



HAL
open science

Algorithmique discrète et réseaux d'automates

Didier Pellegrin

► **To cite this version:**

Didier Pellegrin. Algorithmique discrète et réseaux d'automates. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1986. Français. NNT: . tel-00321866

HAL Id: tel-00321866

<https://theses.hal.science/tel-00321866>

Submitted on 16 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

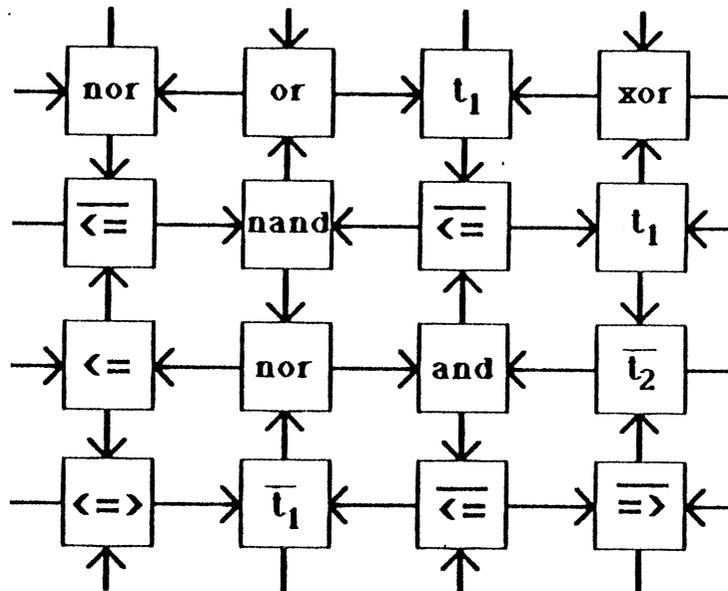
THESE

présentée par
Didier PELLEGRIN

pour obtenir le titre de DOCTEUR
de l'Université Scientifique Technologique et Médicale de Grenoble,

Mathématiques Appliquées

ALGORITHMIQUE DISCRETE ET RESEAUX D'AUTOMATES



Date de Soutenance : 23 Juin 1986

Composition du Jury :

F.Robert

Président

Y.Chiamella

M.Cosnard

J.Della Dora

Examineurs

F.Fogelman-Soulié

G.Weisbuch



UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie
PROFESSEURS DE 2ème CLASSE	
ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie





TABLE DES MATIERES

	pages
Introduction	1 à 16
Chapitre I : Test algorithmique de l'attraction d'un point fixe d'une itération discrète	1 à 40
I-0 Introduction	2
I-1 Principales définitions et propriétés	3
I-2 Conditions nécessaires et suffisantes épurées	6
I-3 Algorithme	11
I-4 Comparaisons, conclusions	18
I-5 Simulations	19
I-6 Un peu de combinatoire	25
Références	
Chapitre II : Racine carrée de matrices booléenne	1 à 31
II-0 Introduction	2
II-1 Racines $p^{\text{ième}}$ de matrices réelles	4
II-2 Résultats pour des matrices booléennes	6
II-3 Algorithme de calcul de racine de matrices booléennes quelconques	9
II-4 Présentation des résultats	20
II-5 Conclusion	25
Références	27
Chapitre III : Utilisation de fonctions d'énergie comme outil d'étude de réseaux à seuil	1 à 31
III-0 Introduction	2
III-1 Définitions	3
III-2 Conditions de convergence vers des points fixes	5
III-3 Iterations parallèles	12
III-4 Iterations séquentielles de fonctions majoritée	15
III-5 Modélisation de verres de spins par des fonctions majoritée	19
III-6 Conclusion	23
Références	24

Chapitre IV	: Dynamique de réseaux booléens aléatoires et applications.	1 à 61
IV-1	Introduction	2
IV-2	Definitions	4
IV-3	Rappels de résultats sur la dynamique de certains réseaux	8
IV-4	Dynamiques des réseaux booléens aléatoires d'interconnectivité 2	11
IV-5	Applications : Classification et Reconnaissance de séquences booléennes	28
IV-6	Conclusion	32
	Références	33

INTRODUCTION

La dynamique de grands systèmes composés "d'atomes" en interaction se situe à l'intersection de nombreuses études scientifiques, tant pratiques (étude d'un phénomène modélisé) que théoriques (étude du modèle).

