois qu'on est sur une place, on coche une transition adjacente, et que, à chaque fois qu'on est sur une transition, on coche un certain nombre de places adjacentes. Pour montrer que $\Delta \subset E$, on utilisera une certaine forme de récurrence sur les sommets de Δ. On montrera :

. $\Delta \cap E = \phi$

. Si une place x de Δ appartient à E_1 alors la transition cochée à partir de x appartient à E.

. Si une transition appartient à E_1, alors les places cochées à partir de t appartiennent à E.

Ainsi $S \in \Delta \cap E \Rightarrow \Delta \cap E = \phi$

Soit x une place de Δ appartenant à E : la transition marquée à partir de x est soit $\bar{\Pi}(x)$, soit $\bar{\phi}(x)$ donc appartient à E par définition. Soit t une transition de Δ appartenant à E : les places marquées à partir de t sont nécessairement dans E puisque E vérifie ii. c.q.f.d.
On se propose maintenant de montrer que :

$E \subset \Delta$: Si on applique Attak($\bar{\Pi}$ $\bar{\phi}$) à R_1 on va trouver un ensemble de sommets cochés qu'on notera Δ^+, tel que $\Delta^+ \subset E$, pour des raisons analogues à celles évoquées ci-dessus,(soit, en l'occurence :

. $S \in E$

. Si une place x de Δ^+ appartient à E alors $\bar{\phi}(x)$ aussi

. Si une transition t de Δ^+ appartient à E alors toutes les places aval appartiennent à E cqfd).

Par ailleurs, on a évidemment $\Delta^+ \subset \Delta$ donc $\Delta^+ \subset \Delta \cap E$.

Or Δ^+ contient un chemin de S à P - par construction de $\bar{\Pi}$ et $\bar{\phi}$ - donc $\Delta \cap E$ est un couvert contenant un chemin de S à P ==> $\Delta \cap E$ est un écoulement
$\Delta \cap E$ est contenu dans E qui est un écoulement minimal donc il lui est égal
$\Delta \cap E$ ==> $E \subset \Delta$.

On a montré que, étant donné un écoulement minimal E, il existe une configuration $\begin{pmatrix} \bar{\Pi} \\ \bar{\phi} \end{pmatrix}$ de $\begin{pmatrix} \Pi \\ \phi \end{pmatrix}$ telle que New($\bar{\Pi}$ $\bar{\phi}$) appliquée à R exhibe E.

Vu que AT active New en passant en revue toutes les combinaisons possibles pout $\begin{pmatrix} \Pi \\ \phi \end{pmatrix}$, donc en particulier $(\bar{\Pi}\bar{\phi})$, on trouvera E en appliquant AT. cqfd.

Prop 14 : L'union des écoulements élémentaires d'un réseau couvre ce réseau.

Dem 14 : D'après Prop 11 les écoulements trouvés par AT couvrent le réseau. D'après Prop 12 tous les écoulements trouvés par AT sont élémentaires.

Remarque 6 : Il a été prouvé à la figure 4 que l'union des minimaux n'était pas égale au réseau. Ainsi AT trouve tous les minimaux du réseau et au moins assez d'élémentaires non minimaux pour couvrir tout le réseau.

Remarque 7 : La proposition 14 met en évidence l'intérêt des élémentaires dans un réseau. En fait il va plus loin : Soient deux écoulements E1 et E2 dans R. Par définition, on peut tester E1 et E2 séparément. Il ne saurait être d'aucun intérêt de tester E1 ∪ E2 séparément. On dira alors que l'on s'intéresse exclusivement aux écoulements qui ne sont pas union de deux autres distincts d'eux -mêmes, à savoir les écoulements élémentaires.

D. Evaluation de performances de AT

Considérons un réseau R. On appelle :

N le nombre d'applications $(\Pi\phi)$ qu'il est possible de définir sur R.

NP le nombre de places de R.

NAP le demi degré maximal sur les places de R.

AT appelle New N fois

Considérons l'organigramme de New à la figure 6.

NEW appelle Δ^+ et ferme une fois

Considérons les organigrammes de Δ^+ et ferme figures 2 et 5

Δ^+ et ferme déroulent au plus NP marquages à eux deux.

La longueur de AT, évaluée en nombre de marquages est donc majorée par N. NP ; N est très largement majoré par $(NP)^{NAP}$, donc $N.\ NP \leq (NP)^{NAP+1}$.

Cette majoration n'est pas satisfaisante car ce nombre peut être trop grand pour des réseaux ordinaires (Il est exponentiel en NAP). Ceci conduit à la nécessité de modifier l'algorithme AT et le réseau sur lequel on l'active : les modifications de l'algorithme sont présentées en hors texte à la fin de la thèse ; quant aux modifications des réseaux, elles sont faibles dans le paragraphe suivant. L'exemple donné en hors texte montre que ces modifications sont efficaces puisqu'on a pu trouver des résultats satisfaisants dans un temps raisonnable.

E. Transformation du réseau

Considérons le réseau présenté au paragraphe D du chapitre 1. Soit à calculer le nombre d'applications $(\overset{\Pi}{\phi}$ que l'on peut définir sur ce réseau par la formule donnée au chapitre précédent $N = \Pi_{i=1}^{NP} k(i)$.

où $k(i) = d^+(i)$ si $i = S$, $d^-(i)$ si $i = P$ et $d^-(i).d^+(i)$ sinon. On trouve N = 378.622.771.200 ; soit autant d'itérations New dans l'algorithme AT appliquée à ce réseau. En outre, on évalue par expérience le temps que mettrait New sur un réseau de cette taille à (1/10) seconde : ces chiffres prouvent la nécessité de modifier le réseau : On modifiera le réseau en regroupant dans une même place certaines places et transitions du réseau initial, en accord avec les règles suivantes :

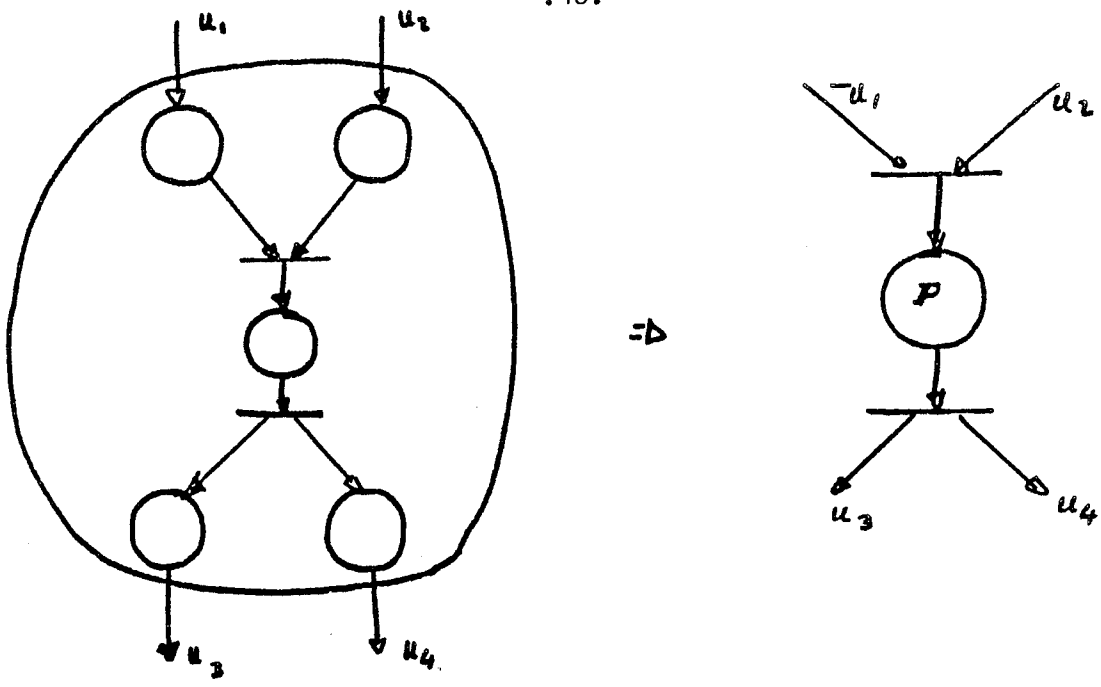

u1
u2
u3
u4
=D
P

fig 9.

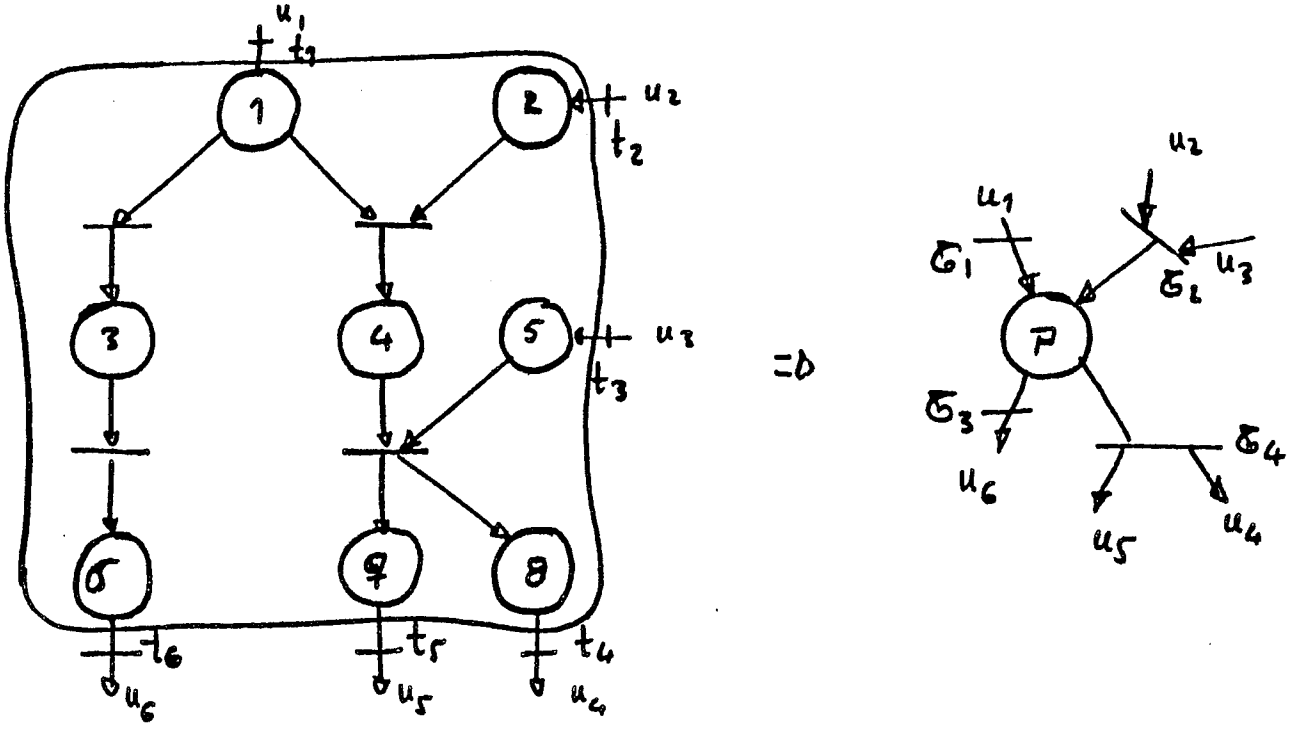

u1
t1
1
2
u2
t2
3
4
5
u3
t3
=D
6
7
8
t6
t5
t4
u6
u5
u4
σ1
σ2
σ3
σ4
P

fig 10

R1 - On regroupe dans la même place du réseau réduit les places et les transitions indiscernables (ceci ne diminue pas le nombre d'itérations mais simplifie le réseau). Les arcs entrant ou sortant de la place du réseau réduit sont connectés en noeud ET - voir figure 1.

R2 - On regroupe dans la même place les places et les transitions du réseau initial dont la séparation est "facile à réaliser à vue d'oeil". (Ceci revient à partager le travail de séparation entre le programme AT et son utilisateur - l'utilisateur se réservant la partie facile consistant en des séparations locales. Ce regroupement diminue le nombre d'itérations). Considérons, par exemple, l'ensemble des arcs sortant du sous réseau que l'on a décidé de regrouper. On définit sur cet ensemble la relation ρ par $u_i \rho u_j$ <==> les transitions portées par u_i et u_j sont indiscernables ($\forall \Pi\phi$, alors New(Π,ϕ) coche t_i <==> New($\Pi\phi$) coche t_j ; t_i et t_j étant les transitions figurant sur les arcs u_i et u_j). A chaque classe d'équivalence modulo ρ on associera une transition qui regroupera les arcs de cette classe en noeud ET. Ainsi si toutes les classes n'ont qu'un élément on obtient un noeud OU autour de la place P du réseau P réduit qui représente le sous réseau que l'on veut regrouper. S'il n'y a qu'une classe d'équivalence, cela fera un noeud ET en sortie de P. Dans les autres cas on aura un noeud mixte. Une fois que le programme AT a été activé sur le réseau réduit, et s'il a coché une place P, on décidera quelle partie du sous réseau regroupé prendre en considérant la ou les transitions sortant de P qui sont cochées. voir figure 2. On ferait de même pour les arcs entrants. Si New($\Pi\phi$) appliquée au réseau réduit coche avec P les transitions T_1 et T_3 alors, dans le réseau initial on prendra les places 1 3 6. S'il coche T_1, T_2 et T_4 on prendra 1 2 4 5 7 8 ; s'il coche. S'il coche T_1, T_2, T_3 et T_4 on prendra 1 2 3 4 5 6 7 8, dans les autres cas l'itération New sera considérée comme nulle. Ceci est dû au fait que dans la démarche de réduction des réseaux, il y a certaines relations d'indicernabilité entre transitions entrantes et sortantes qui ne sont pas exmprimées. Dans le cas

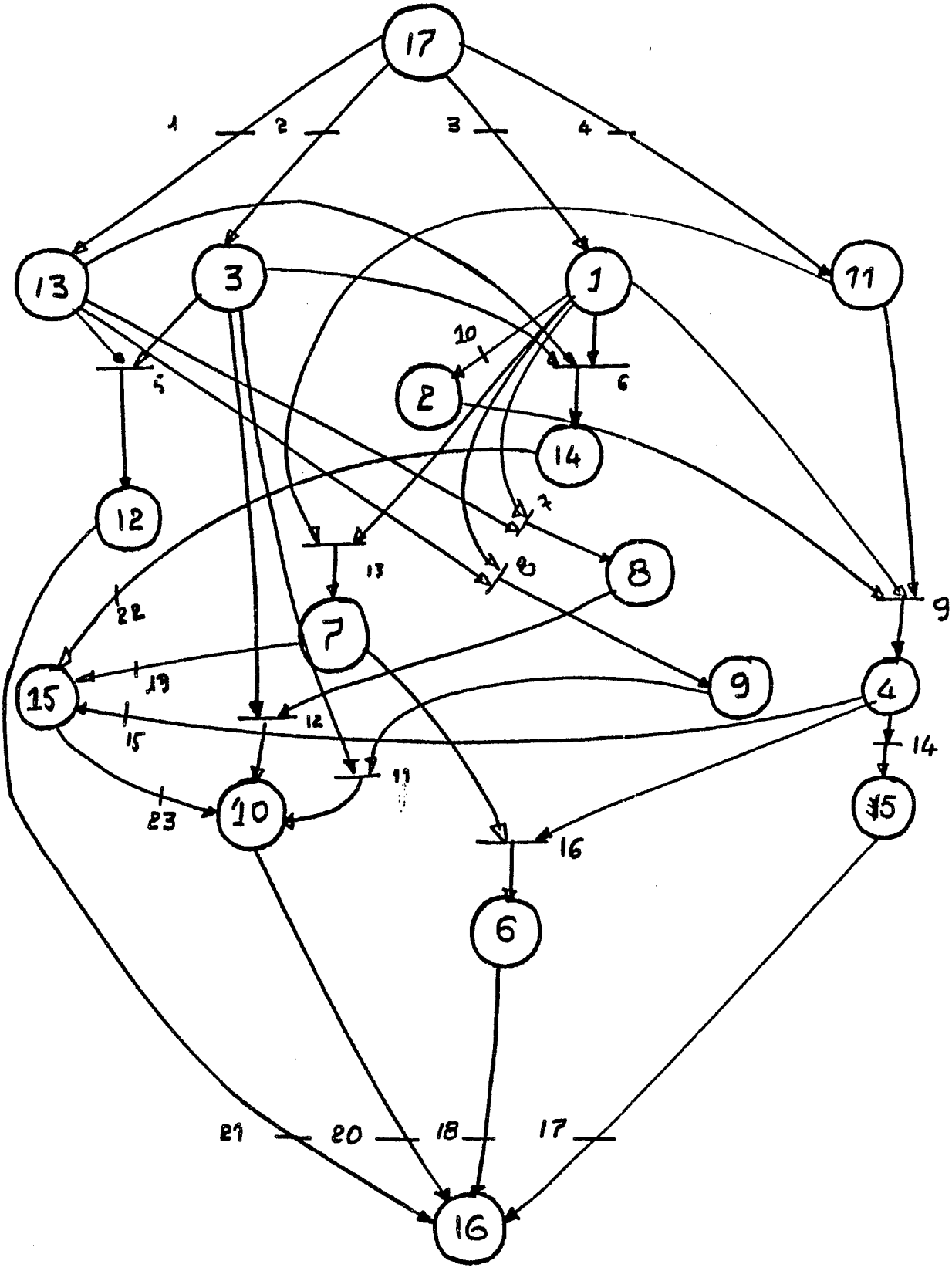


fig 3.2.

Réseau déduit de celui du chapitre 1 par les règles de réduction R1 et R2.

de la figure 2 on pourrait, pour ce faire, adjoindre une place entre les transitions T_2 et T_4 destinée à transmettre le marquage entre T_2 et T_4 sans augmenter le nombre d'itérations New puisque cette place n'aurait qu'un père, T_2, et qu'un fils, T_4.

En appliquant R1 et R2 on a réduit le réseau du chapitre 1 pour obtenir celui donné à la figure 3 qui a 17 places et 23 transitions. Le nombre d'applications (Π_ϕ qu'il est possible de définir sur ce réseau vaut : 157.888. Ceci vaut approximativement $0.5.10^{-6}$ fois le nombre associé au réseau initial mais reste très grand. On verra qu'une légère amélioration du programme permettra de ne faire que 4608 appels new dans un temps très raisonnable : environ 140 Secondes.

On va donner dans la suite la liste des places du réseau réduit en précisant à chaque fois les places regroupées ainsi que la règle appliquée pour ce regroupement.

1. Regroupe 4,5,6,12,16,17,20,24,25,33,50,51,52,56,57 et 60 en application de R2.
2. Regroupe 1,13,15 en application de R1.
3. Regroupe 2,14,29,58 en application de R2.
4. Regroupe 18,22,28 en application de R2
5. Regroupe 7,21,26,27,34,35,37,38,43 et 44 en application de R2.
6. Regroupe 8,45,61 en application de R1
7. Regroupe 9 et 46 en application de R1
8. Regroupe 47 et 53 en application de R1.
9. Regroupe 49 et 55 en application de R1.
10. Regroupe 62 et 63 en application de R2.
11. Regroupe 3 en application de R1.

12. Regroupe 40,41,48, et 54 en application de R1
13. Regroupe 10 et 11 en application de R2.
14. Regroupe 23,30,31, et 36 en application de R1.
15. représente 39
16. représente 64, le puits
et 17. représente 64, la source.

Exemple d'application

On trouve dans la suite la trace d'exécution de l'algorithme AT sur un exemple simple de réseau à 5 places et 5 transitions. On ne s'intéressera pas ici à l'implémentation en Algol W de AT, celle-ci fera l'objet du hors texte à la fin de la thèse. Ainsi on ne s'intéressera pas à DP et D (qui permettent de réaliser la fonction SUIV).

Le réseau à étudier est implanté deux fois en mémoire, dans AP et AT : dans la I° ligne de AP on met la liste des adjacents de la place i avec le signe plus s'il s'agit de fils et avec le signe moins s'il s'agit de pères. Il en est de même pour AT. Le réseau obtenu est représenté figure 13. Nous allons étudier une à une les applications de l'algorithme de New($\Pi\phi$) pour toutes les valeurs intéressantes de ($\Pi\phi$).

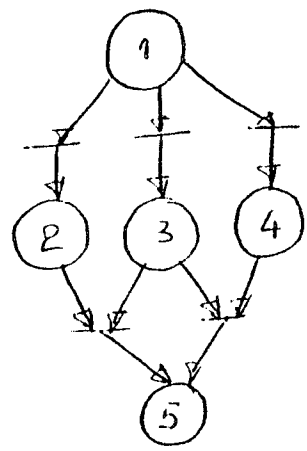


figure 13

Soit à compter le nombre d'applications (Π_ϕ qu'il est possible de réaliser sur ce réseau. On le notera N.

Soient NP et NT le nombre de places et le nombre de transitions respectivement.

On a $N = \prod_{i=1}^{NP} d^+(i) \cdot \prod_{i=1}^{NP} d^-(i)$, par définition.

$d^+(i)$: nombre de transitions en aval de i.

$d^-(i)$: nombre de transitions en amont de i.

dans le cas du réseau qui nous intéresse, N = 12. Or N est aussi le nombre d'appels du segment New dans l'algorithme AT. On voit que ce nombre augmente rapidement avec le nombre de places où il y a un choix entre plusieurs transitions, i.e., avec le nombre de noeuds OU :

Toujours est-il que dans le cas qui nous intéresse, nous avons pu explorer le réseau en seulement 6 appels de New au lieu de 12, et ce grâce à la remarque suivante : A chaque fois que le puits est coché par Δ^+, il a certainement une transition amont qui est cochée, c'est celle qui lui a propagé le marquage, donc 'ferme' n'aura jamais à affecter un père au puits P ; donc $\Pi(P)$ peut être fixé puisque sa valeur n'a aucune espèce d'importance. Ceci nous a permis en l'occurence de diviser le nombre d'appels New par deux.

A chacune des six itérations, on donnera l'état de la fonction $|_{\phi}^{\Pi}$ au moyen des colonnes PERE et FILS ainsi que, si Differ, le nouvel écoulement trouvé à cette itération.

TABLEAU AP

1 /	1	2	3
2 /	-1	4	
3 /	-2	4	5
4 /	-3	5	
5 /	-4	-5	

TABLEAU AT

1 /	-1	2	
2 /	-1	3	
3 /	-1	4	
4 /	-2	-3	5
5 /	-3	-4	6

TABLEAU DP

3	1	2	1	0	0	1	1
1	2						

ITERATION NUMERO 1

TABLEAU D

1	1	1	1	0	0	1	1
1	1						

PERE ET FILS

1	PERE	-1	FILS	1
2	PERE	1	FILS	4
3	PERE	2	FILS	4
4	PERE	3	FILS	5
5	PERE	4	FILS	-4

NOMBRE D'APPELS NEW 1

ECOULEMENT NUMERO 1

PLACES

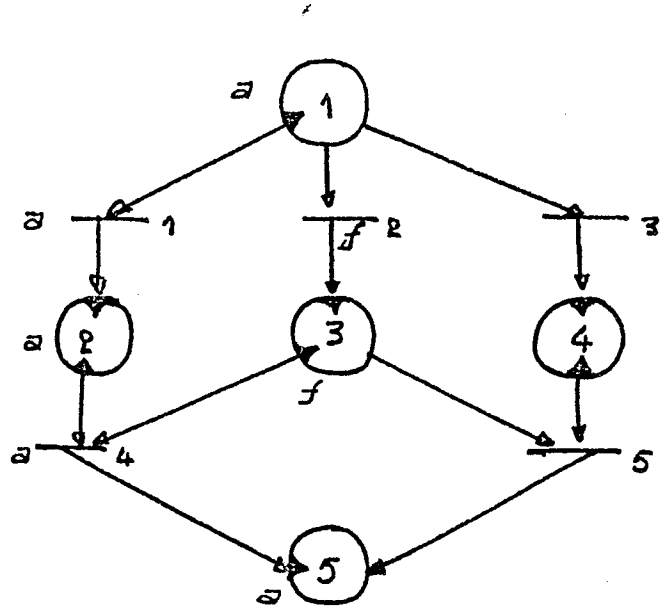
1	2	3	5

TRANSITIONS

1	2	4

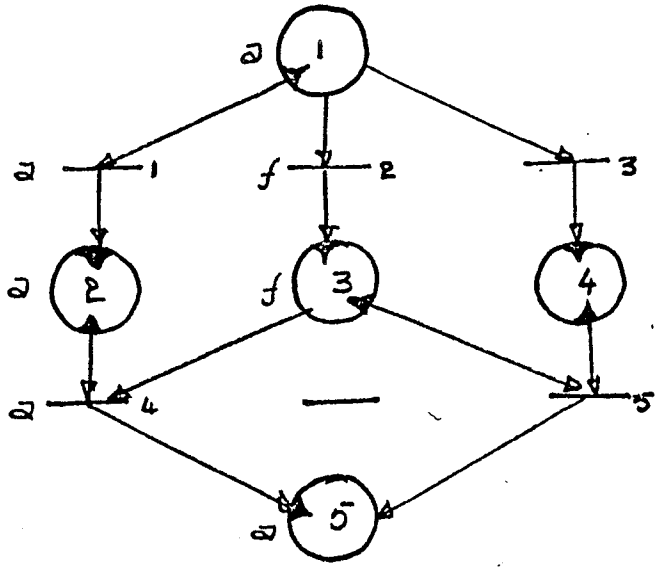
Itération 1.

L'état de $(17, \Phi)$ est matérialisé par les marquages sur chaque place.



On marquera d'un 'Δ' les sommets ouverts par Δ^+ et d'un 'f' ceux ouverts par 'ferme.' L'écoulement trouvé est bien 1235. On le matérialise.

Itération 2



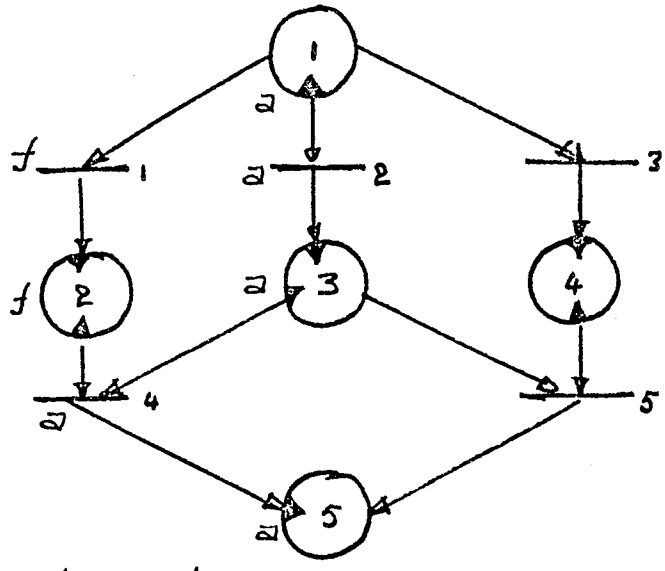
Ce qui a changé entre les itérations 1 et 2, c'est $\Phi(3)$. En effet dans l'itération 1 il n'a pas été fait appel à $\Phi(3)$ car le marquage est arrivé à 3 d'une transition fils $\Rightarrow$ On trouve l'écoulement E2.

```
ITERATION  NUMERO              2
TABLEAU D
          1                   1                   2                   1                   0                   0                   1                   1
          1                   1
PERE ET FILS
          1  PERE              -1  FILS              1
          2  PERE               1  FILS              4
          3  PERE               2  FILS              5
          4  PERE               3  FILS              5
          5  PERE               4  FILS             -4
NOMBRE D'APPELS NEW             2
```

```
ITERATION  NUMERO              3
TABLEAU D
          2                   1                   1                   1                   0                   0                   1                   1
          1                   1
PERE ET FILS
          1  PERE              -1  FILS              2
          2  PERE               1  FILS              4
          3  PERE               2  FILS              4
          4  PERE               3  FILS              5
          5  PERE               4  FILS             -4
NOMBRE D'APPELS NEW             3
```

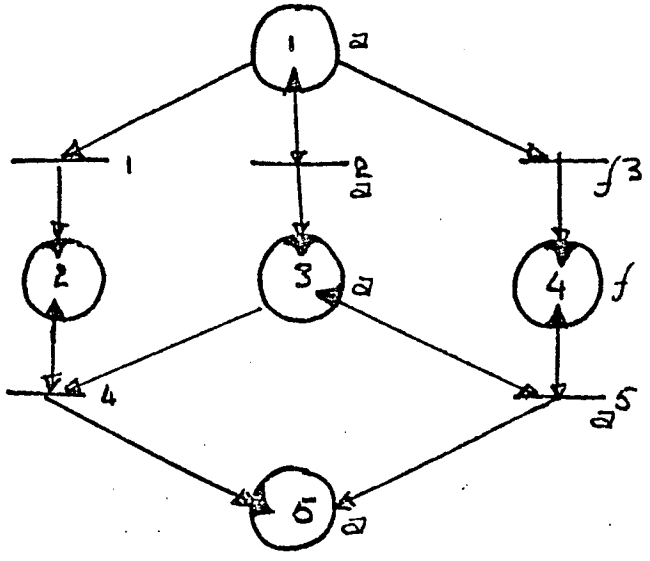
```
ITERATION  NUMERO              4
TABLEAU D
          2                   1                   2                   1                   0                   0                   1                   1
          1                   1
PERE ET FILS
          1  PERE              -1  FILS              2
          2  PERE               1  FILS              4
          3  PERE               2  FILS              5
          4  PERE               3  FILS              5
          5  PERE               4  FILS             -4
NOMBRE D'APPELS NEW.
```

Itération 3



C'est l'écoulement E1, bien que la partie cochée par Δ^+ ait changé.

Itération 4



Nouvel écoulement, noté E2 et contenant, comme c'est indiqué par le programme, 1345 pour les places et 235 pour les transitions.

ECOULEMENT NUMERO 2

PLACES
1 3 4 5

TRANSITIONS
2 3 5

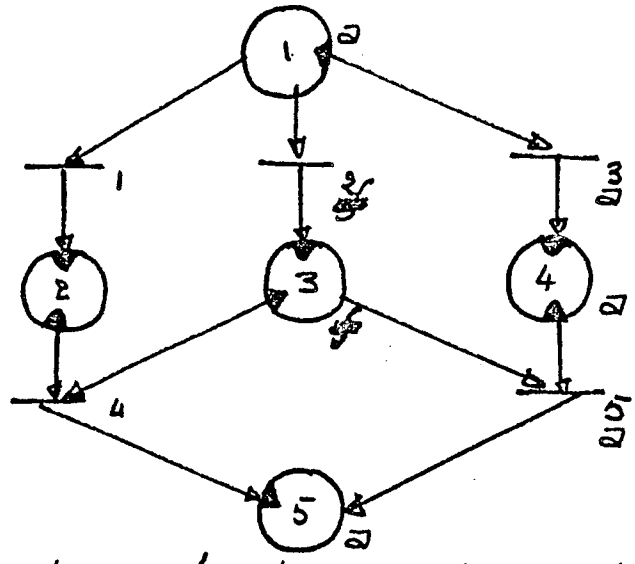
ITERATION NUMERO 5
TABLEAU D
3 1 1 1 0 0 1 1
1 1

PERE ET FILS
1 PERE -1 FILS 3
2 PERE 1 FILS 4
3 PERE 2 FILS 4
4 PERE 3 FILS 5
5 PERE 4 FILS -4
NOMBRE D'APPELS NEW 5

ITERATION NUMERO 6
TABLEAU D
3 1 2 1 0 0 1 1
1 1

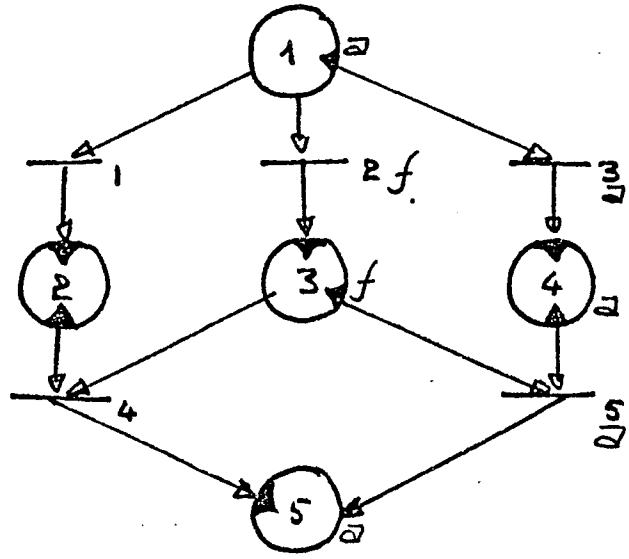
PERE ET FILS
1 PERE -1 FILS 3
2 PERE 1 FILS 4
3 PERE 2 FILS 5
4 PERE 3 FILS 5
5 PERE 4 FILS -4
NOMBRE D'APPELS NEW 6

Itération 5



La partie cochée par Δ^+ est différente de celle de l'itération 4, mais l'écoulement est le même : E2.

Itération 6.



Ce qui a changé entre les itérations 5 et 6, c'est $\Phi(3)$. Vu que il n'est pas intervenu au cours de Δ^+ au cours de "ferme", on trouve le même écoulement que à l'itération 5 à savoir E2. Au cours des six itérations, $\Pi(5)$ a été fixée à 4.

CHAPITRE 3

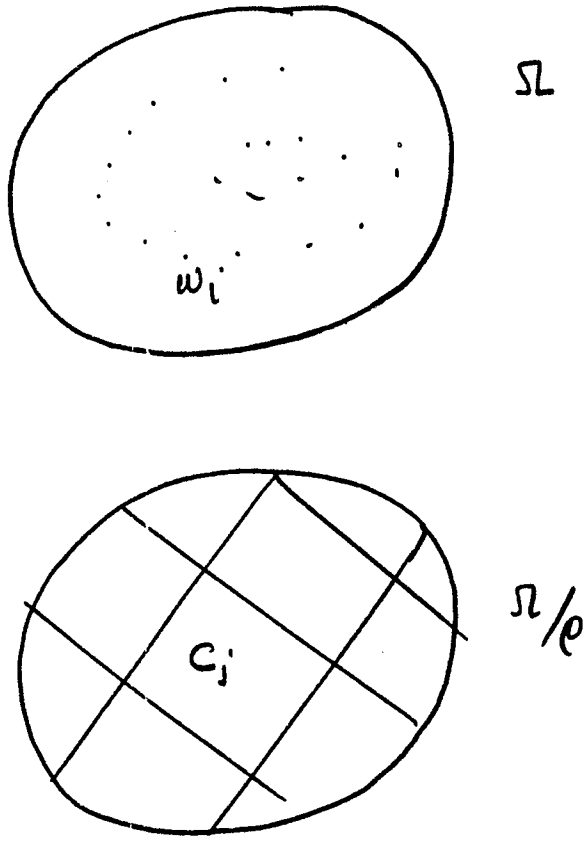
INTRODUCTION A L'ETUDE DES PARAMETRES DE TEST

Pour définir et étudier les paramètres de test, nous avons besoin d'introduire deux notions : celle de "flot d'information" dans un système de communication et celle de "coupe de capacité maximale" dans un réseau. Pour introduire ces deux notions nous utilisons la Théorie de l'information et l'optimisation dans les réseaux.

A - Rappels de Théorie de l'information

A1 - Incertitude d'un événement - Soit $(\Omega\ \mathcal{A}\ P)$ un espace probabilisé fini. $\Omega = \{w_1 — w_n\}$. On pose $P(w_i) = p_i$. Un observateur réalise des expériences dont le résultat est un élément de Ω. Avant d'observer le résultat de son expérience, l'observateur associe à chaque événement w_i l'incertitude $J(w_i)$ qui porte sur la réalisation de w_i. La fonction J vérifie deux propriétés :

- $J(w_i)$ est une fonction décroissante de p_i
- $J(w_i \cap w_j) = J(w_i) + J(w_j)$ si w_i et w_j sont indépendants.



"la réalisation est dans C_j" fournit moins d'information que "la réalisation est ω_i" $\Rightarrow H(\Omega/\ell) \leq H(\Omega)$.

fig 1

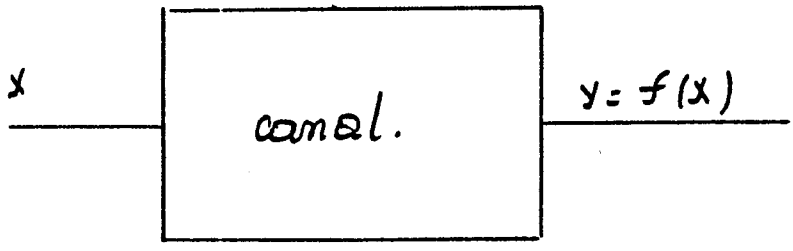


fig 2

On montre que $J(w_i) = -\text{Log } p_i$ qui vérifie les deux critères ci-dessus ; Le log est en base quelconque.

A2 - Incertitude d'une source - On s'intéresse dans ce paragraphe à l'incertitude de l'ensemble Ω qu'on définit comme l'espérance de la variable aléatoire J :

$$H(\Omega) = \sum_{i=1}^{n} p_i J(w_i) = - \sum_{i=1}^{n} p_i \log p_i \qquad (1)$$

la quantité $H(\Omega)$ est dite quantité d'information de Ω ou incertitude de Ω car elle mesure à la fois l'incertitude qu'on a sur la réalisation de Ω ainsi que la quantité moyenne d'information que l'on s'attend à recevoir a priori de la réalisation de Ω.

Remarque 1 : La fonction H a été introduite par annon en 1948 sous le nom d'entropie de la distribution P à cause de l'analogie formelle que son expression présente avec celle de l'entropie introduite par Boltzmann en Thermodynamique et qui mesure le désordre d'un système : l'analogie informelle est aussi claire.

On parlera indifféremment de quantité d'information, d'incertitude ou d'entropie de la distribution P, de l'ensemble Ω ou de la variable aléatoire qui prend ses valeurs dans Ω.

On a $0 \leq H(\Omega) \leq \text{Log}n \qquad (2)$

$H(\Omega)$ est nul si un élément w_i a une probabilité 1, étant convenu que $0\text{Log}0 = 0$

$H(\Omega) = \text{Log}n$ dans le cas où $\forall\, i \;\; p_i = \frac{1}{n}$. (Le meilleur menteur est celui qui ment une fois sur deux).

Soit $(\Omega\, \mathcal{A}\, P)$ un ensemble probabilisé. On définit sur Ω une relation d'équivalence ρ et on définit sur $\Omega/\rho = (c_1\ c_2\ \ c_p)$ une distribution de probabilité Π par $\Pi(c_j) = \sum_{w_i \in c_j} P(w_i)$. Alors on peut s'intéresser à l'entropie de la distribution Π qui vaut $-\sum_{j=1}^{P} \Pi c_j \text{ Log } \Pi c_j$ et on a $H(\Pi) \leq H(P)$. (3)

A3 - Entropie de lois composées - Soient deux variables X et Y prenant leurs valeurs dans deux ensembles $\{x_i,\ 1 \leq i \leq n\}$ et $\{x_j,\ 1 \leq j \leq m\}$. Les couples de réalisations (x,y) prennent leurs valeurs dans l'ensemble $\{x_i\} \times \{y_i\}$ avec une distribution de probabilité $p_{ij} = p(x_i\ y_i)$. L'entropie conjointe est définie par :

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} \text{ Log } p_{ij} = H(Y,X) \qquad (4)$$

On montre que $H(X\ Y) \leq H(X) + H(Y)$, l'égalité ne se produit que si X et Y sont indépendantes ($p_{ij} = p_i.p_j$). La différence $H(X) + H(Y) - H(X,Y)$ est d'autant plus grande que X est stochastiquement lié à Y.

A4 - Entropie conditionnelle - Soient deux variables X et Y. On connaît une réalisation y_i de Y. On appelle entropie conditionnelle de X si $Y = y_j$ l'entropie de la distribution $p(x_i/Y = y_j)$, à savoir,

$$H(X/y_j) = -\sum_{i=1}^{n} p(x_i/Y = y_j) \text{ Log } p(x_i/Y = y_j) \qquad (5)$$

L'entropie conditionnelle de X si Y se note H(X/Y) et est définie comme l'espérance de $H(X/y_j)$ pour toutes les réalisations Y.

$$H(X/Y) = \sum_{j=1}^{m} H(X/y_j).\ p(y_j) \qquad (6)$$

On a $0 \leq H(X/Y) \leq H(X)$ (7)

L'incertitude sur X sachant Y ne peut être plus grande que l'incertitude sur X ignorant Y).

$H(X/Y) = 0$ la réalisation de X est connue si on connaît celle de Y.

$H(X/Y) = H(X)$ X et Y sont indépendantes.

On a les relations $H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y) = H(Y,X)$

d'où l'on déduit $H(X) - H(X/Y) = H(Y) - H(Y/X)$: (8)

La diminution d'entropie de X consécutive à l'observation de Y est égale à la diminution d'entropie de Y consécutive à l'observation de X. Cette quantité est symétrique en X et Y, s'appelle transinformation de X,Y et se note T(X,Y). Elle vérifie :

$$0 \leq T(X,Y) \leq \min(H(X), H(Y)) \qquad (9)$$

$T(X,Y) = 0$ <==> X et Y sont indépendantes.

$T(X,Y) = H(X)$ <==> la connaissance de Y détermine X

$T(X,Y) = H(Y)$ <==> la connaissance de X détermine Y

A5 - <u>Capacité d'un canal, capacité d'une ligne</u> - Un canal d'information est un opérateur univoque qui, à toute entrée X associe une sortie $Y = f(X)$. On appelle capacité du canal et on note C(XY) le maximum de T(X,Y) sur l'ensemble des distributions possibles de X. voir figure 2.

Une ligne est un canal qui réalise la fonction identité Y = X ; sa capacité vaut C(X) = max T(X,X) = max H(X) = Log(X) où (X) est le nombre de valeurs que peut prendre X. Dans la formule 1 on a vu que la base du Log est arbitraire. A chaque base correspond une unité d'information au sens physique du terme. Ainsi on parlera du Nat(natural Unit), de Decit (décimal unit) ou de Bit (binary unit) selon que le log est en base e, 10 ou 2. Etant données les applications qu'on a en vue on adoptera évidemment le bit, si bien que la capacité d'une ligne de n bits binaires vaut Log 2^n = n bits.

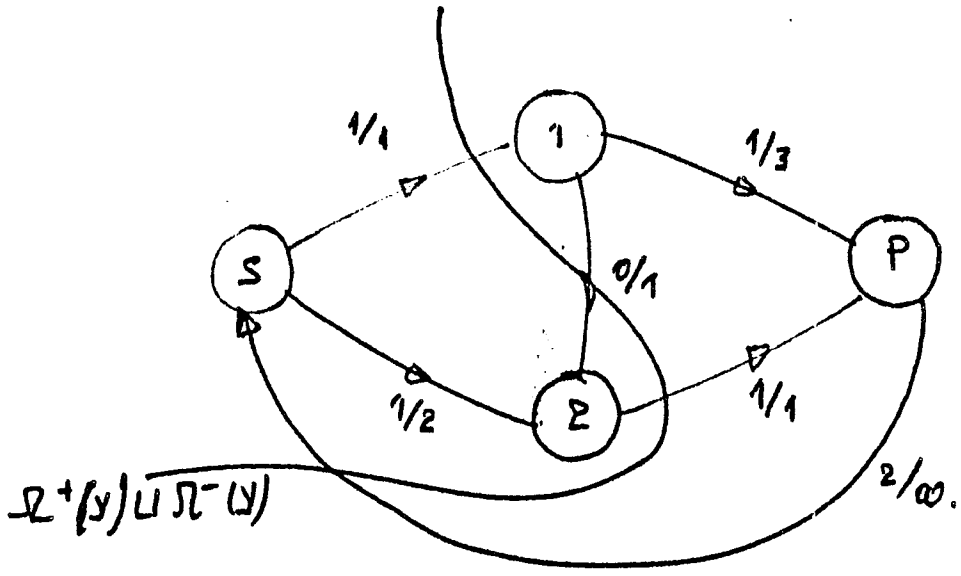
B. Optimisation dans les réseaux

B1 - Position du problème - L'optimisation dans les réseaux peut se présenter sous deux formalismes distincts, l'un étant plus général que l'autre ; dans les paragraphes C et D nous ferons appel à l'un puis à l'autre de ces deux formalismes. On donnera d'abord le formalisme particulier, plus facile à exposer et à comprendre, quitte ensuite à le généraliser pour obtenir le deuxième.

Soit un graphe orienté G = (X,U) admettant une source et un puits : (c'est le cas des réseaux de Pétri que nous étudions). On définit une fonction $c : \tilde{U} \to \mathbb{R}^+ \cup \{+\infty\}$ où $\tilde{U} = U \cup u_r$. u_r est l'"arc de retour" joignant P à S tel que $c(u_r) = +\infty$. Alors le problème de flot maximum de S à P dans le réseau R = $(X,\tilde{U},c)$ consiste à chercher une fonction $f : \tilde{U} \to \mathbb{R}^+ \cup \{+\infty\}$ qui maximise $f(u_r)$ sous les contraintes suivantes :

i) $\forall x \in X, \quad \sum_{u \in \Omega^+(x)} f(u) - \sum_{u \in \Omega^-(x)} f(u) = 0$ (1ère loi de Kirchoff)

ii) $0 \leq f(u) \leq c(u) \qquad \forall u \in \tilde{U}$



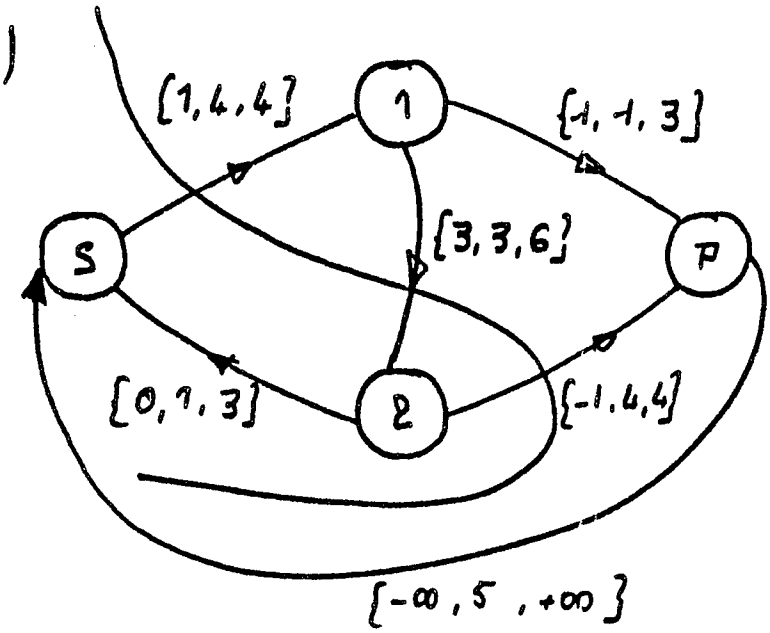
Sur chaque arc on porte $f_{uv}/c(u)$

On pose $y = \{1, 2\}$ alors la capacité de $\Gamma^+(y)$ vaut $2 = f(u_r)$ donc le flot est maximal

fig 3

$\Gamma^+(y) \cup \Gamma^-(y)$

$y = \{s, 2\}$.



$U^+ = \Gamma^+(y) = \{(1,1), (2,P)\}$ $\quad U^- = \Gamma^-(y) = \{1,2\}$

$\sum_{u \in U^+} c(u) = 4 + 4 = 8$ $\quad \sum_{u \in U^-} b(u) = 3$ $\quad 8 - 3 = 5 = f(u_r)$

figure 4.

On appelle flot réalisable toute fonction vérifiant i et ii. Ce formalisme est un modèle adéquat pour le problème qui consiste à maximiser la quantité de fluide qui traverse un réseau de pipe lines dont chaque branche a une section limitée ou le courant qui traverse un montage électrique dont chaque connexion ne peut supporter une intensité plus forte qu'un seuil fixé.

DEF1 : Dans le réseau $R = (X,U,c)$ un ensemble d'arcs C est appelé coupe séparant P de S s'il existe $Y \subset X$ tel que $S \in Y$, $P \notin Y$ et que $\Omega^+(Y) = C$.

DEF2 : On appelle capacité d'une coupe C et on note $c(C)$ la somme des capacités des arcs de cette coupe.

Prop1 : Pour tout "flot réalisable" f sur $R = (X,U,c)$ et pour toute coupe C, on a $f(u_r) \leq c(C)$

Dem1 : Ce théorème sera démontré dans le cadre général de la proposition 3.

Prop2 : La valeur maximum de $f(u_r)$ pour un flot réalisable sur $R = (X,\tilde{U},c)$ est égale à la capacité d'une coupe de capacité minimum séparant P de S.

Ce théorème, dit théorème de la coupe minimum est très utile en pratique. Quand on a défini un flot réalisable f sur un réseau R et qu'on pense que f est optimal, il n'y a qu'à exhiber une coupe dont la capacité vaut $f(u_r)$ pour trancher. Sur la figure 3, on a mis en évidence une fonction $f : U \to \mathbb{R}$ qui vérifie i et ii et une coupe C (en l'occurence $\Omega^+(2,S)$) telle que $c(C) = f(u_r)$. On conclut que f est optimale.