Les travaux initiaux, auxquels J. V. Neumann a largement pris part, ont fourni un support à de telles études : les réseaux d'automates. La définition générale d'un réseau d'automates :

"ensemble de cellules interconnectées localement évoluant dans un temps discret par interactions réciproques"

est donc à la base de modèles discrets étudiés depuis une vingtaine d'années.

L'intérêt reconnu de ces réseaux réside dans leur surprenante capacité à modéliser un comportement fortement complexe alors qu'ils ne sont composés que d'éléments régis par des fonctions simples n'intervenant que sur des liens locaux. Cette propriété est la source de l'impressionnante diversité des domaines d'applications de ces modèles. Nous énumérons ici les différents champs d'études.

1) INFORMATIQUE

J. V. Neumann met au point, dès le début des années 50, un réseau d'automates ayant la propriété d'autoreproduction [NE1], [NE2], [BU]. Sur cette voie on retrouve plus récemment le fameux jeu de la vie de Conway [BE].

On observe, en fait, à l'aube des recherches sur la science des calculateurs une quête de simulation de comportement du monde vivant. Cette quête se poursuit depuis, aux côtés des développements technologiques : le récent succès du connexionisme en est actuellement le reflet [FO3]. Le connexionisme recouvre un ensemble de techniques, basées sur la théorie des automates, qui ont pour but de réaliser des tâches "intelligentes" : reconnaissances de formes, apprentissage, compréhension de langues naturelles..... Ici ce sont des propriétés du système nerveux que l'on tente d'imiter (cf. [AC], [DE], [FE], [FO1], [FU], [HIL], [HIN], [HU], [SEJ], [SEI], [WA]). La question se pose déjà de savoir si ces techniques seront à même de concurrencer les systèmes experts dans certains domaines de l'Intelligence Artificielle.

Les réseaux d'automates occupent également d'autres fonctions en informatique, principalement :

- étude et gestion du parallélisme dans les différents modèles de machines parallèles (cf. [YRO])
- étude et validation de schémas de synchronisation de processus (réseaux de Pétri par exemple) (cf. [BR])

2) PHYSIQUE

Ici aussi l'emploi des réseaux d'automates est d'importance, et il y est de plus en pleine expansion. A tel point qu'un calculateur spécialisé a été mis au point au M.I.T. par T. TOFFOLI pour simuler le comportement de ces réseaux. (cf. [FA])

G. WICHNIAC qui en a fait une démonstration, au colloque "Disordered Systems and Biological Organization" (école de physique des Houches, Février 1985, [BI]) a présenté quelques réseaux d'automates : calculateurs universels, modèles de verres de spins, modèles de phénomènes de transition instables et métastables [WI]. A l'entendre, les réseaux d'automates ne tarderont pas à contester l'égémonie des Equations aux Dérivées Partielles comme outils privilégiés dans la modélisation de phénomènes physique, pour certains types de problèmes. A ce propos, un avantage indéniable et primordial de ces modèles est l'utilisation optimale des capacités de l'ordinateur : rapidité, exactitude, possibilité de parallélisme, voir de "facilité" de conception de machines spécialisées.

3) BIOLOGIE

Mac CULLOCH et PITTS sont les premiers à modéliser le comportement d'un neurone par un automate à seuils [MA]. Cette idée a été largement reprise par la suite [AM], [CO], [PER]. De ces études sont parfois ressorties des techniques de reconnaissance de formes et d'apprentissage; on rejoint par cette voie les modèles connexionistes.

Le système nerveux n'est pas le seul domaine de cette matière qui emploie des réseaux d'automates, citons par exemple :

- la différenciation cellulaire par S. KAUFFMAN [KA1-4]],
- des manifestations d'auto-organisation par H. ATLAN [AT1-2],
- un modèle de l'évolution des espèces par G. WEISBUCH [WE].

A la fin de ce tour d'horizon des principaux champs d'application de ce type de modèles nous ne ferons qu'évoquer leur emploi dans les sciences sociales.

Quelles sont les origines du succès remporté par cet outil ?