Les résultats et les définitions qui viennent d'être exposés sont faciles à comprendre car ils s'apparentent à des problèmes très concrets.

Nous allons nous efforcer de trouver des généralisations de ces résultats car certains problèmes ne sont modélisables que dans un cadre plus large que celui vu ci-haut :

Soit le graphe orienté $G = (X,\tilde{U})$. On définit deux fonctions :

$b : \tilde{U} \to \mathbb{R} \cup \{-\infty\}$ et $c : U \to \mathbb{R} \cup \{+\infty\}$. Le problème que l'on se pose est alors de chercher une fonction $f : U \to \mathbb{R} \cup \{-\infty,+\infty\}$ qui maximise $f(u_r)$ sous les contraintes suivantes :

i) $\forall x \in X \quad \sum_{u \in \Omega^+(\{x\})} f(u) - \sum_{u \in \Omega^-(\{x\})} f(u) = 0.$

ii) $b(u) \leq f\,u \leq c(u) \qquad \forall u \in \tilde{U}.$

On appelle flot toute fonction vérifiant i et flot réalisable toute fonction vérifiant i et ii. (Le problème précédent revient à fixer de celui-là $b(u) = 0 \quad \forall u \in \tilde{U}$).

Def3 : On appelle coupe séparant S de P un sous ensemble C de U (donc ne contenant pas u_r) tel qu'il existe un ensemble de sommets $Y \subset X$ vérifiant :

+ $C = \Omega^+(Y) \cup \bar{\Omega}^-(Y)$

+ Y contient S et ne contient pas P.

$\bar{\Omega}^-(Y)$ étant défini par $\Omega^-(Y) - \{u_r\}$. En effet $u_r \in \Omega^-(Y)$ puisque Y contient S et ne contient pas P.

Def4 : Soit U' un sous ensemble de U. On note $b(U') = \sum_{u \in U'} b(u)$

$c(U') = \sum_{u \in U'} c(u).$

Prop3 : Pour tout "flot réalisable" f sur $R = (X,\tilde{U},b,c)$, et pour toute coupe C séparant P de S, on a, sachant que C est définie par $Y \subset X$,

$$f(u_r) \leq c(\Omega^+(Y)) - b(\bar{\Omega}^-(Y)).$$

Dem3 : On a $\forall x \in X \quad \sum_{u \in \Omega^+(x)} f(u) - \sum_{u \in \Omega^-(x)} f(u) = 0$ on en déduit aisément :

$\sum_{u \in \Omega^+(Y)} f(u) - \sum_{u \in \Omega^-(Y)} f(u) = 0$. Puisque Y contient S et ne contient pas P, $u_r \in \Omega^-(Y)$, donc :

$$f(u_r) = \sum_{u \in \Omega^+(Y)} f(u) - \sum_{u \in \bar{\Omega}^-(Y)} f(u).$$

f est réalisable, donc :

$$\sum_{u \in \Omega^+(Y)} f(u) \leq \sum_{u \in \Omega^+(Y)} c(u)$$

$$- \sum_{u \in \bar{\Omega}^-(Y)} f(u) \leq - \sum_{u \in \bar{\Omega}^-(Y)} f(u)$$

d'où le résultat annoncé. La proposition 1 est un cas particulier de ceci pour $b(u) = 0$.

Propo4 : La valeur maximum de $f(u_r)$ pour un réseau réalisable sur $R = (X,U,b,c)$ est égale au minimum sur l'ensemble des parties Y de X contenant S et ne contenant pas P de la quantité $c(\Omega^+(Y)) - b(\bar{\Omega}^-(Y))$.

Dem4 : Le problème de la recherche d'un flot maximum s'écrit simplement comme un programme linéaire ayant $|\tilde{U}|$ variables, $2|\tilde{U}|$ contraintes de type inégalité et $|X|$ contraintes de type égalité (respectivement les variables $f(u)$ pour $u \in \tilde{U}$, les contraintes de type ii et celles de type i).

La proposition 4 n'est qu'une expression du théorème de la dualité connu en programmation linéaire.

Prop5 : (Théorème d'Hoffman). Une condition nécessaire et suffisante pour qu'il existe un flot réalisable sur $R = (X,\tilde{U},b,c)$ est que pour tout $Y \subset X$, on ait $\sum_{u \in \Omega^-(Y)} f(u) \leq \sum_{u \in \Omega^+(Y)} c(u)$

Dem5 : La condition nécessaire est montrée aisément en écrivant que :

$\sum_{u \in \Omega^-(Y)} f(u) = \sum_{u \in \Omega^+(Y)} f(u)$. Or $\sum_{u \in \Omega^-(Y)} f(u) \geq \sum_{u \in \Omega^-(Y)} b(u)$

et $\sum_{u \in \Omega^+(Y)} f(u) \leq \sum_{u \in \Omega^+(Y)} c(u)$.

Quant à la condition suffisante, elle est montrée en construisant un algorithme qui trouve un flot vérifiant i et ii à partir d'un flot vérifiant i.

Le théorème d'Hoffman est une expression particulière d'un théorème de programmation linéaire qui donne la condition d'existence d'une solution réalisable à un programme donné.

On trouvera à la figure 4 un exemple de réseau $\mathbb{R} = (X,\tilde{U},b,c)$ sur lequel on a défini un flot f vérifiant la proposition 4, donc optimal : on présentera sur chaque arc (b(u), f(u), c(u)).

Pour clore ce paragraphe, on va présenter dans la suite l'algorithme de Ford et Fulkerson destiné à trouver un flot maximal à partir d'un flot réalisable. Cet algorithme est pris dans [16].

"Agorithme Ford et Fulkerson"

Dans la description de cet algorithme on utilisera les notations suivantes :

$Y \subset X$; $\delta : Y \to \mathbb{R}^+$ $\quad$ $A : Y \to U$

Y représente l'ensemble des sommets (dits sommets marqués) qu'on peut atteindre par une chaîne C dite "chaîne augmentante". $\partial(x)$ est la valeur dont on peut augmenter le flot de s à $x \in Y$. A(x) est l'arc à partir duquel on a pu marquer le sommet x_0. L'algorithme, tel qu'il est décrit ici permet de trouver une chaîne augmentante de S à P si une telle chaîne existe. Pour construire effectivement une solution optimale du problème du flot maximum il convient d'appliquer cet algorithme de manière répétitive c'est-à-dire en fait de remplacer l'instruction "terminer" en fin de pas 3 par "Aller en 1".

1. Poser $Y := \{\partial\}$; $\partial(s) = \infty$

- Si $p \in Y$, poser $x = p$ $\quad$ $C^+ = \{u_r\}$ $\quad$ $C^- = \phi$ aller en 2
- Si $p \notin Y$ on essaie de marquer un sommet hors de Y (c'est-à-dire d'augmenter Y par l'un des procédés suivants :

a) Processus de marquage direct : il existe un arc $u = (xy)$ avec $x \in Y$, $\quad y \notin Y$ et tel que $f(u) < c(u)$.

Poser $Y := Y \cup \{y\}$; $\partial(y) = \min(\partial x, c(u) - f(u))$; $A(y) = u$; allera 1.

b) Processus de marquage indirect : il existe un arc $u = (yx) = u_r$ avec $x \in Y$, $y \notin Y$ et tel que $f(u) > b(u)$

Poser $Y := Y \cup \{y\}$; $\partial(y) = \min(\partial u$; $f(u) = b(u))$; $A(y) := u$; allera 1.

S'il est impossible de marquer un nouveau sommet, c'est-à-dire si on ne peut appliquer ni le processus de marquage direct ni le processus de marquage inverse, terminer. Le flot actuel est optimal.

2. Si x = s aller en 3.

Si $x \neq s$ soit $u = A(x)$: Si $x = T(u)$, poser $C^+ := C^+ \cup \{u\}$; $x := J(u)$ allera 2

Si $x = J(u)$ poser $C^- := C^- \cup \{u\}$ $x := T(u)$ allera 2.

3. Poser $\varepsilon = \partial(P)$. On définit le nouveau flot par $f(u) :=$

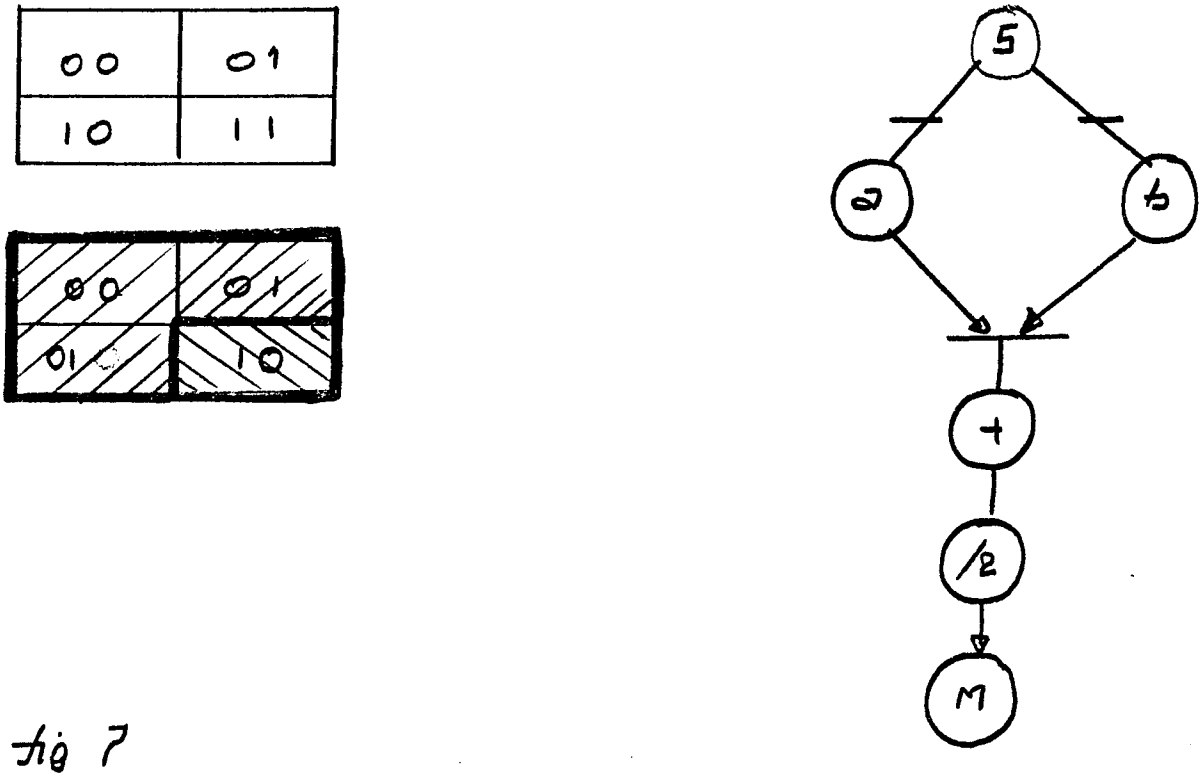
$f(u)+\varepsilon$ si $u \in C^+$, $f(u) - \varepsilon$ si $u \in C^-$, $f(u)$ sinon.

Poser $Y := \{\alpha\}$; allera 1".

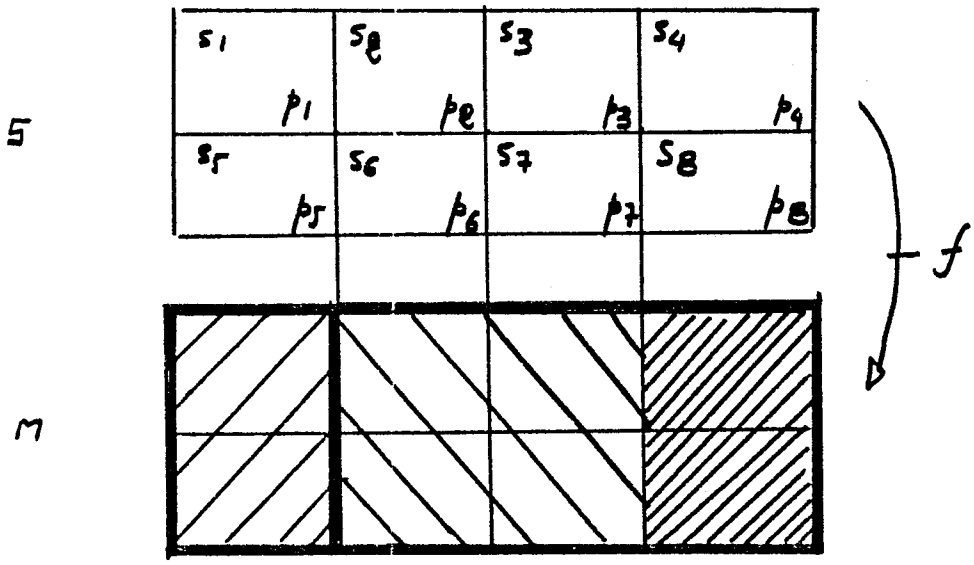
C - FLOT D'INFORMATION -

C1 - Introduction : Soit un additionneur 1 bit - 1 bit dont la sortie est sur deux bits. Il est connecté à un diviseur par deux. Un observateur s'intéresse à cette sortie. On trouvera le réseau de Pétri de ce système à la figure 5. On notera S et M la source et le puits de ce réseau. Supposons que les quatre valeurs de l'entrée sont équiprobables ; alors chaque réalisation de S émet une quantité d'information égale à 2 bits, alors que l'observateur reçoit une quantité d'information plus petite que 1 bit car la sortie du réseau est sur un digit binaire.En effet, ce qui intéresse l'observateur n'est pas de savoir si S vaut 00, 01, 10 ou 11 mais seulement si $S \in \{00, 01, 10\}$ ou $S \in \{11\}$. La quantité d'information reçue par l'observateur est donc $\frac{3}{4} \text{ Log } \frac{4}{3} + \frac{1}{4} \text{ Log } 2 = \frac{1}{2} + \frac{3}{4} \text{ Log } \frac{4}{3}$.

On dira que la quantité d'information émise par S vaut 2 alors que le "flot d'information" émis de S vers M à travers le réseau vaut $\frac{3}{4} \text{ Log } \frac{4}{3} + \frac{1}{2}$.


00
01
10
11
00
01
10
5
a
b
+
/2
M

Фиг 7


5
M
s1
p1
s2
p2
s3
p3
s4
p4
s5
p5
s6
p6
s7
p7
s8
p8
f

Фиг 8.

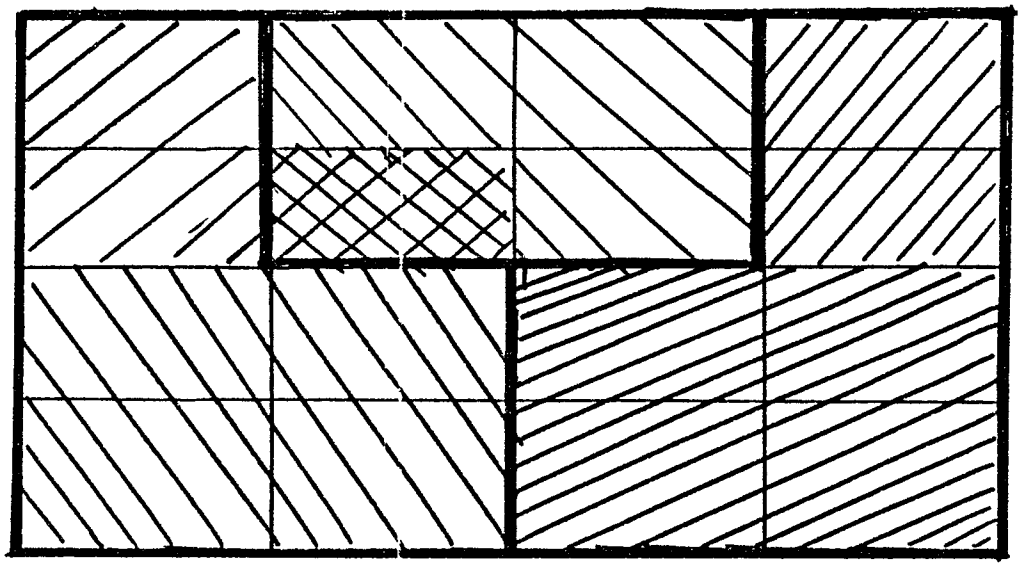
Remarque 2 : Si la quantité d'information émise par la source est intrinsèque à la source, car elle est déterminée par sa distribution de probabilité, le "flot d'information" émis de S vers M mesure l'entropie des significations que S émet vers M par delà la forme de ses messages et dépend bien entendu du moyen de communication qui sépare S de M ainsi que de la signification que M accorde à chacun des messages qu'il reçoit.

C2 - **Définition** : Soit R = (X,U) un réseau réalisant une application f entre les valeurs que prend S, la source, et celles que prend M, le puits. On se propose de définir une fonction notée w et dite "flot d'information".

Soit P_S une distribution de probabilité définie sur S et Γ_M une relation d'équivalence définie sur M. On définit sur M la distribution de probabilité par $P_M(m_i) = 0$ si $f^{-1}(\{m_i\}) = \phi$ ou = la somme des $P_S(s_i)$ sur l'ensemble des s_i vérifiant $m_j = f(s_i)$. On définit sur M/Γ_M, l'ensemble des classes d'équivalence de M modulo Γ_M, La distribution de probabilité Π_M qui associe à chaque élément de M/Γ_M la somme des probabilités - par P_M - des éléments de M qui le composent. Alors :

Def5 : Le "flot d'information" émis de S vers M pour la distribution P_S sur S et la partition Γ_M sur M se note $W_{P_S\Gamma_M}(S,M)$ et vaut $H(\Pi_M) = H(M/\Gamma_M)$.

Remarque 3 : Γ_M est la relation d'équivalence sur M qui met dans la même classe les éléments de M qui ont même signification. L'introduction de Γ_M n'est pas indispensable. Simplement elle permet une généralisation du modèle. En effet, si M accorde une signification différente à chacun de ses éléments, on prendrait Γ_M = Identité.



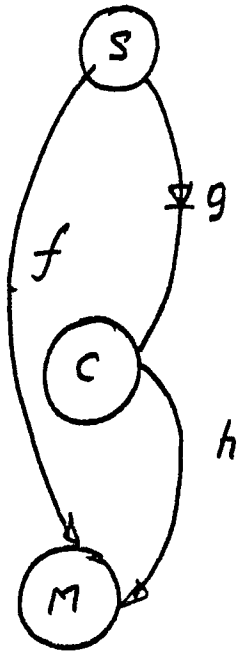
5.

Soit ϱ la relation $n_i \varrho n_j \Leftrightarrow f(n_i) = f(n_j)$

σ_5 la relation $n_i \sigma_5 n_j \Leftrightarrow f(n_i) \sigma_M f(n_j)$.

alors $|S| = 16$ $|S/\varrho| = 6$ $|S/\sigma_5| = 3$.

fig 9.



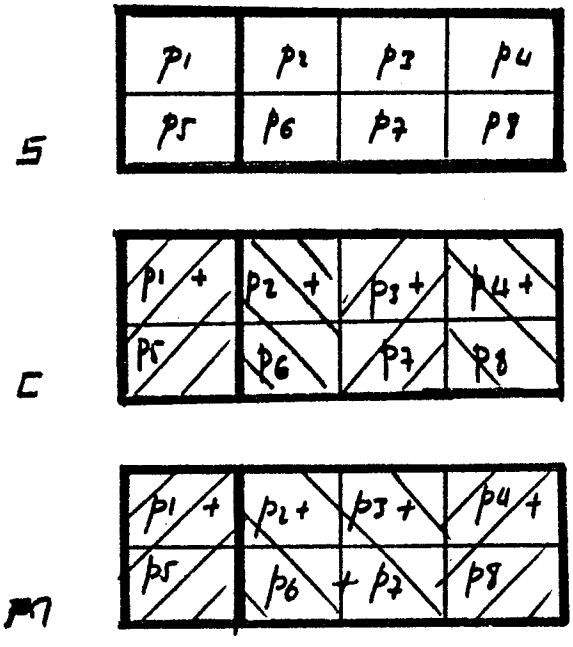


figure 10.

On calcule $W_{P_S \Gamma_M}(SM)$ pour l'exemple donné à la figure 8.

Trois valeurs de M seulement sont commandables à partir de S :

Il s'agit de $M_1 = f(s_1) = f(s_5)$ $M_2 = f(s_2) = f(s_3) = f(s_6) = f(s_7)$

et $M_3 = f(\partial_4) = f(\partial_8)$. Les autres valeurs de M ne sont pas représentées sur le schéma. P_M est définie par :

$P_M(M_1) = p_1 + p_5$ $P_M(M_2) = p_2 + p_3 + p_6 + p_7$ $P_M(M_3) = p_4 + p_8$

$P_M(M_i) = 0$ pour $i > 3$.

Γ_M partitionne M en $\{M_1\}$ et $\{M_2\ M_3\}$, pour ne citer que les classes d'équivalence qui contiennent des éléments commandables.

Le vecteur Π_M a donc deux composantes non nulles $\Pi_{M_1} = P_M(M_1)$ et

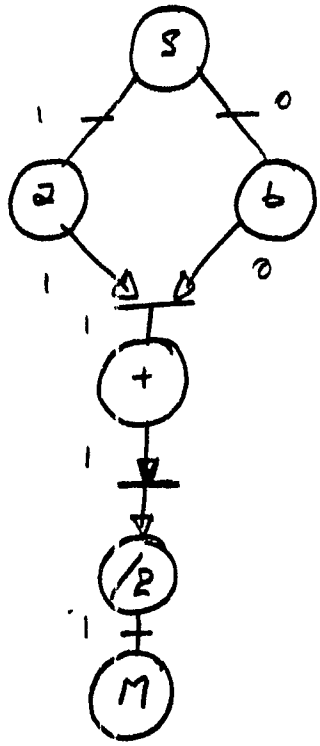
$\Pi_{M_2} = P_M(M_2) + P_M(M_3)$. Alors $W_{\Gamma_M P_S}(S,P) = - \Pi_{M_1} \text{Log } \Pi_{M_1} - \Pi_{M_2} \text{Log } \Pi_{M_2}$

$= -(p_1 + p_5) \text{Log}(p_1 + p_5) - (1 - p_1 - p_5) \text{Log}(1 - p_1 - p_5)$.

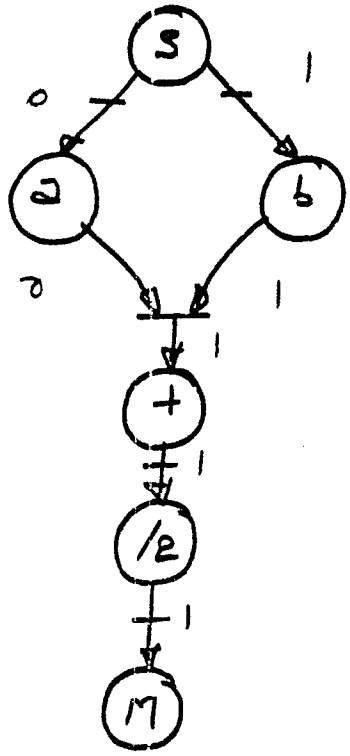
Prop6 : Le flot d'information émis d'une source S vers un récepteur M est majoré par l'entropie de cette source.

Dem6 : Le flot d'information émis de S vers M vaut $H(M/\Gamma_M)$. L'entropie de S vaut H(S). Vu que toute réalisation de S détermine la réalisation qui a lieu en M/Γ_M, on a $H(S) \geq H(M/\Gamma_M)$.

Remarque 4 : Si on appelle f la fonction de S vers M qui à toute réalisation de S associe la réalisation qui en résulte en M et g_{Γ_M} l'application de M dans M/Γ_M qui à toute valeur de M associe sa classe d'équivalence modulo Γ_M, alors $M/\Gamma_M = g_{\Gamma_M} \circ f(S)$.



a



b.

fig 11

On considère une variable aléatoire C telle que C = g(s) et M = h(C). On a alors f = hog. A l'égard de M, C se somporte comme une source donc on peut calculer le flot d'information que C émet à M pour une distribution de probabilité donnée sur C, en l'occurence celle déduite de P_S par $P_C(c_i)$ = si $g^{-1}(\{c_i\}) = \phi$ alors 0 sinon la somme des $P_S(s_j)$ pour $s_j \in g^{-1}(\{c_i\})$. On a alors :

Prop7 : $W_{P_S \Gamma_M}(S,M) = W_{P_C \Gamma_M}(C,M)$

Dem7 : On a $M/\Gamma_M = g_{\Gamma_M}$ o h o g(S)

$$M/\Gamma_M = g_{\Gamma_M} \text{ o } h(C)$$

Or, d'une part, on a C = g(S).

d'autre part P_S et P_C sont construites de sorte à induire la même distribution P_M en M donc :

$$W_{P_S \Gamma_M}(S,M) = H(M/\Gamma_M) = W_{P_C} = \Gamma_M(C,M).$$

Remarque 5 : Interprétation : considérons le cas du réseau présenté figure 10 et supposons que les huit messages de la source soient équiprobables ; alors H(S) = Log8 = 3 bits ; H(C) = Log_2 (4) = 2 bits ;

$$H(M) = \frac{1}{4} \text{Log } 4 + \frac{1}{2} \text{Log } 2 + \frac{1}{4} \text{Log } 4 = \frac{3}{2} .$$

alors que $W_{P_S \Gamma_M}(S,M) = W_{P_C \Gamma_M}(C,M) = W_{P_M \Gamma_M}(M,M) = \frac{1}{4} \text{Log } 4 + \frac{3}{4} \text{Log } \frac{4}{3}$.

L'entropie a changé entre SC et M alors que le flot d'information est resté constant. En effet, si la forme du message change en S,C et M, la signification qu'il porte à M se conserve.

C3 - Flot d'information et flot : Considérons l'exemple du réseau introduit au paragraphe C1 dans lequel on aurait fixé b à 1, par exemple. On suppose qu'alors $p_S(a = 0) = p_S(a = 1) = \frac{1}{2}$. Soit Y_a l'arc sortant de la place a, b étant fixé, M est fonction certaine de Y_a; donc on peut calculer le flot émis depuis la source Y_a vers le puits M. On trouve 1 bit. Vu que M est fonction certaine de tous les autres arcs du réseau sauf ceux entrant et sortant de b, on peut calculer le "flot d'information" émis depuis ces arcs vers M. La proposition 7 nous permet de conclure rapidement que ce flot vaut 1 pour tous les arcs.Si on fixe a à 1, on aura un schéma symétrique. La figure 11 représente les deux réseaux dans lesquels on a fixé tour à tour b puis a. Dans chacun des cas on place à côté de chaque arc le flot d'information émis depuis cet arc jusqu'au puits M. Ceci suggère l'idée de flot d'un liquide à travers un réseau de canalisations. En effet, fixer b correspondrait à fermer le "robinet" b auquel cas tout le flot passe par la branche a.

D'autre part considérons une coupe C séparant S de M dans un réseau de Pétri. M est fonction de la valeur affichée sur les arcs de C - par définition d'une coupe - donc on peut calculer le flot d'information que C émet vers M. D'après la proposition 7, ce flot d'information est égal à celui émis depuis S, donc à celui émis depuis n'importe quelle coupe : Or ceci est une idée force en optimisation dans les réseaux (et découle aisément de la première loi de kirchoff : on considère deux coupes C_1 et C_2 définies par $Y_1 \subset X$ et $Y_2 \subset X$ alors :

Etant donné un flot f sur le réseau, on a, d'après Kirchoff :

$$\sum_{\Omega^-(Y_1)} f(u) = \sum_{\Omega^+(Y_1)} f(u) \quad \text{et} \quad \sum_{\Omega^-(Y_2)} f(u) = \sum_{\Omega^+(Y_2)} f(u). \text{ Or } \Omega^+(Y_1) = C_1$$

$$\Omega^+(Y_2) = C_2 \quad \Omega^-(Y_1) = \Omega^-(Y_2) = \{u_r\} \quad \text{donc } \sum_{C_1} f(u) = f(u_r) = \sum_{C_2} f(u)).$$

Ceci suggère que la fonction "flot d'information" se comporte sur un réseau comme un flot dans le sens défini au paragraphe B. Ce rapprochement trouve une justification concrète dans l'interprétation suivante : on a vu que l'entropie d'une source est plus grande que le flot d'information que cette source peut émettre vers le puits M. Ceci est valable pour toutes les sources intermédiaires, i.e. les coupes séparant S de M. La différence entre le flot d'information et l'entropie au niveau de chaque coupe mesurerait la redondance du message sous la forme qu'il a au niveau de cette coupe. Par analogie à un réseau de canalisation, l'entropie de chaque coupe s'identifie à la capacité de cette coupe, le "flot d'information" qui y transite et qui mesure la quantité de signification donc de bien s'apparente au liquide donc au bien qui traverse cette coupe, enfin la redondance s'identifie au vide dans les canalisations qui résulte de la différence entre la capacité de la coupe et le flot qui la traverse.

Nous allons confirmer cette présomption en douant le réseau d'une capacité et en mettant en évidence d'autres analogies entre flot et flot d'information.

Soit un réseau de Pétri représentant la structure matérielle d'un système. $R = (X,U)$. On définit une fonction $\gamma : U \to \mathbb{N}$ qui à tout arc $u \in U$ associe le nombre de fils binaires de cet arc, à l'exclusion des fils redondants de parité, de CRC ou autres ... γ sera dite la fonction capacité du réseau. Alors :

Prop8: $W_{P_S \Gamma_M}(S,M) \leq \gamma(C)$, quelle que soit la coupe C du réseau, et quels que soient P_S et Γ_M.

Dem8 : On utilise les notations de la démonstration 7.

$$M/\Gamma_M = g_{\Gamma_M} h(C) \text{ donc } H(M/\Gamma_M) \leq H(C).$$

Or : $H(M/\Gamma_M) = W_{P_S \Gamma_M}$

et $H(C) \leq \log$(Nombre de vecteurs qu'il est possible d'afficher à la coupe C).

$$\leq \gamma(C)$$

c.q.f.d.

Soit $\tilde{P}_S$ une distribution de probabilité définie sur S et vérifiant :
soient $m_1 \ldots m_L$ les L éléments commandables en M à partir de S,

Alors, $\forall\, j \leq L \quad \sum_{s_i \in f^{-1}(\{m_j\})} \tilde{P}_S(s_i) = \frac{1}{L}$

(En d'autres termes la distribution $\tilde{P}_M$ déduite de $\tilde{P}_S$ est uniforme sur $m_1 \ldots m_L$ et nulle en dehors).

Soit $\tilde{\Gamma}_M$ la relation d'équivalence identité sur M. Alors :

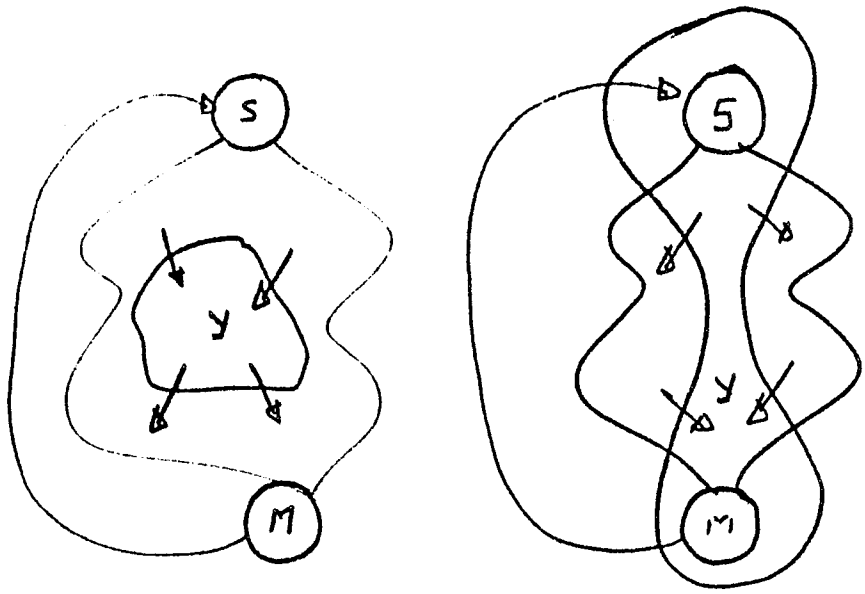
Prop9 : $\forall\, P_S$ défini sur S, $\forall\, \Gamma_M$ défini sur M, on a $W_{P_S\Gamma_M}(S,M) \leq W_{\tilde{P}_S\tilde{\Gamma}_M}(S,M)$

Démonstration 9 : $\forall\, P_S$, , $W_{P_S\Gamma_M}(S,M) \leq W_{P_S\tilde{\Gamma}_M}(S,M)$ puisque $\tilde{\Gamma}_M$ est la relation la plus fine qui soit sur M. D'autre part, $\forall\, \Gamma_M$, $W_{P_S\Gamma_M}(S,M) \leq W_{P_S^+\,\Gamma_M}(S,M)$ où P_S^+ est une probabilité définie sur S qui "uniformise" les éléments de M/Γ_M.

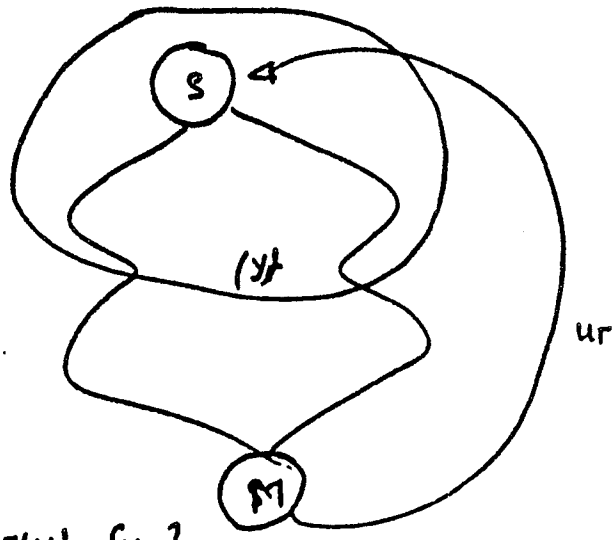
Remarque 6 : Soit à calculer $W_{\tilde{P}_S\tilde{\Gamma}_M}(S,M)$.

La distribution $\tilde{P}_M$ définie sur M à partir de $\tilde{P}_S$ associe $\frac{1}{L}$ aux éléments accessibles de M et 0 aux autres. $\tilde{\Gamma}_M$ étant l'identité , on a
$\tilde{\Pi}_M(\{m_i\}) = \tilde{P}_M(m_i) = \frac{1}{L} \Longrightarrow W_{\tilde{P}_S\tilde{\Gamma}_M}(S,M) = H(\tilde{\Pi}_M) = \text{Log } L.$

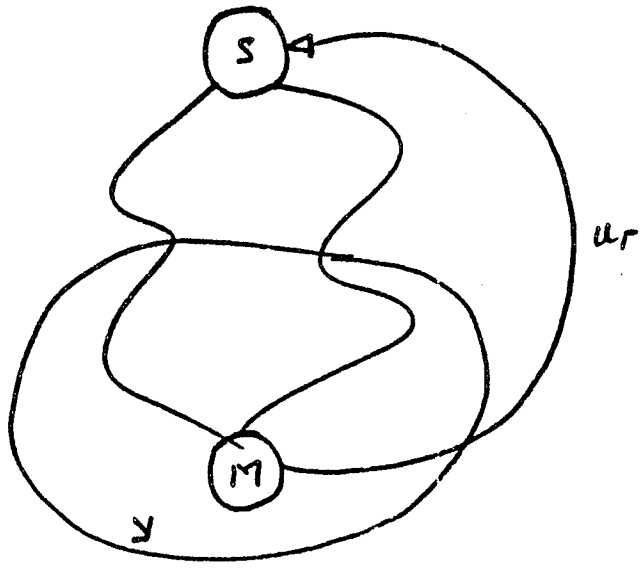
Vu qu'il n'y a que L valeurs qu'il est possible de faire afficher en M sachant qu'on n'agit que sur S, on a $H(M) \leq \text{Log } L$. Par ailleurs, si on émet des messages en S avec la distribution de probabilité $\tilde{P}_S$, on va



dans les deux cas $\Omega^-(y) \subset U$; $\Omega^+(y) \subset U$.



$\Omega^+(y) \subset U$; $\Omega^-(y) = \{ur\}$



$\Omega^+(y) = ur$; $\Omega^-(y) \subset U$

Ajoj 12.

trouver $\tilde{H}(M) = \text{Log } L \Longrightarrow$ donc $\text{Log } L = \max\limits_{P_S} H(M) = \text{Max}\limits_{P_S} H(M) - H(M/S)$,

puisque $H(M/S) = 0$ car $M = f(S)$.

$\text{Log } L = \max\limits_{P_S} T(M,S) = C(M,S)$ d'après les notations du paragraphe A.

Ainsi $W_{P_S \tilde{\Gamma}_M}(S,M)$ = la capacité du canal d'information qu'est le réseau =

Log (nombre de valeurs commandables en M à partir de S).

Des propositions 8 et 9 on déduit la proposition 10 :

Prop10 : $W_{P_S \tilde{\Gamma}_M}(S,M) = \max\limits_{P_S \Gamma_M} W_{P_S \Gamma_M}(S,M) \leq \text{Min}\limits_{C} \gamma(c)$

Remarque 7 : L'inégalité de la proposition 10 suggère une formule d'optimisation dans les réseaux : $\max\limits_{f} f(u_r) \leq \text{Min}\limits_{C} \gamma(c)$

deux questions viennent à l'esprit : Peut-on associer un flot à tout flot d'information (défini par P_S et Γ_M) et un flot d'information à tout flot ? La proposition 11 répondra oui à la première et la proposition 14 dira sous quelles conditions on peut répondre oui à la deuxième.

Prop11 : P_S et Γ_M étant donnés, il existe un flot f sur le réseau tel que $f(MS) = W_{P_S \Gamma_M}(S,M)$.

Dem11 : On associe le réseau $R = (X,U)$ sur lequel on définit la fonction capacité γ et auquel on adjoint l'arc de retour u_r de M vers S. On pose $\hat{U} = U \cup \{u_r\}$ et $\gamma(u_r) = W_{P_S \Gamma_M}(S,M)$. En outre, on définit une fonction $\beta : \tilde{U} \to \mathbb{R}$ tq $\beta(u) = 0$ si $u \in U$, $\beta(u_r) = W_{P_S \Gamma_M}(S,M)$. Alors s'il existe un flot réalisable sur $R = (X \tilde{U} \beta\gamma)$, ce flot satisfait à la condition imposée.

Considérons le théorème d'Hoffman donné au paragraphe B, soit Y un sous ensemble de X.

- Si Y contient S et M ou ne contient ni S ni M alors $u_r \phi$ ($\Omega^-(Y)$, et $\Omega^+(Y)$) donc l'inégalité du Théorème d'Hoffman est automatiquement satisfaite car γ est nul sur U alors que β est positif. Y contient S et ne contient pas M alors $\Omega^-(Y) = \{u_r\}$ et $\Omega^+(Y)$ est une coupe du réseau séparant S deM.

Alors l'inégalité d'Hoffman n'est autre que la proposition 8. Enfin si Y contient M et ne contient pas S, $\Omega^+\{Y\} = \{u_r\}$ et $\Omega^-(Y) \subset U$ alors

$\gamma(\Omega^+(Y)) = W_{P_S \Gamma_M}(S,M)$: $\beta(\Omega^-(Y)) = 0$. $W_{P_S \Gamma_M}(S,M)$ étant une entropie,

elle est positive. c.q.f.d. - voir figure 12.

Remarque 8 : La réciproque de la proposition 11 est fausse : prenons l'exemple présenté à la figure 13 : il s'agit du circuit réalisant la fonction $x \to x + \bar{x}$. Sur le réseau de Pétri de ce circuit on présentera un flot réalisable - en l'occurence le flot optimal obtenu en le douant de sa capacité γ.

Or $\forall P_S$, on a toujours $P_M(1) = 1$ et $P_M(0) = 0$ donc $\forall \Gamma_M$ défini sur M, on a $W_{P_S \Gamma_M}(S,M) = 0$; ce qui s'explique par le fait qu'aucune information n'arrive à M. Voilà donc un cas où on a pris un flot f sur le réseau et qu'on n'a pas pu lui associer de flot d'information (en exhibant P_S et Γ_M) qui lui soit égal.

Deuxième exemple : on considère le schéma logique de la figure 14, et son réseau de Pétri (la place Id réalise la fonction identité et a été adjointe afin de satisfaire la contrainte que toute place n'est adjacente qu'à des transitions et réciproquement). Sur chaque arc u on portera f(u)/ c(u) où f est le flot optimal. On a trouvé $f(u_r) = 2$ alors que les seules valeurs affichables en M à partir de S sont 00 et 10, donc $\forall P_S$ et $\forall \Gamma_M$, on a $W_{P_S \Gamma_M}(S,M) \leq 1$ bit.

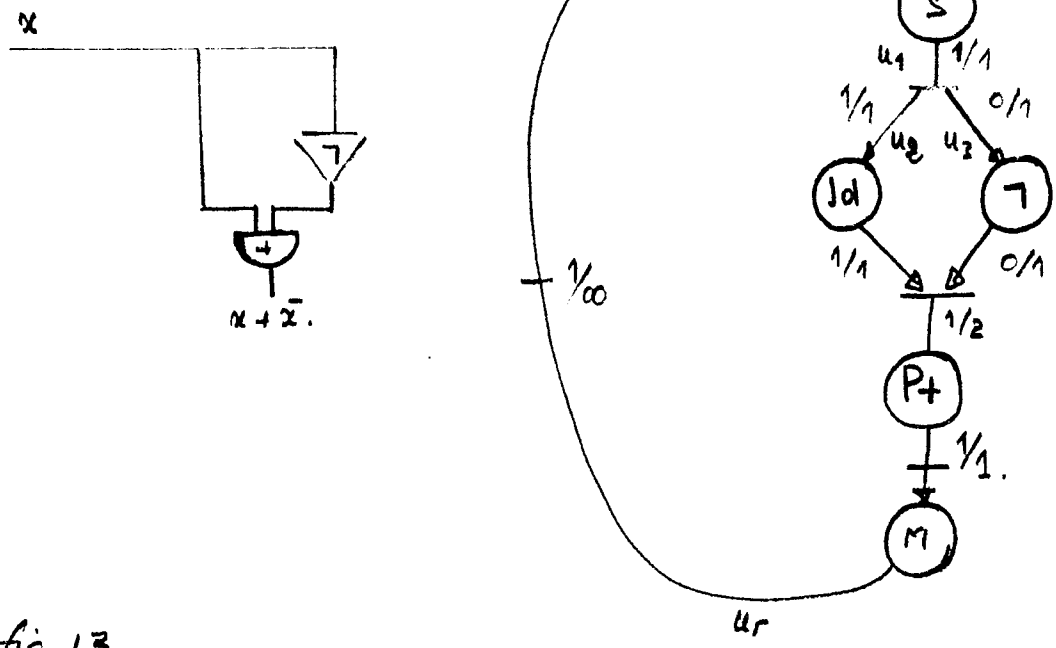

x
x + x̄.
S
u1
1/1
1/1
0/1
u2
u3
Id
T
1/1
0/1
1/∞
1/2
P+
1/1.
M
ur

fig 13

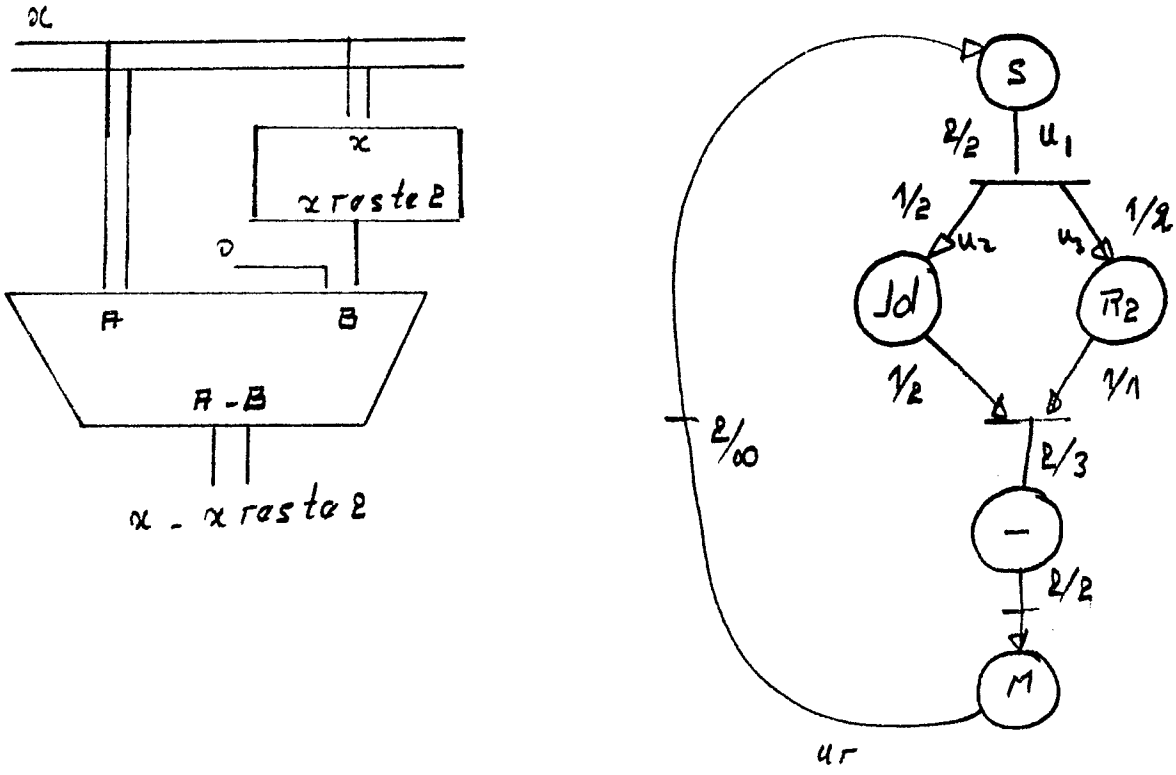

x
x
x resta 2
0
A
B
A - B
x - x resta 2
S
2/2
u1
1/2
1/2
u2
u3
Id
R2
1/2
1/1
2/∞
2/3
—
2/2
M
ur

fig 14.

Nous allons énoncer dans la suite quatre conditions et nous allons montrer que, sous ces conditions, à tout flot, on peut associer une distribution de probabilité sur S et une équivalence Γ_M sur M telles que $W_{\Gamma_M P_S}(S,M) = f(u_r)$. Nous commenterons ensuite ces quatre conditions une à une et montrerons qu'elles ne diminuent que très peu la valeur du résultat.

Soit un réseau de Pétri vérifiant :

C1 - Toute place n'a qu'un père et un fils

C2 - dans toute configuration α - voir figure 15 - les arcs u_2 et u_3 n'ont pas de fils communs - (toute valeur de (u_2, u_3) est commandable en u_1).

C3 - Dans toute configuration α' - voir figure 15 - tout vecteur de u_1 peut être obtenu en combinant un vecteur de u_2 avec un vecteur de u_3.

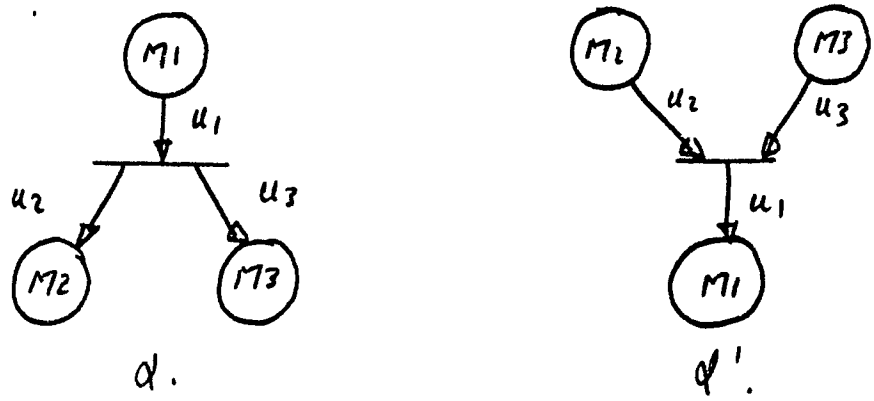
C4 - Etant donné une place P dont l'arc sortant est u_i on peut afficher $2^{\gamma(u)}$ vecteurs distincts en sortie de P en accédant à son entrée.

Remarque 9 : C1 peut être fausse dans le cas où la sortie de la place P est prévue pour être communiquée à plusieurs autres places.

C2 peut être fausse dans le cas où certains digits de u, sont communs aux deux arcs u_2 et u_3.

C3 peut être fausse dans le cas où les places qui donnent les résultats sur les arcs u_2 et u_3 sont mutuellement exclusives.

C4 peut être fausse dans le cas, par exemple, d'un additionneur numérique 1 bit 1 bit dont la sortie est sur 2 bits. Pourtant il n'y a que trois valeurs possibles en sortie 100 01 10) et non $4 = 2^2$. voir figure 16 des cas où C1 C2 et C3 sont fausses.



a. a'.

Fig 15.

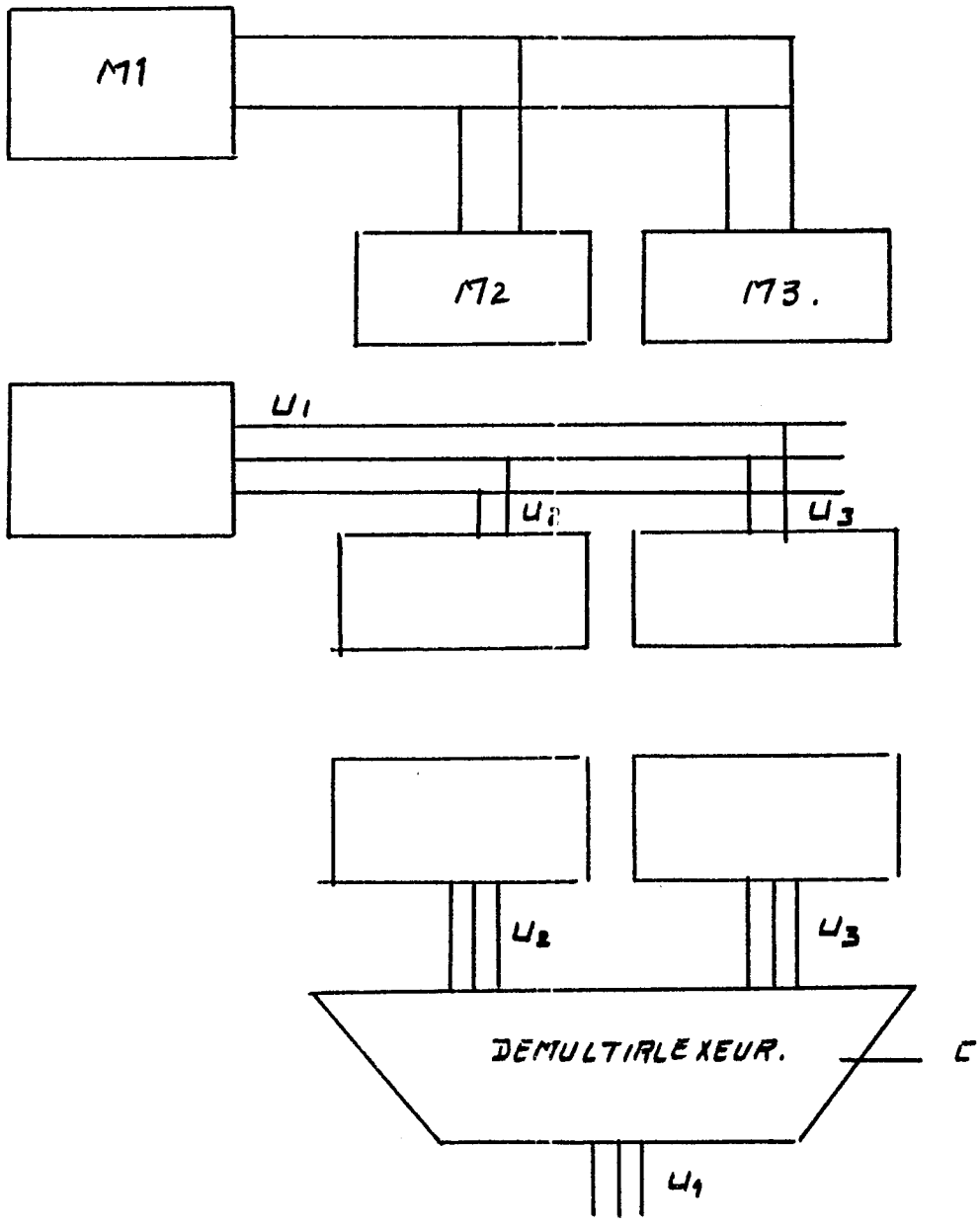


Fig 16.

Sur un réseau vérifiant C1 C2 C3 et C4, la fonction vérifie :

- dans le cas des configurations α et α', on a $\gamma(u_1) = \gamma(u_2) + \gamma(u_3)$.
- soit une place où entre un arc u_1 et dont sort un arc u_2. Alors $\gamma(u_2) \leq \gamma(u_1)$.

Prop12 : Soit un réseau vérifiant C1 C2 C3 et C4 ; et soit ϕ un entier compris entre 0 et $\min_C \gamma$ (C). Alors il existe 2^ϕ vecteurs distincts tels que si on les affiche en S, on obtient 2^ϕ vecteurs distincts en M.

Dem12 : On montrera ce résultat en deux temps :

1. La fonction M = f(S) est surjective
2. La coupe du réseau définie par l'arc X_M entrant dans M est minimale.

1 ==> On peut commander $2^{\gamma(X_M)}$ vecteurs distincts en M à partir de S.

2 ==> $\phi \leq \gamma(X_M)$

1 et 2 ==> On peut commander 2^ϕ vecteurs distincts en M à partir de S.

On suppose que chaque sommet x du réseau réalise une fonction C entre l'ensemble des valeurs affichables sur les arcs $\Omega^-(\{x\})$ et l'ensemble des valeurs affichables sur les arcs de $\Omega^+(\{x\})$. Ces deux ensembles ont pour cardinaux respectifs $2^{\gamma(\Omega^-\{x\})}$ et $2^{\gamma(\Omega^+\{x\})}$.

Les conditions C1 et C4 impliquent que la fonction réalisée par chaque place est surjective ==> $\gamma(\Omega^+\{p\}) \leq \gamma(\Omega^-\{p\})$ pour toute place p.

Les conditions C2 et C3 impliquent que la fonction réalisée par chaque transition est surjective ==> $\gamma(\Omega^+\{t\}) \leq \gamma(\Omega^-\{t\})$ pour toute transition t.

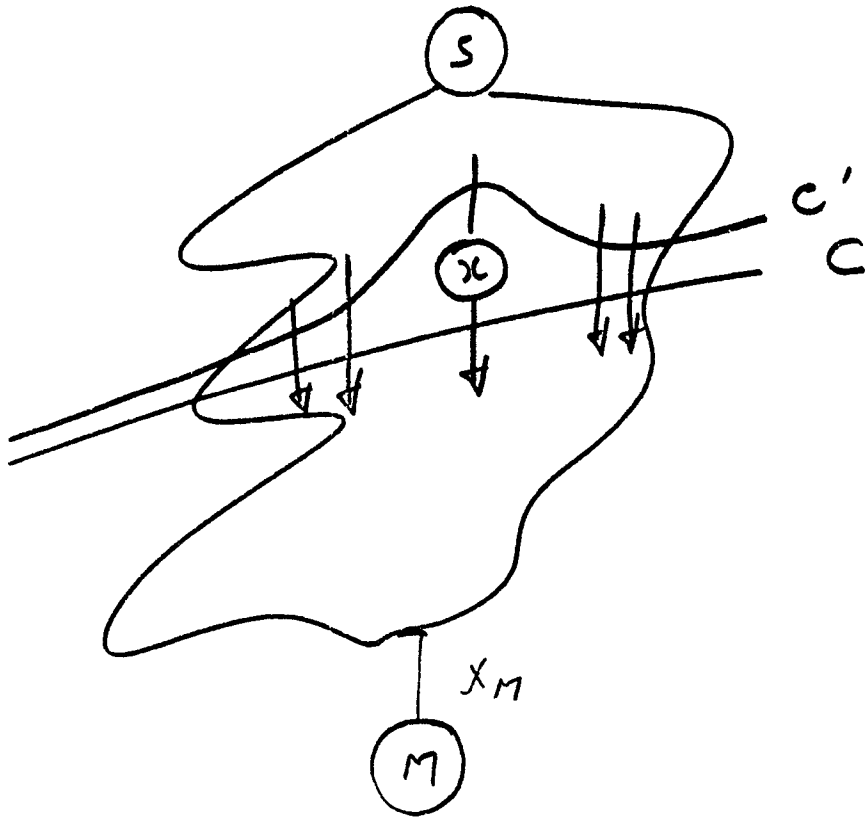


figure 17.

Nous allons montrer par récurrence sur les coupes que la fonction f_C définie par $M = f_C(C)$ (où C est une coupe) est surjective.

Ceci est vrai pour la coupe $\{X_M\}$

Supposons la vraie pour la coupe C définie par l'ensemble de sommets Y.
Soit x un élément de Y tel que $\Omega^+(\{x\}) \subset \Omega^+(Y)$ - voir figure 17.
On définit la coupe C' par $C' = \Omega^+(Y - \{x\})$ et on appelle $f_{C,C'}$ la fonction définie par $C = f_{C,C'}(C')$ alors on a

- $f_{C'} = f_{CC'} \cdot f_C$ et

- $f_{CC'}$ est surjective car elle est le produit cartésien de fonctions surjectives (la fonction f_x et autant de fois la fonction Identité qu'il n'y a d'arcs dans C' autres que $\Omega^+(\{x\})$.

Conclusions.

i) $f_{C'}$ est surjective

ii) $\gamma(C') \geq \gamma(C)$

On propage ainsi la coupe C depuis $\Omega^-(\{M\})$ jusqu'à $\Omega^+(\{S\})$: la propriété i montre 1 et la propriété ii montre 2.

Prop13 : Soit un réseau $R = (X,U,\gamma)$ vérifiant C1 C2 C3 et C4 et réalisant une fonction θ entre sa source S et son puits M et soit ϕ un réel compris entre 0 et $\min_C \gamma(C)$. Alors il existe P_S sur S et Γ_M sur M telles que

$\phi = W_{P_S\Gamma_M}(S,M)$.

Dem13 ; γ étant une fonction à valeurs dans $\mathbb{N}$, $\min_C \gamma(C) \in \mathbb{N}$ donc

si $\phi \leq \min_C \gamma(C)$, alors $\bar{\phi}$, le plus petit entier plus grand ou égal à ϕ vérifie aussi $\bar{\phi} \leq \min_C \gamma(C)$=> d'après le lemme ci-haut, il est possible

d'envoyer $2^{\bar{\phi}}$ vecteurs de S vers M.

On sait que, étant donné un ensemble E à M éléments, et un réel t compris entre 0 et Logn, il existe une distribution de probabilité sur E, $P = (p_1 \ldots p_n)$ telle que $H(P) = t$. Il est donc possible de définir une distribution de probabilité $P_M^* = (p_1 \ldots p_2^{\bar{\phi}})$ sur les $2^{\bar{\phi}}$ éléments de M commandables en M à partir de S, donc une distribution de probabilité $P_M = (p_1 \ldots p_{|M|})$ sur M qui vérifie $H(P_M) = \phi$. Il est aisé d'en déduire une distribution de probabilité P_S sur S qui vérifie :

$\sum_{s_i \in \Theta^{-1}(\{m_j\})} P_S(s_i) = P_M(m_j)$. De plus si on définit sur M la relation d'équivalence Γ_M = Identité, on voit que $\Pi_M = P_M$. Alors :

$W_{P_S \Gamma_M}(S,M) = H(\Pi_M) = H(P_M) = \phi$ c.q.f.d.

Remarque 10 : Considérons le réseau représenté figure 13. Au flot qui est défini sur ce réseau, on n'a pas su associer un flot d'information : c'est à cause de la transition qui se trouve en aval de S ; elle ne satisfait pas la condition C2 car u_3 porte la même valeur que u_2. Ainsi, si en agissant sur la coupe (u_2, u_3) on peut commander M tel que nous le dicte le flot f, c'est- à-dire en fixant u_3 et faisant prendre à u_2 ses deux valeurs 0 et 1, par contre, en agissant sur u_1, on ne peut plus agir comme nous l'indique f car $f(u_3) = 0$ nous impose de fixer u_1 alors que $f(u_1) = 1$ nous impose de faire varier u_1.

Il en est de même du réseau de la figure 14 : la transition qui se trouve en aval de S ne vérifie pas C2. Ainsi quand on fait prendre à u_1 ses $2^{f(u_1)} = 4$ valeurs, aussi bien u_2 que u_3 vont prendre 4 valeurs alors que leurs flots ne valent que 1.

Sous les hypothèses C1 C2 C3 et C4 on sait qu'à tout flot d'information on sait associer un flot et à tout flot on sait associer un flot d'information. De là on déduit le résultat-clé de ce paragraphe.

Prop14 : Sous les hypothèses C1 C2 C3 et C4, on a :

$$W_{\tilde{P}_S \tilde{\Gamma}_M}(S,M) = \tilde{f}(u_r)$$

où $\tilde{f}$ est le flot maximal sur $R = (X,\tilde{U},\gamma)$.

Dem14 : Soit f^* le flot associé au flot d'information $W_{\tilde{P}_S \tilde{\Gamma}_M}(S,M)$. On a

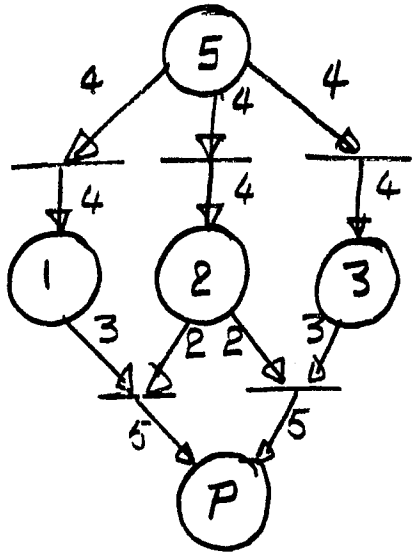
$W_{\tilde{P}_S \tilde{\Gamma}_M}(S,M) = f^*(u_r) \leq \tilde{f}(u_r)$ car $\tilde{f}$ est maximal.

Soit $W_{P^*_S \Gamma^*_M}(S,M)$ le flot d'information associé au flot $\tilde{f}$; on a :

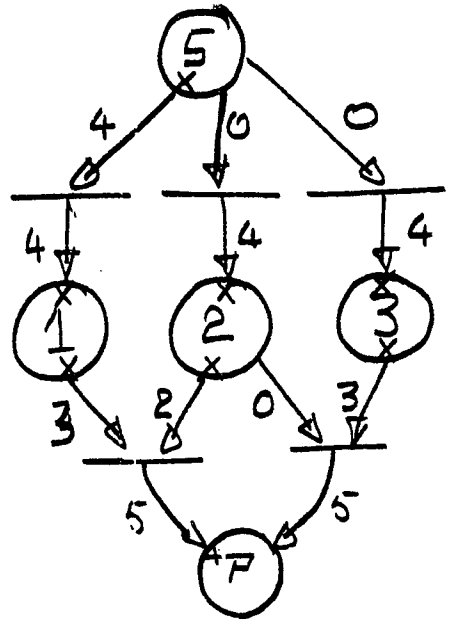
$\tilde{f}(u_r) = W_{P^*_S \Gamma^*_M}(S,M) \leq W_{\tilde{P}_S \tilde{\Gamma}_M}(S,M)$ car $W_{\tilde{P}_S \tilde{\Gamma}_M}(S,M)$ est maximal.

Remarque 11 : La proposition 14 veut donc dire que le "flot d'information" maximal qui peut être émis de S vesr M à travers un réseau, ou la capacité de ce réseau ou le Log(nombre de vecteurs que l'on peut émettre de S vers M à travers ce réseau) vaut, sous les conditions (C1 C2 C3 C4) = C la capacité de la coupe de capacité minimale dans le réseau $R = (X,U,\gamma)$.

Remarque 12 : Le résultat de la proposition 14 pourrait être interprété comme suit. Pour envoyer k messages de S vers M, il faut pouvoir coder k messages sur chacune des coupes intermédiaires entre S et M, donc le nombre maximum de messages que l'on peut émettre de S vers M vaut le nombre de messages que l'on peut coder sur la plus petite coupe, à savoir $2^{\min_C \gamma(C)}$.



(X, U, σ)



$(X, U, \sigma_{\pi\phi})$.

figure 18.

C4 - Flot d'information et écoulement -

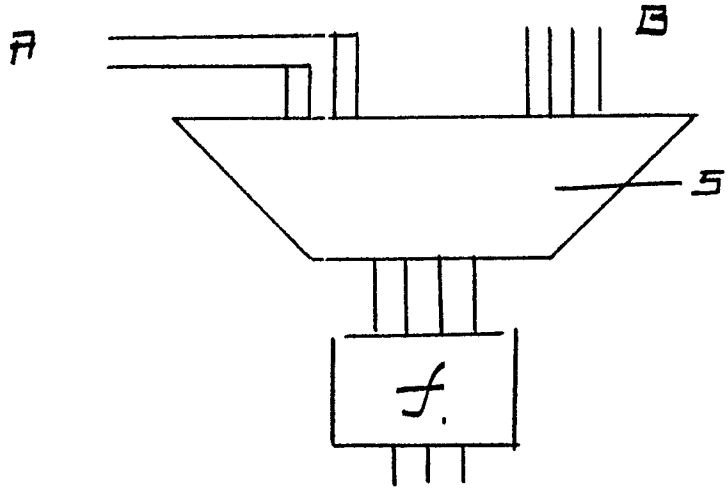
Le résultat de la proposition 14 est essentiellement destiné à être appliqué à des écoulements trouvés par AT, donc à des réseaux qui ne vérifient pas nécessairement les conditions C1 C2 C3 et C4. Il est possible de remédier à un réseau qui ne vérifie pas certaines de ces contraintes. On dira dans la suite ce qu'il y a lieu de faire pour les contraintes remédiables et ce qu'il y a lieu de penser de celles qui ne le sont pas.

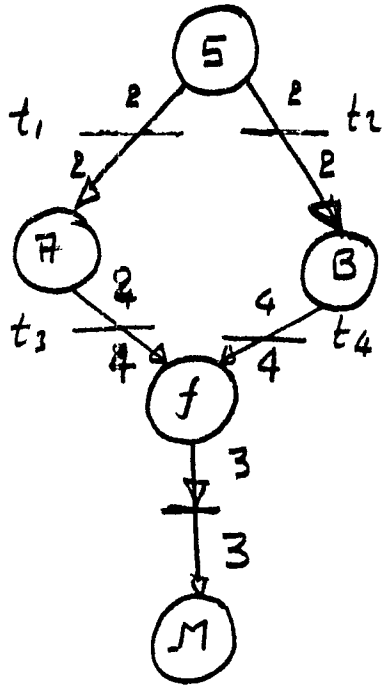
Condition C1 : Soit E un écoulement dans lequel des placès ont plus d'un père, ou plus d'un fils et soit (Π,ϕ) une application qui à toute place associe un père et un fils privilégiés. Toute valeur de la fonction (Π,ϕ) définit de manière biunivoque un chemin de données dans le réseau, i.e., un aiguillage des multiplexeurs et des démultiplexeurs décidé par le contrôle. Sur l'écoulement $E = (X,U)$ on définit la fonction capacité $\gamma_{(\Pi,\phi)}: U \to \mathbb{N}$ définie par $\gamma_{\Pi,\phi}(u)$ = Si J(u) est une place p et que $T(u) = \phi(p)$

où T(u) est une place p et que $Ju = \Pi(p)$ alors $\gamma(u)$
sinon 0.

Soit $f_{\Pi\phi}$ le flot maximal trouvé sur E muni de la capacité $\gamma_{\Pi\phi}$ et soit $E_{\Pi\phi}$ le réseau partiel construit sur l'ensemble des arcs de E tels que $f_{\Pi\phi}(u) > 0$.

$E_{\Pi\phi}$ vérifie la condition C1, par définition ==> sous les conditions C2 C3 C4 , et sachant que les multiplexeurs et les démultiplexeurs ont été commandés tel que le dicte (Π,ϕ), le Log (nombre de vecteurs que l'on peut commander en M à partir de S) vaut $f_{\Pi\phi}$. La question qui se pose est de savoir comment calculer f, le Log(nombre de vecteurs que l'on peut émettre de S vers M dans E) sachant que l'on connaît les $f_{\Pi\phi}$ pour toutes les valeurs possibles de (Π,ϕ). On ne peut répondre à cette question





On définit

$\eta_1(f) = t_3$

$\eta_2(f) = t_4$

$\phi_1(S) = t_1$

$\phi_2(S) = t_2$

$f_{\eta_1 \phi_1} = 2 \qquad f_{\eta_1 \phi_2} = 0 \qquad f_{\eta_2 \phi_1} = 0 \qquad f_{\eta_2 \phi_2} = 3$

$f = f_{\eta_2 \phi_2}$

figure 19.

si on n'a pas davantage d'information sur le réseau. La figure 19 représente un exemple de réseau où $f = \underset{\Pi\phi}{\text{Max}} \; f_{\Pi\phi}$.

La figure 20 représente un exemple de réseau où $f = \sum_{\Pi\phi} f_{\Pi\phi}$.
En général il serait peut être plus intéressant de connaître tous les $f_{\Pi\phi}$.

Condition C2 :

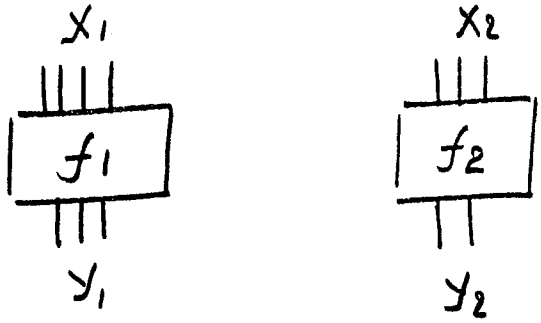
Il arrive assez souvent que des distributions ne satisfont pas la condition C2. Il arrive beaucoup moins souvent que des réseaux ayant des jonctions qui ne vérifient pas C2 donnent des résultats loins de la réalité quand on leur applique la proposition 14. voir les exemples donnés figures 13 et 14. Le regroupement de places peut être salutaire dans ces cas là : Par exemple le regroupement de toutes les places du réseau de la figure 14.

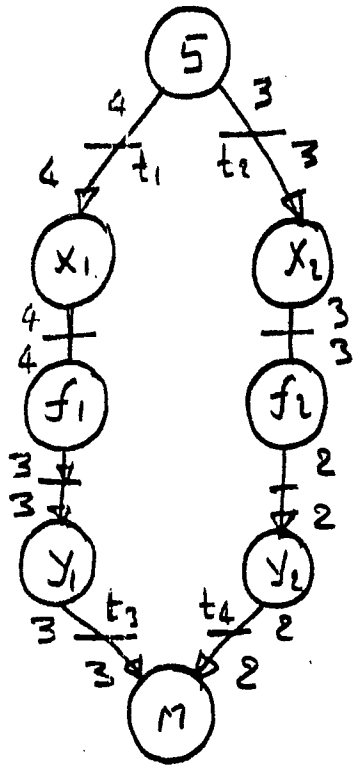
Condition C3 :

Il est extrêmement rare qu'un réseau admette des fonctions qui ne vérifient pas C3. On pourrait alors améliorer le résultat en affectant à U, la capacité $\gamma(u_2) + \gamma(u_3)$ - voir figure 15 a' - au lieu de $\gamma(u_1)$.

Condition C4 :

Il arrive que des arcs ne vérifient pas cette condition. On peut alors, pour remédier partiellement à cet inconvénient, remplacer la capacité de l'arc u sortant de P par le Log du nombre de valeurs distinctes qu'il est possible d'afficher en sortie de la place P en commandant toutes ses entrées. Il semble que, théoriquement, la capacité que l'on doit définir sur chaque arc ne soit pas être le nombre de fils de cet arc. Elle doit être définie à partir d'une distribution de probabilité P_S sur S par $\gamma(u) = $ l'entropie des messages qui arrivent en u à partir de S
P_S





On définit

$\eta_1(M) = t_3$

$\eta_2(\Pi) = t_4$

$\phi_1(S) = t_1$

$\phi_2(S) = t_2$.

$f_{\eta_1\phi_1} = 3 \qquad f_{\eta_1\phi_2} = 0 \qquad f_{\eta_2\phi_1} = 0 \qquad f_{\eta_2\phi} = 2$

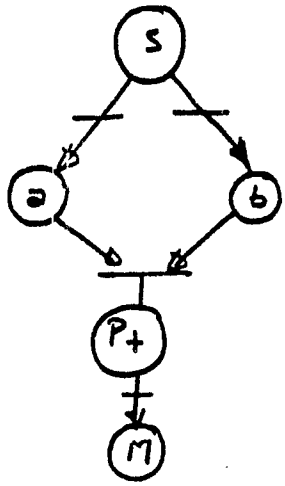
$$f = f_{\eta_1\phi_1} + f_{\eta_2\phi_2}.$$

figure 20.

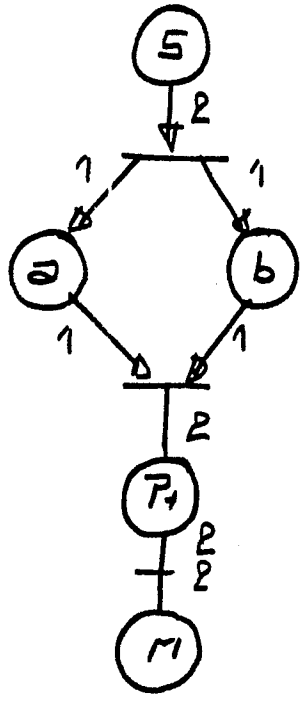
sachant que S émet avec la distribution de probabilité P_S.Dans un tel réseau un flot serait obtenu à partir d'une relation d'équivalence Γ_M et seulement à partir de Γ_M puisque P_S est fixé, et le flot maximal dans le réseau muni de sa capacité γ_{P_S} est obtenu par l'identité sur M, $\forall P_S$. Il est remarquable qu'il n'y a, pour chaque distribution P_S qu'un nombre fixé de "flots d'informations" réalisables (autant que de partitions de M). Dans un tel réseau la condition C4 n'est pas nécessaire pour la démonstration de la proposition 14 mais l'élaboration de γ_{P_S} nécessite que l'on connaisse les fonctions exécutées par chacune des places du réseau - chose que l'on s'est interdite par souci de simplicité. Si l'on ignore cette condition, le flot maximal trouvé sur le réseau est plus grand que le Log du nombre véritable de vecteurs distincts qui peuvent être affichés en M à partir de S. Toutefois, la différence excède rarement l'unité pour des réseaux que l'on peut trouver dans la "nature",même s'il existe plus d'une place qui ne vérifie pas C4 - voir figure 24 -

Conclusion : Dans la pratique, il faut remédier aux situations où C1 et C3 ne sont pas vérifiées par le procédé exposé en haut. Quant aux conditions C2 et C4, l'expérience montre fort heureusement que, si peu de systèmes les vérifient, elle montre aussi que beaucoup donnent un résultat très satisfaisant sans les vérifier. Ceci suggère certains résultats de recherche opérationnelle que l'on ne peut pas affirmer mais qui ne sont faux que dans des cas étudiés pour et qui ont une probabilité nulle d'être faux sur des cas tirés au hasard. Il se trouve qu'en outre, les exemples pour lesquels l'application de Proposition 14 donne un mauvais résultat sont assez visibles. voir l'exemple de la figure 13.

D. Coupe de capacité Maximale - Pour étudier le comportement d'un réseau face à l'acheminement de l'information entre sa source et son puits, nous serons amenés à nous intéresser au "profil informationnel" du réseau,



Le module P_+ réalise la somme algébrique de a et b. Le seul éventuel de ce réseau devient après transformation :



Le flot max dans ce réseau vaut 2 alors que le "flot d'information" max vaut $\log_2 3$. On peut remédier à cette situation en remplaçant la capacité de l'arc sortant de P_+ par log 3 au lieu de 2.

fig 21.

c'est-à-dire l'évolution de sa largeur à mesure qu'on évolue de S à P. voir figure 25. La notion de profil informationnel peut être définie rigoureusement comme suit :

DEF6 : On dit qu'un ensemble de sommets Y contenant S est semi connexe dans R si et seulement si quelque soit x dans Y, il existe dans R un chemin de s à x ayant tous ses sommets dans Y.

DEF7 : Soit Y un ensemble de sommets semiconnexe. On appelle γ coupe définie par Y l'ensemble d'arcs $\Omega^+(Y)$.

DEF8 : Le profil informationnel d'un réseau est la donnée de toutes les capacités de toutes les γ coupes.

Toutefois on s'intéressera à établir des profils informationnels de façon très sommaire sans faire le calcul des capacités de toutes les γ coupes. Pour ce faire, on sera amené à introduire la notion de coupe de capacité maximale dans un réseau. Ce problème ne consiste pas en la recherche de la coupe du réseau qui maximise la capacité sur toutes les coupes du réseau mais seulement sur un certain sous ensemble des coupes. Toute la difficulté réside dans la définition de ce sous ensemble.

Remarque 14 : - figure 26 - Considérons la coupe déterminée par Y = {5,1,2} . Elle contient deux arcs u_1 et u_2. Supposons que la capacité γ définie sur ce réseau soit telle que (u_1 u_2) est la γ coupe qui maximise la capacité sur l'ensemble des γ-coupes. L'information qui transite sur l'arc u_1 est la synthèse de l'information reçue de S et celle reçue de u_2 à travers la place 3. Ainsi dans γ coupe (u_1 u_2) l'information portée par u_2 est "comptée deux fois", une fois par u_2 et une autre fois par u_1 qui est fonction de u_2. Ceci ne correspond pas à l'idée intuitive que l'on se fait de la coupe max. On ne définira donc pas la coupe max comme étant celle qui maximise la capacité sur l'ensemble des γ-coupes, mais sur l'ensemble des déconnectants minimaux".

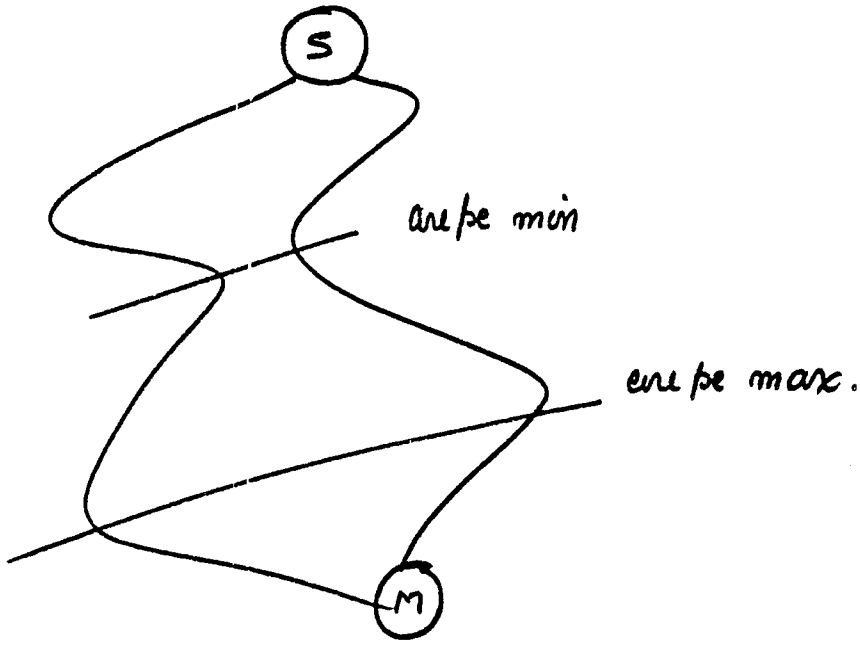


fig 24.

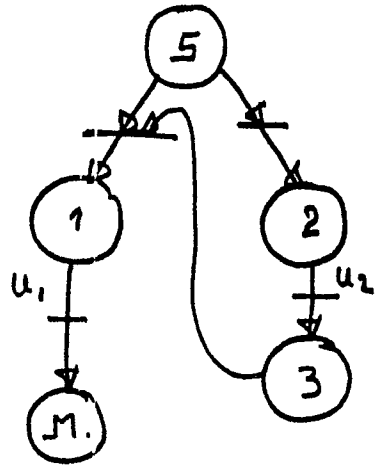


fig 25.

Def9 : On appelle déconnectant minimal dans un réseau un ensemble d'arcs tels que (si on les enlève du réseau aucun chemin ne va plus de S vers M) et qui est minimal pour cette propriété.

Soit un réseau R = (X,U) de source S et de puits M. On lui adjoint un arc de retour u_r = (MS) et on définit deux fonctions :

$\beta : U \to \mathbb{R} \cup \{-\infty\}$ $\quad \beta(u) = -\infty \quad \forall u \in U \cup \{u_r\} = \tilde{U}$

$\gamma : U \to \mathbb{R} \cup \{+\infty\}$ $\quad \gamma(u) =$ si $u \in U$ alors moins le nombre de fils binaires de l'arc u Sinon $(u = u_r)$ $+\infty$.

Prop15 : Il existe un flot réalisable sur $\tilde{R} = (X,\tilde{U},\beta,\gamma)$.

Dem15 : Ceci découle immédiatement du théorème d'Hoffmann et de la définition de β et γ.

Appliquons l'algorithme de Ford et Fulkerson à partir d'un flot réalisable (facile à trouver grâce à β). Soit $\bar{Y}$ l'ensemble des sommets de X cochés par la dernière itération de Ford et Fulkerson.

$\Omega^+(\bar{Y})$ est par définition un déconnectant du réseau - de plus,

Prop16 : $\Omega^+(\bar{Y})$ est un déconnectant minimal.

Dem16 : Soit $\bar{f}$ le flot max trouvé par Ford et Fulkerson.

$$\bar{f}(u_r) = \sum_{u \in \Omega^+(\bar{Y})} \gamma(u) - \sum_{u \in \Omega^-(\bar{Y})} \beta(u).$$

Par ailleurs $\bar{f}(u_r) \leq \sum_{u \in \Omega^+(\{S\})} \gamma(u)$ donc $\bar{f}(u_r)$ est fini.

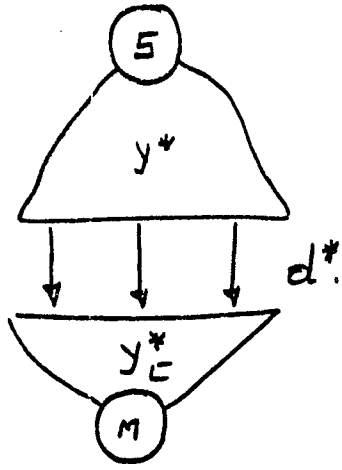
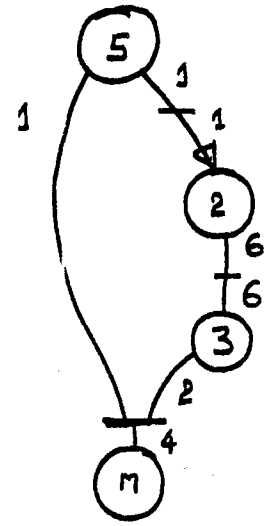
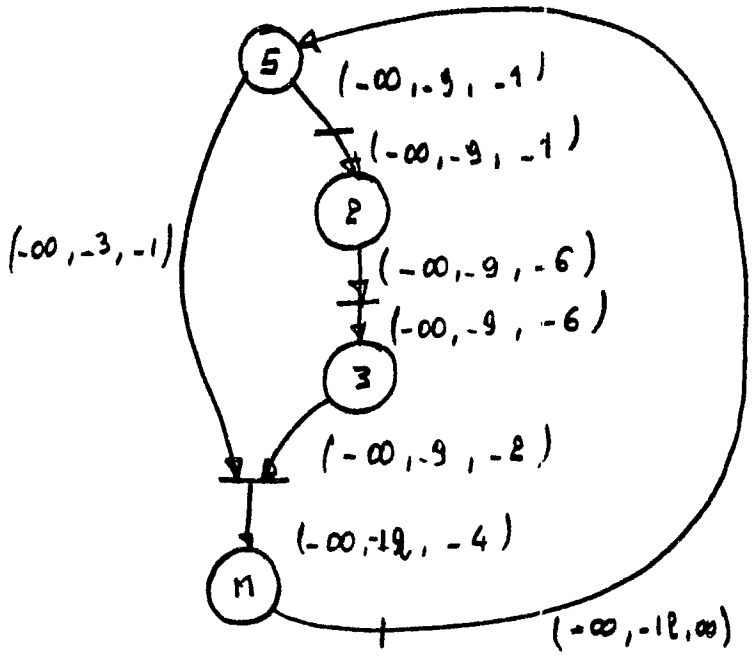


fig 24

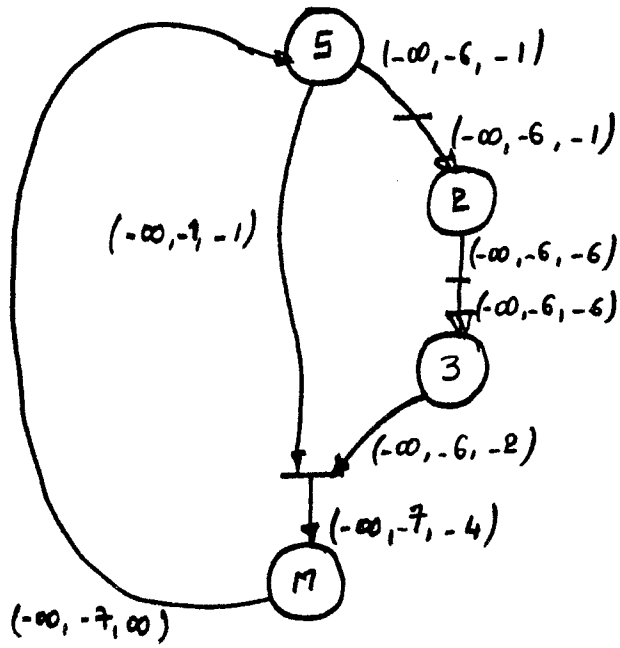


a



Sur chaque arc on porte $(\beta_u, f(u), \sigma_u)$

b



et $(\beta_u\ \bar{f}(u), \sigma_u)$

figure 25

c

Supposons que $\Omega^-(\bar{Y}) \neq \phi$ alors $\bar{f}(u_r) = \sum_{u \in \Omega^+(\bar{Y})} - \gamma(u) - |-\infty| = +\infty$

donc $\Omega^-(\bar{Y}) = \phi$.

Soit u un arc de $\Omega^+(\bar{Y})$ on considère le réseau R auquel on a soustrait tous les ars de $\Omega^+(\bar{Y})$ sauf u. On va exhiber un chemin de S à P.

Il existe un chemin de S à J(u) dont tous les sommets sont dans $\bar{Y}$. Par définition de $\bar{Y}$.

D'autre part, il existe dans le réseau initial R un chemin allant de T(u) à M. Deux cas :

- Ce chemin a un sommet dans $\bar{Y}$, alors il a un arc u tel que $J(u) \in \bar{Y}_C$ (le complémentaire de Y dans X) et $T(u) \in \bar{Y}$. Ceci est en contradiction avec $\Omega^-(\bar{Y}) = \phi$.

- Ce chemin a tous ses sommets dans $\bar{Y}_C$ donc tous les arcs de ce chemin sont dans le réseau auquel on a enlevé $\Omega^+(\bar{Y}) - \{u\}$. La concaténation du chemin de S à J(u), de u, et du chemin de T(u) à M donne un chemin de S à M :

Récapitulons : $\Omega^+(\bar{Y})$ est un déconnectant du réseau R.
Tout sous ensemble de $\Omega^+(\bar{Y})$ n'est pas un déconnectant.
$\Omega^+(\bar{Y})$ est un déconnectant minimal du réseau. c.q.f.d.

Soit un déconnectant minimal d^* . On définit $Y^* \subset X$ par

Y^* = {sommets $x \in X$ tels que il existe un chemin de S vers x n'ayant aucun arc dans d^* }. On suppose que $\Omega^-(Y^*) = \phi$.

Prop17 : La capacité de la coupe définie par Y^* dans $R = (X, U, -\gamma)$ est plus petite que celle de $\bar{Y}$.

Dem17 : Supposons que $-\gamma(\Omega^+(Y^*)) > -\gamma(\Omega^+(\bar{Y}))$

$\Longleftrightarrow$ $\gamma(\Omega^+(Y^*)) < \gamma(\Omega^+(\bar{Y}))$. donc, puisque $\Omega^-(Y^*) = \Omega^-(\bar{Y}) = \phi$,

$\gamma(\Omega^+(Y^*)) - \beta(\Omega^-(Y^*)) < \gamma(\Omega^+(\bar{Y})) - \beta(\Omega^-(\bar{Y}))$.

Ceci est en contradiction avec la définition de $\bar{Y}$ et la proposition 4.

$\bar{Y}$ réalise donc le maximum sur l'ensemble des déconnectants minimaux vérifiant $\Omega^-(Y) = \phi$ de la fonction $-\gamma$.

Remarque 15 : Soit Y un ensemble de sommets contenant S et ne contenant pas M. Les arcs de $\Omega^+(Y)$ portent l'information qui résulte de S par le traitement des sommets modules de Y. Les arcs de $\Omega^-(Y)$ portent de l'information portée par un ou plusieurs arcs de $\Omega^+(Y)$ traitée par des sommets modules de Y_C qui va être à nouveau traitée par des modules de Y et acheminée par des arcs de $\Omega^+(Y)$. Ainsi quand $\Omega^-(Y)$ n'est pas vide, la capacité de $\Omega^+(Y)$ peut être plus grande que l'information qui y transite effectivement. Ceci nous encourage à poser la définition suivante :

Def11 : On appelle coupe de capacité maximale dans un réseau R = (X,U, $-\gamma$), l'ensemble d'arcs $\bar{U} \subset U$ et vérifiant :

i. Il existe un ensemble $\bar{Y}$ de sommets tq $S \in \bar{Y}$, $M \notin \bar{Y}$ et $\Omega^+(\bar{Y}) = \bar{U}$.

ii. $\Omega^-(\bar{Y}) = \phi$ et $\Omega^+(\bar{Y})$ est un déconnectant minimal.

iii. $\bar{U}$ réalise le maximum de $-\gamma(\Omega^+(\bar{Y}))$ sous ces conditions.

Prop18 : La coupe de capacité maximale dans un réseau R = (X,U,$-\gamma$) est celle exhibée par Ford et Fulkerson dans le réseau $\tilde{R} = (X,\tilde{U},\beta,\gamma)$.

Dem18 : découle de la définition 11 et de la proposition 17.

Remarque 16 : La figure 28 représente en a,b et c un réseau R = (X,U, $-\gamma$) sur lequel on se propose de trouver la coupe de capacité maximale, un flot réalisable sur le réseau $\tilde{R} = (X,\tilde{U},\beta,\gamma)$ et un flot optimal sur R. $\bar{Y} = \{S,2\}$ vérifie $\Omega^+(\bar{Y})$ est un déconnectant minimal et $\Omega^-(\bar{Y}) = \phi$.

On propose, pour clore ce paragraphe, un moyen pour se faire une idée sommaire du profil informationnel d'un réseau donné, en utilisant les notions de coupe de capacité minimale et de coupe de capacité maximale.

Il est bien entendu que la donnée du profil informationnel tel qu'il est défini à la définition 8 n'est pas intéressante car elle fournirait trop d'information qu'il serait difficile de traiter, donc, en se contenant d'un profil informationnel, on répond à la fois à deux critères :

- la facilité de calcul du profil
- la facilité d'utilisation.

On se donne les capacités $CS = -\gamma(\Omega^+\{S\})$ capacité de la source, C_{max} trouvée comme il vient d'être indiqué.
C_{min} et $C_{\mathfrak{P}} = -\gamma(\Omega^+(U - \{\mathfrak{P}\}))$, capacité du puits.

En général, pour des écoulements de taille ordinaire, ces quatre capacités sont suffisantes. Toutefois, si on veut détailler davantage, on considère, par exemple, dans le cas de la figure 29, le réseau compris entre C_S et C_{max} et on y calcule la coupe min : si cette coupe est C_S, et étant donné que la coupe max dans ce réseau est C_{max}, on admet qu'on a affaire, entre C_S et C_{max}, à un évasement. Si la coupe min du réseau comprise entre C_S et C_{max} est différente de C_S, son adjonction à la figure 29 donnerait un détail de plus sur le profil informationnel. Il est bien entendu que, dans le réseau compris entre C_{max} et C_{min}, la coupe max est C_{max} et la coupe min C_{min}. Dans le cas de la figure 29 on admet qu'on a affaire à un étranglement. Dans le réseau compris entre C_{min} et C_M on cherche la coupe max. Si elle vaut C_M on admet qu'on a affaire à un évasement sinon on l'insère entre C_{min} et C_{max}.

On itère ce processus jusqu'à ce que l'on soit satisfait du niveau de détail ou que l'on aboutisse à une suite alternée d'évasements et d'étranglements.

———————— C_5

———————————————————— C_{max}

———— C_{min}

———————— C_m.

Exemple de profil informationnel d'un niveau.

fig 26.

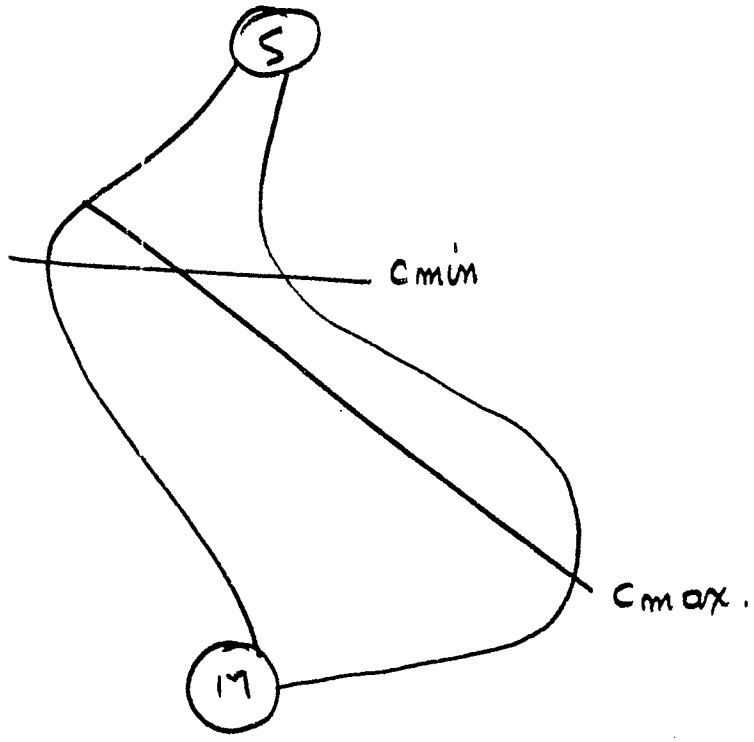


figure 27

Remarque 15 : Il peut arriver qu'une coupe min et une coupe max soient croisées - voir figure 27 - On propose de diviser le réseau en deux selon sa coupe min ou sa coupe max et on recherche les profils informationnels des deux moitiés ainsi trouvées ; désormais on supposera que tout réseau peut être décomposé en une suite alternée d'étranglements et d'évasements.

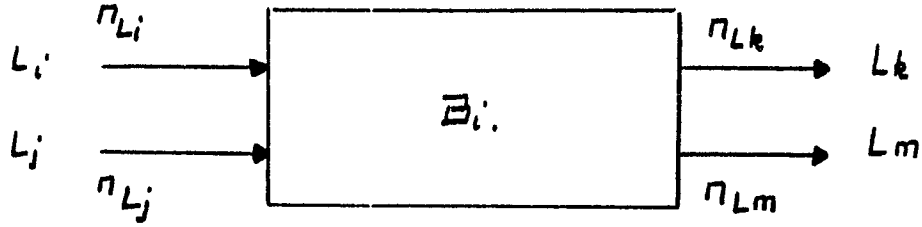


fig 1.

---

temps CPU normalisé.

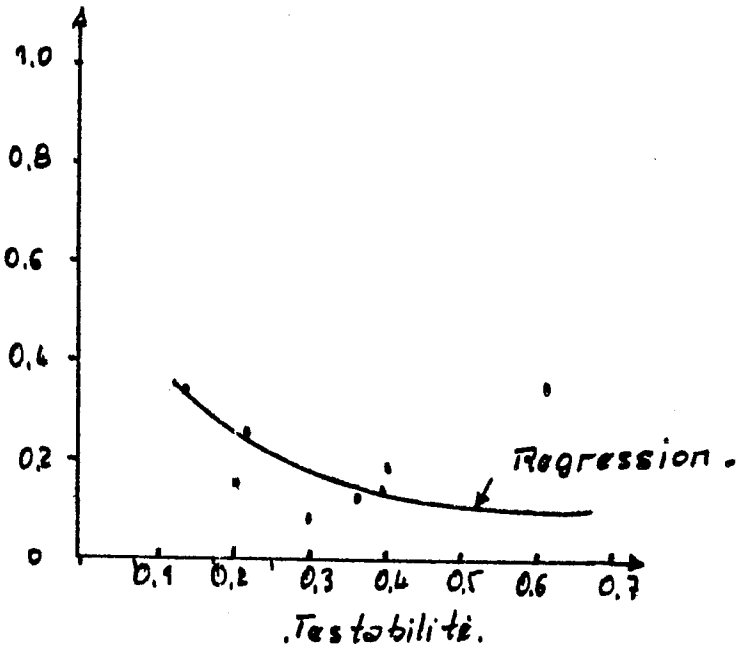


fig 2.

CHAPITRE 4

ELABORATION DES PARAMETRES DE TEST

En utilisant les outils introduits au chapitre 4, nous allons définir des paramètres qui interviennent dans l'évaluation de la difficulté de tester un module isolé ou un module dans un système. Auparavant, nous allons exposer ce qui a été fait dans le même sujet : ce sera l'objet du paragraphe A.

A. Approche de la Testabilité -

Cette formulation de la testabilité est dûe à Stephenson et Grason et a été exposée dans la référence [11] citée en bibliographie. Il est proposé une mesure de testabilité (dans le sens d'une mesure de la difficulté de génération des séquences de test) pour un système représenté au niveau registres de transfert.