- Possibilité d'obtenir un comportement complexe et global à l'aide d'actions simples et locales.
- Réduction d'un grand ensemble d'états possibles (conditions initiales) pour un système, à un ensemble restreint d'états admissibles (états limites). Ces attracteurs, caractéristiques du réseau, font de ce dernier un selectionneur dynamique d'états possibles.

Cette dernière remarque justifie l'importance des études dans ce domaine qui se concentrent sur les propriétés caractéristiques de ces attracteurs : composition, taille, composition des bassins d'attraction, longueur des transitoires. En France l'école grenobloise a déjà largement contribué à cette recherche [GO], [TC], [FO3], [FRO], [YRO], [LE], [SN].

La diversité des matières concernées est à la source d'un autre trait symptomatique de ces modèles : l'étroite coopération entre chercheurs de formation très différentes. L'étude des mémoires associatives fournit un exemple significatif de cette coopération. Ces mémoires adressables par contenu ont la caractéristique de reconnaître comme éléments mémorisés des "données bruitées" (partiellement erronées). Biologistes, physiciens, informaticiens et mathématiciens se sont largement penchés sur ce problème ([AM], [FO1], [HOP], [KO]). Pour donner un exemple de complémentarité nous pouvons citer F. FOGELMAN [FO1] utilisant le concept de dérivée discrète introduit par F. ROBERT [FRO] pour établir un résultat de convergence sur le modèle de mémoires associatives proposé par J.HOPPFIELD [HO].

Il est certain que l'origine des chercheurs concernés par les réseaux d'automates influence le type des travaux produits : observation du comportement par simulation, étude statistique, modification du réseau pour en faire un outil de reconnaissance efficace, étude théorique de la convergence vers l'état limite.....

Néanmoins les attracteurs d'un réseau donné constituent toujours le centre de ces études. Dans ce cadre le travail présenté dans cette thèse se divise en deux parties :

Algorithmique et Itérations Discrètes : chapitres 1 et 2,

Etude de deux types de Réseaux (à seuils et aléatoires) : chapitres 3 et 4.

Contenu de la thèse

Chapitre I

Test algorithmique de l'attraction d'un point fixe d'une itération discrète

On trouve dans [FRO1] une étude complète de la théorie des itérations discrètes, cette référence se poursuit depuis par plusieurs rapports de recherche TIM3/IMAG ([FRO2]). Dans un de ces derniers figure l'expression de conditions nécessaires et suffisantes à l'attraction d'un point fixe dans son voisinage second.

Soit F une application booléenne de $\{0,1\}^n \rightarrow \{0,1\}^n$ et ξ un point fixe de F (i.e. $F(\xi)=\xi$). A l'aide des conditions nécessaires et suffisantes on peut vérifier que :

Quelque soit x_0 de $\{0,1\}^n$ différant de ξ par au plus deux composantes, l'itération $x_{n+1} = F(x_n)$ partant de x_0 converge vers ξ sans qu'aucun des itérés de x_0 à ξ ne diffère de ξ par plus de deux composantes.

Ces conditions utilisent la dérivée discrète de F calculée en x et dans son voisinage premier. Nous avons déjà mentionné l'intérêt de ce type de résultat pour l'étude des réseaux d'automates booléens. Dans ce premier chapitre nous proposons un test algorithmique simple permettant d'effectuer la même vérification. Ce test présente deux avantages sur la démarche théorique :

i) le coût de vérification est inférieur dans le test. Nous avons défini un coût basé sur les deux opérations élémentaires que sont l'addition et la comparaison de deux vecteurs de $\{0,1\}^n$. Le coût calculé est celui "du plus mauvais cas".

ii) le test ne tient pas compte du contexte topologique, il s'applique donc également à la vérification de l'attraction d'un attracteur quelconque (point fixe ou cycle limite), et vis à vis d'un ensemble de points quelconque (voisinage ou non de l'attracteur).

A la fin de ce chapitre nous dénombrons les différents types de bassins d'attraction comportant n éléments, pour n allant de 1 à 170.