Le système est donc décrit par des modules B_i et des lignes L_i auxquelles sont attachés les nombres de fils de ces lignes M_{L_i}, reliant ces modules. Pour chaque module B_i on définit deux coefficients calculés à partir de la fonction réalisée par B_i :

- Le facteur de commande (CTF_i) (commandability transfert factor) représente la difficulté de générer une valeur en sortie de B_i à partir des

valeurs d'entrée ; le CTF_i est donné comme une moyenne pondérée de l'écart entre le nombre de valeurs d'entrée produisant une valeur de sortie donnée et la moyenne de ce même nombre. Le facteur est compris entre 0 et 1, 1 représentant une commandabilité parfaite.

- Le facteur d'observation (OTF_i) représente de même la difficulté de discerner si une valeur erronée est présentée en entrée de B_i, à partir des valeurs de sortie.

La commandabilité C_i du module s'obtient par moyenne pondérée des commandabilités des lignes d'entrées CL_i. Pour l'exemple de la figure 1, on obtient :

$$C_i = \frac{M_{L_i} \; C_{L_i} + M_{L_j} \; C_{L_j}}{ML_i + ML_j}$$

La commandabilité des lignes de sortie est identique pour toutes les lignes et s'écrit pour l'exemple donné $C_{L_k} = C_{L_m} = C_i \; CTF_i$

On a, pour l'observabilité, des relations analogues :

$$O_i = \frac{M_{L_k} \; O_{L_k} + M_{L_m} \; O_{L_m}}{M_{L_k} + M_{L_m}}$$

$$O_{L_i} = O_{L_j} = OTF.O_i$$

Des formules spéciales permettent de traiter les divergences de lignes. Soit un système comportant N modules B_i, $1 \leq i \leq N$;
On cherche à calculer les commandabilités C_i et les observabilités O_i de tous les modules. Ces valeurs s'obtiennent en résolvant deux systèmes linéaires de N équations à N inconnues.

La commandabilité C et l'observabilité O du système complet sont les moyennes des C_i et des O_i. Enfin, la testabilité T du système global est définie par $T = \sqrt{CO}$

Les auteurs ont procédé à une étude expérimentale de leur coefficient de testabilité, étude qui montre son intérêt pratique : ils ont considéré un certain nombre de circuits de complexité variée pour lesquels ils ont calculé la testabilité T_i. Ils ont procédé à la génération automatique de séquences de test en notant le temps calcul t_i (temps CPU normalisé) pour chaque circuit. Enfin ils ont approximé le nuage (t_i, T_j) par régression linéaire à l'aide de la fonction $\text{Log } t_i = p \text{ Log } T_i + \log a$ trouvant un coefficient de corrélation significatif au seuil de 10% le résultat de cette régression est donné dans la figure 2.

Ce résultat montre que l'on peut utiliser ce coefficient de testabilité pour prévoir la difficulté d'élaboration d'un test, chercher des modifications améliorant la testabilité, et même guider des algorithmes du type chemin sensible en les aiguillant vers les voies les plus commandables ou les plus observables.

L'approche qui sera utilisée dans la présente étude, est quelque peu différente. On élaborera tout d'abord un paramètre intrinsèque à un module, qui mesure la difficulté de tester ce module, en évaluant, en l'ocurrence, le nombre de vecteurs de test qu'il lui faut. Ensuite on définit de façon tout à fait indépendante, les paramètres qui s'imposent quand ce module se trouve dans un système et qu'il faut acheminer les vecteurs de test de l'entrée de ce module vers la source du système et les résultats de test de sa sortie vers le puits du système. Ces paramètres seront étudiés dans un contexte de test déterministe mais gardent leur signification en dehors de ce contexte si bien que certains d'entre eux seront utilisés dans des stratégies de test aléatoire (voir le chapitre 5).

Ces paramètres peuvent se classer dans trois catégories essentiellement :

1. Les paramètres d'accessibilité
2. Les paramètres de rendement
3. les paramètres de travail de propagation.

Nous les présentons un à un :

1a. Commandabilité : considérons un module isolé. On peut forcer à son entrée tous les vecteurs nécessaires à son test. Considérons ce même module M_i dans un écoulement et appelons f_i la fonction qui à toute valeur prise par la source S de cet écoulement associe la valeur qui en résulte à l'entrée X_i de M_i. f_i peut être non surjective. Alors $|f_i(S)|$ mesure le nombre de vecteurs que l'on peut commander à l'entrée de M_i en agissant sur S. On définira la commandabilité de M_i par une fonction croissante de $|f_i(S)|$. (Dans cette notation S est considéré comme un ensemble).

1b. Observabilité : Considérons un module isolé, en accédant à sa sortie, on peut lire toutes les réponses qu'il donne à ses entrées. Considérons ce même module M_i dans un écoulement et soit Y_i sa sortie. P est fonction de Y_i et d'autres variables. Supposons les autres variables fixées et appelons g_i la fonction qui à toute valeur de Y_i associe la valeur qui en résulte en P. g_i peut être non injective. Ainsi une même valeur en P peut avoir plus d'un antécédent en Y_i par g_i. On définira l'observabilité de M_i par une fonction croissante de $|g_i(Y_i|$.

2a. Rendement amont : Soit un module M_i contenu dans un écoulement. Supposons que l'on a décidé de lui appliquer un vecteur de test. Le premier problème qui se pose est alors de lui trouver un antécédent par f_i, si celui-ci existe. On suppose que ceci se fait d'une coupe à la suivante en essayant à chaque fois de commander à partir de l'entrée d'un module une valeur que l'on veut faire afficher à sa sortie. La fonction exécutée par ce module peut être non surjective, auquel cas la propagation s'arrête là.

Le rendement amont sera fonction de la probabilité qu'une propagation amont partie de X_i aboutisse jusqu'en S. Le rendement amont est relié à la commandabilité. Ce lien sera étudié dans le paragraphe E1.

2b. Rendement aval : Soient $y_1, y_2, \ldots, y_k$ les réponses que livre un module M_i à un vecteur de test x quand il est dans l'état de santé f_1 f_2 ou f_k. On suppose que M_i est dans un écoulement et on note g_i la fonction définie comme au paragraphe 1b. g_i peut être non injective. Il peut arriver alors que l'on ait $g_i(y_1) = g_i(y_2)$: en lisant $g_i(y_1)$ on ne sait pas si y_1 ou y_2 est affiché en Y_i, donc on ne sait pas si le module M_i est dans l'état de santé f_1 ou f_2. Le rendement aval est fonction de la probabilité que $g_i(y_1)$ soit égal à $g_i(y_2)$ sachant que $y_1 \neq y_2$. Le rendement aval est lié à l'observabilité (voir § E2).

3a. Charge de travail amont : Nous nous sommes inspirés d'une étude qui évalue le travail d'une fonction (voir [20]) pour évaluer l'espérance du travail que l'on doit exécuter afin de propager un vecteur de X_i vers S sachant que l'on risque d'être bloqué au cours de la propagation par le défaut de surjection d'un module.

3b. Charge de travail . Nous évaluerons aussi l'espérance du travail à effectuer pour propager y_1 et y_2 de Y_i vers P sachant que l'on risque, à chaque nouvelle coupe, de leur trouver des images identiques (et donc d'arrêter la propagation).

B. Evaluation du nombre de vecteurs de test -

Soit un module M à tester. Soient :

- X la variable aléatoire qui prend ses valeurs dans l'ensemble des valeurs d'entrée possibles du module M.
- Y la variable aléatoire qui prend ses valeurs dans l'ensemble des valeurs de sortie possibles du module M.

- F la variable aléatoire qui prend ses valeurs dans l'ensemble des pannes possibles du module F = $\{f_0, f_1, \ldots, f_n\}$; f_0 : état sain.

Le test de M est d'autant plus facile que la séquence de test est plus courte. On se propose de définir un coefficient compris entre 0 et 1 qui mesure la facilité du test et qui est en l'occurence une évaluation de l'inverse du nombre de vecteurs nécessaires au test. Nous allons donner une formule de ce coefficient, pour laquelle nous fournirons une justification a priori et une justification a postériori.

Justification a priori : on présente ci-dessous un modèle de test pratiquement irréalisable mais qui présente l'avantage d'être un modèle : général et rigoureux, il nous servira de canevas pour justifier l'évaluation de la charge de test proposée par CASPI dans [8]

```
Débutest
entier K, Kfin ; logique Satisfait :
K := 1 ; Satisfait := faux ;
A chaque panne f_i on associe sa probabilité p_i = P(F = f_i) ;
tantque   Satisfait faire
    début
    appliquer X_k ; cø k° vecteur de test ;
    observer Y_k_i
    calculer les p_i^(k+1) connaissant les p_i^k et la réalisation (X_k Y_k) ;
    k := k+1 ;
    fin
kfin := k.
fintest.
```

En fonction de l'objectif du diagnostic on décidera de la condition d'arrêt de l'itération : c'est là que réside la généralité du modèle.

Ainsi :

1. Pour localiser la panne, on pose Satisfait 1 <==> $\exists$ i tel que $p_i = 1$,

2. Pour réaliser un diagnostic à la carte, on pose Satisfait 2 <==> Eviter les pannes envisageables, compte tenu des itérations précédentes, sont sur une même plaque.

3. Pour réaliser un test Go/No Go on pose Satisfait 3 <==> $p = 0$ ou $p_0 = 1$.

A toute distribution de probabilité p_i^k $1 \leq k \leq kfin$, on associe la quantité d'information $V_k(F)$ qui mesure l'incertitude sur F à l'instant k. On écrira les conditions relatives à ces trois exemples en termes de théorie de l'information.

1. Satisfait 1 <==> $V_{kfin}(F) = 0$ acquisition de la quantité d'information H(F).

2. On partitionne l'ensemble des pannes par plaques $F = \{f_0, f_1, \ldots, f_n\}$ $\rightarrow \phi_2 = \{f_0, P_1, P_2, \ldots, P_p\}$ et on pose : Satisfait 2 <==> $V_{kfin}(\phi_2) = 0$

le test consacre l'acquisition de la quantité d'information $H(\phi_2)$ sur F.

3. On partitionne F en deux classes : $\phi_3 = \{f_0, \bar{f}_0\}$ et on pose :

Satisfait 3 <==> $V_{kfin}(\phi_3) = 0$ acquisition de $H(\phi_3)$

Ces exemples dégagent une idée commune : le diagnostic consiste en l'application séquentielle de vecteurs de test jusqu'à accumuler une quantité d'information suffisante sur F. Un système est d'autant plus testable que le nombre d'itérations nécessaires à l'acquisition de cette quantité d'information est plus petit. Par souci de normalisation, on écrira l'expression du nombre d'itérations kfin dans le cas de l'exemple 1.

$$kfin = \frac{V_0(F) - V_{kfin}(F)}{V_0(F) - V_1(F)}$$ en supposant que l'application de

X, et l'observation de Y_1 a fourni une quantité d'information "moyenne" sur F.

$V_0(F) = H(F)$; $V_{kfin}(F) = 0$; et $V_1(F) = H(F/X=X_1, Y=Y_1)$;

d'après les développements du chapitre 4, et vu que (X_1, Y_1) à été supposé être un test moyen, $V_1(F)$ s'écrit $H(F/XY)$, d'où :

$$kfin = \frac{H(F)}{H(F) - H(F/XY)}$$

et, si on appelle T la mesure de testabilité, on a :

$$T = \frac{H(F) - H(F/XY)}{H(F)}$$

En utilisant les résultats du chapitre 4, on transforme H(F).

$H(F/XY)$ est $= H(F) + H(XY) - H(F/XY)$

$= H(F) + H(XY) - H(F/X) - H(Y/F,X)$

Vu que les variables aléatoires F et X sont indépendantes, on écrit :

$= H(XY) - H(X) - H(Y/F,X) = H(Y/X) - H(Y/F,X)$

Pour un module combinatoire ou un module séquentiel dont on connaît l'état interne, on a $H(Y/F,X) = 0$ car la donnée de l'état de santé et de l'entrée détermine la sortie Y.

donc $T = \frac{H(Y/X)}{H(F)}$: T sera appelé coefficient de test.

Dans cette formule H(Y/X) mesure l'incertitude qui porte sur Y quand on a affiché un vecteur X en entrée. Cette quantité dépend étroitement de l'ensemble dans lequel F prend ses valeurs. Pour achever les transformations menées ci-haut, on a invoqué que X et F sont indépendantes.

Ceci appelle deux remarques :

+ la réalisaticn de F est indépendante de toute réalisation de X car F intervient avant X.

+ de plus, quand on calcule des paramètres de test, on n'est pas supposé connaître la relation qui existe entre la réalisation de F et celles de X car celle ci dépend de la méthode utilisée pour réaliser le test ; si bien que, même si on connaît la réalisation de F, on n'en déduira aucune présomption sur la réalisation de X. Ceci explique :

- d'une part qu'il est significatif de prendre des hypothèses probabilistes sur X sachant F

- d'autre part que, pour nous, X est indépendant de F.

Def1 : Le coefficient de test du module M sous les hypothèses de panne cataloguées dans F s'écrit :

$$T = \frac{H(Y/X)}{H(F)}$$

Justification a Posteriori

Soit x un vecteur de test moyen dans le sens $H(Y/X=x) = H(Y/X)$ alors :

$T = 0 \Rightarrow$ sous les hypothèses F, la connaissance de X détermine parfaitement Y. Donc Y ne dépend pas de F $\Rightarrow$ l'observation de Y ne nous informe pas sur F : Module impossible à tester.

Par contre, dans le cas de la testabilité idéale, x étant affiché en entrée du module, il s'établit une bijection entre Y et F, d'où :

$H(Y/X=x) = H(F) \Rightarrow 1$.

Calcul du coefficient de test d'un module isolé

On a : $T = \frac{H(Y/X)}{H(F)}$: Vu qu'il est difficile d'évaluer la relation stochastique qui existe entre X , Y et F, on est amené à établir des règles de calcul approché qui sont présentées et justifiées dans [8] que nous réécrivons et que nous appliquons à certains exemples :

$$T = \frac{\min(H(Y), H(F)}{H(F)}$$

En effet, étant donné un vecteur x moyen,

$H(Y/X) = H(Y/X=x) \leq H(Y)$, par définition, en outre :

$H(Y/X=x) \leq H(F)$, car, le vecteur x étant affiché en entrée du module , il ne saurait y avoir plus de sorties possibles que d'états de panne.

Soit un module M_1 tel que $H(Y_1) = 5$ alors que $H(F_1) = 6$; $T_1 = \frac{5}{6}$

aucun vecteur ne suffit à détecter toutes les pannes.

Soit un module M_2 tel que $H(Y_2) = 8$ alors que $H(F_1) = 6$; $T_1 = 1$

Il est permis d'espérer qu'il existe un vecteur d'entrée X tel que la sortie correspondante soit l'image inverse de F par une certaine fonction μ : $F = \mu(Y)$. A toute sortie on associerait une panne de F_i. Ce serait un test idéal. Toujours est-il que $T = 1$ est révélateur d'une bonne testabilité car la sortie Y fournit plus d'information que l'incertitude sur F.

Coefficient de test d'un module dans un écoulement

Soit à tester un module m_i dans un écoulement E. Ceci revient à tester l'écoulement E et à accuser le module m_i chaque fois qu'un vecteur x_k donne une sortie y_k non conforme. Autrement dit, ceci revient à tester l'écoulement E en définissant son ensemble de pannes comme étant

F_i, celui de m_i. Si bien que l'on pose, tout naturellement :

$$T_i^E = \frac{H(Y_P/X_S)}{H(F_i)}$$ où X_S est la variable aléatoire source et Y_P la variable aléatoire puits du réseau.

Coefficient de test d'un écoulement : On admet aisément :

$$T^E = \frac{H(Y_P/X_S)}{H(F)} \quad \text{où } F = \bigcup F_i$$

On montre que T^E peut s'écrire en fonction des T_i^E

$$T^E = \sum_{i=1}^{P} \frac{M_i}{N} \left(\frac{\text{Log } M_i}{\text{Log } N} T_i^E \right)$$

avec $M_i = (F_i)$: $N = \sum_{i=1}^{P} n_i$; on remarque que T^E n'est pas l'espérance des T_i^E, mais l'espérance des $\left(\frac{\text{Log } M_i}{\text{Log } N} T_i^E \right) \leq T_i^E$. Ceci rendrait compte du fait que certains vecteurs destinés à tester m_i peuvent informer sur l'état de santé d'un autre module m_j.

Remarque 1 : La formule de T présente une lacune assez grave:elle ne fait pas intervenir H(X). En outre H(Y) est une majoration très brutale de l'information que l'on peut recueillir sur l'état de santé du module par une expérience de test (X = x, Y = y). En outre, dans la formule de T_i^E, $H(Y_P)$ peut être beaucoup plus grand que l'information que fournit effectivement le module m_i à P. Ces inconvénients ne mettent pas en cause la formulation rigoureuse de T mais son approximation ; dans [2] Caspi propose des rectifications à cette formule. En outre, pour la testabilité d'un module dans un écoulement, nous allons nous donner dans la suite des moyens de l'évaluer avec plusieurs paramètres indépendants.

C - Commandabilité - Soit un écoulement E dans lequel on s'intéresse à un module M particlier. On se propose de définir une quantité réelle C(M) qui mesure la facilité de commander le module M.

Def2 : La commandabilité de M dans l'écoulement E est le Log_2 du nombre de vecteurs distincts qu'il est possible d'afficher en entrée de M en affichant en entrée de l'écoulement toutes les valeurs possibles.
Considérons l'algorithme de marquage suivant : dit algorithme $A_\Pi^-(M)$
0 cocher M
1 si une place X cochée autre que S n'a pas de transition père cochée, alors cocher $\Pi(X)$ sinon fin.

2 si une transition cochée n'a pas ses places pères cochées, cocher les et allera 1.

Soit $E_\Pi^-(M)$ le sous graphe de E construit sur les sommets cochés par $A_\Pi^-(M)$.

On définit sur $E_\Pi^-(M)$ la fonction capacité γ qui associe à chaque ligne le nombre de ses fils binaires, alors

Prop1 : $C(M) = \underset{\Pi}{Max}$ (flot max qui peut transiter dans $E_\Pi^-(M)$).
En particulier, si $E_\Pi^-(M)$ vérifie C1, C(M)= le flot max de $E_\Pi^-(M)$.

Dem1 : A. Par construction de $E_\Pi^-(M)$, il existe un chemin de tout sommet de $E_\Pi^-(M)$ vers M ==> M est le puits de $E_\Pi^-(M)$.

B. D'autre part, et toujours pas construction, tous les arcs et les modules intervenant dans l'élaboration de M, pour la selection déterminée par Π, retrouvent dans $E_\Pi^-(M)$. D'après B le flot d'information max dans $E_\Pi^-(M)$ vaut le flot d'information max qu'il est possible d'acheminer de S vers M dans le réseau E si on a pris la sélection Π. En outre, la détermination de Π se fait-elle aussi par S - aiguillage des multiplexeurs -

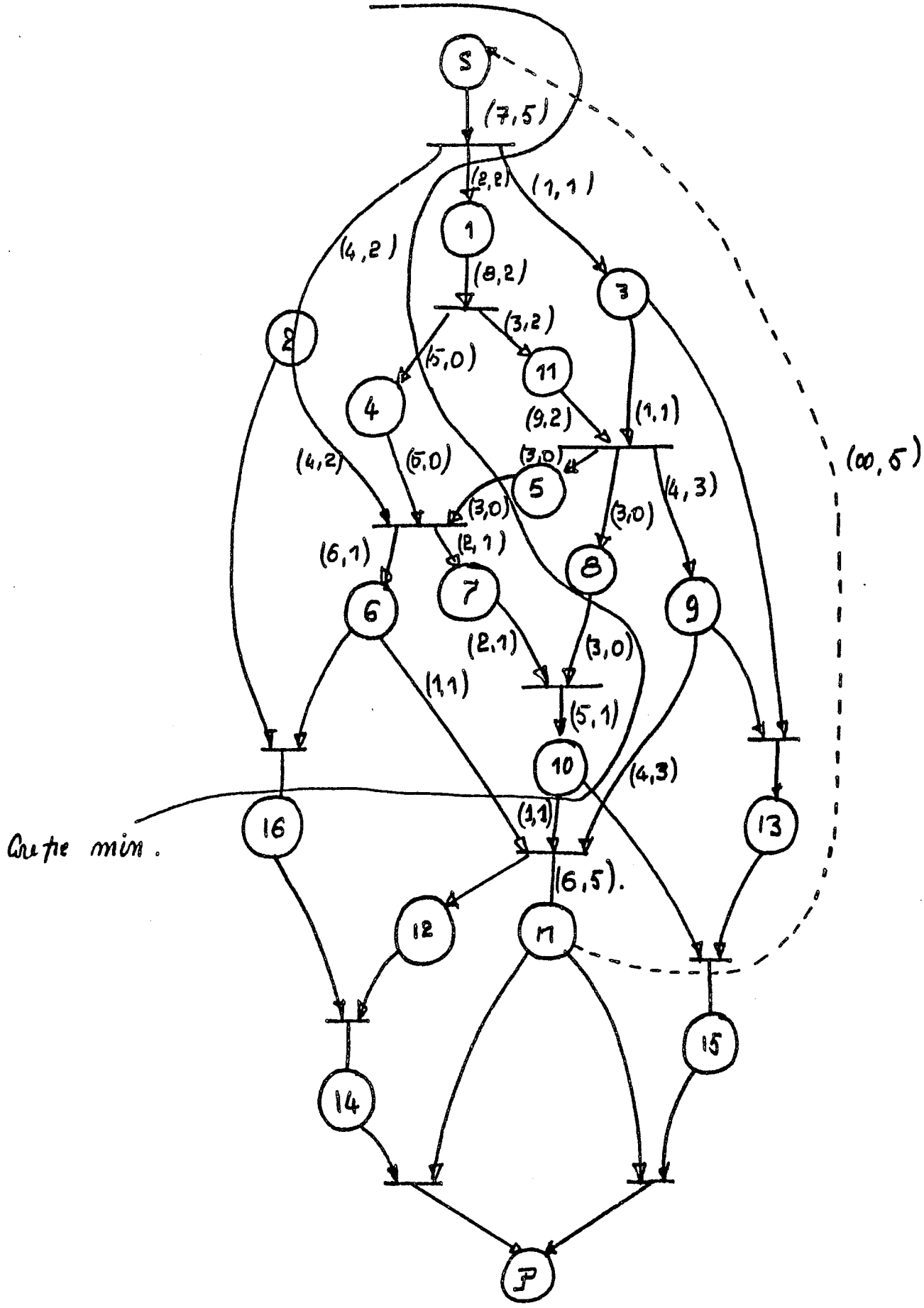

S
(7,5)
(2,2)
(1,1)
(4,2)
1
(8,2)
3
2
(3,2)
(5,0)
11
4
(9,2)
(1,1)
(4,2)
(5,0)
(3,0)
5
(4,3)
(3,0)
(3,0)
(6,1)
(2,1)
8
6
7
9
(2,1)
(3,0)
(1,1)
(5,1)
10
(4,3)
16
(1,1)
13
(6,5).
12
17
15
14
F
(∞,5)
Coupe min.

fig 3.

donc le flot d'information qui peut transiter de S vers M dans E est égal au max sur Π des flots d'information qui traversent $E_\Pi^-(M)$.
D'après A, le flot d'information max qu'il est possible d'acheminer de S vers M dans $E_\Pi^-(M)$ vaut, sous les réserves d'usage concernant C1 C2 C3 C4, le flot max dans $E_\Pi^-(M)$. Enfin, le flot d'information max dans un réseau vaut le nombre de vecteurs qui peuvent y transiter. c.q.f.d.

Soit un module M plongé dans un écoulement E. On se propose de lui associer un réel dit rapport de commandabilité, compris entre 0 et 1 et qui mesure le rapport entre sa commandabilité dans E et sa commandabilité intrinsèque, à savoir la capacité de son entrée, H(XM). Il vient :

Def3 : Le rapport de commandabilité d'un module M dans un écoulement E se note $CC_E(M) = \frac{C(M)}{H(XM)}$

Sur la figure 3 ci-contre, on trouvera un exemple de calcul de la commandabilité d'un module M dans un écoulement. $E_\Pi^-(M)$ est le sous graphe construit sur les sommets S, M les places numérotées de 1 à 11 et les transitions adjacentes à deux places parmi celles-là. Sur chacun des arcs de $E_\Pi^-(M)$ on écrit $(c(u), \bar{f}(u))$.

On voit que $\Omega^+(\{S,2,4,6,7,10\}) = 5 = \bar{f}(MS)$ donc $\bar{f}$ est maximal.

On trouve $C(M) = 5$ bits
$CC(M) = \frac{5}{6}$

En effet $E_\Pi^-(M)$ vérifie la condition C1 ⟹ on ne calcule le flot max qu'une fois. Ceci est un cas courant.

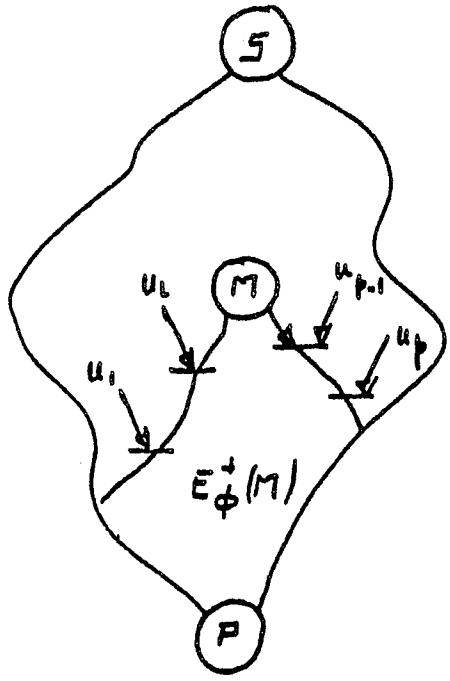


Figure 4.

$(Y \times U)/\mathcal{C}$

y_1	u_1
y_2	u_1
y_3	u_1
y_1	u_2
y_2	u_2
y_3	u_2
y_1	u_3
y_2	u_3
y_3	u_3

$n_1 = 1$

$n_2 = 2$

$n_3 = 3$

Figure 5.

D. Observabilité

Soit Y_M la variable aléatoire qui prend ses valeurs dans l'ensemble des vecteurs de sortie du module M. A chaque observation de P on associe un certain nombre de réalisations possibles de Y_M. Un module est d'autant plus observable que le nombre de réalisations de Y_M envisageables pour une réalisation donnée de P est réduit (car l'information obtenue sur Y_M est d'autant plus précise).

Nous allons mesurer l'observabilité de M, O(M) par une quantité décroissante par rapport à ce nombre évalué dans un contexte très particulier. Nous dirons ensuite dans quelle mesure l'évaluation proposée garde sa signification en dehors de ce contexte.

Considérons l'algorithme de marquage suivant : dit algorithme $A^+_\phi(M)$.

0 - Cocher M

1 - Si une place autre que P est cochée et n'a pas de transition fils cochée, alors cocher $\phi(M)$ sinon fin

2. Si une transition est cochée et n'a pas tous ses fils cochés, cocher tous ses fils. allera 1.

Soit $E^+_\phi(M)$ le sous réseau construit sur l'ensemble des sommets cochés et soit U_M la variable aléatoire qui prend ses valeurs dans $U_1 \times U_2 \times \ldots U_p$ avec : u_i : ensemble des valeurs que peut prendre l'arc u_i vérifiant

$T(u_i) \in E^+_\phi(M)$ et $J(u_i) \notin E^+_\phi(M)$. Dans le cas de $E^-_\Pi(M)$, on a vu que M est fonction de S. Dans $E^+_\phi(M)$, on ne peut pas dire que P est fonction de Y_M;

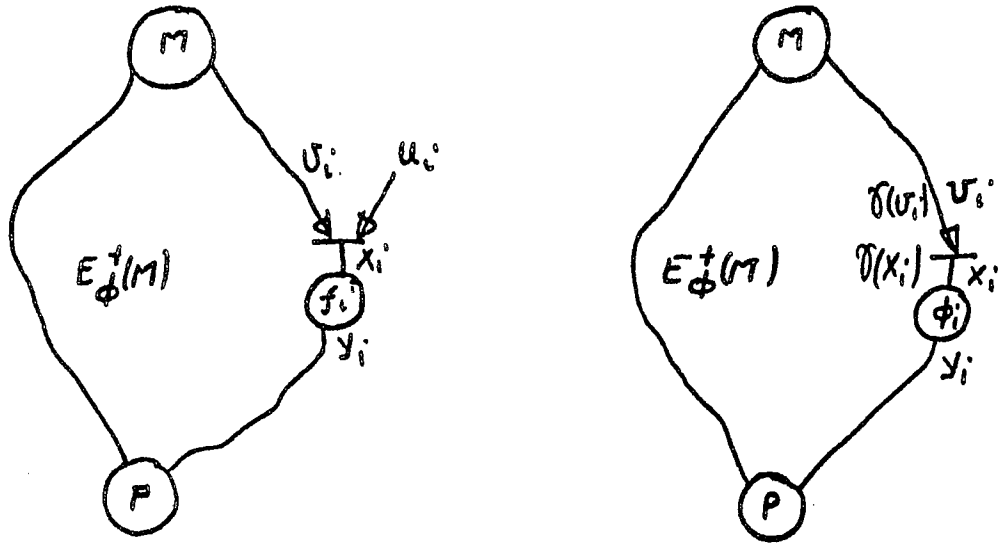
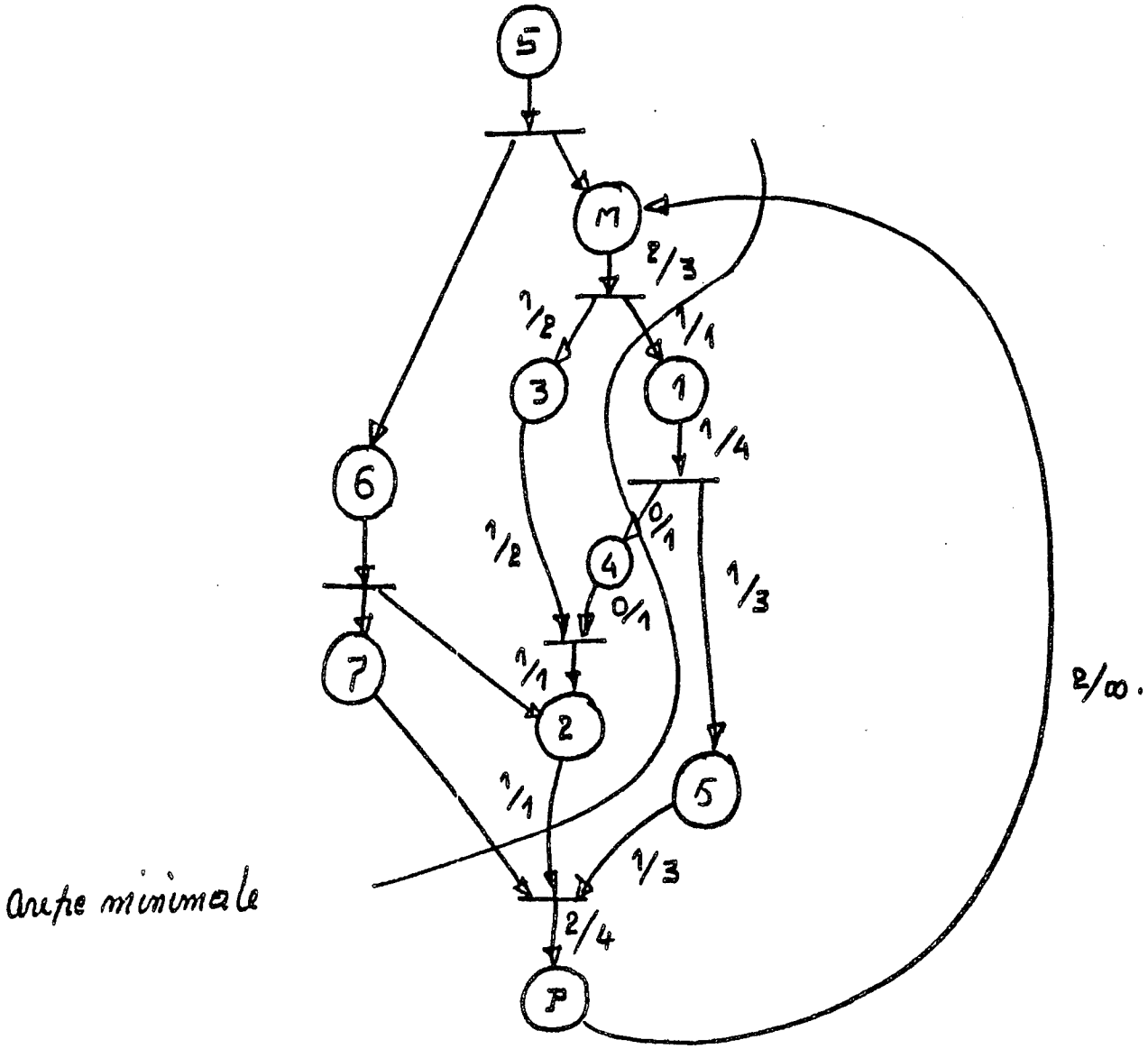


figure 6.



Arête minimale

figure 7.

On pose $P = g(Y_M, U_M)$. On partitionne l'ensemble des valeurs que peut prendre la variable conjointe (Y_M, U_M) par la relation d'équivalence :

$$(y_{i_1}, U_{j_1}) \,\&\, (y_{i_2}, U_{j_2}) \Longleftrightarrow g(y_{i_1}, U_{j_1}) = g(y_{i_2}, U_{j_2}).$$

A chaque valeur u_j on associe l'entier n_j = nombre de classes d'équivalence contenant un élément ayant u_j en deuxième argument. Le nombre n_j mesure le pouvoir de discrimination associé à u_j. La meilleure observabilité est obtenue en donnant à U la valeur u_i telle que

$n_j* = \max_j n_j$. (u_i* sera noté u_* et n_j* sera notée n_*).

U étant fixé à U^* ; on définit sur Y_M la relation d'équivalence

Γ_* par $y_1 \,\Gamma_*\, y_2 \Longleftrightarrow g(y_1\, u_*) = g(y_2, u_*)$

et on pose :

Def4 : L'observabilité : $O(M) = \mathrm{Log}_2 |Y_M/\Gamma_*|$

En effet $(Y_{M/\Gamma_*}) = \dfrac{|Y_M|}{T(\Gamma_*)}$ où $T(\Gamma_*)$ est la taille moyenne d'une classe d'équivalence de Y_M modulo Γ_*. $T(\Gamma_*)$ est le nombre moyen de valeurs envisageables en Y_M quand on connaît une réalisation de P, correspondant à la valeur de U la plus discriminatoire. Ceci prouve que, comme on l'a annoncé plus haut, O(M) est une fonction décroissante de $T(\Gamma_*)$.

(Y_M/Γ_*) mesure le nombre de classes d'équivalence de Y_M que l'on peut discerner en lisant P, sachant que $U = u_*$. Des considérations de perte de commandabilité peuvent faire que u_* ne soit pas commandable à partir de S.

En outre, l'interdépendance de Y_M et U peut faire que l'on ne peut pas dérouler toutes les valeurs de Y_M en gardant U fixé. Ceci tend à prouver que $|Y_M/\Gamma_*|$ est une majoration du nombre de classes de Y_M que l'on peut distinguer en lisant P. En fait il n'en est rien car, dans la réalité, on peut être amené, pour identifier un vecteur de Y_M, à afficher en U plusieurs valeurs et à opérer par intersection de classes. Toutefois, après avoir exposé une méthode de calcul de O(M), on remarquera que le résultat est très facile à interpréter physiquement, donc qu'il est significatif.

Calcul de l'observabilité : Considérons le réseau $E^+_\phi(M)$ et fixons U à U_*. On se propose de calculer $Log_2|Y_M/\Gamma_*|$. U étant fixé, P est fonction de M, donc $W_{P_M,\Gamma_P}(M,P)$ a un sens. Soient $\tilde{P}_M$ la probabilité sur Y_M telle que si on émet les messages de Y_M avec la probabilité P_M, tous les messages réalisables en P apparaissent avec une probabilité uniforme, et $\tilde{\Gamma}_M$ la relation d'équivalence Identité sur P.

Prop2 : $Log_2 |Y_M/\Gamma_*| = W_{\tilde{P}_M \tilde{\Gamma}_P}(M,P)$

Dem2 : Ceci résulte immédiatement de la définition de Γ_* et du fait que :

$$H(\frac{1}{n},\ldots,\frac{1}{n}) = Log_2\ n$$

Par construction, le sous réseau par lequel transite l'information entre M et P dans E est $E^+_\phi(M)$. De plus, soit une place P_i exécutant une fonction f_i à deux arguments , $f_i(v_i,u_i)$

v_i est un arc tel que $J(u_i) \in E^+_\phi(M)$

u_i est un arc tel que $J(u_i) \in E^+_\phi(M)$. cf. figure 6.

u_i étant fixé, on peut poser $\phi_i(v_i) = f_i(v_i - u_i)$. La place réalisant f_i dans E peut être supposée réaliser ϕ_i dans $E^+_\phi(M)$ sans qu'on change quoique ce soit au graphe si ce n'est d'ignorer u_i. Il n'est même pas utile de changer la capacité $\gamma(X_i)$ (sous prétexte que P_i ne reçoit plus que $\gamma(v_i)$) car, en amont de X_i se trouve u_i donc le flot qui peut passer dans X_i est limité par $\gamma(v_i)$ et ce $\forall\ \gamma(X_i)$.

Donc le réseau par lequel transite l'information de M vers P dans E est $E^+_\phi(M)$. On a alors, avec les réserves d'usage (C1 C2 C3 C4),

Prop3 : O(M) = flot max de M vers P dans $E^+_\phi(M)$.

Remarque 1 : Pour la simplicité de l'exposé on a supposé ϕ fixé ou unique sur $E^+_\phi(M)$. Il est bien entendu que, s'il y a plusieurs valeurs de ϕ, l'observabilité de M est le max sur ϕ des observabilités trouvées ; de même que le ϕ qui réalise le max indique le cheminement qui permet la meilleure observabilité.

Remarque 2 : Supposons que Y_M émet des messages avec la probabilité P_M. En observant P on reçoit sur Y_M la quantité d'information $T_{P_M}(Y_M, P)$ où T est la fonction transinformation. Cette transinformation est maximale pour $P_M = \tilde{P}_M$ car c'est alors que la probabilité sur les occurences de P est uniforme.

Si Y_M émet avec la distribution $\tilde{P}_M$, l'observation de P fournit sur Y_M la quantité d'information $T_{\tilde{P}_M}(Y_M,P)$. M est d'autant plus observable que cette quantité est grande. On pourrait donc définir :

$$O(M) = T_{\tilde{P}_M}(Y_M,P)$$

$$= \max_{P_M} T_{P_M}(Y_M,P) \text{ qui vaut, par définition :}$$

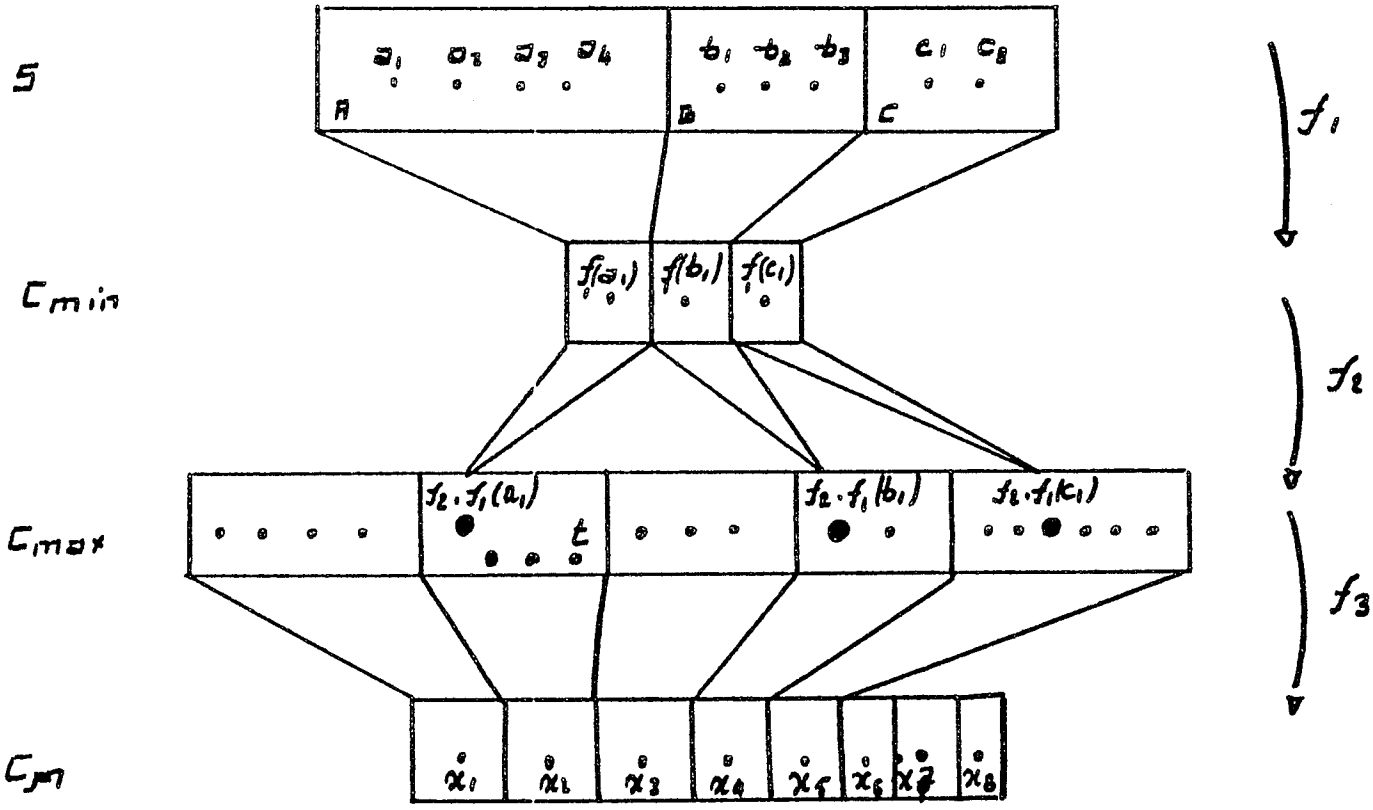
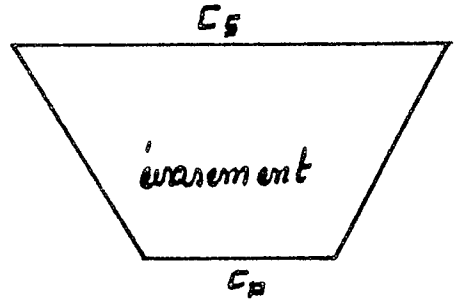


figure 8.



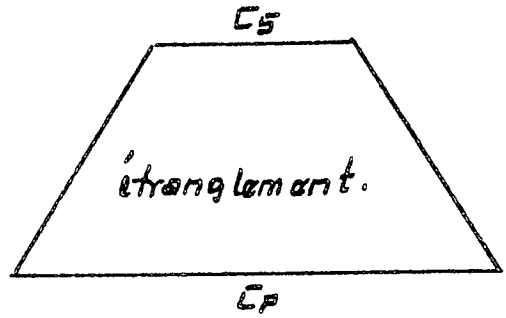


figure 9.

Ce que nous appelons évasement (étranglement) est symétrique de ce que nous avons appelé évasement (étranglement) dans le chap 4 car dans ce cas-ci

$= C(Y_M,P)$, la capacité du canal $E^+_\phi(M)$,

$= W_{\tilde{P}_M \tilde{\Gamma}_P}(Y_M,P)$, d'après la remarque 6 du chapitre 3.

$O(M) = W_{\tilde{P}_M \tilde{\Gamma}_P}(Y_M,P)$: on retrouve la formulation déduite de DEF4 de même que pour la commandabilité , on définit :

Def5 : Le rapport d'observabilité de M : $DO(M) = \frac{O(M)}{H(Y_M)}$

On trouvera à la figure 7 un exemple de calcul de l'observabilité dans le cas où ϕ n'a qu'une valeur.

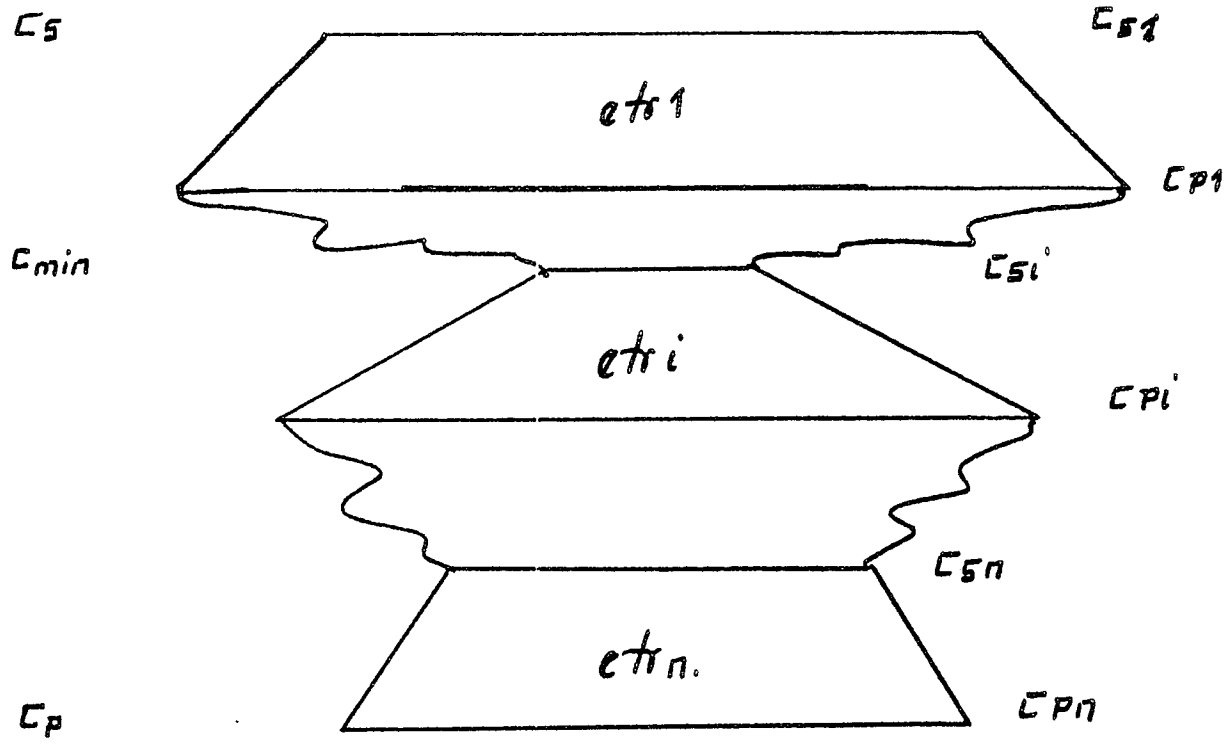
$E^+_\phi(M)$ est le sous graphe construit sur les places numérotées de 1 à 5, P, M et les transitions adjacentes à deux places de cet ensemble Sur chaque arc de $E^+_\phi(M)$ on a porté $f(u)/c(u)$. On a mis en évidence une coupe telle que tous les arcs sortants soient saturés et tous les arcs entrant aient un flot nul :

On a alors : $O(M) = 2$ bits

et $DO(M) = \frac{2}{3}$

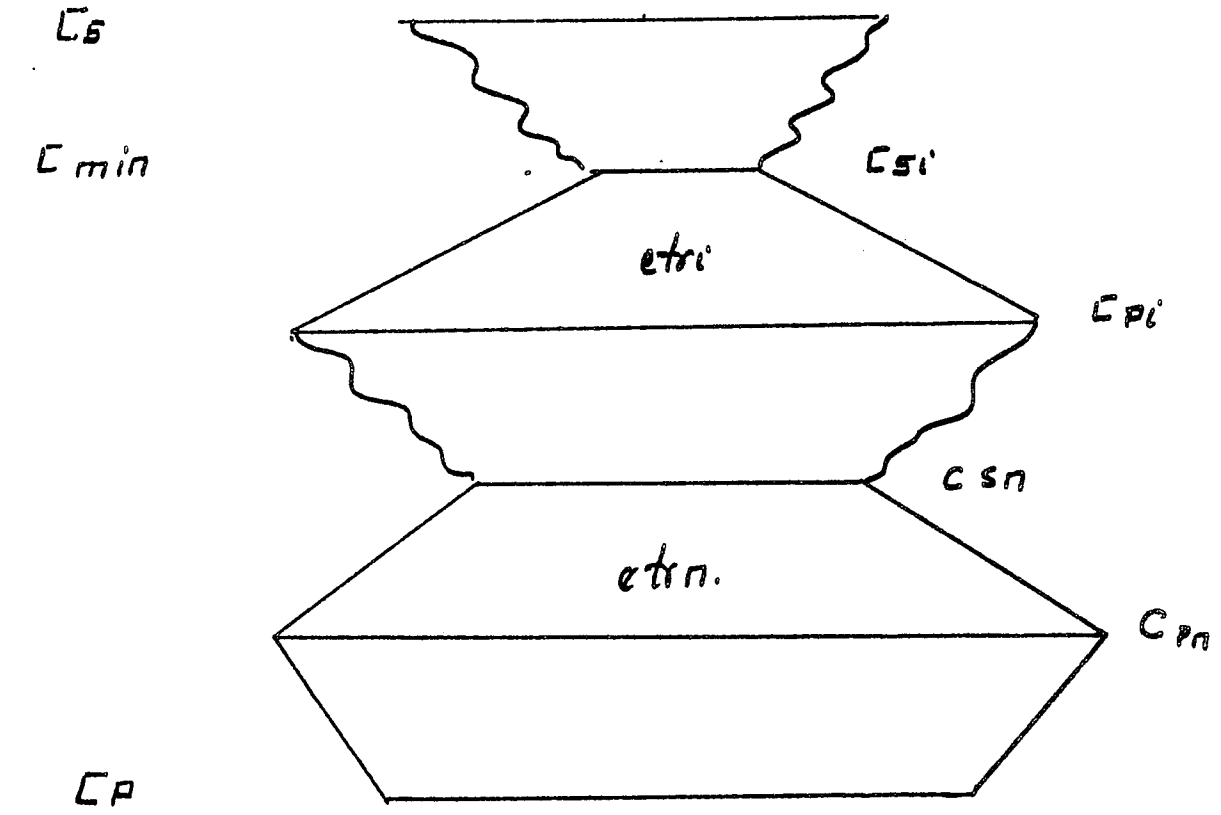
E. Rendement -

E1. Rendement amont : Considérons un module M (d'entrée X_M) que l'on veut tester dans un écoulement E de source S et de puits P. Et soit x un vecteur de test de ce module. On se propose de trouver une valeur s telle que si on l'affiche en S, X_M prend la valeur x. La valeur $s \in S$ est déduite de $x \in X_M$ par propagations d'une coupe à la suivante dans le réseau $E^-_\Pi(M)$. Position du problème :



$$p = 2^{C_{s1} - C_{P1} + C_{si'} - C_{pi} + C_{sn} - C_{PN}} = 2^{C_{si'} - C_{PN}} \cdot 2^{\alpha} = 2^{C_{min} - C_P} \cdot 2^{\alpha}$$

a. $\alpha \leq 0$ donc $2^{\alpha} \leq 1$



$$h = 2^{C_{sn} - C_{Pn} + C_{si'} - C_{Pi'} + \cdots} = 2^{C_{Pn} - C_{si'}} \cdot 2^{\alpha} \leq 2^{C_{Pn} - C_{si'}} \leq C.$$

b.

Figure 10

Soit un réseau R exécutant une fonction f de S à M. Soit $x \in X_M$. Calculer la probabilité p de trouver, par propagation d'une coupe à l'autre, une valeur $s \in S$ telle que $f(s) = x$.

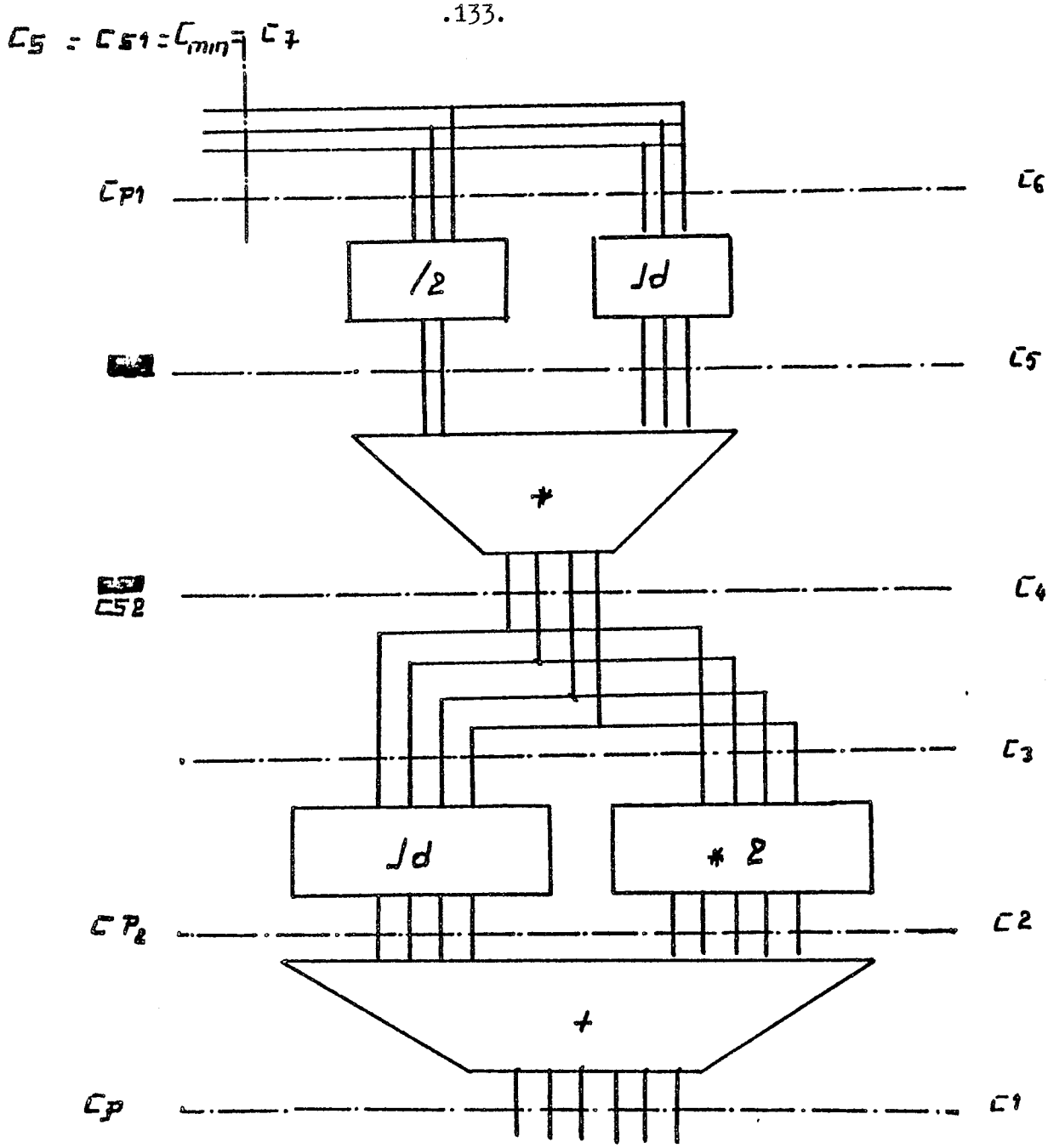
Remarque 3 : La notion de rendement semble avoir un lien avec la commandabilité. Soient $C(X_M)$ et $C(M)$ la capacité de X_M et la commandabilité de M. Alors ${}_2C(M)/{}_2C(X_M)$ mesure, par définition la probabilité qu'un vecteur pris dans X_M ait un antécédent alors que p mesure la probabilité de trouver un antécédent à x en une seule expérience de propagation. Ainsi, sur la figure 8, l'élément x_2 de X_M a 4 antécédents par f ; pourtant si on lui choisit l'antécédent t dans C_{max}, on ne peut plus "remonter" vers S.

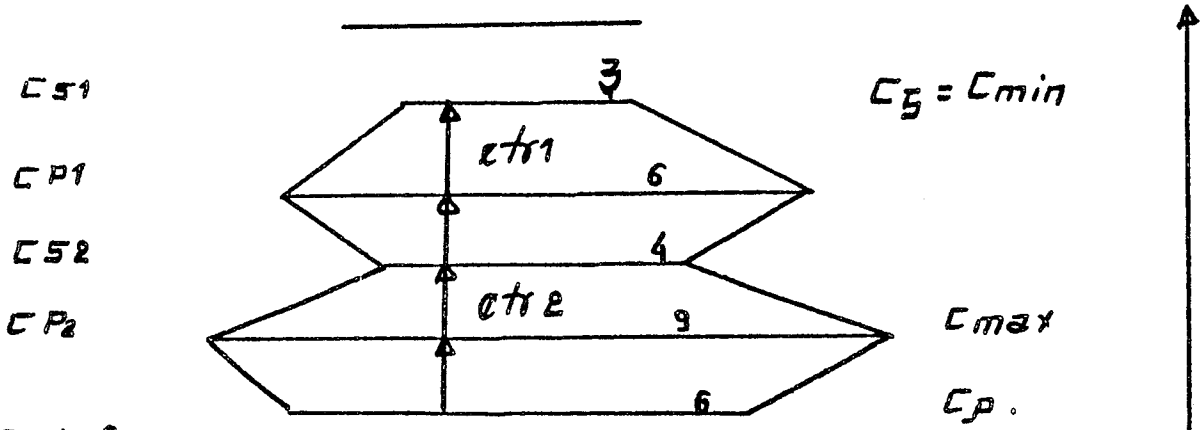
Pour résoudre le problème posé ci-dessus, on pose, par axiome

A1 - Soit un réseau dont les capacités puits et source sont C_P et C_S et qui vérifie, Y_1 et Y_2 étant deux sous ensembles de ses sommets. $[Y_1 \subset Y_2 \iff \gamma(\Omega^+(Y_1)) \geq \gamma(\Omega^+(Y_2))]$; alors tout vecteur de P est commandable en S. Le réseau est alors dit évasement et il est mis en évidence quand la coupe max est C_S et la coupe min est C_P.

A2 - Soit un réseau vérifiant $[Y_1 \subset Y_2 \implies \gamma(\Omega^+(Y_1)) \leq \gamma(\Omega^+(Y_2))]$ alors seulement 2^{C_S} vecteurs de P sont commandables en S. On dit alors que le réseau est un étranglement (voir figure 9).

On montre aisément que l'axiome A1 est vrai sous les conditions C1 C2 C3 et C4. En particulier le réseau compris entre C_{max} et C_M dans la figure 8 ne vérifie pas A1. Car x_6, x_7 et x_8 n'ont pas d'antécédent. Supposons, pour fixer les idées, qu'un seul module exécute la fonction f_3 qui transforme C_{max} en C_M. La sortie de cette place serait sur 3 digits binaires (pour coder les 5 valeurs x_1 à x_5) alors que cette place ne code que cinq valeurs, donc ne vérifie pas la condition C4.





$p = 2^{-6+3+4-9} = 2^{-8}$ si le cheminement est aléatoire.

Profil Informationnel du réseau.

Sens du transfert d'information.

figure 11

Tout réseau peut être décomposé en suite alternée d'évasements et d'étranglements. Soient etr_i les étranglements d'un réseau donné et soient p_i les probabilités pour que un vecteur de P_i soit commandable en S_i. Selon les axiomes A1 et A2, la probabilité cherchée p vaut

$$p = \Pi_{i=1}^{n} p_i. \text{ Or } p_i = \frac{2^{C_{S_i}}}{2^{C_{P_i}}} = 2^{C_{S_i} - C_{P_i}}$$

$$p = \Pi_{i=1}^{n} 2^{C_{S_i} - C_{P_i}}$$

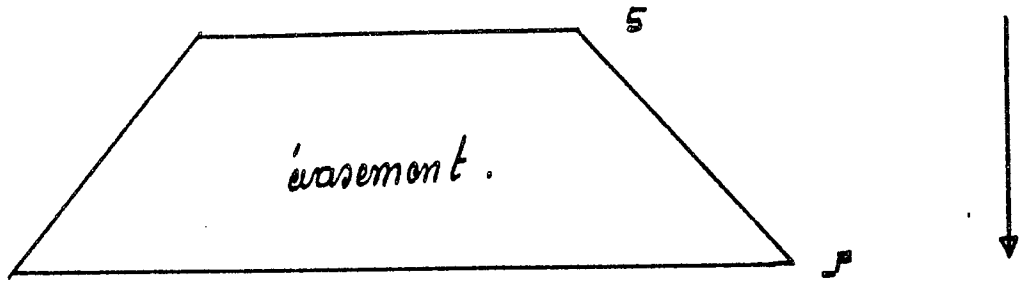
$$= 2^{\Sigma_{i=1}^{n} C_{S_i} - C_{P_i}}$$

Remarque 4 : L'axiome A2 n'est autre que la proposition 14 du chapitre 4, appliquée à un réseau dont on sait que la coupe min est C_S.

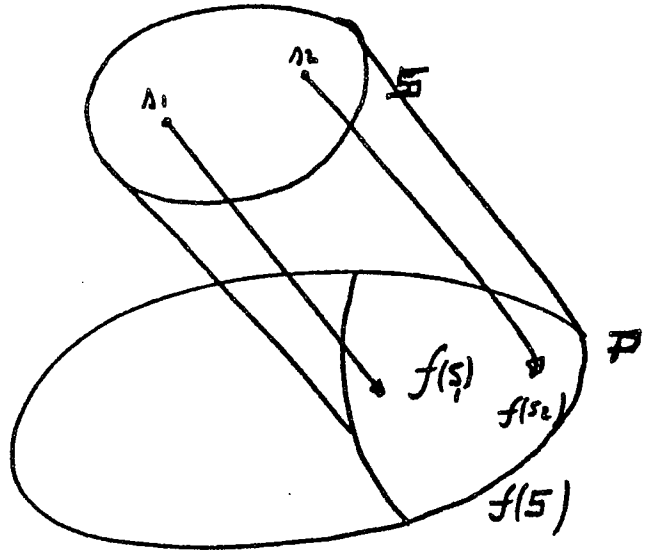
Remarque 5 : On fait référence à la remarque 3. On considère un réseau ayant la source S et le puits P et composé d'étranglements etr_i $1 \leq i \leq n$ et d'évasements - cf. figure 10 - on pose $\rho = 2^{C_{min} - C_P}$ et $p = 2^{\Sigma_{i=1}^{n} C_{S_i} - C_{P_i}}$. Alors deux cas sont possibles.

a) voir figure 10a - Si C_P se trouve être la capacité maximale d'un étranglement, ρ figurera comme facteur dans l'expression de p, et on a $p \leq \rho$.

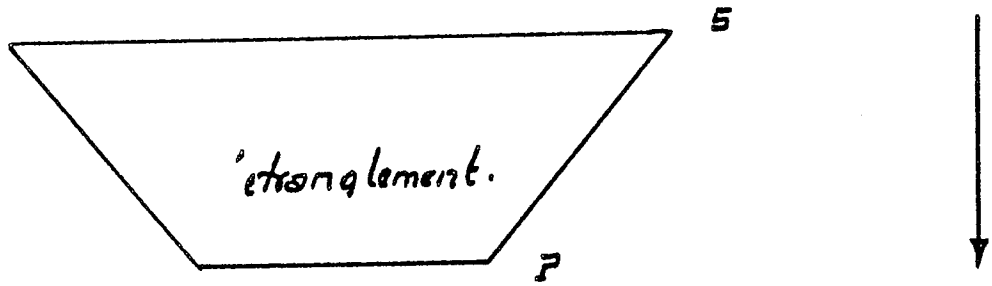
b) voir figure 10b - Si C_P est, au contraire, la coupe min d'un évasement, on peut mettre $2^{C_{min} - C_{S_n}}$ en facteur dans l'expression de p. Vu que $C_P < C_{S_n}$, $\rho = 2^{C_{min} - C_P} \geq 2^{C_{min} - C_{S_1}}$, lui-même étant plus grand que P_1 dans ce cas aussi $p \leq \rho$.



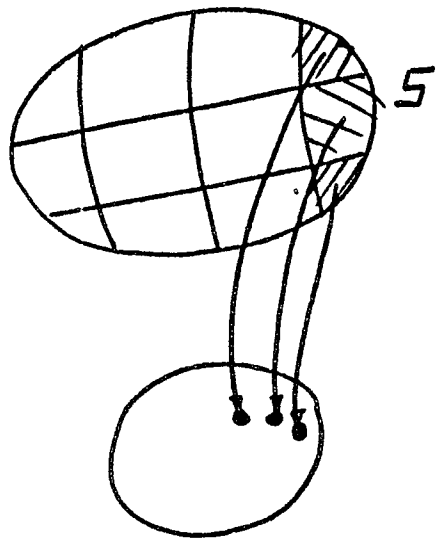
Sens du transfert d'information.



12 - 1



Sens du transfert d'information.



12 - 2.

fig 12

Pour le réseau donné à la figure 11 on donne un exemple de cheminement qui échoue au premier étranglement

en C1 : 8, en C2 : 2,6, en C3 : 2,3, en C4 : échec

un exemple de cheminement qui échoue au deuxième étranglement

en C1 : 12 en C2 : 4,8 en C3 : 4,4 en C4 : 4

en C5 : 2,2 en C6 : 4,2 en C7 : échec.

Un exemple de cheminement qui réussit :

C1 : 24 C2 : 8,16 C3 : 8,8 C4 : 8
C5 : 2,4 C6 : 4,4 C7 : 4

Un exemple de valeur de P non commandable : 5.

Il appartient à l'utilisateur de détailler comme il l'entend la description du profil informationnel du réseau et d'en déduire une formule de p aussi précise qu'il le veut en fonction de son critère coût / performance. On pense que la taille des systèmes courants que l'on doit étudier ne justifie pas de détailler plus que la mise en évidence de C_S C_P, C_{min} et C_{max}. Deux cas se présentent :

a. Si C_{max} est en amont de C_{min} alors $p_a = 2^{C_{min} - C_P + C_S - C_{max}}$

b. Si C_{max} est en aval de C_{min} alors $p_b = 2^{C_{min} - C_{max}}$.

E2. <u>Nombre moyen d'essais</u> - On va utiliser p pour évaluer le nombre moyen d'essais qu'il faut faire afin de trouver un cheminement réalisable d'une valeur en X_M vers une valeur S. Position du problème : Soit un rationnel p tq : $0 \leq p \leq 1$. Soient k boules dont k_1 sont blanches et k_2 sont noires et telles que $p = \frac{k_1}{k}$. Soit à calculer le nombre moyen

de boules qu'il faut tirer sans remise afin de trouver la première boule blanche. Pour simplifier le problème on suppose que $k_1 = 1$. Ceci est d'autant plus légitime que $\frac{1}{p}$ est proche d'un entier ou qu'il est grand. On pose X = la variable aléatoire "nombre de boules noires tirées avant la boule blanche". On se propose de calculer l'espérance de X.

$$P(X = 0) = \frac{1}{k} = p$$

$$P(X = 1) = \frac{k_2}{k} \cdot \frac{1}{k}$$

$$P(X = 2) = \frac{k_2}{k} \cdot \frac{k_2-1}{k} \cdot \frac{1}{k}$$

$$P(X = 3) = \frac{k_2}{k} \cdot \frac{k_2-1}{k} \cdot \frac{k_2-2}{k} \cdot \frac{1}{k}$$

$$P(X = k_2) = \frac{k_2}{k} \cdot \frac{k_2-1}{k} \cdot \frac{k_2-2}{k} \cdots\cdots \frac{k_2-(k_2-2)}{k} \cdot \frac{k_2-(k_2-1)}{k} \cdot \frac{1}{k}$$

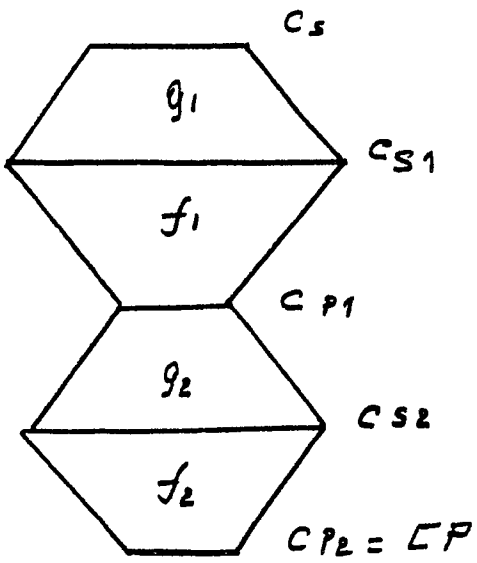
$$P(X = i) = \frac{(k_2)!}{(k_2-i)!k^{i+1}}$$

d'où $E(X) = \sum_{i=0}^{k_2} i \frac{(k_2)!}{(k_2-i)!k^{i+1}}$, soit encore, en fonction de p,

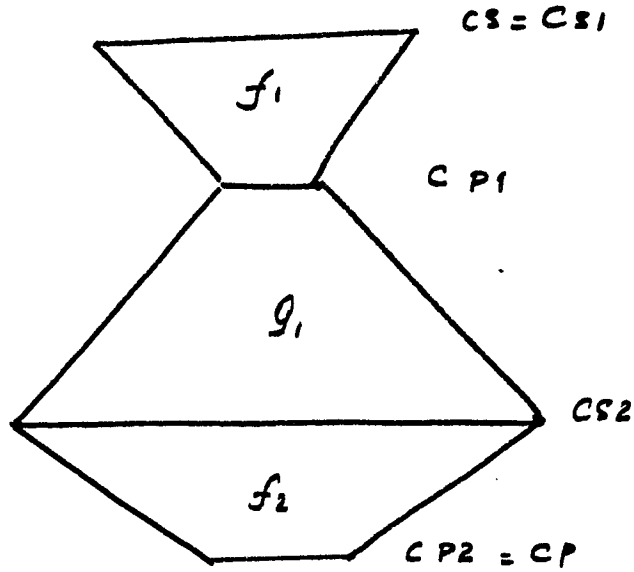
$$E(X) = \sum_{i=0}^{\frac{1-p}{p}} i \left(\frac{1-p}{p}\right)! \frac{1}{\left(\frac{1-p}{p} - i\right)!\left(\frac{1}{p}\right)^{i+1}}$$

sachant bien entendu que $\frac{1}{p} \in \mathbb{N}$, par définition.

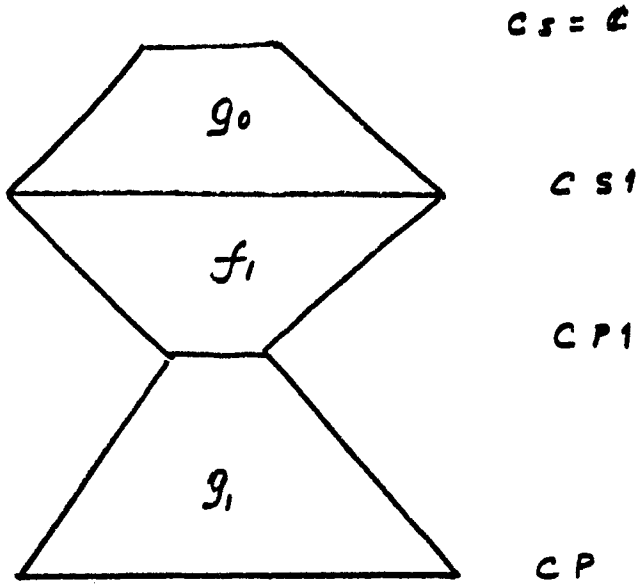
Ainsi, si k représente le nombre de cheminements possibles de X_M vers S et p la probabilité qu'un cheminement choisi au hasard aboutisse jusqu'en S, alors E(X) + 1 mesure la longueur moyenne d'une séquence d'essais



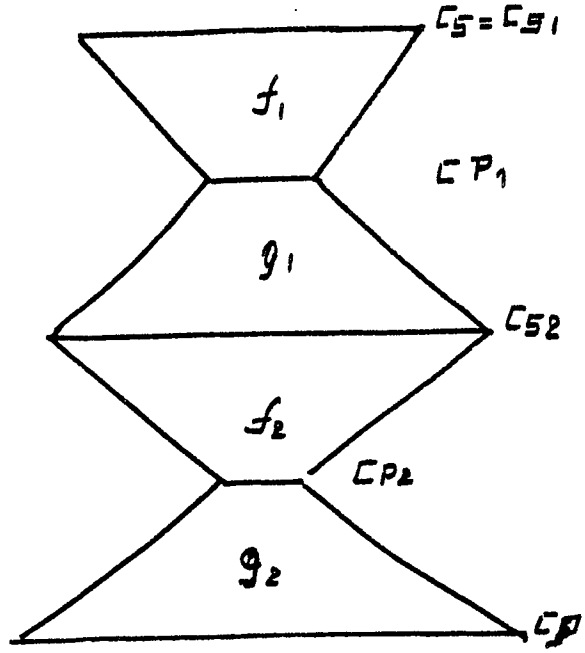
a



b



c



d.

$$\vartheta_{(a)} = \left(\frac{1-2^{-c_{p1}}}{1-2^{-c_{s1}}}\right)\left(\frac{1-2^{-c_{s2}}}{1-2^{-c_{s2}}}\right)$$

$$\vartheta_b = \left(\frac{1-2^{-c_{p1}}}{1-2^{-c_{s1}}}\right) \Big/ \left(\frac{1-2^{-c_{p2}}}{1-2^{-c_{s2}}}\right).$$

$$\vartheta_c = \left(\frac{1-2^{-c_{p1}}}{1-2^{-c_{s1}}}\right)$$

$$\vartheta_d = \left(\frac{1-2^{-c_{p1}}}{1-2^{-c_{s1}}}\right) \Big/ \left(\frac{1-2^{-c_{p2}}}{1-2^{-c_{s2}}}\right).$$

fig 13 -.

nécessaire pour trouver un cheminement qui aboutit. La quantité $\rho = E(X)+1$ sera utilisée plus loin dans le calcul de la charge de travail nécessaire à l'élaboration d'un vecteur de test dans une méthode déterministe.

E3. Rendement aval - Soit un vecteur de test $x \in X_M$ et soient Y_M et $\bar{Y}_M$ les sorties de M dans le cas où M est sain et dans le cas où il est en panne. En supposant que x ne détecte qu'une seule panne, x est un vecteur de test pour le module isolé car Y_M est différent de $\bar{Y}_M$. Une fois que M est plongé dans un écoulement, x continue à être un vecteur de test pour M à deux conditions :

- x est commandable à partir de S (probabilité $2^{C_{min} - C_M}$) dans $(E_{\Gamma}^{-}(M))$.

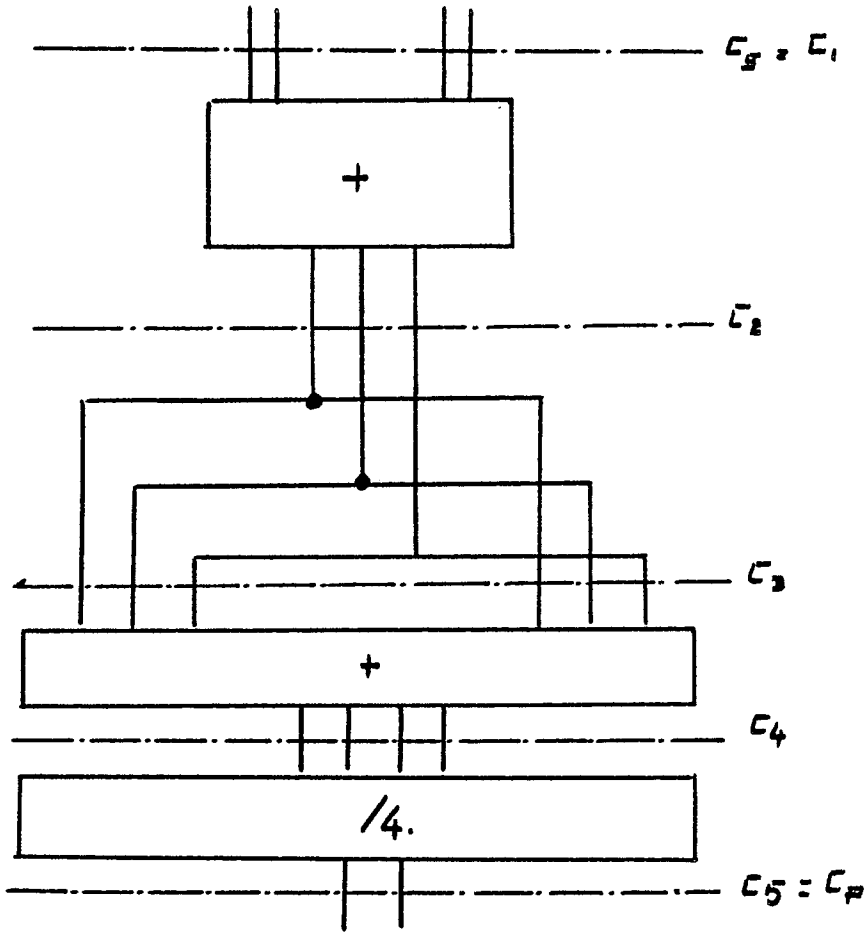
- les images de Y_M et $\bar{Y}_M$, pour la meilleure valeur commandable de U (voir notations du paragraphe C) sont différentes en P. (Ceci se réalise avec la probabilité que nous allons calculer).

Considérons le réseau $E_{\phi}^{+}(M)$ dont on a mis en évidence le profil informationnel, i.e. une suite d'évasements et d'étranglements. On prend les axiomes suivants :

A1. Dans un évasement de source S et de puits P la fonction P = f(S) est injective . i.e. $s_1 \neq s_2 \Rightarrow f(s_1) \neq f(s_2)$.

(Il y a bijection de S à f(S); voir figure 12.1)

A2. Dans un étranglement de source S et de puits P, la probabilité que $f(s_1) \neq f(s_2)$ sachant que $s_1 \neq s_2$ est inférieure ou égale à 1 et se calcule comme suit. Soit S l'ensemble des valeurs que peut prendre S et P l'ensemble des valeurs que peut prendre P. On partitionne S par la relation Γ_S définie par $s_1 \Gamma_S s_2 \Leftrightarrow f(s_1) = f(s_2)$. Alors q = probabilité que s_1 et s_2 étant différents, $f(s_1)$ soit différent de $f(s_2)$ = probabilité que s_1 étant pris dans S, s_2 ne soit pas dans la même classe d'équivalence



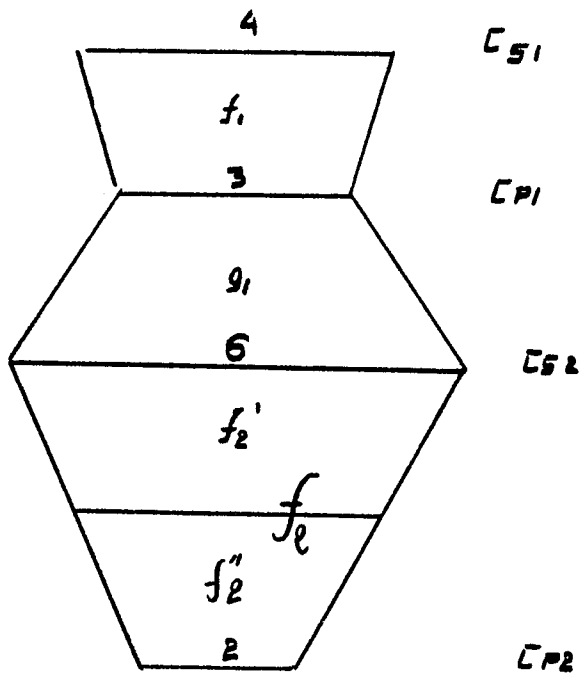


figure 14.

$$q = \left(\frac{1-2^{-3}}{1-2^{-4}}\right)\left(\frac{1-2^{-2}}{1-2^{-6}}\right) = \frac{32}{45}.$$

modulo Γ_S que lui. Par approximation - et parce que la connaissance du réseau ne nous autorise pas à plus que cela - nous supposerons que toutes les classes d'équivalence de S modulo Γ_S ont même cardinal $2^{C_S - C_P}$ - voir figure 12.2.

si bien que $q = \frac{(2^{C_P} - 1) \cdot 2^{C_S - C_P}}{2^{C_P} \cdot 2^{C_S - C_P} - 1} = \frac{2^{C_S} - 2^{C_S - C_P}}{2^{C_S} - 1} = \frac{1 - 2^{-C_P}}{1 - 2^{-C_S}}$

Considérons un réseau dont on a le profil informationnel et supposons qu'il contient k étranglements de source S_i et de puits P_i $1 \leq i \leq k$. Soient deux valeurs s_1 et s_2 de la source, on se propose de calculer la probabilité que leurs images en P soient distinctes. On appellera f_i la fonction définie par le réseau compris dans chaque étranglement entre S_i et P_i, et g_j la fonction exécutée par chaque évasement ; 0 ou $1 \leq j \leq k$ ou k+1. Si les images de s_1 et s_2 en P_i sont distinctes, alors leurs images en S_{i+1} sont aussi distinctes, d'après A1 . Par ailleurs, si les images de s_1 et s_2 en S_i sont distinctes, alors leurs images en P_i ont une probabilité $q_i = \frac{1 - 2^{-C_{P_i}}}{1 - 2^{-C_{S_i}}}$ d'être distinctes si bien que la probabilité que les images de s_1 et s_2 en P soient distinctes est $q = \Pi_{i=1}^{k} q_i$

$$q = \Pi_{i=1}^{k} \left(\frac{1 - 2^{-C_{P_i}}}{1 - 2^{-C_{S_i}}} \right).$$

La figure 13 représente les quatre principaux cas d'alternance entre évasements et étranglements avec les formules correspondantes de q.

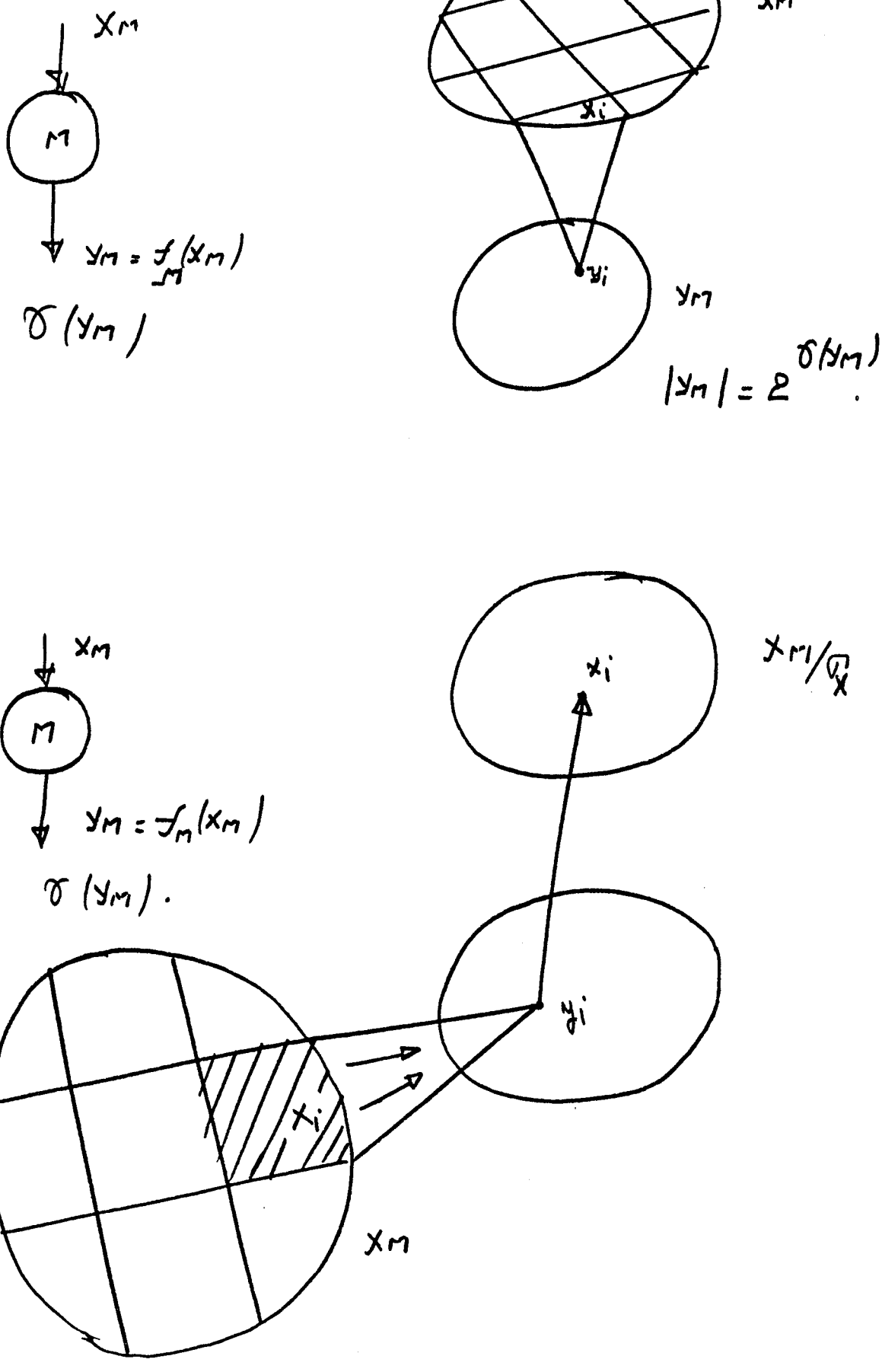

X_m
m
$y_m = f_m(x_m)$
$\sigma(y_m)$
x_m
x_i
y_i
y_m
$|y_m| = 2^{\sigma(y_m)}$.
x_m
m
$y_m = f_m(x_m)$
$\sigma(y_m)$.
x_i
$x_m/\sim_X$
y_i
x_i
x_m

Figure 15.

Considérons la figure 14 . On donne un exemple :

- de cheminement qui échoue au premier étranglement

  - en C1 $s_1 = (2,3)$ $s_2 = (3,2)$
  - en C2 $f_1(s_1) = 5$ $f_2(s_2) = 5$ Echec

- de cheminement qui échoue au deuxième étranglement

  - en C1 $s_1 = (2,0)$ $s_2 = (2,1)$
  - en C2 $f_1(s_1) = 2$ $f_2(s_2) = 3$
  - en C3 $g_1 \circ f_1(s_1) = (2,2$ $g_1 \circ f_1(s_2) = (3,3)$
  - en C4 $f_2' \circ g_1 \circ f_1(s_1) = 4$ $f_2' \circ g_1 \circ f_1(s_2) = 6$
  - en C5 $f_2 \circ g_1 \circ f_1(s_1) = 1$ $f_2 \circ g_1 \circ f_1(s_2) = 1.$ Echec

Le cheminement qui réussit $s_1 = (2,2)$ $s_2 = (2,1)$

$f(s_1) = 2$ $f(s_2) = 1$

F - Travail de propagation -

F1 . Introduction : Soit M un module isolé, x un vecteur de test de ce module et $y_1, y_2, \ldots, y_k$ les vecteurs de sortie de M selon son état de santé. Considérons M dans un écoulement E.

Pour vérifier si x est un vecteur de test de M dans E, on propage x en amont de x_M vers S et $y_1, \ldots, y_k$ en aval, de Y_M vers P moyennant une quantité de travail intellectuel que nous allons tenter d'évaluer. Pour ce faire, nous faisons appel à un article publié par Leo Hellerman dans IEEE transactions on computers et cité en bibliographie [20] . L'essentiel de son contenu est le suivant :

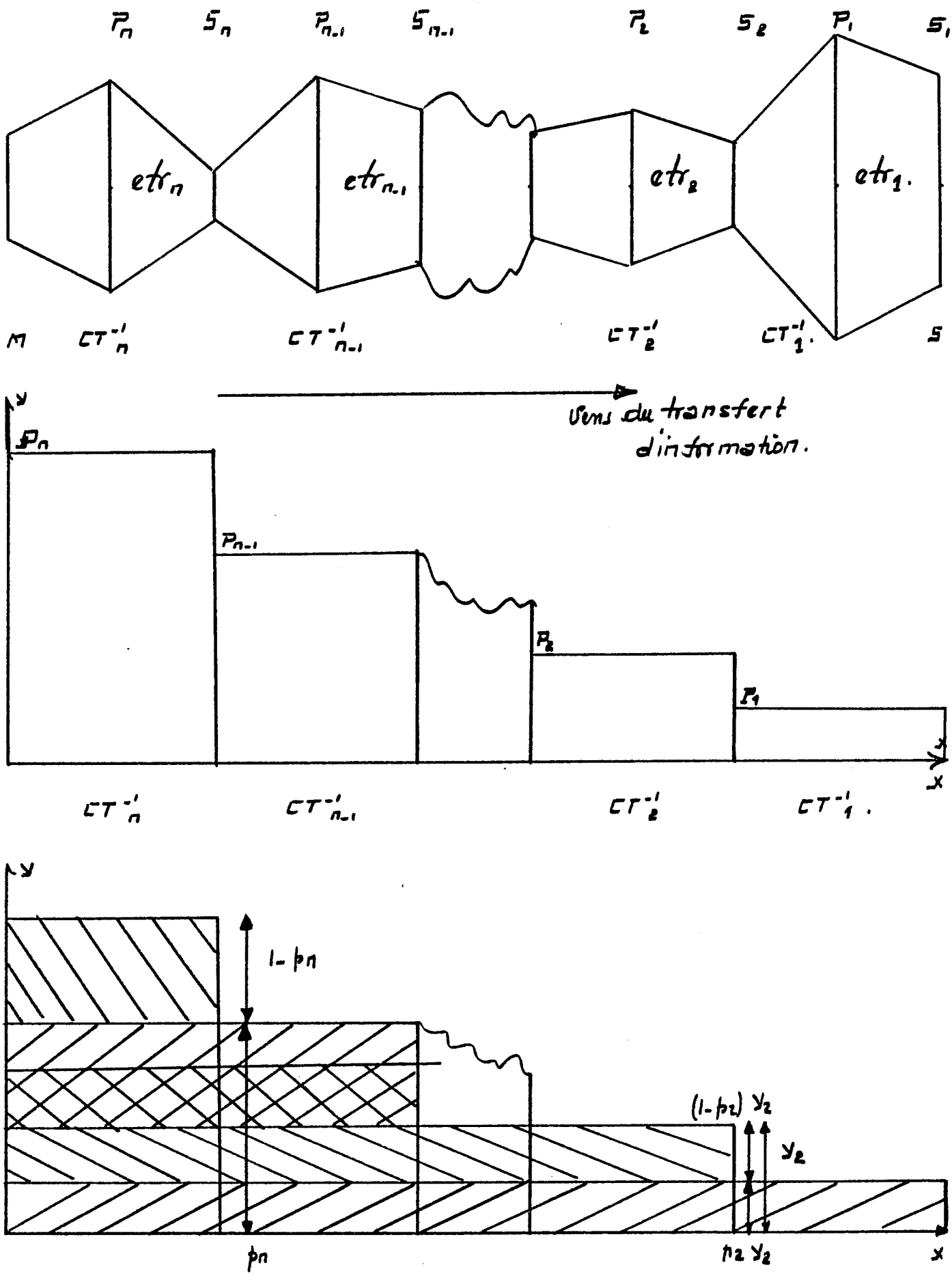

P_n S_n P_{n-1} S_{n-1} P_2 S_2 P_1 S_1
etr_n
etr_{n-1}
etr_2
etr_1.
M CT^{-1}_n CT^{-1}_{n-1} CT^{-1}_2 CT^{-1}_1. S
y
Sens du transfert d'information.
P_n
P_{n-1}
P_2
P_1
x
CT^{-1}_n CT^{-1}_{n-1} CT^{-1}_2 CT^{-1}_1.
y
$1-p_n$
$(1-p_2)\ y_2$
y_2
p_n
$p_2\ y_2$
x

Figure 16.

Soit f une fonction de X vers Y. On partitionne X par la relation Γ_X définie par $x_1 \Gamma_X x_2 \iff f(x_1) = f(x_2)$, et on appelle X_i $1 \leq i \leq n$ les m classes d'équivalence de X modulo Γ_x. Le travail de f est alors par

définition $w(f) = \sum_{i=1}^{n} |X_i| \text{ Log} \frac{|X|}{|X_i|} = |X| \sum_{i=1}^{n} \frac{|X_i|}{|X_i|}$.

Si on pose $p_i = \frac{|X_i|}{|X|}$ $\quad w(f) = X \; H(p_1, p_2, \ldots, p_n)$

Le travail d'une fonction s'entend toujours pour une implémentation particulière. w(f) est calculé pour l'implémentation la plus simple qui soit, à savoir l'implémentation par une table de valeurs, i.e., une table dont les cases sont adressées par les éléments de X et dont le contenu de la case adressée par x est f(x). Il est remarquable que w(f) mesure aussi la quantité d'information fournie par cette table si on tire au hasard ses adresses. w(f) se mesure donc en bit quoi que Hellerman propose d'appeler cette unité le wit afin de spécifier le contexte.

w(t) mesure le travail global d'une fonction définie sur X. Le travail fourni pour trouver l'image d'un élément $x \in X$ et $\bar{w}(t) = \frac{w(f)}{|X|}$

$$\bar{w}(t) = H\left(\frac{|X_1|}{|X|}, \ldots, \frac{|X_k|}{|X|}\right).$$

Soit une bijection $f : X \to Y$, on a $w(f) = |X| \text{ Log}|X|$; $\bar{w}(f) = \text{Log}|X|$.

Soit un module M qui réalise la fonction f_M entre son entrée X_M et sa sortie Y_M. Soit $\gamma(Y_M)$ la capacité de l'arc sortant de M et supposons que f_M soit injective (c'est la condition C4 étudiée dans le chapitre

précédent). Il en résulte $|Y_M| = 2^{\gamma(Y_M)}$

On se propose de calculer le travail de la recherche d'une image y_i à un élément $x \in X_M$ et le travail de la recherche d'un antécédent quelconque x à un élément $y_i \in Y_M$. On appellera $CT(M)$ et $CT^{-1}(M)$ ces deux quantités. voir figure 15.

$$CT(M) = \bar{w}(f_M) = \sum_{i=1}^{n} \frac{|X_i|}{|X|} \text{ Log } \frac{|X|}{|X_i|}$$

On suppose que les X_i ont même cardinal, alors $\frac{|X|}{|X_i|} = 2^{\gamma(Y_M)} = n$

d'où $CT(M) = \bar{w}(f_M) = \gamma(Y_M)$ wit

Par ailleurs, on définit la fonction ϕ de Y_M dans (X_M/Γ_M) par $\phi(Y)$ = la classe d'équivalence de X_M modulo Γ_X dont les éléments x vérifient $f(x) = y$. Rechercher un antécédent quelconque à y consiste à chercher l'image de y par ϕ, qui coûte un travail $\bar{w}(\phi)$, puis prendre un élément x au hasard dans $\phi(y)$, qui coûte un travail nul.

$CT^{-1}(M) = \bar{w}(\phi)$. ϕ étant une bijection, $\bar{w}(\phi) = \text{Log } |Y_M| = \gamma(Y_M)$

$CT^{-1}(M) = \gamma(Y_M)$ wit

F2 - Propagation amont :

Considérons un réseau R de source S et de puits M. On se propose d'évaluer le travail à fournir afin de propager une valeur de M vers la source ou vers un échec à un étranglement ; dans la pratique il s'agira de trouver un antécédent à $x \in X_M$ dans $E_{\Pi}^{-}(M)$.

Considérons le profil informationnel du réseau donné par la succession alternée d'étranglements et d'évasements. On note les étranglements etr_1 à etr_n, etr_1 étant du côté de la source. On appelle S_i et P_i la source et le puits de etr_i et C_{S_i}, C_{P_i} leurs capacités.

On pose $p_i = \frac{2^{C_{S_i}}}{2^{C_{P_i}}}$. On suppose que (voir figure 16).

H1. Pour propager un vecteur s de s_i vers s_{i-1}, on effectue une recherche d'image inverse dans tous les modules séparant s_i de s_{i-1} et que, s'il y a échec, celui-ci se perçoit à la coupe s_{i-1}. On associera donc à chaque portion du réseau contenant un étranglement etr_i et l'évasement qui se trouve à son aval la quantité CT_i^{-1} = (la somme des capacités des lignes de sortie de tous les modules compris entre s_i et s_{i-1}) qui mesure le travail de propagation d'un vecteur de s_i à s_{i-1}) sous l'hypothèse vue ci-haut.

H2. Tout vecteur affichable en s_i a la probabilité $p_{i-1} = \frac{2^{C_{S_{i-1}}}}{2^{C_{P_{i-1}}}} = 2^{C_{S_{i-1}} - C_{P_{i-1}}}$

d'être commandable à partir de s_{i-1}.

Ceci résulte des axiomes A1 et A2 pris dans le paragraphe D1.

Considérons un repère ox, oy sur lequel on représente les points p_i

des coordonnées $\begin{vmatrix} X_i \\ Y_i \end{vmatrix}$ avec $X_i = \sum_{k=i+1}^{n} CT_k^{-1}$; $Y_i = \prod_{k=i+1}^{n} p_k$

pour $1 \leq i \leq n-1$

et $X_n = 0 \qquad Y_n = 1$.