Chapitre II

Racine carrée de matrices booléennes

Le sujet abordé ici n'a pas de lien direct avec les réseaux d'automates, mais les questions d'existence et de calcul de la racine carrée de matrices se trouvent fréquemment posée dans la théorie des itérations. Parmi les principales références nous trouvons le mathématicien G. Targonski ([TA]) qui s'intéresse à l'équation fonctionnelle, et deux autres canadiens : C. Reischer et D.A. Simovici qui ont publié récemment un survey "Iterations properties of multivalued switching functions". [RE]

Le problème général est le suivant :

soit F une application d'un ensemble fini S dans lui-même
soit n un entier supérieur à 1

Existe-t-il une application G de S dans S telle que $G^n = F$?

Si la réponse est oui : calculer G

Cette question a été entièrement traitée pour des applications de \mathbb{R}^n dans \mathbb{R}^n par E. D. Denman (voir les références du second chapitre).

Nous présentons succinctement les techniques employées dans le contexte réel (Newton accéléré) et nous constatons l'impossibilité de les adapter aisément au contexte discret.

Targonski et G. Zimmerman ont résolu le problème de l'existence dans certains cas, mais le théorème caractérisant cette existence n'est absolument pas constructif.

Reischer et Simovici proposent un algorithme de calcul de la racine $n^{\text{ième}}$ d'une permutation. Ici nous présentons un algorithme qui répond à la question pour le cas $n = 2$ et qui calcule la racine carrée d'une application booléenne quelconque lorsqu'elle existe. L'algorithme est décrit en terme de calcul de racine de graphe.

Les résultats obtenus recouvrent bien évidemment ceux de [RE] et les étendent aux applications quelconques. L'exécution d'un tel algorithme accredit l'idée de K. H. Kim sur la complexité de cette question (il conjecture que ce problème est NP-Complet pour les applications quelconques).

Chapitre III

Utilisation de fonctions d'énergie décroissantes comme outil d'étude de réseaux à seuil

Nous nous intéressons dans ce chapitre à un type de réseaux d'automates parmi les plus fréquemment utilisés : les réseaux à seuil.

Définitions

On appelle *réseau à seuil* une application $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ où F est un n -uplet de fonctions booléennes de n variables, à seuil.

$$F = (f_1, \dots, f_i, \dots, f_n) \quad \text{où}$$

$$\forall i \in \{1, 2, \dots, n\} \text{ et } \forall x \in \{0, 1\}^n \quad f_i(x) = \mathbb{1} \left[\sum_{j=1}^n a_{ij} x_j - b_i \right]$$

$$\text{où } \mathbb{1}[u] = \begin{cases} 1 & \text{si } u \geq 0 \\ 0 & \text{sinon} \end{cases}$$

$A = ((a_{ij}))$ est une matrice réelle (n, n) , appelée matrice de connexion, et b est un vecteur réel (n) , appelé vecteur de seuil.

$$\text{L'itération générale} \quad x(t+1) = F(x(t))$$

partant d'une condition initiale $x(0) \in \{0, 1\}^n$ s'effectue selon l'un des trois modes suivants :

$$\textit{parallèle} : \quad \forall i \in \{1, 2, \dots, n\}, \quad x_i(t+1) = f_i(x(t))$$

$$\textit{série} : \quad \forall i \in \{1, 2, \dots, n\}, \quad x_i(t+1) = f_i(x_1(t+1), \dots, x_{i-1}(t+1), x_i(t), \dots, x_n(t))$$

en modifiant l'ordre des composantes on obtient toutes les itérations séries possibles.

bloc - série :

C'est le mode mixte : l'ensemble des composantes est partitionné en blocs que l'on itère en parallèle, et à l'intérieur desquels l'itération se mène en série. Les deux modes précédents sont en fait des cas particulier de celui-ci.

E. Goles a déjà démontré d'importantes conditions de convergence vers un point fixe pour de tels réseaux. Dans [FO4] on trouve une borne de la taille du transitoire dans le cas d'une itération série pour un réseau dont la matrice de connexion est symétrique à éléments diagonaux positifs ou nuls.

Là, en définissant une énergie liée au réseau

$$\forall x \in \{0,1\}^n, \quad E(x) = -1/2 \sum_{i=1}^n x_i \sum_{j=1}^n a_{ij} x_j + \sum_{i=1}^n b_i x_i$$

nous généralisons les résultats précédents à une itération bloc-série, et nous donnons une autre démonstration de la propriété établie par E. Goles pour les itérations parallèles : "avec une matrice de connexion symétrique l'itération parallèle d'un réseau à seuil conduit soit à un point fixe soit à un cycle limite à deux états".