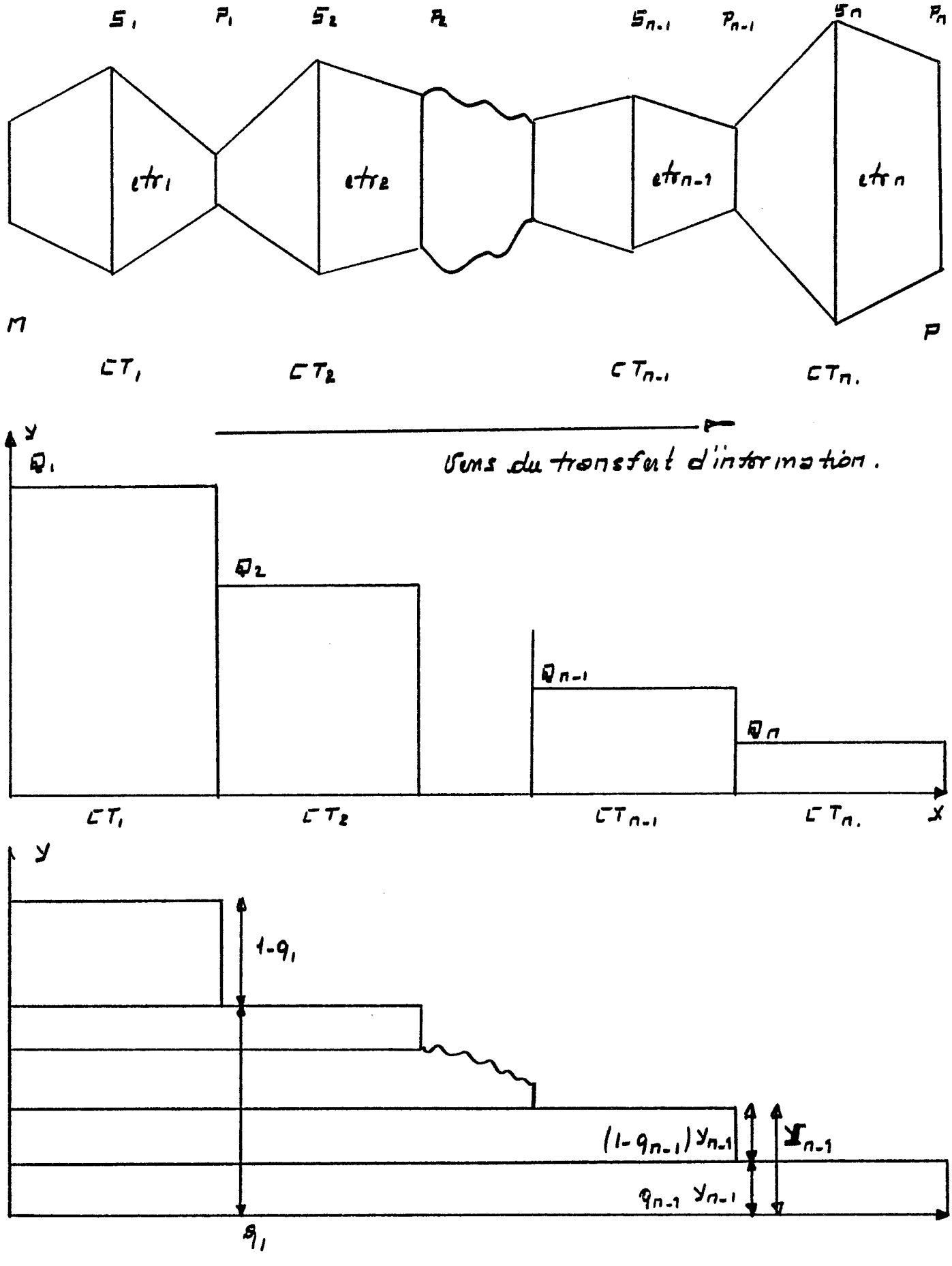

S1
P1
S2
P2
Sn-1
Pn-1
Sn
Pn
etr1
etr2
etrn-1
etrn
M
P
CT1
CT2
CTn-1
CTn.
Sens du transfert d'information.
Q1
Q2
Qn-1
Qn
CT1
CT2
CTn-1
CTn.
X
Y
1-q1
(1-qn-1) yn-1
Yn-1
qn-1 yn-1
q1

figure 17

Ces points sont représentés à la figure 16.

On a $X_{n-1} = CT_n^{-1}$ $\quad Y_{n-1} = p_n$

$X_{n-2} = CT_n^{-1} + CT_{n-1}^{-1}$ $\quad Y_{n-2} = p_n . p_{n-1}$

$\vdots$

$X_2 = CT_n^{-1} + CT_{n-1}^{-1} + \ldots + CT_3^{-1}$ $\quad Y_2 = p_n \cdot p_{n-1} \cdots p_4 p_3$

$X_1 = CT_n^{-1} + CT_{n-1}^{-1} + \ldots + CT_2^{-1}$ $\quad Y_1 = p_n . p_{n-1} \cdots p_3 p_2$

Ces points définissent la fonction en escalier $y = e(x)$ qui vaut Y_i sur l'intervalle compris entre X_i et X_{i-1} (non compris). On affirme :

l'intégrale de e entre 0 et $\sum_{i=1}^{n} CT_i^{-1}$ mesure l'espérance mathématique de la charge de travail qu'il faut exécuter pour essayer une propagation amont à partir de M.

En effet, si on quadrille la surface représentant l'intégrale des droites horizontales d'ordonnées Y_i, on trouve :

$$\int_0^{\sum_{i=1}^{n} CT_i^{-1}} e(x)\, dx$$

$= (p_n\, p_{n-1} \cdots p_2)\, [CT_n^{-1} + \ldots + CT_1^{-1}]$

$+ (p_n\, p_{n-1} \cdots p_3)\, [CT_n^{-1} + \ldots CT_2^{-1}]\, (1 - p_2)$

....

$+ (p_n \qquad\qquad)\, [CT_n^{-1} + CT_{n-1}^{-1}]\, (1 - p_{n-1})$

$+ (1 \qquad\qquad)\, [CT_n^{-1} \qquad\qquad]\, (1 - p_n)$

Or $(p_n \cdot p_{n-1} \ldots p_2)$ mesure la probabilité qu'une propagation "franchisse" les étranglements $etr_n \ldots etr_2$ avec succès et $[CT_n^{-1} + \ldots + CT_1^{-1}]$ mesure la quantité de travail qu'elle aura alors arrêté (d'après H1, une propagation qui échoue en S_1 est aussi coûteuse qu'une propagation qui y réussit).

$(p_n \cdot p_{n-1} \ldots p_3)(1 - p_2)$ mesure la probabilité qu'une propagation "franchisse" les étranglements $etr_n \ldots etr_3$ avec succès et échoue à l'étranglement etr_2, en S_2 ; $[CT^{-1} + \ldots + CT_1^{-1}]$ mesure le travail que cette propagation a alors coûté.

.....

$p_n(1 - p_{n-1})$ mesure la probabilité qu'une propagation franchisse l'étranglement etr_n avec succès mais échoue en S_{n-1} ; $CT_n^{-1} + CT_{n-1}^{-1}$ mesure le travail qu'elle a coûté.

Enfin $(1 - p_n)$ mesure la probabilité qu'une propagation échoue en S_n et CT_n^{-1} mesure le travail de calcul qu'elle coûte.

Dans la figure 16, chaque zone hachurée par des traits parallèles correspond à un terme de l'expression de l'intégrale e(x) donnée ci-haut ; la zone hachurée par des traits croisés correspond aux pointillés dans l'expression de e(x).

La figure 16 présente donc un moyen rapide d'évaluer la charge de travail nécessaire pour propager un vecteur de M vers S ou vers un échec.

F3 - <u>Propagation aval</u> : Soit un réseau $E_\phi^+(M)$ réalisant la fonction f de M à P dans un écoulement E. Un vecteur de test x appliqué à l'entrée X_M de M sort $y_1\ y_2 \ldots y_k$ en Y_M selon que l'état de santé de M est $f_1\ f_2 \ldots$ ou f_k.

Pour trouver quelles pannes x détecte une fois que M est plongé dans E, on calcule l'image de y_1 correspondant à f_1, l'état sain, puis l'image par f de tous les y_i, $2 \leq i \leq k$. Si $f(y_i) \neq f(y_1)$, on dira que x détecte la panne f_i. (Pour discriminer les pannes individuellement, on peut s'intéresser à comparer les $f(y_i)$, $2 \leq i \leq k$ entre eux. On étudiera exclusivement le problème de comparer $f(y_1)$ à $f(y_i)$ $2 \leq i \leq k$).

Le problème de la propagation aval consiste donc en k-1 problèmes qui se modélisent comme suit :

Soit un vecteur $\tilde{y}$ de Y_M et soient $\tilde{y}_1$ $\tilde{y}_2$ ... $\tilde{y}_n$ les valeurs des images de y au puits P_i des étranglements etr_1 etr_2 ... etr_n respectivement - voir figure 17. On se propose de chercher à savoir si l'image d'un vecteur $y^* \in Y_M$ par f est distincte ou non de celle de y_1. Trouver la solution la plus économique qui permet de conclure si oui ou non $f(\tilde{y}) = f(y^*)$ et évaluer son coût.

Résolution : Considérons le profil informationnel du réseau. On appelle etr_1 etr_2 ... etr_n les étranglements du réseau, S_i, P_i la source et le puits de l'étranglement etr_i et C_{S_i} C_{P_i} leurs capacités. On a $\forall$ i $C_{S_i} > C_{P_i}$ et on appelle y_i l'image de y à la coupe P_i. On suppose :

H1 - Si $\tilde{y}_{i-1}$ et y^*_{i-1} sont distincts, alors leurs images à toutes les coupes qui se trouvent en amont de P_i sont distinctes (on dit que la coupe C1 est en amont de la coupe C2 si et seulement si les ensembles de sommets Y_1 et Y_2 contenant M sur lesquels sont construits C1 et C2 vérifient $Y_1 \subset Y_2$). Seules leurs images en P_i peuvent être égales, et ce avec la

probabilité $1 - q_i = 1 - \dfrac{1 - 2^{-C_{P_i}}}{1 - 2^{-C_{S_i}}}$. Ceci résulte des axiomes A1 et A2

du paragraphe D3 et des développements qui l'ont suivi.

H2. La charge de travail nécessaire pour trouver y_i connaissant y_{i-1} sera notée CT_i et vaut la somme des capacités sortantes de tous les modules compris entre P_{i-1} et P_i.

Alors, si les images de $\bar{y}$ et y^* sont égales à une coupe donnée, elles sont égales à toutes les coupes en aval de celle-ci (i.e. à toute coupe bâtie sur un ensemble de sommets contenant celui de cette coupe). donc on propage y^* jusqu'à ce que l'on trouve une coupe où l'image de y^* est égale à l'image de $\tilde{y}$.

D'après H1, il n'est pas utile de vérifier si les images de $\tilde{y}$ et y^* sont égales ailleurs que sur une coupe parmi les n P_i : c'est cela la solution la plus économique. On se propose d'évaluer son coût : Pour cela on considère le graphe 0x 0y sur lequel on porte les points $Q_1\ Q_2\ \ldots\ Q_n$ de coordonnées $\binom{X_1}{Y_1} \ldots\ldots \binom{X_n}{Y_n}$

avec $X_i = \sum_{k=1}^{i-1} CT_k$

$Y_i = \prod_{k=1}^{i-1} q_k$ pour $2 \leq i \leq k$ et $X_1 = 0$, $Y_1 = 1$

on a $X_1 = 0$ $Y_1 = 1$

$X_2 = CT_1$ $Y_2 = q_1$

...

$X_{n-1} = CT_1 + CT_2 + \ldots + CT_{n-2}$ $Y_{n-1} = q_1, q_2, \ldots, q_{n-2}$

$X_n = CT_1 + CT_2 + \ldots + CT_{n-1}$ $Y_n = q_1, q_2, \ldots, q_{n-1}$

Les points Q_i définissent une fonction en escalier $y = e(x)$ qui vaut Y_i entre X_i et X_{i+1}. On annonce : l'intégrale de e entre 0 et $CT = \sum_{i=1}^{n} CT_i$ vaut l'espérance mathématique de la charge de travail nécessaire pour vérifier si le vecteur y_i détecte la panne f_i une fois que le module est plongé dans l'écoulement E (sachant bien entendu que l'on opère par la procédure proposée plus haut).

Pour démontrer ce résultat, on quadrille l'aire représentant cette intégrale par des droites horizontales d'abcisses Y_i et on écrit :

$$\int_0^{CT} e(x)dx.$$

$$= [CT, \quad] (1 - q_1)$$

$$+ [CT_1 + CT_2 \quad] q_1(1 - q_2)$$

...

$$+ [CT_1 + CT_2 + \ldots + CT_{n-1}] (q_1 q_2 \ldots q_{n-2})(1 - q_{n-1})$$

$$+ [CT_1 + CT_2 + \ldots + CT_n \ ] (q_1 q_2 \ldots q_{n-1})$$

Or

$(1 - q_1)$ mesure la probabilité que y_1^* soit égal à $\tilde{y}_1$; CT, mesure le travail fourni pour trouver y_1^* à partir de y^*.

$q_1(1 - q_2)$ mesure la probabilité que $y_1^* = \tilde{y}_1$ et $y_2^* = \tilde{y}_1$; $CT_1 + CT_2$ mesure le travail fourni pour trouver y_2^* à partir de y^*.

...

$(q_1 q_2 \ldots q_{n-2})\ (1 - q_{n-1})$ mesure la probabilité que $y_i^* = \tilde{y}_i$ pour $1 \leq i \leq n-2$ et $y_{n-1}^* = \bar{y}_{n-1}$. L'échec de la propagation s'est fait en

P_{n-1}° $[CT_1 + CT_2 + \ldots + CT_{n-1}]$ mesure le travail fourni pour calculer y_{M-1}^* à partir de y^*.

$(q_1 \bullet q_2 \ldots q_{n-1})$ mesure la probabilité que la propagation de y^* franchisse avec succès tous les étranglements $etr_1 \ldots etr_{M-1}$. Qu'elle ait, en plus, franchi ou non l'étranglement etr_n, la charge de travail est de toute façon $CT = \sum_{i=1}^{n} CT_i$.

La figure 17 montre un procédé très simple permetatnt d'évaluer la charge de travail d'une propagation aval faite sous les conditions proposées et à laquelle il faudrait éventuellement ajouter CT, la charge de travail nécessaire à l'élaboration de $f(\tilde{y})$ et des images de y aux diverses coupes P_i du réseau.

F4 - <u>Conclusion</u> : Le paramètre qui a été établi en F2 et qu'on notera C_m mesure l'espérance de la charge de travail de propagation d'un vecteur de x_M (l'entrée de M) vers S, sachant que cette propagation peut échouer avant S. Il serait en fait plus intéressant de calculer l'espérance γ_m de la charge de travail de propagation amont nécessaire pour trouver une propagation qui réussit. On propose pour γ_m une formule doublement approchée.

- $\gamma_m = C_m \times$ l'espérance du nombre d'essais nécessaires pour trouver une propagation qui réussit jusqu'en S. Cette égalité est approximative car elle est le produit de deux moyennes faites sur la même loi.

- En outre, si on appelle p la probabilité de propagation calculée au paragraphe E1, on pose, par approximation $\frac{1}{p}$ = l'espérance du nombre d'essais nécessaires pour trouver une bonne propagation. En effet, si l'on pense que la formule donnée au paragraphe E2 est elle-même approximative, on se convaincra que la formule exacte est très compliquée.

Ainsi , $\gamma_m = \frac{1}{p} C_m$.

G. Conclusion -

Les paramètres que nous venons d'élaborer présentent des caractéristiques communes que nous allons exposer :

- Ils ont été étudiés sur une description très sommaire du système, donc une description facile à obtenir et qui contient beaucoup moins d'information que le schéma logique.

- Ils sont tous très faciles à interpréter physiquement : ceci est une garantie supplémentaire de leur validité.

- Une fois que les coupes min et max sont mises en évidence sur le réseau, les paramètres se calculent de façon très simple. Or dans les réseaux de faible taille ces coupes peuvent être trouvées très rapidement et dans les réseaux de grande taille il est possible d'appliquer des codes performants pour les trouver. En outre le profil informationnel du réseau peut éventuellement servir à d'autres fins qu'à l'élaboration de paramètres de test.

CHAPITRE 5

ELABORATION DE POLITIQUES DE TEST

Dans ce chapitre, on se propose de définir, à partir des paramètres de test mis au point au chapitre 5, des critères numériques de décision auxquels on appliquera des outils mathématiques de prise de décision afin d'aider à concevoir un test. Il n'est pas question d'élaborer une politique, ce qui nécessiterait d'ailleurs de connaître le système que l'on se propose de tester ainsi que les moyens que l'on s'est donnés pour cela, mais de faire un ensemble de propositions qui aident le concepteur du test à prendre sa décision aux diverses étapes de la conception.

A. Test déterministe.

Dans ce paragraphe on propose des solutions aux divers problèmes qui se posent dans le cadre d'un test déterministe.

A1. Choisir une couverture

A11. Optimisation discrète : On considère un ensemble fini L à k éléments $l_1, \ldots, l_k$ et on définit une fonction $\gamma : L \to \mathbb{R}$. L'optimisation discrète,

sous sa forme la plus générale consiste à chercher l'élément l de L qui réalise le min de $\gamma(l_i)$ pour $1 \leq i \leq k$. Le cas particulier du problème d'optimisation discrète qui nous intéresse est celui où L est un sous ensemble de $[0,1]^N$ défini par $A.l = b$ où A est une matrice booléenne [m × N] et b un m-vecteur colonne booléen et où γ est une application définie par un m-vecteur ligne $C \in \mathbb{R}^N$ par $\gamma(l) = C.l$.
Le problème s'écrit :

$$A\,l \geq b$$
$$C\,l = \gamma_{max}$$

Quand la taille de la matrice le permet, on peut calculer $\gamma(l)$ pour les valeurs - 2^n au maximum - que peut prendre l. Cette méthode cède rapidement le pas à des méthodes d'énumération implicite dont on a construit des implémentations très performantes.

A12. Application : Soit un réseau R sur lequel AT a mis en évidence n écoulements $e_1, \ldots, e_n$. On suppose que R a m places. On définit un m-vecteur colonne booléen l et on établit une bijection entre les 2^n valeurs de l et les 2^n partitions de $(e_1, \ldots, e_n)$ définie par : A toute partie de $(e_1 \ldots e_n)$ on associe l tel que $l_i = 1$ si e_i appartient à la partition considérée, 0 sinon.

On définit le m-vecteur colonne U tq $u_i = 1$ et on appelle A la matrice de couverture définie par les écoulements $e_1 \ldots e_n$:

A est une matrice m-lignes, n-colonnes telle que $A_{ij} = 1$ si et seulement si l'écoulement e_j couvre la i° place. Un vecteur l est en bijection avec une couverture du réseau si et seulement si $A.l \geq U$ (condition de réalisibilité).

Il appartient à l'utilisateur d'évaluer , selon la méthode de test envisagée et en utilisant les paramètres de test proposés au chapitre 5, une quantité numérique associée à chaque écoulement et qui mesure sa charge globale de travail pour la méthode de test envisagée. Soit C le n vecteur ligne

dont la composante C_i mesure le travail nécessaire pour l'écoulement e_i. La charge de travail nécessaire pour tester le sous ensemble de $(e_1 \ldots e_n)$ défini par l est $\sum_{i=1}^{n} l_i C_i$ ou encore le produit scalaire C.l.

La recherche d'une couverture qui minimise la charge globale de travail s'écrit donc :

$$A\,l \geq U \qquad l_i = 0 \text{ ou } 1$$
$$C\,l = w \text{ minimiser.}$$

Cette formulation du problème est justiciable des méthodes de séparation et évaluation progressive quoi que, une fois que l'on a enlevé les colonnes obligatoires de la matrice A (celles telles qu'il y a une place qu'elles sont seules à couvrir) ainsi que les lignes correspondant aux places que ces écoulements couvrent, la taille de la matrice obtenue devient en général telle qu' il est possible d'opérer par énumération exhaustive des solutions réalisables.

A2. Associer un écoulement à chaque Place.

A21. Décision floue : On se propose d'appliquer une méthode qui permet de déterminer la solution optimale dans un problème de choix multicritère. Cette méthode s'appelle "ElectreII" et elle est extraite de [19].

"ElectreII permet d'établir un classement sur un ensemble fini d'éléments (candidats ou objets) en se basant sur plusieurs critères non aisément réductibles à un seul". "Soit (a, b,..., x, y, z ...) l'ensemble des p éléments à classer : 1,2,..., i, j, ...n sont les n critères selon lesquels chaque élément est noté. Diverses échelles de notation sont possibles". "L'échelle pouvant changer avec le critère i, évaluer un élément x selon le critère i revient à donner une note $g_i(x)$ qui appartient à l'échelle choisie pour le critère i. Il est possible que certains des n critères

Poids des Critères	Eléments à classer / Critères	a	b		x	y	z.
P(1)	1	|	|			|	|
P(2)	2						
	⋮						
P(i)	i	$g_i(a)$	$g_i(b)$	———		$g_i(y)$	$g_i(z)$
P(j)	j	$g_j(a)$	$g_j(b)$	———		$g_j(y)$	
	⋮						
P(n)	n						

fig 1

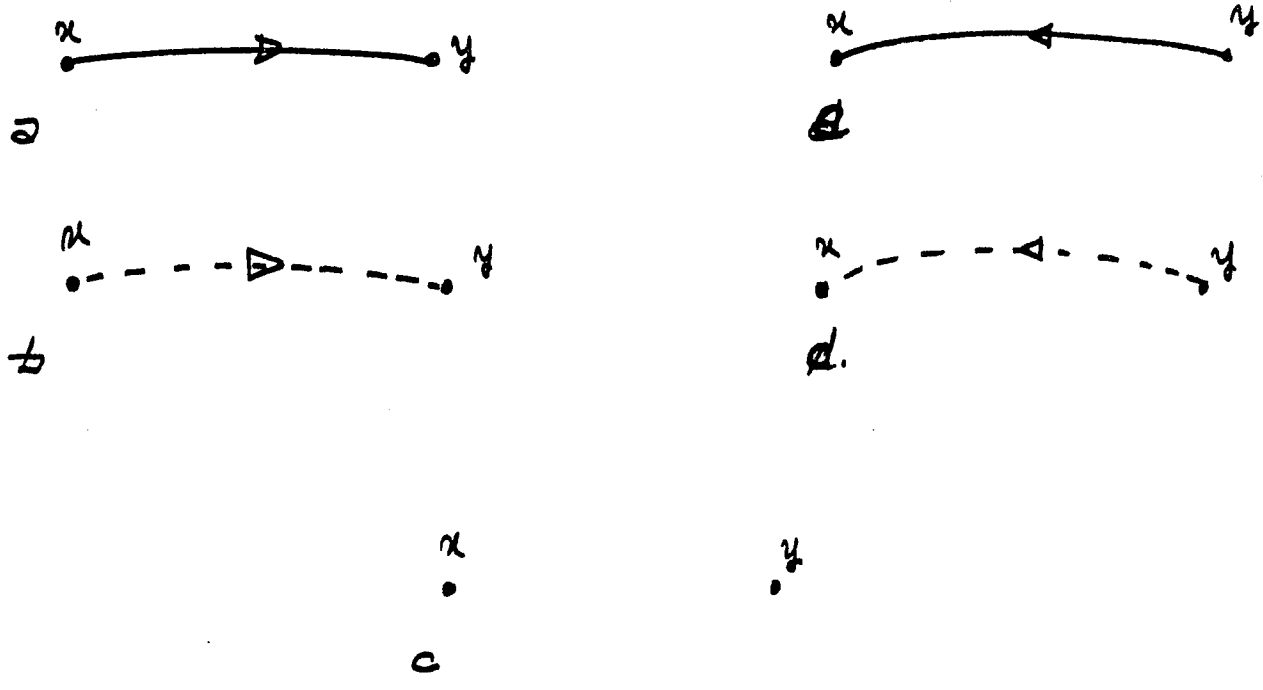


fig 2.

(ou des points de vue) aient plus d'importance que d'autres. On traduit ces choix politiques du décideur en affectant chaque critère i d'un poids noté P(i) et dont la valeur est d'autant plus élevée que le décideur désire donner plus d'importance au critère i considéré (P(i) = 0 traduit le fait que le critère i n'est pas pris en compte.

Toutes ces données peuvent se résumer dans le tableau de la figure 1.

"La relation de surclassement :

Il va de soi qu'on conclura à la supériorité d'un élément x sur un autre, y, lorsque l'élément x est au moins aussi bon que l'élément y selon chaque critère. Mais lorsque :

- l'élément x est meilleur que l'élément y selon certains critères
- l'élément y est meilleur que l'élément x selon d'autres critères
- les éléments x et y sont équivalents selon un troisième groupe de critères,

alors il est utile de définir les conditions qui permettent de conclure à la supériorité globale de l'un des éléments sur l'autre ou qui ne permettent pas une telle conclusion, le risque d'erreur est grand. D'après électre II, on dira que x surclasse u ou est meilleur ou est préférable (on notera x > y), si et seulement si les deux conditions suivantes sont simultanément réalisées :

a - Condition de concordance : "la somme des poids des critères selon lesquels x est considéré comme au moins aussi bon que y est suffisamment élevée ;

b - condition de non discordance : il n'y a pas de discordance majeure venant à l'encontre du consensus précédent, ce qui se traduit par le fait que pour tout critère selon lequel x est moins bon que y, la différence de valeur n'est pas trop importante".

"1.3.1. La condition de condition de concordance

Soit P^+ la somme des poids des critères pour lesquels x est mieux noté que y $P^+ = \sum_i P(i)$ pour Hi tq $g_i(x) > g_i(y)$

Soit P^- la somme des poids des critères pour lesquels x est moins bien noté que y $\quad P^- = \sum_i P(i)$ pour tout i tel que $g_i(x) < g_i(y)$

Soit enfin $P^=$ la somme des poids des critères pour lesquels x et y ont la même note $P^= = \sum_i P(i)$ pour tout i tq $g_i(x) = g_i(y)$. La condition de concordance est vérifiée si et seulement si les deux inégalités suivantes sont simultanément vérifiées :

1°) $\dfrac{P^+}{P^-} \geq 1$

2°) $\dfrac{P^+ + P^=}{P^+ + P^= + P^-} \geq C$ c'est-à-dire si et seulement si une certaine majorité se dégage en faveur de l'élément x. La constante C est appelée seuil de concordance. Plus cette constante est faible, plus la condition de concordance est facile à vérifier.

1.3.2. La condition de non discordance

Cette condition est vérifiée si et seulement si, pour tout critère où l'élément x est moins bien noté que l'élément y_1 l'avantage de ce dernier n'est pas trop grand :

$g_i(y) - g_i(x) \leq D_{i,g_i(x)}$ pour tout i tq $g_i(x) < g_i(y)$

c'est-à-dire si et seulement si il n'y a pas d'opposition majeure à une préférence de l'élément x.

Le terme $D_{i,g_i(x)}$ est appelé paramètre de discordance ; les indices i et $g_i(x)$ signifient que ce paramètre dépend de l'échelle de notation du critère i considéré et aussi de la valeur de la note $g_i(x)$ donnée à l'élément x.

Plus les valeurs que prend ce paramètre sont élevées, plus la condition de non discordance est facile à vérifier.

1.3.3. Les paramètres de sur classement

En fixant le seuil de concordance C et le paramètre de discordance $D_{ig_i(x)}$ à divers niveaux, ELECTRE II permet de distinguer un sur-classement fort et un sur-classement faible.
On fixe trois niveaux : C1 C2 C3 pour le seuil de concordance, tels que $1 > C1 \geq C2 \geq C3 > 0$ et deux niveaux $D^1_i\ g_i(x)$ et $D^2_i\ g_i(x)$ pour le paramètre de discordance, tels que $0 \leq D^1_i\ g_i(x) \leq D^2_i\ g_i(x)$ pour tout i et $g_i(x)$.

Le surclassement est dit "fort" lorsque les deux paramètres qui interviennent dans les conditions de surclassement sont :

$C = C_1$ et $D_{i,g_i(x)} = D^2_{i,g_i(x)}$ ou $C = C_2$ et $D_{i,g_i(x)} = D^1_{i,g_i(x)}$.

Le classement est dit "faible" lorsque ces paramètres sont

$C = C3$ et $D_{i,g_i(x)} = D^2_{i,g_i(x)}$ ou, en considérant que pour

$C = C1$, la concordance est forte

$C = C2$, la concordance est moyenne

$C = C3$, la concordance est faible,

$D_i\ g_i(x) = D^1_i\ g_i(x)$, la concordance est faible,

$D_i\ g_i(x) = D^2_{i,g_i}(x)$, la concordance est moyenne,

Le surclassement est fort lorsque :

- la concordance est forte et la discordance faible ou moyenne.
- la concordance est moyenne et la discordance est faible.

Le surclassement est faible lorsque :
- la concordance et la discordance sont moyennes
- la concordance est faible et la discordance est faible ou moyenne

Les valeurs standard proposées pour les seuils de concordance sont
C1 = 3/4 C2 = 2/3 C3 = 3/5.

Pour les paramètres de discordance, la solution la plus simple est de leur attribuer des valeurs constantes qui seront, bien entendu, fonction de l'échelle de notes considérée. Les valeurs standard proposées sont :

Echelle	0.4	D_i^1	2	D_i^2	3.
	0.8		3		4.
	0.12		5		8.
	0.20		9		14".

1.3.4. Construction du graphe de surclassement

Il ressort de ce qui précède que deux éléments x et y peuvent entretenir les rapports suivants :

- x est fortement meilleur que y (ou surclasse fortement y), ce qui s'écrit x F y et se traduit sur un graphe par "la figure 2a.

-"x est faiblement meilleur que y (ou surclasse faiblement y), ce qui s'écrit x f y et se traduit sur un graphe par "figure 2b.

-"x et y ne sont pas comparables compte tenu de l'information disponible, ce qui se traduit sur un graphe par " figure 2c.

-"y est faiblement meilleur que x (ou surclasse faiblement x), ce qui s'écrit y f x et se traduit sur un graphe par " la figure 2d.

-"y est fortement meilleur que x(ou surclasse fortement x), ce qui s'écrit y F x et se traduit sur un graphe par " la figure 2e.

"1.4. Le classement

Le surclassement fort offrant une meilleure sécurité, on va chercher un classement compatible avec le graphe du surclassement fort :

$\forall x, \forall y \quad x \; F \; y \iff C(x) < C(y)$" où C(x) est le rang de x.

"Un tel classement étant donné, les relations de surclassement faible ne seront prises en compte que dans la mesure où elles permettront de départager les ex-aequo.

Première conséquence : Lorsque le graphe de surclassement fort comporte un circuit, il ne peut exister de classement compatible avec ce graphe : tous les pôles du circuit sont considérés comme ex-aequo et on opère sur le graphe réduit (le représentant unique de la classe ainsi constituée dominera tout élément dominé par l'un au moins des éléments du circuit. Il sera dominé par tout élément dominant l'un d'entre eux au moins).

Deuxième conséquence : En l'absence de préférence directe, la préférence indirecte est suffisante pour classer deux éléments. Supposons par exemple qu'après réduction éventuelle du graphe on ait " "x F y y F z, z F t ; Ces données sont suffisantes pour autoriser à écrire C(x) < C(t) et ceci que l'on ait ou non la relation de préférence directe x F t (la relation t F x est exclue, le graphe étant supposé réduit - chaque composante fortement connexe réduite en un sommet).

1.4.1. Le classement direct :

D'abord, on ne tient compte que du graphe de surclassement fort. Les sommets sont classés en fonction de la longueur des chemins incidents les plus longs qui y aboutissent dans l'ordre croissant de ces longueurs : le sommet qui n'est surclassé fortement par aucun autre est classé premier. Le sommet extrémité du chemin le plus long est classé dernier. On départage ensuite les ex-aequo par les surclassements faibles. La formule donnant le classement C(x) du sommet x s'écrit C(x) = L(x) + 1 où l(x) est la longueur du plus long chemin aboutissant au sommet x. On notera C'(x) le rang de x dans le classement direct".

"1.4.2. Le classement inverse

D'abord on ne tient compte que du graphe de surclassement fort. Les sommets sont classés en fonction de la longueur des chemins les plus longs qui en sont issus dans l'ordre décroissant de ces longueurs : le sommet origine du chemin le plus long est classé premier et tout sommet n'en surclassant aucun autre est classé dernier. Comme pour le classement direct, on départage les ex-aequo par les surclassements faibles.

La formule donnant le classement C(x) du sommet x s'écrit $C(x) = L - L(x) + 1$ où L(x) est la longueur du plus long chemin issu du sommet x et L la longueur du plus long chemin du graphe.

On notera $C^2(x)$ le rang de l'élément x dans ce classement dit "inverse".

"1.4.3. Le classement médian

C'est le classement final retenu. On l'obtient en classant les éléments dans l'ordre croissant de la somme des rangs dans le classement direct et dans le classement inverse.

La somme v(x) relative à un élément x s'écrit :

$u(x) = C^1(x) + C^2(x)$. et on classe les éléments x dans l'ordre croissant des valeurs u(x)".

"1.5. Sensibilité du classement

Le classement obtenu dépend non seulement des notes attribuées et des pondérations choisies mais encore des valeurs retenues pour les seuils de concordance C et les paramètres de discordance D.

Dans le cas où le classement médian est peu ou pas modifié par des variations affectant toutes ces données dans les limites des incertitudes et des imprécisions dont elles sont entachées, ce classement n'est pas sensible et mérite que le décideur le considère comme faible.

Si, par contre, de légères variations de données bouleversent le classement, celui-ci est trop sensible et on en déduit que l'information initialement fournie est insuffisante pour départager valablement les candidats.

Aussi les auteurs de cette méthode conseillent-ils d'effectuer une "analyse de sensibilité" (en modifiant les pondérations des critères et en introduisant éventuellement des valeurs des paramètres C et D différentes des valeurs standard)".

A22 - Application -

Considérons un module M ayant un ensemble de pannes $f_1,\ldots, f_n$ et supposons que l'on connaisse un moyen de détecter chacune des pannes de M en agissant sur son entrée et en observant sa sortie. Considérons ce même module dans un réseau R sur lequel on a activé l'algorithme AT pour isoler des écoulements. Deux cas sont possibles :

- le module M appartient à un seul écoulement E. On a alors deux possibilités :

  - soit toutes les pannes $(f_1 ,\ldots, f_M)$ de M sont détectables dans un test qui ne fait intervenir que l'entrée et la sortie de l'écoulement E, auquel cas on dira que M est complètement testable dans E.

  - soit certaines pannes , par exemple $(f_1,\ldots, f_p)$ ne sont pas détectables dans E. Ceci veut dire que, quelque soit la valeur affichée en entrée de E, la sortie de E en présence de f_i, $1 \leqslant i \leqslant p$ est la même que sa sortie si M est sain. On dira alors que les pannes $f_1 \ldots f_p$ n'ont aucune incidence fonctionnelle sur le comportement de l'écoulement E, donc il ne présente aucun intérêt de savoir si oui ou non elles ont lieu.

- Le module M appartient à plusieurs écoulements E1 E2...E_n. Supposons que c'est dans E1 qu'il est le "mieux" testable. Ceci n'exclut pas que certaines pannes de M_1 disons $(f_1 ,\ldots, f_p)$ ne soient pas détectables dans E1 alors qu'elles ont une incidence fonctionnelle sur les autres écoulement

	S_1	S_2	S_3	S_4	
A_1	5	25	26	28	$u_1 = 28$
A_2	23	8	26	28	$u_2 = 28$
A_3	24	26	8	24	$u_3 = 26$
→ A_4	24	25	22	7	$u_4 = 25.$

a-

	S_1	S_2	S_3	
A_1	5	18	16	$u_1 = 18$
→ A_2	14	8	16	$u_2 = 16$
$A_3.$	17	16	8	$u_3 = 17$

b

Figure 4.

Il est proposé alors de tester (f_{p+1} ... f_n) par l'écoulement E1 puis de tester (f_1 ... f_p) par E2..E_n. Pour cela on sélectionnera l'écoulement parmi E2 ... E_n dans lequel M est le "mieux" testable et on exclut de (f_1 ... f_p) les pannes que celui-ci détecte... etc... Ceci met en évidence la nécessité de classer pour chaque module M les écoulements qui le couvrent.

Pour établir ce classement, on dispose de critères de qualité de test :

- la commandabilité
- l'observabilité

et des critères de coût du test

- coefficient de test dans un écoulement
- rendement amont
- rendement aval
- travail de propagation amont
- travail de propagation aval

Pour faire le lien avec le formalisme du paragraphe A21, on dira que l'on classe, pour un module M donné, les candidats écoulements E_1 E_2 ... E_M en associant à chaque candidat E_j 1 j M les notes suivantes :

$g_1(E_j)$ = la commandabilité de M dans E_j (ou les rapports de commandabilité de M dans E_j)

$g_2(E_j)$ = la commandabilité (ou le rapport de commandabilité) de M dans E_j

$g_3(E_j)$ = le coefficient de test de M dans E_j

$g_4(E_j)$ = le rendement amont de M dans E_j

$g_5(E_j)$ = le rendement aval de M dans E_j.

$g_6(E_j)$ = le travail de propagation amont de M dans E_j.

$g_7(E_j)$ = le travail de propagation aval de M dans E_j.

A chacun des critères g_i est associé un poids P(i) qui mesure l'importance accordée à ce critère. Selon que l'on met l'accent sur la qualité du test ou son coût, on augmentera les poids P_1 P_2 ou les poids P_3, P_4, P_5, P_6, P_7.

Pour tester M, on utilisera les écoulements E_j dans l'ordre $u(E_j)$ croissants ($u(E_j)$ est le classement médian de E_j).

A3 - Elaborer un test adaptatif -

A31 Prise de décision dans l'incertain : Prendre une décision dans l'incertain. N'est-ce-pas, par essence, la conduite d'un testeur ? La prise de décision dans l'incertain est une branche d'une Théorie globale de recherche opérationnelle dite "théorie des jeux". La particularité de cette branche est double :

1. Le joueur à qui on a affaire est la nature, donc un joueur non intelligent dont on ne peut deviner le comportement.

2. Il n'est pas nécessaire, dans ce formalisme, de faire des hypothèses probabilistes sur les coups possibles que peut jouer la nature. Ceci est un avantage appréciable car, non seulement il est difficile de définir une distribution de probabilité qui tienne compte de la réalité mais il arrive souvent que la notion de probabilité n'ait aucun sens dans le contexte auquel on s'intéresse.

Le modèle général de la prise de décision dans l'incertain est le suivant : La nature peut être dans n états distincts s_1, s_2,..., s_n et nous avons à prendre une décision parmi m, A_1, A_2 ... A_m.

On définit une matrice U ayant m lignes et n colonnes par U_{ij} = le profit que nous réalisons si nous prenons la décision A_i alors que la nature est dans l'état s_j. Le problème qui se pose est de savoir, étant donnée la matrice U, quelle décision A_i prendre sachant que l'on ignore l'état s_j, dans lequel se trouve la nature et qu'en plus on ignore les probabilités d'occurence de chaque état.

Remarque 1 : Si on connaissait la distribution de probabilité p_j $1 \leq j \leq M$ associée aux états s_j, on calculerait pour chaque décision A_i $1 \leq i \leq m$, la quantité $u_i = \sum_{j=1}^{M} p_j U_{ij}$ qui mesure l'espérance du profit associée à l'acte A_i. On prendrait alors une décision A_i qui maximise u_i pour $1 \leq i \leq m$. On dit alors q'uil s'agit d'une prise de décision sous risque.

Il existe deux méthodes permettant de résoudre ce problème : Elles sont communément appelées maximim et minimax et qui ont pour buts respectifs de minimiser le risque et de maximiser le profit.

Maximin : A chaque décision A_i on associe son niveau de sécurité u_i égal au minimum sur j des u_{ij}. La décision A_i que l'on prendra est celle qui maximise u_i, i.e., celle qui maximise le profit minimal. Cette stratégie est une bonne défense contre une nature qui joue la politique du pire : c'est une stratégie prudente et pessimiste.

Minimax : A chaque colonne j de la matrice U on associe u_j = le maximum sur i des u_{ij}. On construit une matrice R de même dimension que U telle que $R_{ij} = u_j - u_{ij}$ - R est dite la matrice des risques. A chaque décision A_i on associe le réel r_i = le maximum sur j des u_{ij} et on sélectionne la décision A_i qui réalise le minimum des r_i - voir figure 3.

A32 - Application - Le test adaptatif consiste à envoyer séquentiellement des vecteurs au système sous test, à observer la réaction du système aux vecteurs déjà appliqués et à en déduire le ou les vecteurs suivants à appliquer. Les méthodes usuelles de test adaptatif sont généralement basées sur le calcul de probabilités [12]; Il s'agit à chaque étape du test de déterminer la probabilité d'occurence de chaque panne du système connaissant la séquence de test déjà appliquée ainsi que la réponse du système à cette séquence. Cette distribution de probabilité détermine la suite du test. Nous proposons dans la suite une méthode de test adaptatif qui se passe de considérations probabilistes :

Soient deux modules M1 et M2 appartenant à un même écoulement E. Supposons que, isolé, M1 nécessite 50 vecteurs pour être testé alors que M2 en nécessite 80 et que :

- si dans l'écoulement E on a testé M1 (et qu'on a eu de l'information sur M2) il suffit d'envoyer 60 vecteurs à l'intention de M2 pour le tester : Tester M1 puis M2 coûte 110 vecteurs de test.

- Si dans l'écoulement E on a testé M2 (et qu'on a eu de l'information sur M1), il suffit d'envoyer 40 vecteurs à l'intention de M1 pour le tester : Tester M2 avant M1 coûte 120 vecteurs.

Si aucune hypothèse n'est faite sur la probabilité avec laquelle M1 ou M2 tombent en panne, on affirme qu'il vaut mieux tester M1 avant M2. C'est ce problème que nous allons tenter de résoudre dans le cas de plusieurs modules dans un même écoulement. Nous considérons un écoulement E contenant n modules $M_1, \ldots, M_n$. On définit n états de la nature par S_j <==> M_j est en panne. Cette définition est rendue possible par l'hypothèse de panne simple ... de même, on définit n décisions à prendre par A_i <==> on envoie la séquence de test complet du module M_i.

La matrice U que nous allons construire n'est pas une matrice de profit mais une matrice de coût :

u_{ii} = la longueur de la séquence de test nécessaire pour M_i (c'est en particulier le travail que l'on fournit si l'on a pris la décision A_i et que la nature est dans l'état S_i).

$U_{ij}(i \neq j)$ = la longueur de la séquence de test nécessaire pour tester i puis tous les autres modules puis j avec l'ordonnancement le plus défavorable. Cette quantité peut aussi être remplacée par un majorant. (U_{ij} est en particulier le travail maximal que risque de faire le testeur S'il choisit A_i alors que M_j est en panne).

La construction de U fait que ce problème n'est justiciable, de préférence, que de la méthode qui réalise le minimum du risque. Dans le formalisme, vu plus haut il s'agit du maximum. En fait vu que U est une matrice de coût, et non de profit , on fera le minimum sur les colonnes du maximum des lignes.

Après avoir exécuté l'action A_i, deux cas sont possibles :

- le résultat du test de M_i est positif, alors M_i est en panne.
- le résultat du test de M_i est négatif, on supprime la colonne S_i et la ligne A_i de la matrice U, on modifie ses termes extradiagonaux et on réitère l'algorithme du maximin (qui consiste à prendre le minimum sur les colonnes du maximum sur les lignes) sur la matrice réduite.

Cette procédure tend à minimiser la longueur totale par des minimisations locales mais simples.

Considérons, à titre d'exemple, la matrice donnée à la figure 4a. En optant A4, on s'assure que, quelque soit le module en panne, on ne fera jamais plus de 25 vecteurs de test (alors que le test indépendant des 4 modules nécessiterait 28 vecteurs). Supposons que le test de M_4 est négatif

. états de la nature.

Décisions	S_1 S_2	S_j	S_n
A_1			
A_2			
⋮			
A_i		u_{ij}	
⋮			
A_m			u_{mn}

Maximin

	S_1	S_2
A_1	2	8
A_2	3	4

$u_1 = 2 \quad u_2 = 3 \Rightarrow$ On décidera A_2.

Minimax

$$U = \begin{array}{c|cc} & S_1 & S_2 \\ A_1 & 2 & 8 \\ A_2 & 3 & 4 \end{array} \Rightarrow \Pi = \begin{vmatrix} 1 & 0 \\ 0 & 4 \end{vmatrix}$$

$v_1 = 3 \quad v_2 = 8 \quad ; \quad r_1 = 1 \quad r_2 = 4 \Rightarrow$ On décidera A_2.

figure 3.

(M_4 est sain) et que la matrice réduite est celle donnée en 4b. On voit que le maximum des u_i est 18 qui est inférieur ou égal (puisque c'est égal) à 25-7. 25 est le nombre de vecteurs que l'on était sûr de ne pas dépasser en optant A4, 7 est le nombre de vecteurs envoyés à M_4. En décidant A2, on s'assure que l'on ne dépassera pas 16 vecteurs à appliquer.

Présentée ainsi, cette méthode semble être un simple moyen d'ordonnancer le test des modules car les matrices consécutives sont figées. En fait quand la décision A_i donne lieu à un test positif, cela n'implique pas nécessairement que M_i est en panne. La présomption sur l'état de santé du système dépend de la réponse qu'il donne à la séquence A_i. Ainsi, à chaque réponse on pourrait associer une matrice distincte de sorte à construire un test adaptatif qui minimise la longueur totale de la séquence appliquée par des minimisations locales.

B - Test déterministe Généré aléatoirement -

B1 - Rendement : Dans cette méthode on dispose d'une liste initiale de pannes à tester. On tire au sort un vecteur source et on le propage dans le circuit en notant la liste des pannes qu'il détecte. Ces pannes sont éliminées de la liste et la procédure est itérée jusqu'à épuisement de cette liste.

Dans cette procédure la difficulté est essentiellement dûe, en première approximation au goulot d'étranglement caractérisé par C_{min} Si à la source on dispose de 2^{C_S} vecteurs distincts, en aval de C_{min} il n'en reste plus que $2^{C_{min}}$. En moyenne $2^{C_S - C_{min}}$ vecteurs distincts à la source donnent le même résultat en sortie. Le goulot d'étranglement C_{min} affecte la commandabilité du sous réseau en aval de C_{min} et l'observabilité du sous réseau en amont de C_{min}. Des pannes détectées en amont de C_{min} ne le sont plus en sortie et pour les pannes en aval de C_{min}, en moyenne $2^{C_S - C_{min}}$ vecteurs en S donnent le même vecteur de test.

On associera donc à cette méthode un rendement de $2^{C_S - C_{min}} = R$. On pourrait raffiner la précision de R en se donnant le profil informationnel du réseau et en affectant à R le produit des $2^{C_{Si} - C_{Pi}}$ sur tous les étranglements etr_i du réseau.

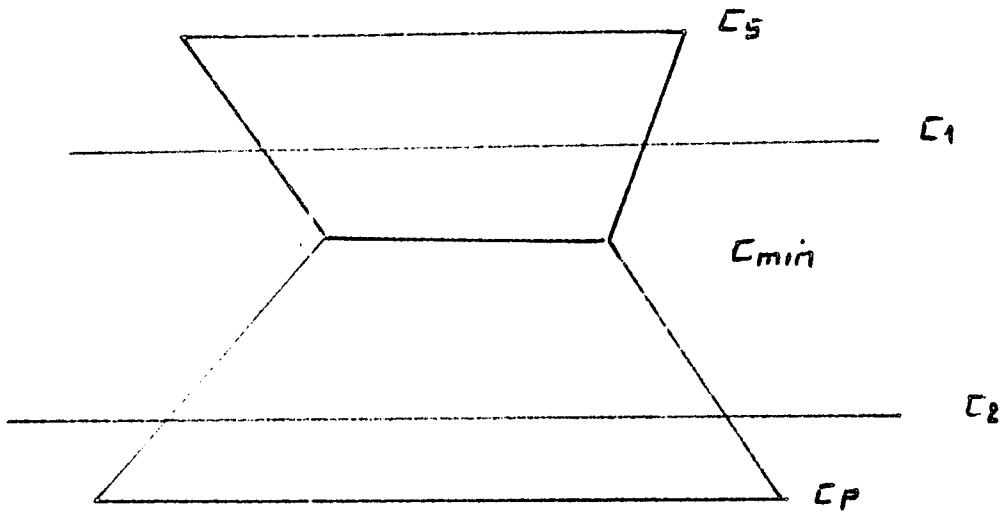
Voir à la figure 5 comment C_{min} peut affecter l'observabilité et la commandabilité du réseau.

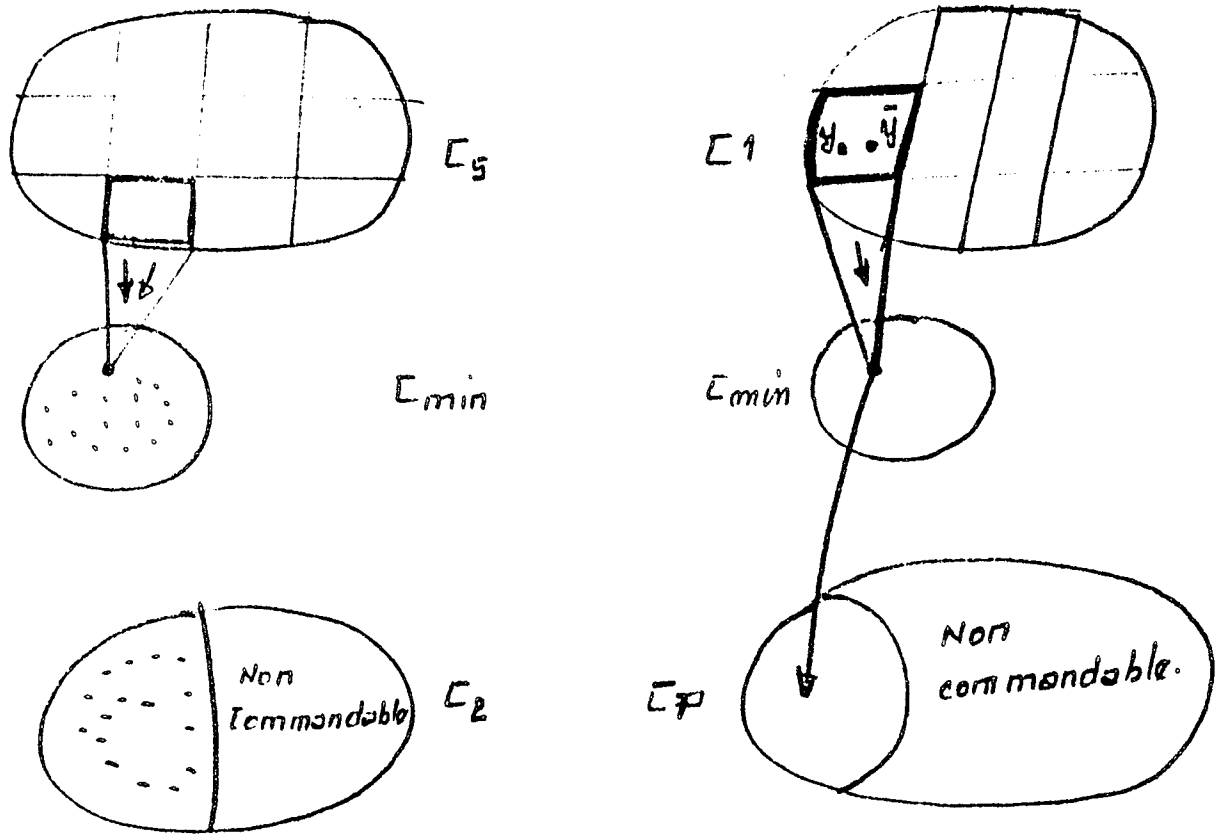
B2 - Application : La combinaison du nombre de pannes que l'on se propose de détecter et de ce coefficient de rendement permet une évaluation de la charge de travail à fournir si l'on opte pour le test déterministe généré aléatoirement donc donne les moyens de décider de la méthode de test à adopter par comparaison avec le coût des autres.

C - Test aléatoire -

C1 - Détermination de la longueur d'une séquence de test aléatoire :

dans la référence bibliogaraphique [13], David et Blanchet proposent une méthode qui permet de déterminer la longueur d'une séquence de test aléatoire nécessaire pour s'assurer que toute panne d'un circuit considéré n'a qu'une probabilité Q_D de ne pas être détectée , Q_D étant fixé d'avance. Soient n modules M_1 M_2 ... M_n qui composent un écoulement E. A chaque module M_i on peut appliquer l'algorithme proposé par David et Blanchet afin de trouver la longueur L_i qui lui correspond. Pour compléter cette démarche, nous allons proposer une méthode qui détermine la longueur de la séquence de test aléatoire qu'il faut émettre à l'entrée de l'écoulement E afin de simuler à l'entrée de chacun des modules M_i les conditions de test analogues à un test aléatoire de longueur L_i. En fait, la méthode de David et Blanchet peut théoriquement être appliquée à l'écoulement E tout entier.





Perte de Commandabilité Perte d'Observabilité.

figure 5.

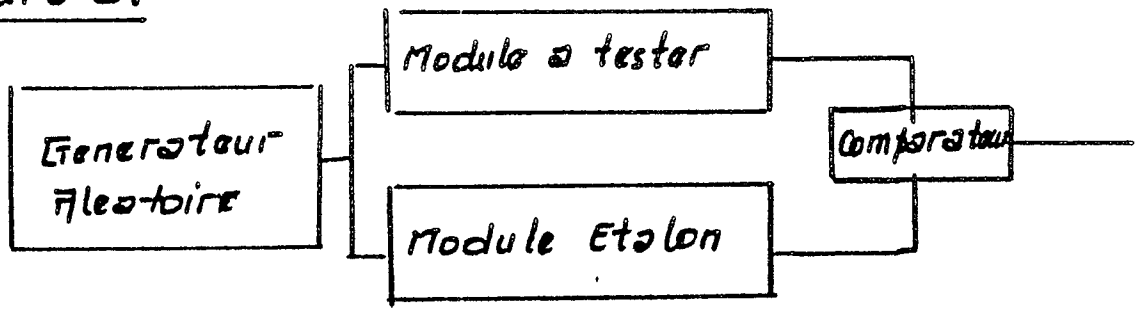


figure 6. Schemas de Principe du test aléatoire.

Or,

- d'une part elle risque d'être fastidieuse si l'écoulement E est grand car elle opère sur la description du système au niveau "porte logique".

- d'autre part il est très commode d'associer à chaque module M_i la quantité L_i qui lui est intrinsèque et d'en déduire pour chaque montage de ces modules dans E une longueur de test propre à E.

Soit M_i un modèle isolé. Soient X_i et Y_i ses arcs d'entrée et de sortie, $\gamma(X_i)$ et $\gamma(Y_i)$ les capacitésde X_i et Y_i. Considérons L_i éléments tirés aléatoirement avec remise dans un ensemble ayant $2^{\gamma(X_i)}$ éléments. Soit $E(L_i)$ l'espérance mathématique du nombre d'éléments distincts tirés.

Pour tester M_i avec la qualité Q_D, il faut y envoyer L_i vecteurs indépendants et en observer les sorties. On affirme que ceci est équivalent à dire qu'il faut envoyer dans M_i $E(L_i)$ vecteurs distincts. Considérons M_i dans l'écoulement E. Si l'on suppose avoir une observabilité totale de M_i , on peut dire : Pour tester M_i avec la qualité Q_D, il suffit d'y commander $E(L_i)$ vecteurs distincts à partir de l'entrée S de E ; la question qui se pose est de savoir combien de vecteurs indépendants il faut émettre à l'entrée de l'écoulement afin de faire afficher $E(L_i)$ vecteurs distincts à l'entrée X_i de M_i. Ce problème se formalise comme suit :

Problème 1 : On considère un ensemble S ayant 2^{C_S} éléments partitionné en $2^{C(M_i)}$ classes ayant même cardinal $2^{C_S - C(M_i)}$ ($C(M_i)$ étant la commadabilité de M_i). On tire avec remise $\bar{L}_i$ éléments indépendants de S, calculer l'espérance $\bar{E}(\bar{L}_i)$ du nombre de classes ayant au moins un élément tiré. Déterminer $\bar{L}_i$ pour que $\bar{E}(\bar{L}_i) = E(L_i)$. Ceci n'a une solution que si $C(M_i) \geq E(L_i)$.

Si S est partitionné de telle sorte que dans chaque classe on trouve les éléments de S qui donnent la même valeur en X_i, on voit que $\bar{E}(\bar{L}_i)$ mesure le nombre moyen de vecteurs distincts affichés en X_i si on affiche $\bar{L}_i$ vecteurs indépendants en S. Si à chaque module M_i on associe la quantité $\bar{L}_i$, alors en émettant en S une séquence de vecteurs indépendants de longueur $\bar{L} = \max_i (\bar{L}_i)$ on est sûr que chaque module M_i reçoit en moyenne plus que $E(L_i)$ vecteurs. Ceci ne suffit pas à assurer la qualité de test Q_D si on veut tenir compte de la perte d'observabilité entre Y_i et le puits P de l'écoulement. On serait alors amené à rectifier $E(L_i)$ en le multipliant par un coefficient de rendement, en l'occurence, par approximation, l'inverse de q_i ; voir le chapitre 5, paragraphe D3.

C2 - Application : Deux problèmes mathématiques se sont posés à nous au paragraphe C1 : donner une expression à $E(L_i)$ et $\bar{E}(L_i)$. Nous n'en donnerons pas une expression analytique - pour autant que ceci soit possible. Néanmoins nous présenterons les moyens de les calculer.

Pbo : Soit E un ensemble à n éléments. On tire dans E, avec remise, L éléments indépendants. Calculer E(L), l'espérance du nombre d'éléments distincts tirés.

Résolution : Un tirage de L éléments avec remise dans E (de cardinaln) est une application de [1,L] dans E. Le nombre d'applications de [1,L] dans E vaut n^L.

Le nombre d'applications f de [1,L] dans E telles que $|f[1,L]| = k$ = (nombre de suites non ordonnées de k éléments dans) × (nombre de surjections de [1,L] → [1,k]).

$= C_M^k \cdot k!\, S(L,k) = A_M^k\, S(L,k)$

où S(M,k) est le nombre de Stirling de deuxième espèce étudié par Louis Comtet dans [23] et [24]. Voir figure 7.

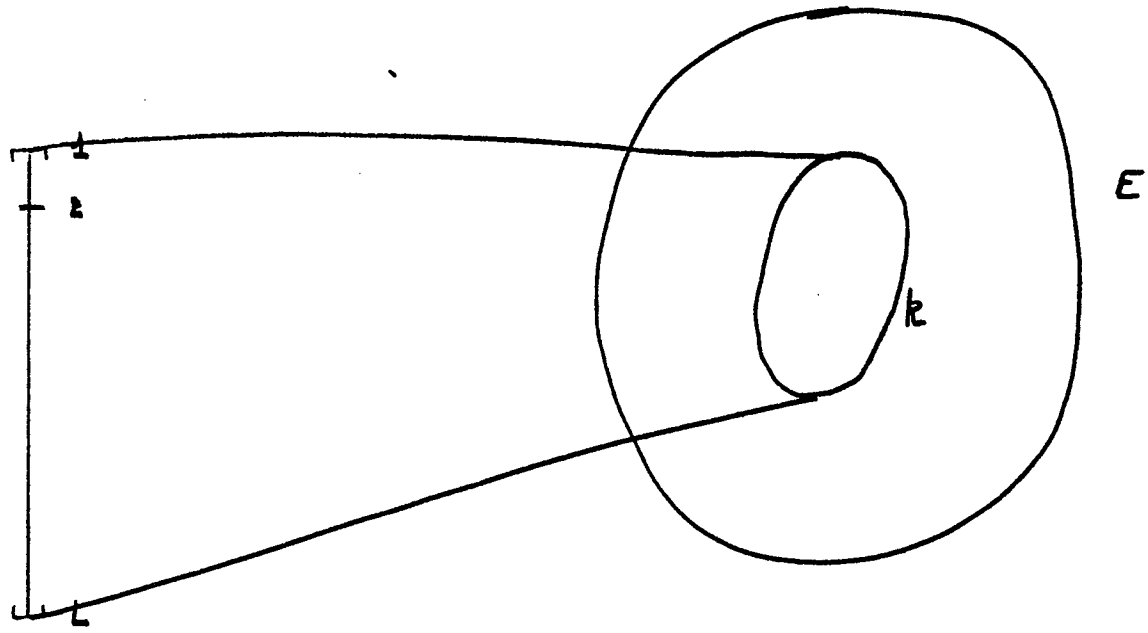


fig 7

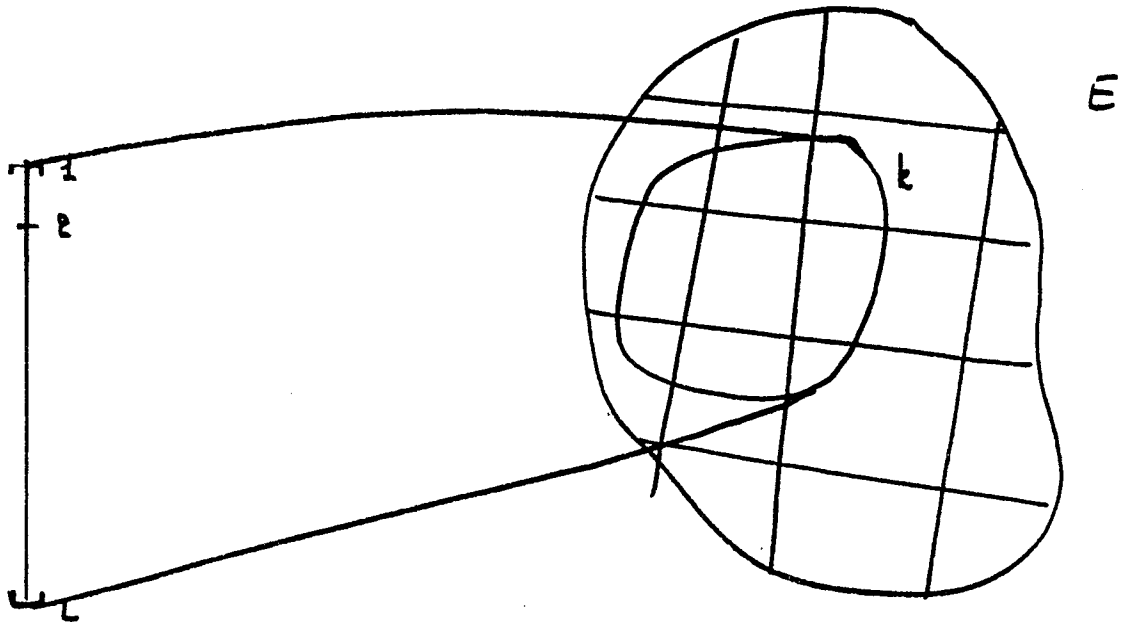


figure 8

Ainsi, si on appelle p_k la probabilité qu'en tirant L éléments avec remise on ait touché k éléments distincts, on a :

$$p_k = \frac{A_M^k \, S(L,k)}{M^L}$$

Ceci est confirmé par la formule citée dans [23] $\sum_{k=1}^{L} A_M^k \, S(L,k) = M^L$.

$$p_k = \frac{1}{N^L} \; A_M^k \; \frac{1}{k!} \; \sum_{j=0}^{k} (-1)^j \, C_j^k \, (k-j)^L$$

$$= \frac{1}{N^L} \; C_M^k \; \sum_{j=0}^{k} (-1)^j \, C_j^k \, (k-j)^L .$$

Nous avons traité un cas "à la main" pour L = 5 et N = 10.

$$p_1 = 10^{-4}$$

$$p_2 = 135 . \; 10^{-4}$$

$$p_3 = 1800 . \; 10^{-4}$$

$$p_4 = 5040 . \; 10^{-4}$$

$$p_5 = 3024 . \; 10^{-4}$$

d'où on tire $E(L) \simeq 4{,}2$ ce qui est très vraisemblable.

Pb1. Soit E un ensemble à n éléments partitionné en C classes de p éléments chacune. On tire dans E, avec remise, L éléments indépendants. Calculer $\bar{E}(L)$, l'espérance du nombre de classes où un élément au moins a été tiré.

Solution : On énonce le problème autrement : on tire sans remise E(L) éléments dans E, calculez l'espérance du nombre de classes touchées - voir figure 8.

On propose la formule approximative très simple $\bar{E}(L) = \frac{C}{M} E(L)$.

On remarque que, dans la réalité, l'ensemble S (correspondant à E) n'est pas partitionné uniformément.

On déduit que l'approximation proposée n'en est que plus légitime. Elle est d'autant plus correcte, en fait, que $p = \frac{n}{C}$ est petit ; d'ailleurs, dans le cas où $\frac{n}{C}$ est grand, le problème peut se résoudre rigoureusement (en utilisant des formules d'analyse combinatoire) et ce de façon économique.

HORS TEXTE

IMPLEMENTATION DE AT EN ALGOLW ET MISE EN OEUVRE SUR UN EXEMPLE

Mise en oeuvre de AT

Ce paragraphe est une description détaillée du programme AT en même temps qu'un mode d'emploi. On y présentera, sur l'exemple de réseau réduit donné au chapitre 2, l'implantation du réseau en mémoire, la réalisation des diverses fonctions Π, ϕ, suiv ... l'explication de certaines procédures. Le texte complet du programme ainsi que les résultats qu'il a livrés sur l'exemple considéré ; enfin une étude sommaire de ses performances en termes de temps calcul associé à un réseau d'une taille donnée. Au début des données du programme, on donne, dans cet ordre :

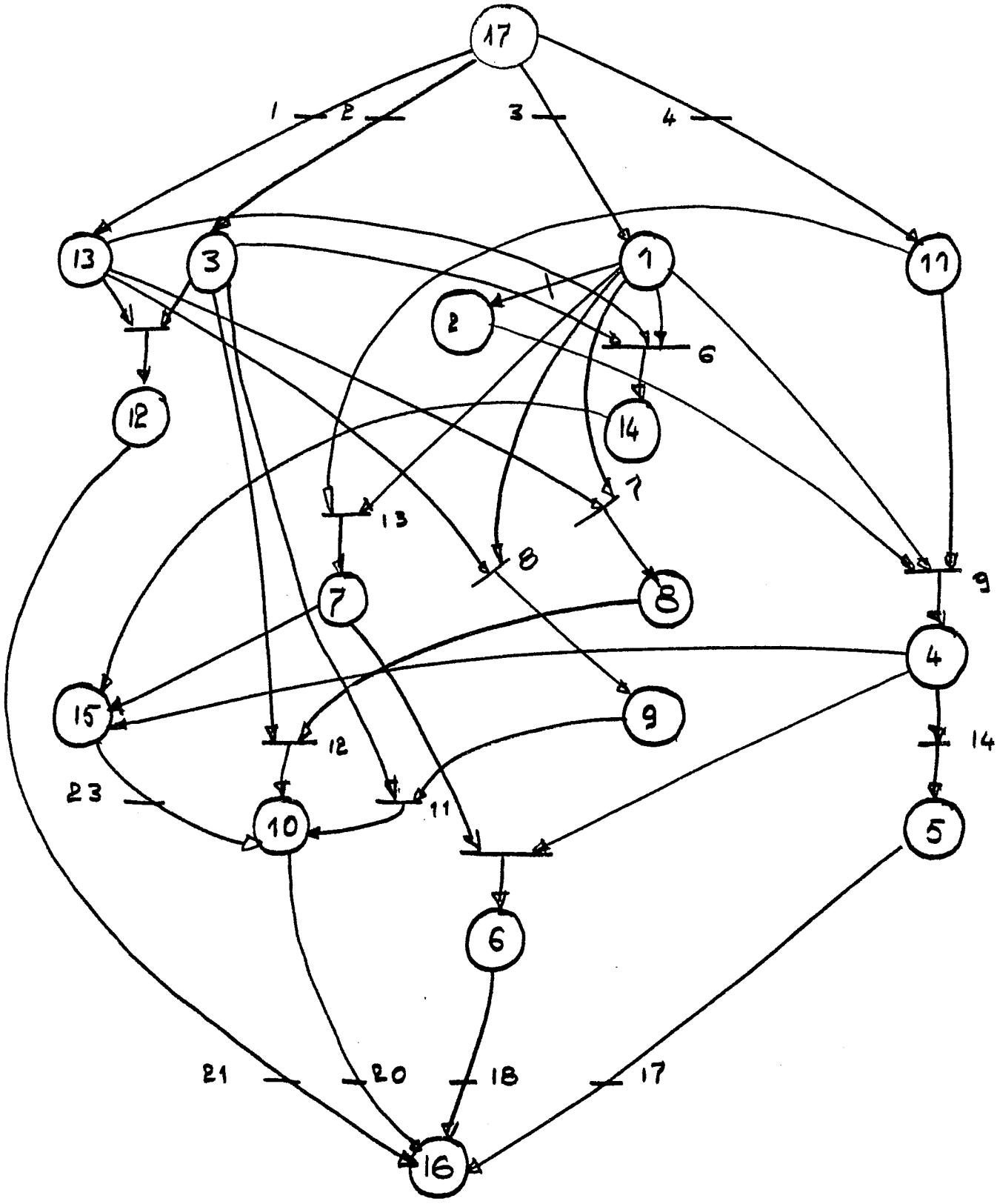
NP , le nombre de places du réseau (17, dans notre cas)

NAP, majorant du nombre d'adjacents à une place (9)

NT, le nombre de transitions (23)

NAT, majorant du nombre d'adjacents à une transition (9)

NBE, majorant du nombre présumé d'écoulements élémentaires (8)



Réseau déduit de celui présenté au chapitre 1 par les règles de réduction R1 et R2.

1	-3	13	6	7	8	9	10		
2	-10	9							
3	-2	5	6	11	12				
4	-9	14	15	16					
5	-14	17							
6	-16	18							
7	-13	16	19						
8	-7	12							
9	-8	11							NP
10	-11	-23	-12	20					
11	-4	9	13						
12	-5	21							
13	-1	5	6	7	8				
14	-6	22							
15	-19	-22	-15	23					
16	-20	-21	-17	-18					
17	1	2	3	4					

NAP

AP

DP

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	1	4	3	1	1	2	1	1	1	2	1	4	1	1	0	4
1	1	1	1	1	1	1	1	1	3	1	1	1	1	3	4	0

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

NP.

fig 4

1	13	-17							
2	3	-17							
3	1	-17							
4	11	-17							
5	-3	12	-13						
6	-1	-3	-13	14					
7	-1	8	-13						
8	-1	9	-13						
9	-1	-2	4	-11					
10	-1	2							
11	-3	-9	10						
12	-3	-8	10						
13	-1	7	11						
14	-4	5							
15	-4	15							
16	-4	6	-7						
17	-5	16							
18	-6	16							
19	-7	15							
20	-10	16							
21	-12	16							
22	-14	15							
23	10	-15							

NT

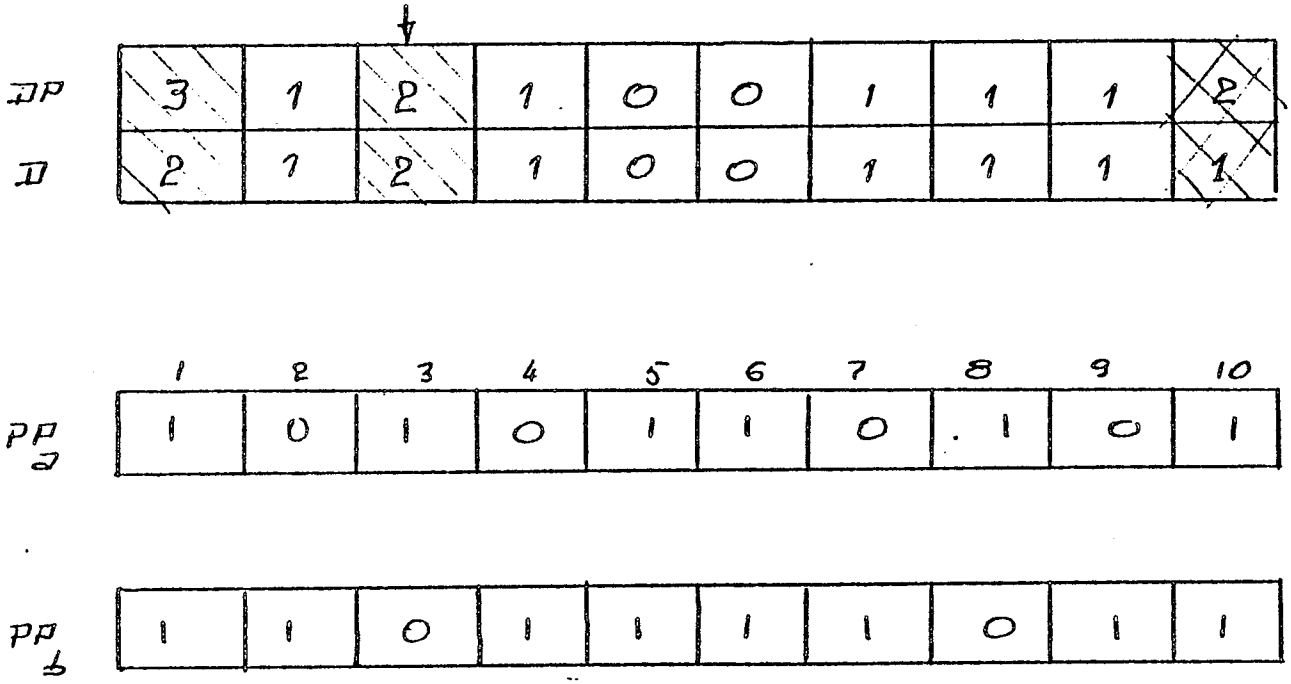
NAT.

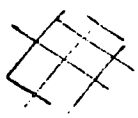
AT

fig 5.

DP	3	1	2	1	0	0	1	1	1	2
	1	2	3	4	5	6	7	8	9	10
Itération 1	1	1	1	1	0	0	1	1	1	1
2	1	1	2	1	0	0	1	1	1	1
3	2	1	1	1	0	0	1	1	1	1
4	2	1	2	1	0	0	1	1	1	1
5	3	1	1	1	0	0	1	1	1	1
6	3	1	2	1	0	0	1	1	1	1

fig 6.





est hachurée doublement la case qui est tq DP(i) > 1 mais qui ne change pas de valeur d'une itération à l'autre.

fig 7

Le réseau est entré dans l'ordinateur de la manière suivante :

Place i, liste des adjacents à cette place (avec le signe + s'il s'agit de fils et le signe - s'il s'agit de pères) puis 0 pour marquer que l'on a fini avec la place i. A partir de ces données le programme remplit les tableaux AP AT et DP de la façon suivante :

AP(i,j) : j° adjacent à la place i
AT(i,j) : j° adjacent à la transition i
DP(k) : Si $k \leq NP$ alors $d^+(k)$ sinon $d^-(k-NP)$

On vérifie que le tableau AP est conforme au réseau de la figure 3 et que AT et DP lui sont conformes. En fait AT et DP sont redondants pour AP mais sont très utiles pour le traitement.

Réalisation de la fonction (Π,ϕ) : L'implantation du réseau en mémoire induit fortuitement, pour chaque place i, un ordre ρ_i sur les fils de la place i, et un ordre θ_i sur ses pères :

$t\ \rho_i\ t'$ <==> En définissant j et j' par AP(i,j) = t et AP(i,j') = t', on a j > j'

$t\ \theta_i\ t'$ <==> En définissant j et j' par AP(i,j) = -t et AP(i,j') = -t', on a j > j'.

Considérons un tableau D de même dimension que DP et qui vérifie si DP(I) > 1 alors D(I) ≤ DP(I) sinon D(I) = DP(I). A toute valeur du tableau D (considéré comme 2NP vecteur), on associe de façon univoque un père et un fils privilégiés à toute place du réseau, de la manière suivante :

$\phi(I)$ = (D(I))° fils de la place i d'après ρ_i.

$\theta(I)$: (D(I+NP))° père de la place i d'après θ_i

Le mécanisme de passage du tableau D à la fonction (Π,ϕ), au moyen du tableau AP est très facile à comprendre sur les six itérations de l'exemple donné au paragraphe D du chapitre 2.

```
0000 1-      DEBUT
0001 --      CO MILI ALI RECHERCHE D'ECOULEMENTS DANS RESEAU DE PETRI.CALCUL
0001 --      DE PARAMETRES DE TEST.ET DETERMINATION D'UNE POLITIQUE DE TEST
0001 --      SCHEMAS DE PROGRAMME
0001 --      INITIALISATIONS
0001 --      IMPLANTATION
0001 --      TANTQUE
0001 --         DEBUT
0001 --         SI UTILE ALORS NOUVEL ECOULEMENT
0001 --         SI TROUVE ET DIFFERENT ALORS TRAITEMEN
0001 --         MISAJOUR
0001 --         IL Y A ENCORE DES ECOULEMENTS
0001 --         FIN
0001 --         FAIRE ITER:=ITER+1 ;
0001 --      ENTIER NP,NAP ; CO NB PLACES ET DEGRE MAX ;
0002 --      ENTIER NT,NAT ; CO NB TRANS ET DEGRE MAX ;
0003 --      ENTIER NBE CO NB D'ECOULEMENTS MAX ;  ;
0004 --      LIRE(NP,NAP,NT,NAT,NBE) ;
0005 2-         DEBUT
0006 --         CO PLACES ;
0006 --         ENTIER TABLEAU DP(1::2*NP) ; CO D+ ET D- ;
0007 --         LOGIQUE TABLEAU OK (1::2*NP); CO CF MISAJOUR ;
0008 --         ENTIER TABLEAU AP(1::NP,1::NAP);
0009 --         CO TRANS ;
0009 --         ENTIER TABLEAU DT(1::2*NT) ; CO D+ ET D- ;
0010 --         ENTIER TABLEAU AT(1::NT,1::NAT);
0011 --         CO ECOUL ;
0011 --         ENTIER TABLEAU PL(1::NP) ; CO MARQUAGE ECOUL ;
0012 --         ENTIER TABLEAU TR(1::NT) ; CO MARQUAGE TRANS ;
0013 --         ENTIER TABLEAU D(1::2*NP) ; CO MOT D'ETAT PROG ;
0014 --         ENTIER TABLEAU EC(1::NP,1::NBE) ; CO COUVERTURE ;
0015 --         CO DIVERS ;
0015 --         ENTIER S,P ; CO SOURCE PUITS ;
0016 --         ENTIER ITER ; CO NB ITERATIONS ;
0017 --         ENTIER Q ; CO RETENUE MISAJOUR ;
0018 --         LOGIQUE TROUVE CO CF PP ; ;
0019 --         ENTIER N ; CO NB D'ECOUL ;
0020 --         ENTIER NBN ; CO NB D'APPELS NEW ;
0021 --         CO PROCEDURES ;
0021 --         PROCEDURE IMP ;
0022 3-            DEBUT
0023 --            ENTIER PC,T,C ; LOGIQUE SO,PU ;
0025 --            POUR I:=1 JUSQUA 2*NP FAIRE D(I):=0;
0026 --            POUR I:=1 JUSQUA NP FAIRE
0026 4-               DEBUT
0027 --               LIRE(PC) ; PU:=VRAI ; SO:=VRAI ;
0030 --               TANTQUE
0030 5-                  DEBUT
0031 --                  LIREC(T) ;
0032 --                  TA=0
0032 -5                  FIN
0032 --               FAIRE
0032 5-                  DEBUT
0033 --                  SI T>0 ALORS
0033 6-                     DEBUT
0034 --                     DP(PC):=DP(PC)+1 ;
0035 --                     AP(PC,DP(PC)+DP(PC+NP)):=T ;
0036 --                     DT(T+NT):=DT(T+NT)+1;
```

Réalisation de suiv - Le passage d'une configuration de D à la "suivante" se fait par "incrémentation" du tableau D au niveau des indices k tels que DP(k) > 1. La procédure Misajour qui modifie D après chaque appel de New s'inspire de la numération arabe : si on veut,par exemple, passer en revue toutes les configurations possibles de trois chiffres, on écrit 000 et on incrémente ce nombre de 1 autant de fois qu'il faut pour arriver à 999. On détecte qu'on les a tous passés en revue quand on obtient une retenue sortante égale à 1. A la figure 6 on voit l'évolution du tableau D au cours des six itérations de AT. On voit que D(10) n'a pas changé : ceci est dû aux raisons invoquées dans l'exemple d'application du paragraphe D - chapitre 2.

Procédure utile : Si les places qui ont changé de père ou de fils à l'itération k n'ont pas été cochées par New, il n'est pas utile de réactiver New à l'itération k+1. Si on prend la précaution avant chaque appel de New, de s'assurer qu'il y a au moins une place précédemment cochée qui a changé de père ou de fils, on peut économiser certains appels New.
A la figure 7 on montre, dans le cas a une configuration, où Util sortira vrai et dans le cas b une configuration où Util sortira faux ; le tableau PP représente une duplication du tableau PL indiquant les places cochées par New. Les colonnes hachurées correspondent aux indices i tels que DP(i) > 1 et celles indiquées d'une flèche sont telles que du passage de la configuration précédente à la présente, le tableau D a eu ces indices modifiés. Il est remarquable :

1. que en regardant l'état actuel du tableau D on sait dire quels indices ont été modifiés entre l'état précédent et l'état actuel.

2. L'efficacité de l'usage d'Util dépend très étroitement de la numérotation des sommets, ainsi, par exemple, si on sait qu'une place a plusieurs pères et que, pour une raison quelconque, elle est appelée à être toujours cochée par New ∀ Πϕ , alors il serait maladroit de lui donner un numéro tel que aucune autre place ayant plusieurs pères n'ait un numéro plus grand que le sien. Car dans ce cas l'indice

qui lui correspond dans le tableau DP serait à l'extrême droite de ceux qui vérifient DP(I) > 1;donc changerait à chaque itération. En outre, par définition, il sera toujours coché, donc Util livrera toujours vrai. Certes, P est coché par New ∀ Πϕ et, dans l'exemple du chapitre 2 il lui a été donné le numéro le plus fort, 5, alors qu'elle a deux pères. Mais on a vu que Π(P) a été fixé donc la restriction ne la concerne pas. On verra au cours de la description du programme que l'on s'est donné les moyens de fixer le père d'autres places X quand on a des raisons pour cela. L'efficacité d'Util a été testée sur un réseau de 12 places et 5 écoulements : New n'a été appelé que dans 80% des cas environ, alors que sur le réseau de la figure 3 on n'a réussi à éviter aucun appel de New : un défaut de numérotation peut être. Toujours est-il que, étant donné le coût en temps calcul du segment Util (qui est très bas), Il est intéressant d'y faire appel pour peu que l'on ait un espoir de son efficacité. C'est ainsi que, le temps calcul consommé par AT activé sur le réseau de la figure 3 avec l'instruction " Si Util alors New" a été de 140 secondes alors que, activé sur le même réseau avec l'instruction "New" (appel systématique), il a consommé 147 secondes pour le même nombre d'appels. Autant dire que le temps d'exécution de 4608 fois Util ne vaut même pas l'erreur avec laquelle est évalué le temps calcul global !.

```
0037 --                    AT(T+DT(T)+DT(T+NT)):=-PC ;
0038 --                    PU:=FAUX
0038 -6                    FIN
0039 --                 SINON
0038 6-                    DEBUT
0039 --                    DP(PC+NP):=DP(PC+NP)+1 ;
0040 --                    AP(PC+DP(PC)+DP(PC+NP)):=T ;
0041 --                    DT(-T):=DT(-T)+1 ;
0042 --                    AT(-T+DT(-T)+DT(NT-T)):=PC ;
0043 --                    SO:=FAUX ;
0044 -6                    FIN
0044 -5                 FIN ; CO TQ ;
0045 --              SI PU ET ^SO ALORS
0045 5-                 DEBUT
0046 --                 P:=PC ;
0047 --                 D(PC+NP):=1 ;
0048 -5                 FIN
0048 --              SINON
0048 --              SI ^PU ET SO ALORS
0048 5-                 DEBUT
0049 --                 S:=PC ;
0050 --                 D(PC):=1 ;
0051 -5                 FIN
0051 --              SINON
0051 --              SI ^PU ET ^SO ALORS
0051 5-                 DEBUT
0052 --                 D(PC+NP):=1 ;
0053 --                 D(PC):=1
0053 -5                 FIN SINON ; CO ERREUR :
0054 -4              FIN ;
0055 --           ECRIRE("TABLEAU AP"); ECRIRE(" ");
0057 --           POUR I:=1 JUSQUA NP FAIRE
0057 4-              DEBUT
0058 --              ECRIRE(" "); ECRIRE(I,"/");
0060 --              POUR J:= 1 JUSQUA DP(I)+DP(I+NP) FAIRE
0060 --              ECRIREC(AP(I+J));
0061 -4              FIN ;
0062 --           ECRIRE("TABLEAU AT"); ECRIRE(" ");
0064 --           POUR I:=1 JUSQUA NT FAIRE
0064 4-              DEBUT
0065 --              ECRIRE(" "); ECRIRE(I,"/");
0067 --              POUR J:=1 JUSQUA DT(I)+DT(I+NT) FAIRE
0067 --              ECRIREC(AT(I+J));
0068 -4              FIN ;
0069 --              POUR I:=1 JUSQUA 2*NP FAIRE OK(I):=VRAI ;
0070 --              OK(NP+P):=FAUX;
0071 4-            DEBUT
0072 --            ENTIER W;TANTQUE
0073 5-               DEBUT
0074 --               LIREC(W); W^=0
0075 -5               FIN
0075 --            FAIRE OK(W):=FAUX;
0076 --           ECRIRE("TABLEAU DP"); ECRIRE(" ");ECRIRE(" ");
0079 --           POUR I:=1 JUSQUA 2*NP FAIRE ECRIREC(DP(I));
0080 -4           FIN;
0081 -3        FIN IMP;
0082 --     PROCEDURE MISAJOUR ;
0083 3-        DEBUT
```

Cartes	Commentaires
0021	Implantation du réseau en mémoire
0082	Exécute la fonction SUIV en modifiant D
0045	PU et ¬SO la place étudiée est le puits
0048	¬PU et SO la place étudiée est la source
0051	PU et SO il s'agit d'une place ordinaire
0070	voir le commentaire page sur la carte 0088
0075	voir le commentaire page sur la carte 0088

```
0084 --            ENTIER I ;
0085 --            I:=2*NP ; Q:=1 ;
0087 --            TANTQUE (I>=1) ET (Q=1) FAIRE
0087 4-               DEBUT
0088 --               SI (DP(I)>1) ET (OK(I)) ALORS
0088 5-                  DEBUT
0089 --                  D(I):=D(I)+Q ;
0090 --                  SI D(I)>DP(I) ALORS
0090 --                  D(I):=1
0090 --                  SINON Q:=0
0090 -5                  FIN ;
0091 --               I:=I-1
0091 -4               FIN ;
0092 -3            FIN MISAJOUR ;
0093 --         LOGIQUE PROCEDURE DIFFER  ;
0094 3-            DEBUT
0095 --            LOGIQUE L ; ENTIER I ;
0097 --            L:=VRAI ; I:=1 ;
0099 --            TANTQUE(L ET (I<=N)) FAIRE
0099 4-               DEBUT
0100 --               SI
0100 5-                  DEBUT
0101 --                  ENTIER X ; X:=0 ;
0103 --                  POUR J:=1 JUSQUA NP FAIRE
0103 --                  X:=X+ABS(PL(J)-EC(J,I)) ;
0104 --                  X=0
0104 -5                  FIN
0104 --               ALORS L:=FAUX  ;
0105 --               I:=I+1;
0106 --               CO SI PL=EC(I) ALORS L:=FAUX ;
0106 -4               FIN ;
0107 --            L
0107 -3            FIN DIFFER ;
0108 --         LOGIQUE PROCEDURE UTIL ;
0109 --         (ITER=1) OU
0109 3-            DEBUT
0110 --            ENTIER I ; LOGIQUE UN,U ;
0112 --            I:=2*NP+1 ; UN:=VRAI ; U:=FAUX ;
0115 --            TANTQUE(I>=2) ET (^U) FAIRE
0115 4-               DEBUT
0116 --               SI
0116 5-                  DEBUT
0117 --                  I:=I-1 ;
0118 --                  SI((DP(I)>1) ET UN) ALORS
0118 6-                     DEBUT
0119 --                     LOGIQUE MODIF ; MODIF:=FAUX ;
0121 --                     SI(D(I)=1)OU(UN) ALORS MODIF:=VRAI ;
0122 --                     SI D(I)^=1 ALORS UN:=FAUX ;
0123 --                     MODIF
0123 -6                     FIN SINON FAUX
0123 -5                  FIN
0123 --               ALORS
0123 5-                  DEBUT
0124 --                  SI PL(
0124 6-                     DEBUT
0125 --                     SI I<=NP ALORS I
0125 --                     SINON I-NP
```

Carte	Commentaires
0088	Certaines places, de par leur position en "bas" du réseau, sont telles que, à l'instant où elles reçoivent leur marquage par New, on peut être sûr qu'elles ont déjà un père marqué ; donc le déroulement de New ne dépend pas du père privilégié qu'elles ont à cette itération => on peut fixer le père de ces places à une valeur quelconque. Pour reconnaître ces places, on crée le tableau DK (1::2 * NP) tel que DK (I + NP) = faux si on veut fixer le père de la I° place. (Un tableau (1::NP) aurait suffi mais avec une taille 2 * NP on permet à l'utilisateur de fixer le fils de certaines places s'il le désire). Si bien que la procédure Misajour ne s'applique qu'aux places I telles que DP(I) > 1 et DK(I) . Le puits n'a pas de fils => le marquage ne peut lui parvenir que d'une transition aval => on n'a jamais besoin de connaître M(P) => On le fixe (pour cela on fait OK(NP + P) := faux ceci explique l'instruction de la carte 0070). Considérons dans le réseau de la figure 3 la place 10 : elle n'a qu'une transition aval, la transition 20. La transition 20 ne peut être marquée avant la place 10 car dès que 16 est marqué, il a déjà un père => le marquage ne peut venir à 10 que d'une transition amont =>M(10) n'a aucune espèce d'importance => on met DK(27) := faux dans le réseau on a fait aussi DK(32) := faux pour fixer le père de la place 15. A la fin des données du programme, on met la liste des indices de OK où on veut mettre faux suivie de 0. (ceci explique la carte 0075).

```
0125 --                  =1 ALORS U:=VRAI
0125 -5                  FIN
0125 -4               FIN ;
0126 --               U
0126 -3            FIN ;
0127 --         PROCEDURE NEW  ;
0128 3-            DEBUT
0129 --            PROCEDURE ATTAK ; ∞ Δ⁴ ;
0130 4-               DEBUT
0131 --               LOGIQUE PAFINI ;
0132 --               TANTQUE
0132 5-                  DEBUT
0133 --                  PAFINI:=FAUX ;
0134 --                  POUR I:=1 JUSQUA NP FAIRE
0134 6-                     DEBUT
0135 --                     SI(PL(I)=1) ET (SI(IA=P) ALORS(TR(SUIV(I))=0) SINON
0135 --                     FAUX) ALORS
0135 7-                        DEBUT
0136 --                        TR(SUIV(I)):=1 ;
0137 --                        PAFINI:=VRAI
0137 -7                        FIN
0137 -6                     FIN ;
0138 --                  POUR I:=1 JUSQUA NT FAIRE
0138 6-                     DEBUT
0139 --                     SI TR(I)=1 ALORS
0139 7-                        DEBUT
0140 --                        POUR J:=1 JUSQUA DT(I)+DT(I+NT) FAIRE
0140 --                        SI AT(I,J)>0 ALORS
0140 --                        SI PL(AT(I,J))=0 ALORS
0140 8-                        DEBUT
0141 --                        PL(AT(I,J)):=1 ;
0142 --                        PAFINI:=VRAI ;
0143 -8                        FIN
0143 --                        SINON ;
0144 -7                        FIN
0144 -6                     FIN ;
0145 --                  PAFINI
0145 -5                  FIN
0145 --               FAIRE ;
0144 -4               FIN ATTAK ;
0147 --            PROCEDURE FERME ;
0148 4-               DEBUT
0149 --               LOGIQUE PAFINI ;
0150 --               TANTQUE
0150 5-                  DEBUT
0151 --                  PAFINI:=FAUX ;
0152 --                  POUR I:=1 JUSQUA NP FAIRE
0152 6-                     DEBUT
0153 --                     SI((PL(I)=1) ET (IA=S) ET (IA=P)) ALORS
0153 7-                        DEBUT
0154 --                        ENTIER PER,FIL ;
0155 --                        PER:=0 ; FIL:=0 ;
0157 --                        POUR K:=1 JUSQUA DP(I)+DP(I+NP) FAIRE
0157 --                        SI AP(I,K)>0 ALORS FIL:=FIL+TR(AP(I,K))
0157 --                        SINON PER:=PER+TR(-AP(I,K)) ;
0158 --                        SI FIL=0 CO IL MANQUE FILS ; ALORS
0158 8-                        DEBUT
0159 --                        TR(SUIV(I)):=1 ;
```

Cartes	Commentaires
	Cette manoeuvre nous a permis de diviser le temps calcul de AT par 4.3.3 = 36 (=$\bar{d}$(P). $\bar{d}$(10). $\bar{d}$(15)) soit de ramener le nombre de configurations (Π/ϕ) de 157.888 à 4608.
0093	Chaque passage de New met 1 aux indices correspondant aux places cochées et 0 ailleurs dans le tableau PL(1::NP). EC contient en colonnes les écoulements déjà trouvés, la procédure Differ calcule la norme de (PL - chacune des colonnes de EC) et livre vrai si toutes ces normes sont non nulles.
0109	Util délivre vrai à l'itération 1
0145	Pafini est vrai tant que l'une des deux conditions suivantes est vérifiée. la première, carte 0135, est qu'il existe une place I cochée (PL(I)=1), différente de P et dont le fils privilégié n'est pas coché (TR(SUIV)=0). La deuxième, carte 0139 est qu'il existe une transition I qui soit cochée (TR(I)=1) et telle que, carte 0140 ,une place fils soit non cochée PL(AT(i,j))=0.
0158	FIL contient alors le nombre de fils de la place I déjà cochés par New. Si FIL = 0 alors I n'a pas de fils : on lui en coche un ,(ϕ(I)), et on met Pafini à vrai.
0161	Il en est de même pour PER.

```
0160 --                        PAFINI:=VRAI
0160 -8                        FIN;
0161 --                        SI PER=0 CO IL MANQUE PERE ; ALORS
0161 8-                        DEBUT
0162 --                        TR(PREC(I)):=1 ;
0163 --                        PAFINI:=VRAI
0163 -8                        FIN
0163 -7                        FIN
0163 -6                     FIN ;
0164 --                  POUR I:=1 JUSQUA NT FAIRE
0164 6-                     DEBUT
0165 --                     SI TR(I)=1 ALORS
0165 7-                        DEBUT
0166 --                        POUR J:=1 JUSQUA DT(I)+DT(I+NT) FAIRE
0166 --                        SI PL(ABS(AT(I,J)))=0 ALORS
0166 8-                        DEBUT
0167 --                        PL(ABS(AT(I,J))):=1;
0168 --                        PAFINI:=VRAI
0168 -8                        FIN
0168 -7                        FIN
0168 -6                     FIN ;
0169 --                  PAFINI
0169 -5                  FIN
0169 --               FAIRE CO NON FERME ; ;
0170 -4               FIN FERME ;
0171 --            CO INITIALISATIONS ;
0171 --            NBN:=NBN+1;
0172 --            POUR I:=1 JUSQUA NP FAIRE
0172 --            PL(I):=0 ;
0173 --            POUR I:=1 JUSQUA NT FAIRE
0173 --            TR(I):=0 ;
0174 --            PL(S):=1 ;
0175 --            CO CORPS DE PROC ;
0175 --            ATTAK ; cp Δ+1
0176 --            SI PL(P)=0 ALORS TROUVE:=FAUX
0176 --            SINON
0176 4-               DEBUT
0177 --               TROUVE:=VRAI ;
0178 --               FERME
0178 -4               FIN ;
0179 -3            FIN NEW ;
0180 --            ENTIER PROCEDURE PREC(ENTIER K) ;
0181 3-               DEBUT
0182 --               ENTIER R,L ;
0183 --               R:=D(K+NP) ; L:=1 ;
0185 --               TANTQUE
0185 4-                  DEBUT
0186 --                  SI AP(K,L)<0 ALORS R:=R-1 ;
0187 --                  L:=L+1 ; R>=1
0188 -4                  FIN FAIRE ;
0189 --               -AP(K,L-1)
0189 -3               FIN ;
0190 --            ENTIER PROCEDURE SUIV(ENTIER K) ;
0191 3-               DEBUT
0192 --               ENTIER R,L ;
0193 --               R:=D(K) ; L:=1 ;
0195 --               TANTQUE
```

Cartes	Commentaires
00169	Pafini est mis à faux au début de chaque itération de la boucle tant que qui commence carte 150. Si à la carte 169 il est encore faux c'est que les instructions 160, 163, et 168 n'ont pas été exécutées. Ces trois instructions signifient que le sous réseau coché est ouvert. On dit qu'on a réalisé la "fermeture ouverte" du sous réseau coché par Attak. Ceci justifie le nom du segment ferme.
0180	Prec(k) déduit de D et de AP le père privilégié de la place k, $\Pi(k)$.
0190	Suiv(k) déduit de D et de AP le fils privilégié $\phi(k)$ de la place k.

```
0196 --                  SI AP(K,L)>0 ALORS R:=R-1 ;
0197 --                  L:=L+1 ; R>=1
0198 -4                  FIN FAIRE ;
0199 --                AP(K,L-1)
0199 -3                FIN ;
0200 --          PROCEDURE INIT ;
0201 3-             DEBUT
0202 --             N:=0 ;
0203 --             NBN:=0;
0204 --             ITER:=1 ;
0205 --             POUR K:=1 JUSQUA 2*NP FAIRE
0205 --             OP(K):=0 ;
0206 --             POUR K:=1 JUSQUA 2*NT FAIRE
0206 --             OT(K):=0 ;
0207 -3             FIN INIT ;
0208 --          PROCEDURE TRAITER ;
0209 3-             DEBUT
0210 --             PROCEDURE ADJONCTION ;
0211 4-                DEBUT
0212 --                N:=N+1    ;
0213 --                POUR I:=1 JUSQUA NP FAIRE
0213 --                EC(I,N):=PL(I) ;
0214 -4                FIN ;
0215 --             ADJONCTION      ;
0216 --             ECRIRE("/");POUR I:=1 JUSQUA 40 FAIRE ECRIREC("/");ECRIRE(" ");
0219 --             ECRIRE("ITERATION  NUMERO",ITER);
0220 --             ECRIRE("ECOULEMENT NUMERO",N);   ECRIRE(" ");
0222 --             ECRIRE("PLACES"); ECRIRE(" ");
0224 --             POUR I:=1 JUSQUA NP FAIRE SI PL(I)=1 ALORS ECRIREC(I);
0225 --             ECRIRE(" ");ECRIRE(" ");
0227 --             ECRIRE("TRANSITIONS"); ECRIRE(" ");
0229 --             POUR I:=1 JUSQUA NT FAIRE SI TR(I)=1 ALORS ECRIREC(I);
0230 --             POUR I:=1 JUSQUA 20 FAIRE ECRIRE(" ");
0231 -3             FIN ;
0232 --          CO CORPS PROG PRINCIPAL ;
0232 --          INIT ;
0233 --          IMP ;
0234 --          TANTQUE
0234 3-             DEBUT
0235 --             SI UTIL ALORS NEW ;
0236 --             SI A TROUVE ALORS ECRIRE("LE RESEAU COMPORTE UN CIRCUIT")
0236 --             SINON  SI DIFFER ALORS TRAITER ;
0237 --             MISAJOUR ;
0238 --             Q=0
0238 -3             FIN
0239 --          FAIRE
0238 --             ITER:=ITER+1 ;
0239 --          ECRIRE("NOMBRE D'ITERATIONS              ",ITER);
0240 --          ECRIRE("EFFICACITE DE LA MISE EN OEUVRE ",NBN/ITER);
0241 --          ECRIRE("EFFICACITE DE L'ALGORITHME       ",N/ITER);
0242 --          ECRIRE("TEMPS D'EXECUTION PROPRE         ",TEMPS(1)/ 60."S");
0243 -2          FIN
0243 -1       FIN.

OPTIONS D'EXECUTION: TRACE.1
```

Cartes	Commentaires
0210	On adjoint l'écoulement trouvé à la matrice de couverture EC.
0238	Q est la "retenue" sortante de l'"incrémentation" de D, tant que Q = 0, D n'est pas saturé.
0240	NBN/ITER mesure la proportion d'appels de New. Il teste l'efficacité de l'utilisation de Util. Il révèle en particulier si le choix de la numérotation des sommets est judicieux.
0241	N/ITER mesure l'inverse du nombre moyen d'itérations nécessaires pour trouver un écoulement.
0242	Ce temps est en général légèrement inférieur à celui donné par le système.