Enfin, nous appliquons ces résultats à deux réseaux à seuil particuliers :

- itération série pour des fonctions majorité
- itération série pour des "verres de spins généralisés".

Chapitre IV

Dynamique de réseaux booléens aléatoires et applications

Dans ce dernier chapitre nous menons une étude détaillée de réseaux booléens aléatoires. Ces réseaux aléatoires ont été introduits par le biologiste américain S. Kauffman qui travaillait sur la différenciation cellulaire (voir [KA1 - 4]). Il a spécialement porté son attention sur les réseaux de connectivité 2 : il justifie cet intérêt en détail dans [KA1].

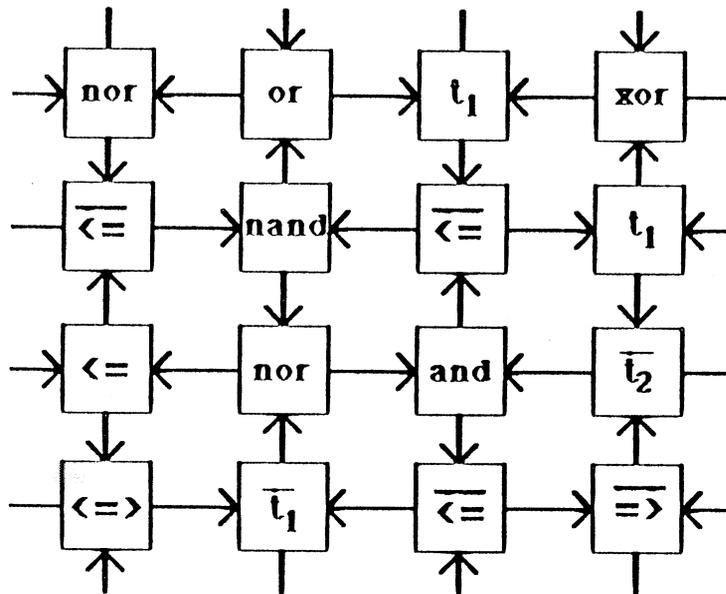
Définitions

$F \{0, 1\}^n \rightarrow \{0, 1\}^n$, est un réseau booléen d'interconnectivité 2 si F est un n -uplet de fonctions booléennes à n variables, et si chacune de ces fonctions ne dépend que de 2 de ses variables.

C'est le *graphe de connexion* qui précise ces dépendances.

Il y a 16 lois booléennes à 2 variables elles sont choisies aléatoirement pour former un réseau à connexions régulières.

Exemple



Parmi les nombreux travaux sur ce sujet (voir les références du chapitre 4) les plus récents ont été conduits au Laboratoire de Dynamiques des Réseaux (Paris). Ils traitent de l'étude du comportement dynamique de ces réseaux et de leurs capacités à différencier ("reconnaître") des séquences booléennes. Du point de vue biologique il s'agit d'observer un comportement d'auto-organisation [AT].

L'étude la plus complète est réalisée par F. Fogelman [FO2]. Elle emploie la notion de fonction forçante (provenant des travaux de Kauffman) pour mettre au point un outil d'étude du comportement dynamique du réseau : *le domaine forçant*. La dynamique de ces réseaux offre en effet une surprenante robustesse au changement de conditions initiales. Il est alors naturel de partitionner le réseau en trois classes de cellules :

i) les cellules qui ont un état constant dans tous les cycles limites forment le cœur stable.

ii) les cellules n'ont pas le même comportement dans tous les cycles limites.

La connaissance du cœur stable du réseau se révèle être capitale dans l'étude de la

dynamique, malheureusement sa définition n'est pas constructive ; il est impossible de disposer de l'ensemble des cycles limites. Le domaine forçant est une approximation de ce coeur.

Nous comparons les avantages respectifs des différents moyens d'approcher cette structure toujours stable : simulations de nombreuses dynamiques, domaine forçant,..