Diagnostics de compilation

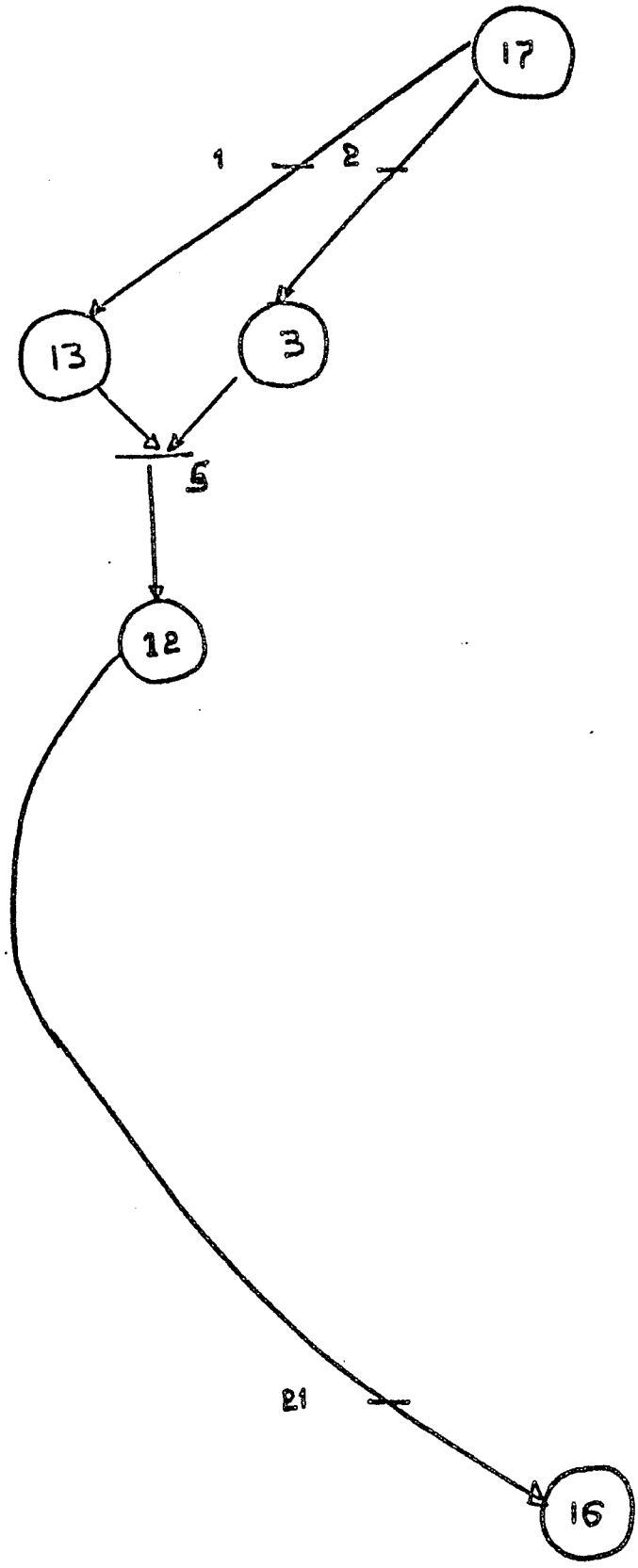
2.16 secondes de compilation. (10980,00 000) octets de code générés.

TABLEAU AP

1 /	-3	13	6	7	8	9	10	
2 /	-10	9						
3 /	-2	5	6	11	12			
4 /	-9	14	15	16				
5 /	-14	17						
6 /	-16	18						
7 /	-13	16	19					
8 /	-7	12						
9 /	-8	11						
10 /	-11	-23	-12	20				
11 /	-4	9	13					
12 /	-5	21						
13 /	-1	5	6	7	8			
14 /	-6	22						
15 /	-19	-22	-15	23				
16 /	-20	-21	-17	-18				
17 /	1	2	3	4				

TABLEAU AT

1 /	13	-17						
2 /	3	-17						
3 /	1	-17						
4 /	11	-17						
5 /	-3	12	-13					
6 /	-1	-3	-13	14				
7 /	-1	8	-13					
8 /	-1	9	-13					
9 /	-1	-2	4	-11				
10 /	-1	2						



Ecoulement numéro 1.

12	/	-3	-8	10
13	/	-1	7	-11
14	/	-4	5	
15	/	-4	15	
16	/	-4	6	-7
17	/	-5	16	
18	/	-6	16	
19	/	-7	15	
20	/	-10	16	
21	/	-12	16	
22	/	-14	15	
23	/	10	-15	

TABLEAU DP

6	1	4	3	1	1	2	1
1	1	2	1	4	1	1	0
4	1	1	1	1	1	1	1
1	1	3	1	1	1	1	3
4	0						

//

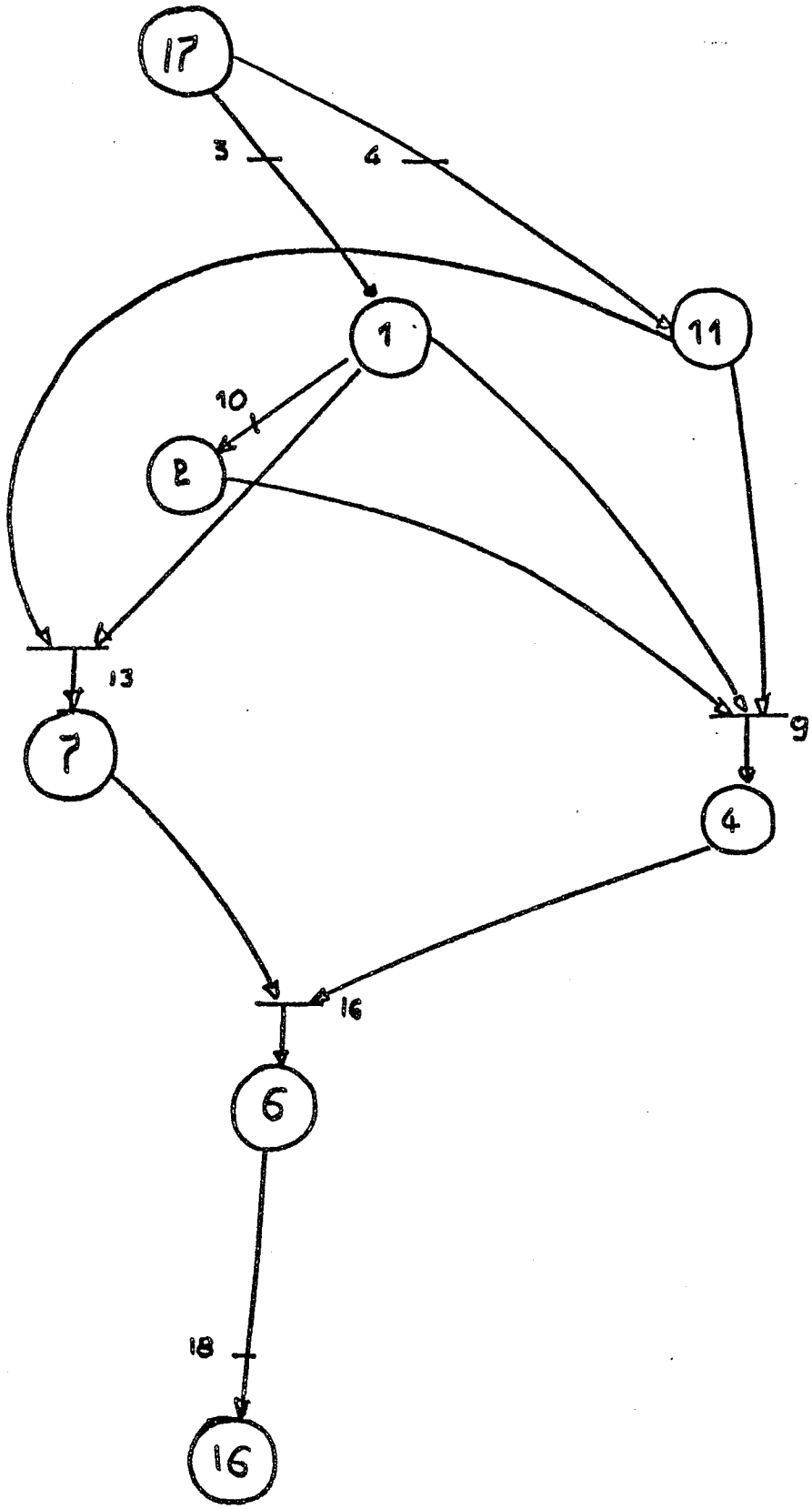
ITERATION NUMERO 1
ECOULEMENT NUMERO 1

PLACES

3	12	13	16	17

TRANSITIONS

1	2	5	21



Ecoulement numero 2

//

ITERATION NUMERO 3
ECOULEMENT NUMERO 2

PLACES

1	2	4	6	7	11	16	17

TRANSITIONS

3	4	9	10	13	16	18

//

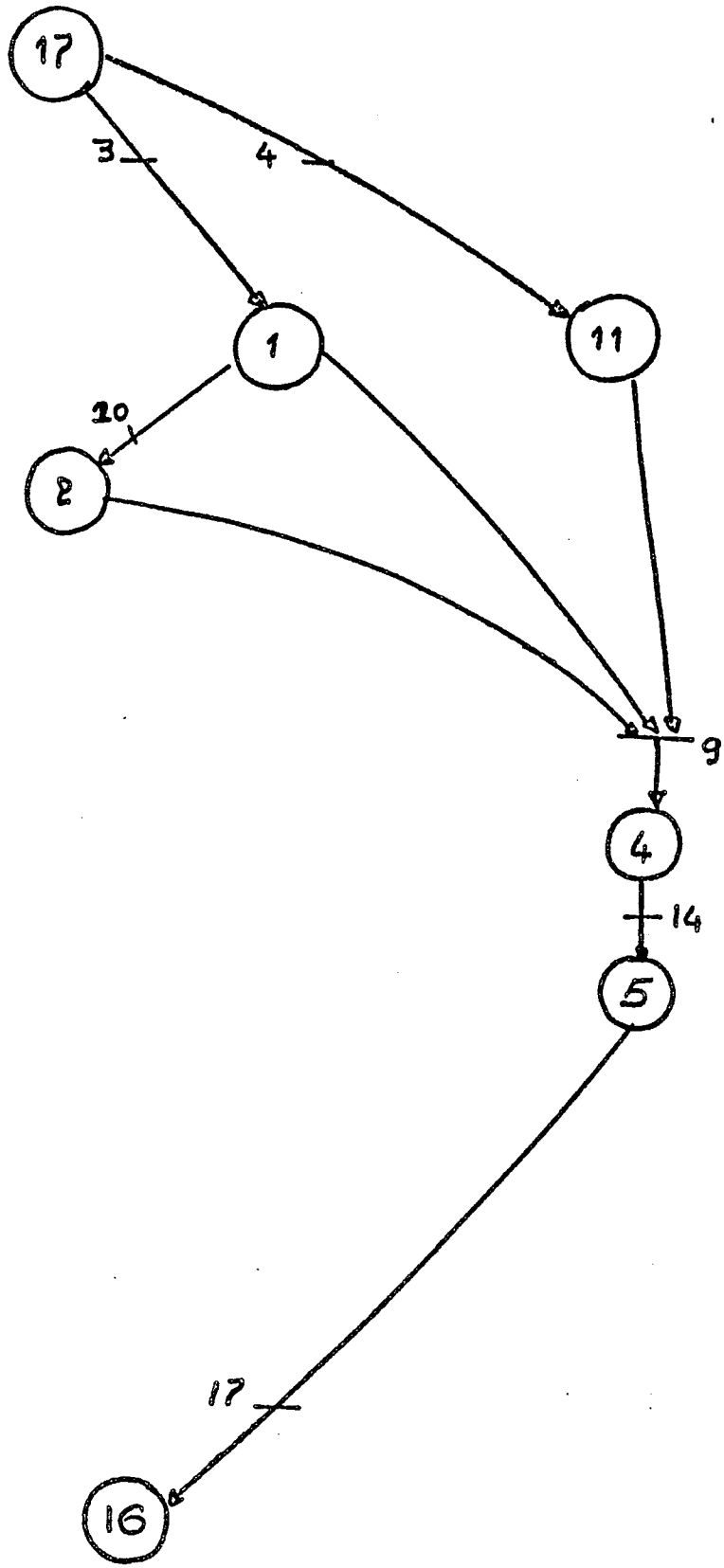
ITERATION NUMERO 4
ECOULEMENT NUMERO 3

PLACES

1	2	4	5	11	16	17

TRANSITIONS

3	4	9	10	14	17



Enseignement numéro 3.

//

ITERATION NUMERO 5
ECOULEMENT NUMERO 4

PLACES
1 3 10 13 14 15 16 17

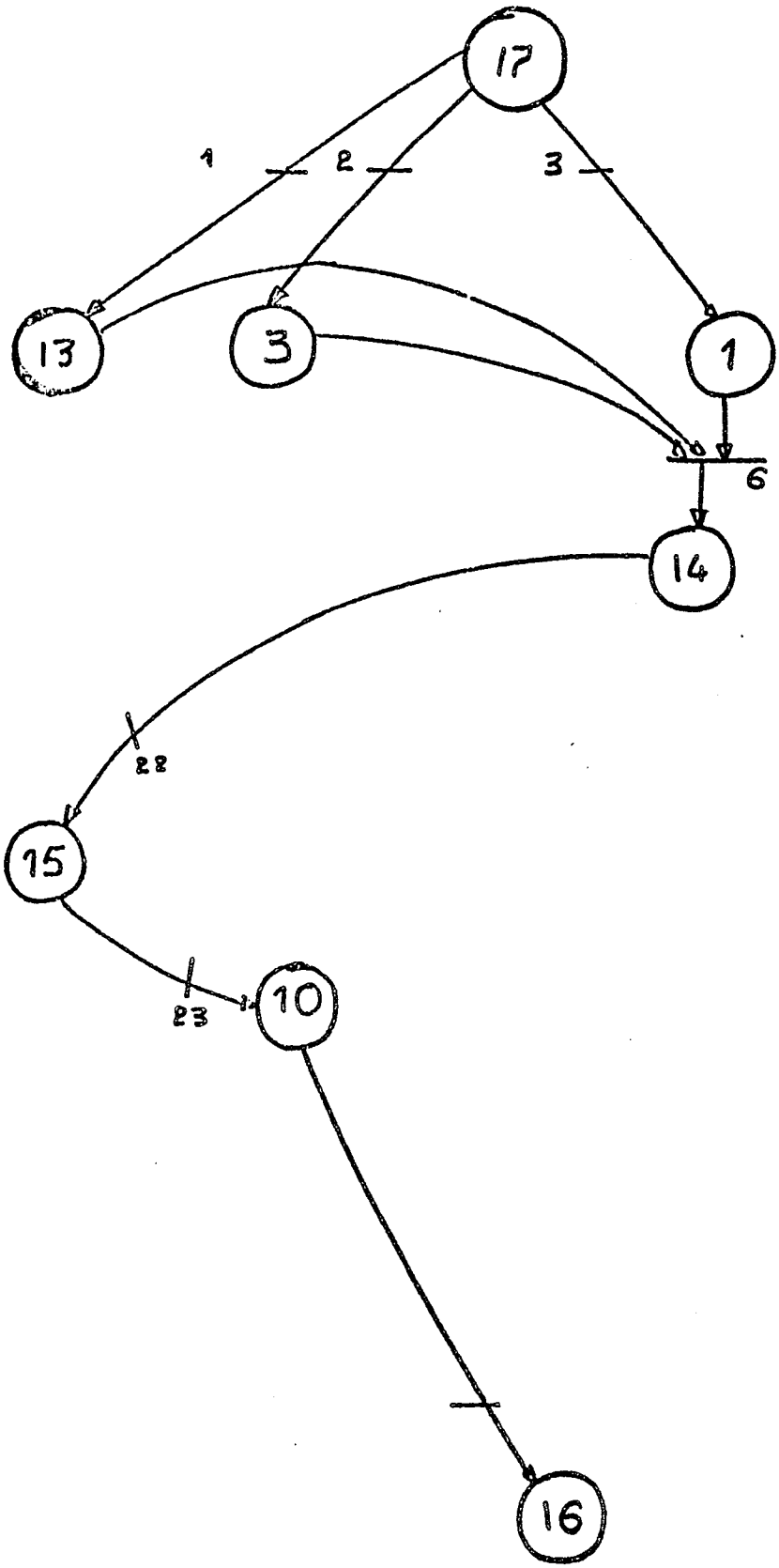
TRANSITIONS
1 2 3 6 20 22 23

//

ITERATION NUMERO 9
ECOULEMENT NUMERO 5

PLACES
1 3 8 10 13 16 17

TRANSITIONS
1 2 3 7 12 20



Seulement numéro 4.

//

ITERATION NUMERO 13
ECOULEMENT NUMERO 6

PLACES

1 3 9 10 13 16 17

TRANSITIONS

1 2 3 8 11 20

//

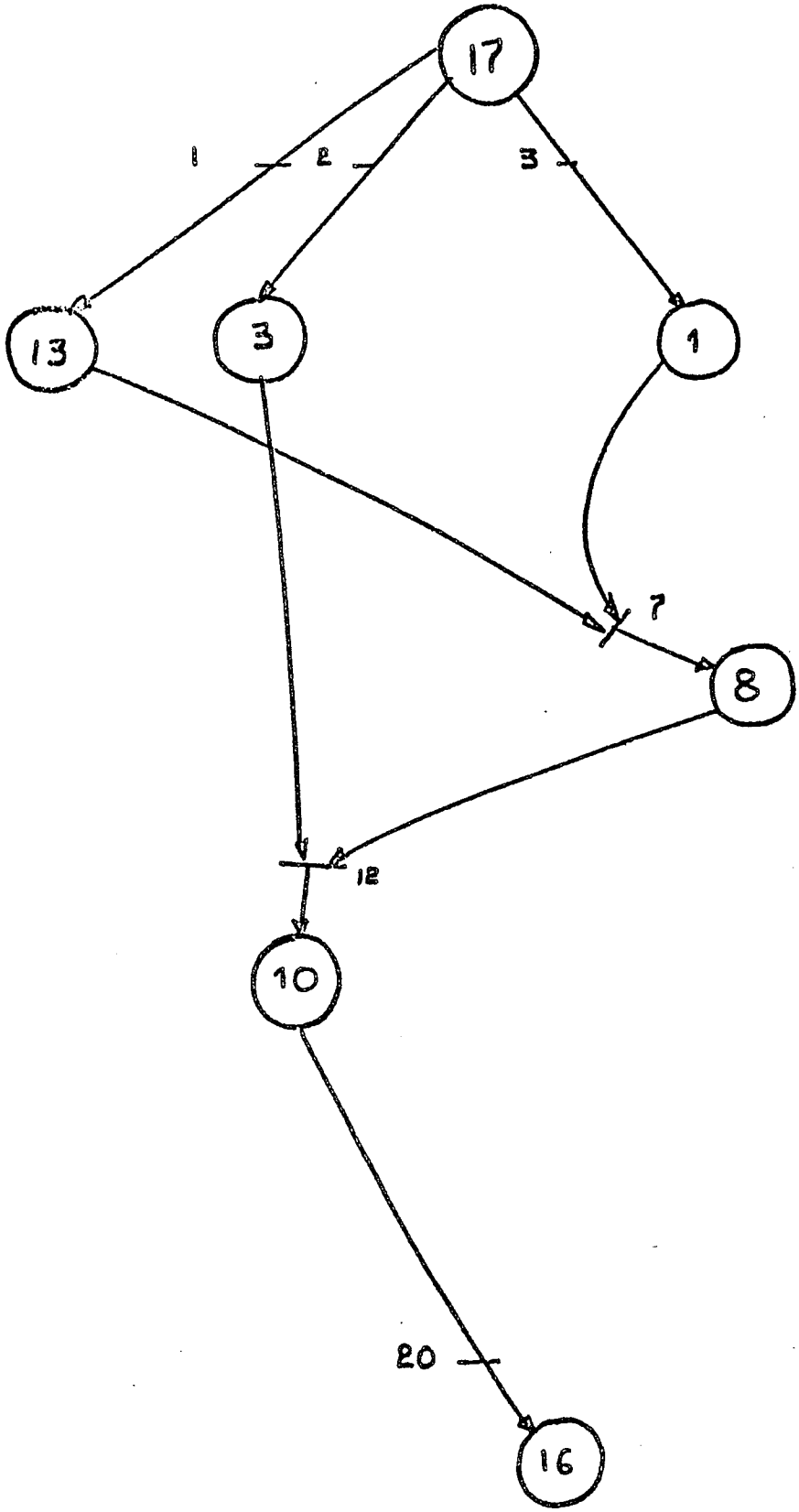
ITERATION NUMERO 35
ECOULEMENT NUMERO 7

PLACES

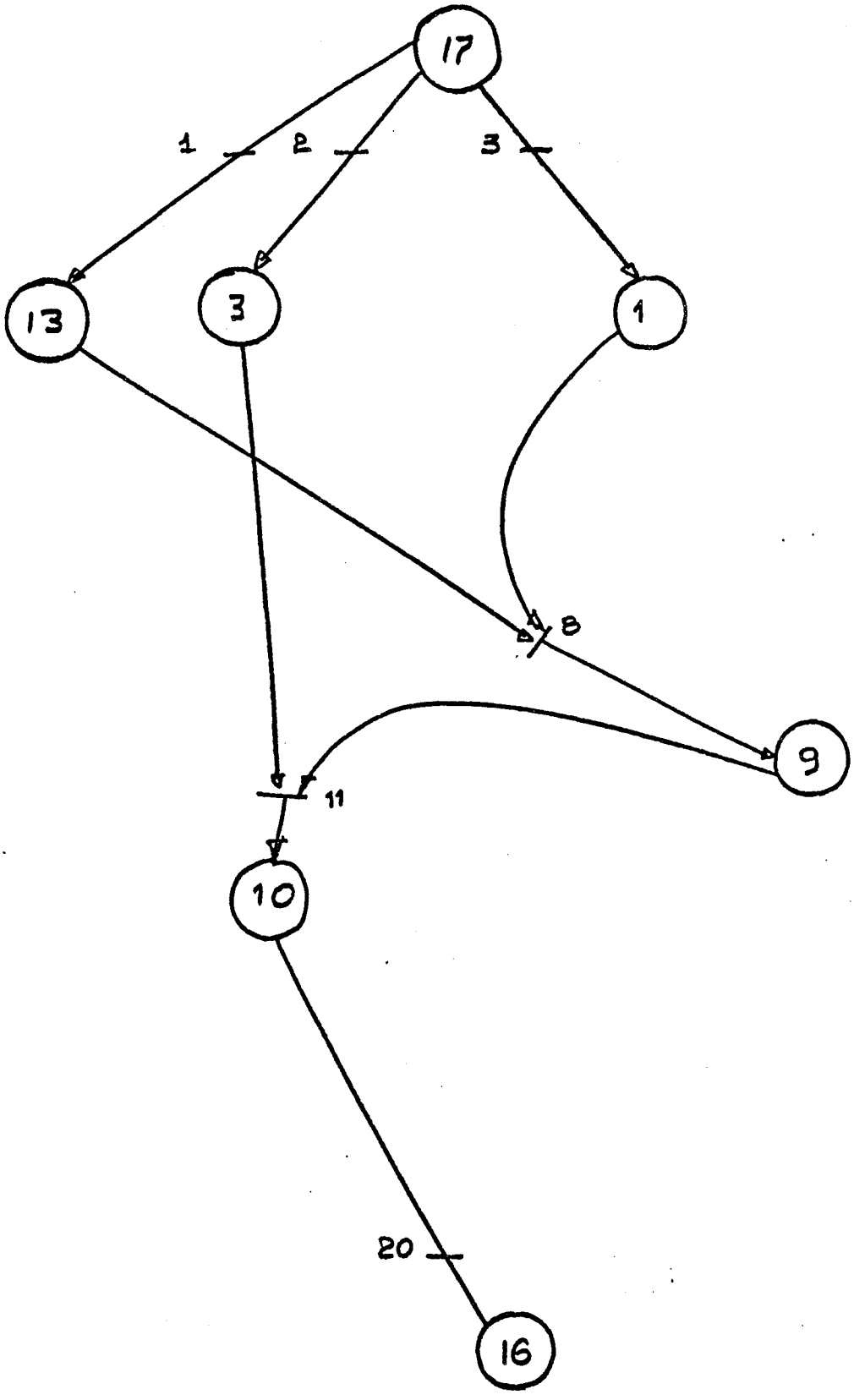
1 7 10 11 15 16 17

TRANSITIONS

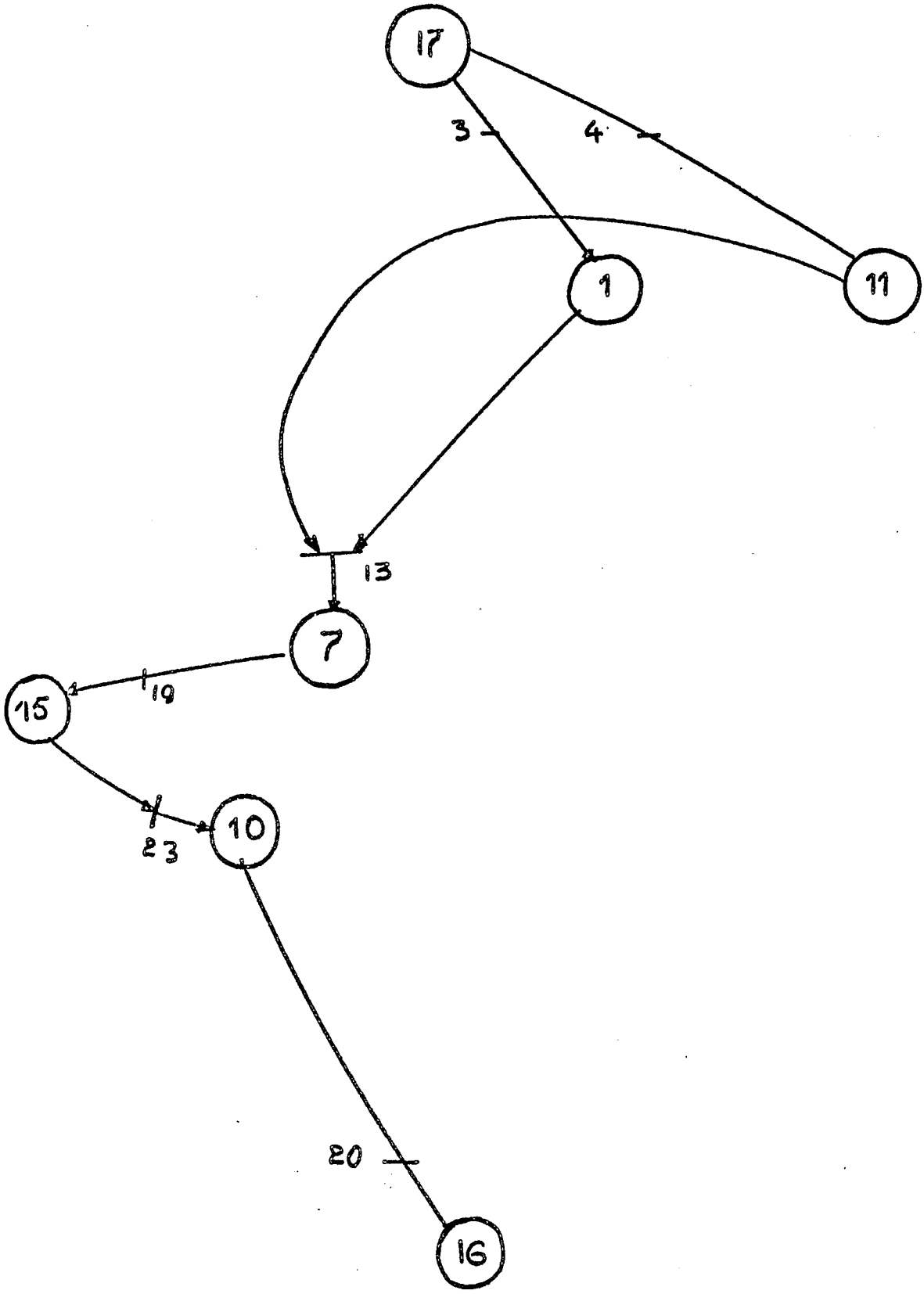
3 4 13 19 20 23



Earlement numero 5.



Enrolement Numero 6



Ecoulement numéro 7.

```
//////////////////////////////////////////

ITERATION  NUMERO          68
ECOULEMENT NUMERO           8

PLACES
          1          2          4          10         11         15         16         17

TRANSITIONS
          3          4          0          10         15         20         23

NOMBRE D'ITERATIONS                    4608
EFFICACITE DE LA MISE EN OEUVRE        1.000000
EFFICACITE DE L'ALGORITHME             0.001736111
TEMPS D'EXECUTION PROPRE               147.8000  S

147.81 SECONDES D'EXECUTION
```

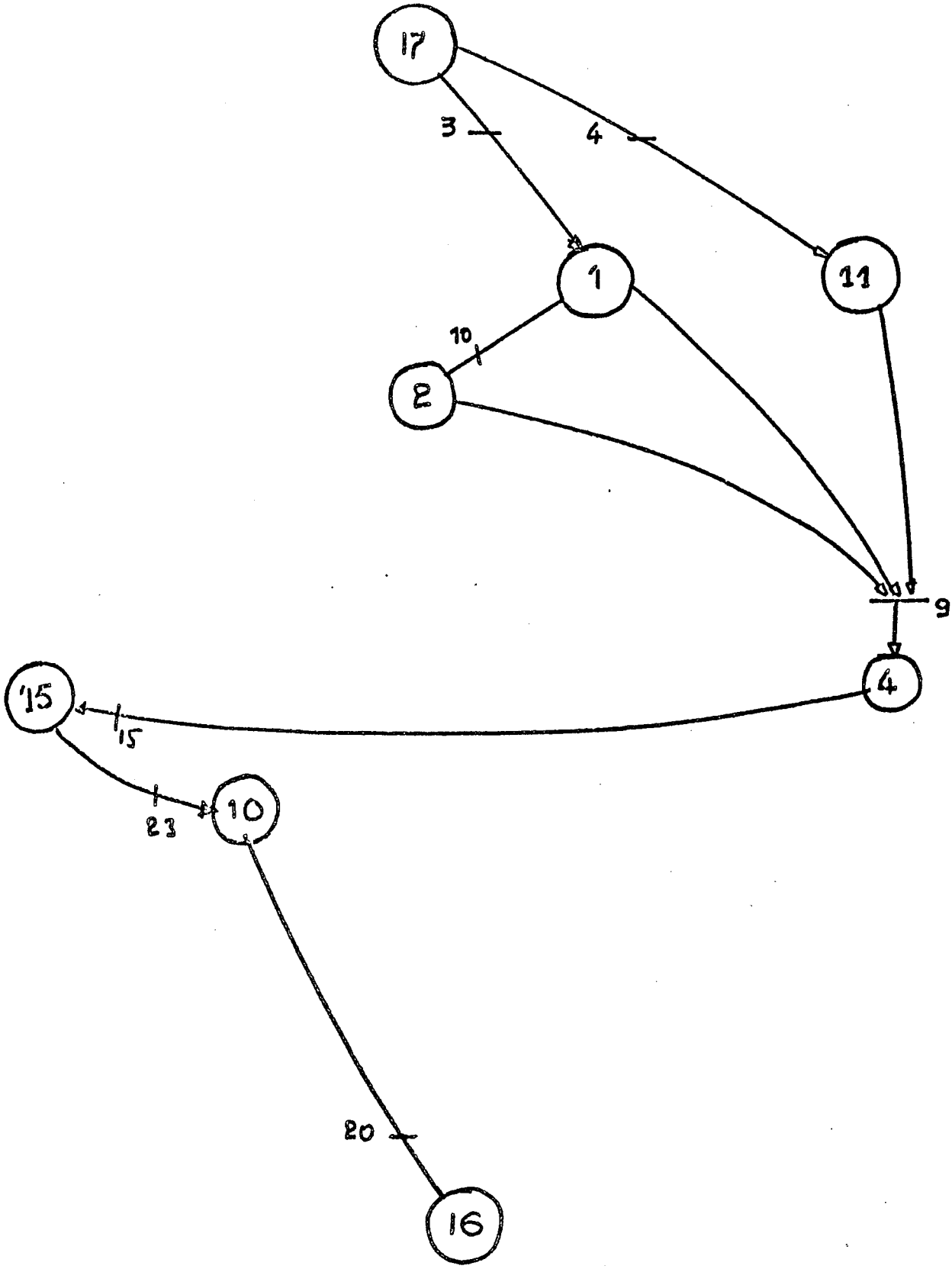
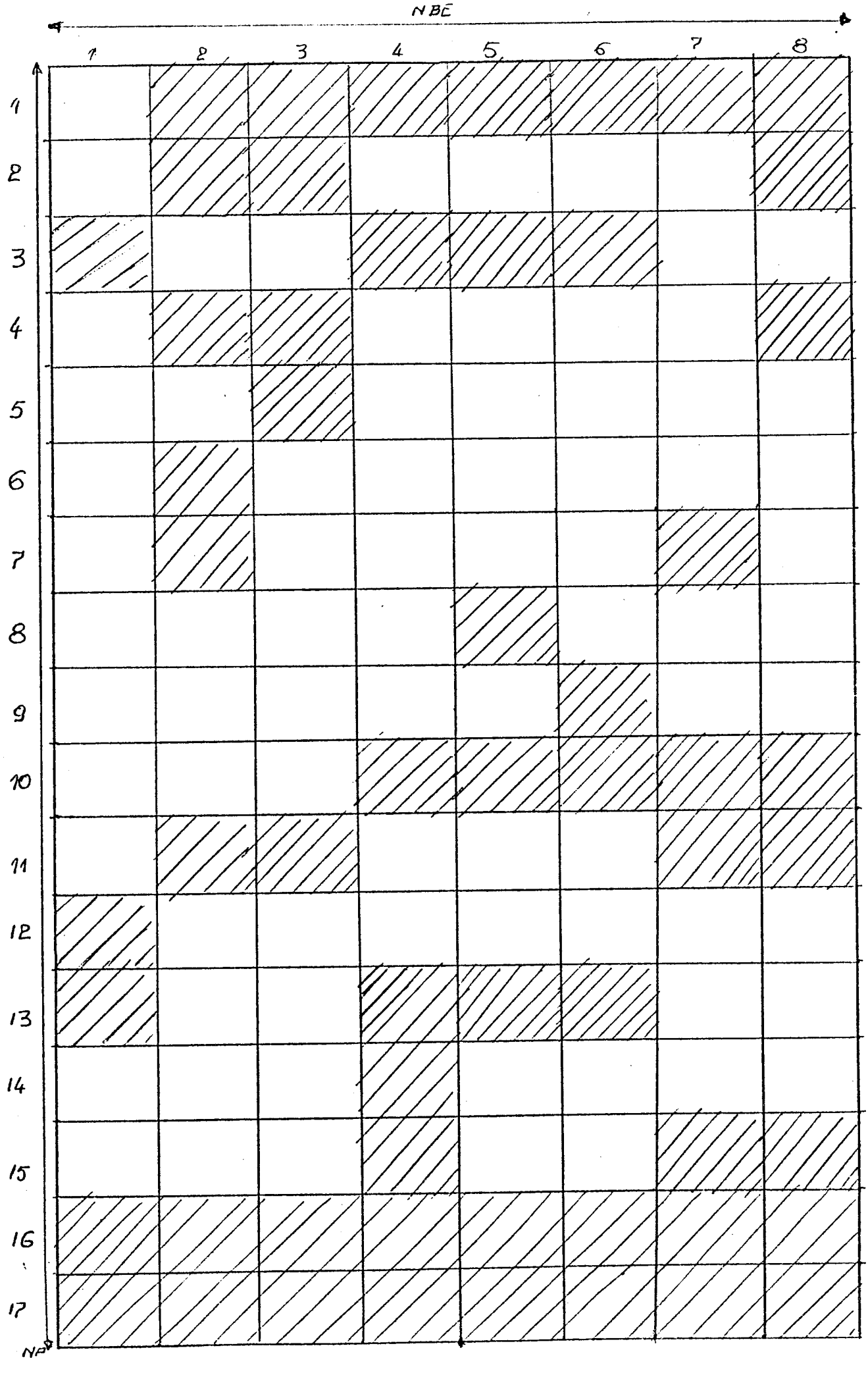


Tableau numéro 8.


NBE
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
NB

On conclut ce paragraphe par quelques remarques.

* Les écoulements trouvés sont tous élémentaires puisqu'ils sont tous minimaux.

* La figure 8 représente la matrice de couverture du réseau par les écoulements. C'est une copie de l'entier tableau EC (1::NP, 1 ::NBE) définie au début du programme.

* Le nombre d'itérations est 4608 ; c'est le produit des éléments plus grands que 1 du tableau DP sauf ceux d'indice 27, 31 et 32 pour lesquels la case du tableau DK est mise à faux. On voit que le huitième et dernier écoulement a été trouvé à la 68° itération, au bout de 2 secondes environ, et que, entre la 69° et la 4608° itération, New n'a fait que retrouver des écoulements déjà stockés dans EC.

* NBN/ITER vaut 1 => la numérotation des places est telle que Util a toujours délivré vrai => il y a autant d'appels New que d'itérations.

* En moyenne, il a fallu 600 itérations pour trouver un écoulement

* Enfin nous allons voir à titre d'exemple, l'écoulement du réseau initial que représente un écoulement du réseau réduit, en l'occurence l'écoulement 1. Il est composé, outre 16 et 17, des places 3,12 et 13.

3 regroupe 2, 14, 29 et 58 en application de R2

12 regroupe 40, 41, 48 et 54 en application de R1

13 regroupe 10 et 11 en application de R2.

Toutes les places regroupées dans 12 sont prises intégralement car 12 est formé en application de R1. En vertu des règles de séparation énoncées sur le mode de regroupement R2, on prendra les deux places de 13,

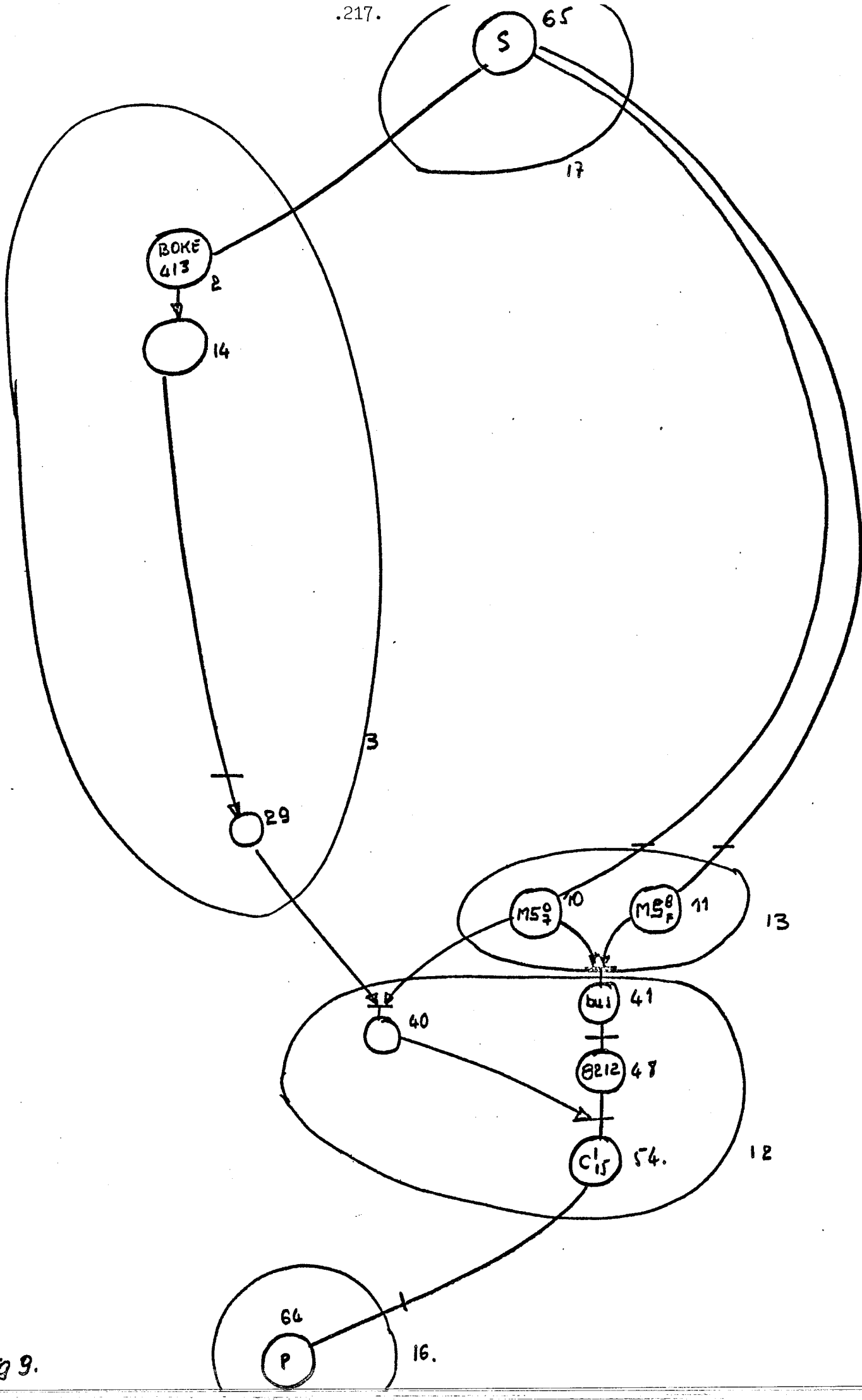

S
65
17
BOKE
413
2
14
3
29
MS^{0}_{7}
10
MS^{8}_{F}
11
13
bu1
41
40
8212
48
C^{1}_{15}
54.
12
64
P
16.

fig 9.

(10 et 11), et 2, 14 et 29 (et non 58) dans la place 3 du réseau réduit. On trouve l'écoulement du réseau initial représenté ci-contre, figure 9. Sur le schéma logique présenté au chapitre 1, ceci suggère de tester le bus de largeur 15 qui fait transiter l'information depuis (MS_7^0 - MS_F^8) jusqu'à C1 C15 avec les latches qui sont sur le bus ainsi que la circuiterie destinée à générer l'entrée validation DS1 des latches.

CONCLUSION

Cette thèse a consisté essentiellement en l'élaboration de paramètres numériques de test associés à un système logique, et en l'application à ces paramètres d'outils mathématiques de prise de décision.

La poursuite de ce travail pourrait se focaliser sur deux points :

- valider expérimentalement les propositions faites au chapitre 5
- Rationnaliser davantage les méthodes de test en applicant aux paramètres de test introduits au chapitre 4 des outils de décision chaque fois que c'est possible.

BIBLIOGRAPHIE

Les références sont classées par ordre chronologique de parution.

Méthodologies de test et paramètres de test.

[1] "Le test logique des composants"
Ch. ROBACH , G.SAUCIER
A paraître dans l'onde électrique 1978

[2] "Etude de la testabilité d'un système"
Rapport de contrat au CNET
P.CASPI Déc. 1977

[3] "Modèles pour l'étude de la testabilité des systèmes informatiques"
congrès AFCET : "Modélisation et maitrise des systèmes techniques, économiques, sociaux". Versailles
C.ROBACH et P.CASPI Nov. 1977

[4] "Constructeurs et utilisateurs face au test de microprocesseurs"
Rapport de recherche n°94. ENSIMAG
C.BELLON, P.CASPI
Ch.ROBACH, G.SAUCIER Nov.1977

[5] "Etude des réseaux porteurs d'information : leur application à la testabilité des systèmes informatiques". IFAC Work stop on Information and Systems. Compiègne.
P.CASPI, Ch.ROBACH, A.MILI Oct. 1977

[6] "Sur la vérification des systèmes digitaux" Thèse présentée pour obtenir le grade de Docteur D'état es sciences - USMG -
A.VERDILLON Sept. 1977

[7] "Application de Techniques mathématiques à la détermination de stratégies de test D.E.A. Génie Informatique - INPG -
A.MILI Juin 1977

[8] "Application de la Théorie de l'information à l'évaluation de la testabilité d'un système " Rapport interne n°69
ENSIMAG Grenoble
P.CASPI, et Ch.ROBACH Mars 1977

[9] "L'approche fonctionnelle dans la vérification des systèmes informatiques. Proposition d'un ensemble de méthodologies"
Thèse de docteur ingénieur INPG
M.MOALLA Déc. 1976

[10] "Partitionning for testability". International Symposium on fault tolerant computing. Pittsburgh ;
SHELDON B. AKERS Juin 1976

[11] "A testability measure for register transfer level digital circuits", International symposium on fault tolerant computing, Pittsburgh.
J.E. STEPHENSON and J.GRASON Juin 1976

[12] "Adaptive random test generation". Technical note N°73
digital systems laboraty. department of Electrical Engineering and computer Science, Standford University,
Standford, California
Kenneth P. Parker Oct. 1975

[13] "About random fault detection of combinational networks" Communication à IEEE transactions on computers
R.DAVID et G.BLANCHET Juil.1975

[14] "Fault detection indigital circuits" - F.F. KUO
A.D. Friedman et P.R. Menon 1970

[15] "diagnostic engineering requirements". Proceedings AFIPS, SJCC,
J.J. DENT 1968

Modèles et méthodes mathématiques

[16] "Techniques mathématiques de la recherche opérationnelle Optimisation dans les réseaux". Polycopié USM-INP-G
M.SAKAROVITCH 1977

[17] "Cours de DEA : Théorie de l'information" ENSEGP - INPG
G.JOURDAIN 1976

[18] "Introduction à l'étude des graphes"
Polycopié USMG-INPG
M.SAKAROVITCH 1975

[19] "Contribution à l'analyse de la décision floue : deux exemples d'application". Mémoire de D.E.S. de sciences économiques.
E.JOLLES 1975

[20] "A measure of computational work", IEEE transactions on computers
L.HELLERMAN - (vol C21 n°5 May 1972) 1972

[21] "Introduction à la théorie de la communication - Théorie de l'information". Masson
E.ROUBINE 1970

[22] "Integer and non linear-programming"
Electricité de France - Paris - et Institut de Statistiques universitaire - Paris -
J.ABADIE 1970

[23] "Analyse combinatoire - tome premier" - PUF
L.COMTET 1970

[24] "Analyse combinatoire - tome second" P.U.F.
L.COMTET 1970

[25] "Games and Decisions : Introduction and critical survey" - Harward University" - Wiley and sons
R.D. LUCE and H.RAIFFA 1958

TABLE DES MATIERES

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

Vu le rapport de présentation de Madame :

- G. SAUCIER, Maître de Conférences à l'Institut National Polytechnique de GRENOBLE

Monsieur Ali M I L I

est autorisé à présenter une thèse en soutenance pour l'obtention du titre de DOCTEUR de TROISIEME CYCLE, spécialité "Génie Informatique".

Grenoble, le 9 Juin 1978

Ph. TRAYNARD
Président
de l'Institut National Polytechnique