En s'appuyant sur la comparaison précédente, nous proposons une modification de la définition du domaine forçant. Partant des états stables du coeur nous pouvons poursuivre *l'étude des autres composantes du réseau* :

- le coeur oscillant (recherche d'une structure de base : le circuit frustré)
- les autres cellules (regroupement par degré d'oscillation identique : les paliers).

Enfin, nous pouvons étudier les réactions de ces dynamiques face à une perturbation extérieure : on force l'état d'une cellule sans tenir compte de sa fonction régissante. La réception d'un tel bruit permet de considérer qu'un réseau aléatoire peut "reconnaître" des séquences booléennes. Ce dernier point est plus développé dans [AT2].

REFERENCES

- [AC] D.HACKLEY, G.E.HINTON, T.J.SEJNOWSKI
A learning Algorithm for Boltzmann Machines
Cognitive Science 9, pp 147-169, 1985.
- [AM] S.AMARI, M.A.ARBIB Eds
Competition and Cooperation in neural nets.
Lecture Notes in Biomathematics, 45, Springer Verlag, 1982.
- [AT1] H.ATLAN, F.FOGELMAN-SOULIE, J.SALOMON, G.WEISBUCH :
Random Boolean Networks
Cybernetics and Systems 12, 1981 pp103-121.
- [AT2] H.ATLAN, E.BEN-EZRA, F.FOGELMAN-SOULIE, D.PELLEGRIN,
G.WEISBUCH
*Emergence of classification procedures in automata networks as a model
for functional self-organization.*
Theoretical Biology 1986, à paraître .
- [AT3] H.ATLAN
Two instance of self_ organization in probabilistic automata networks
Dynamical Systems and Cellular Automata, J.DEMONGEOT, E.GOLES,
M.TCHUENTE Eds, Academic Press, 1985
- [BE] E.R. BERLEKAMP, J.H. CONWAY, R.K. GUY
Winning Ways
Academic Press, 1982.
- [BI] E.BIENENSTOCK, F.FOGELMAN-SOULIE, G.WEISBUCH Eds.
Disordered Systems and Biological Organization
Nato ASI Series in Computer and Systems Sciences, Springer Verlag
1986.
- [BR] G.W.BRAHMS
Réseaux de Petri, Théorie et Pratiques
Masson 1983.
- [BU] A.W BURKS
Essays on Cellular Automata
University of Illinois Press, 1970.

- [CO] M.COSNARD, E.GOLES-CHACC
Dynamique d'un automate à mémoire modélisant le fonctionnement d'un neurone
C.R. Acad. Sciences Paris, T299, serie 1n°10, pp459-461, 1984
- [DE] M.F. DEERING
Architecture for AI
Byte pp193-206, avril 1985.
- [FA] D.FARMER, T.TOFFOLI, S.WOLFRAM Eds.
Cellular Automata
North Holland 1984.
- [FE] J.A.FELDMAN
Connections
Byte pp277-283, avril 1985
- [FO1] F.FOGELMAN-SOULIE
Pattern Recognition by Threshold Networks
Dans les Actes du Colloque International d'Intelligence Artificielle,
Marseille 1984, à paraître.
- [FO2] F.FOGELMAN-SOULIE
Contributions à une théorie du Calcul sur Réseaux
Thèse d'Etat, Grenoble fév. 1985.
- [FO3] F.FOGELMAN-SOULIE
Cerveau et Machines : des Architectures pour demain
Les Actes du Colloque COGNITIVA 1985, CESTA-AFCET Eds Paris
1985.
- [FU] K.FUKUSHIMA, S.MIYAKE
*Neocognitron : a self organizing neural network model for a mechanism of
visual pattern recognition*
Lectures notes in Biomathematics *Competition and Cooperation in Neural
Nets*, S.AMARI, M.A.ARBIB Eds, Springer Verlag 1982.
- [GO1] E.GOLES-CHACC
Comportement Dynamique des Réseaux d'Automates
Thèse d'Etat, Grenoble fév. 1985.
- [GO2] E.GOLES-CHACC
Fixed point behaviour of threshold function on a finite set.
SIAM J. on Alg. and Appl. Maths, 3, n°4

- [HIL] W.D.HILLIS
The Connection Machine : a computer architecture based on cellular automata.
Cellular Automata D.FARMER, T.TOFFOLI, S.WOLFRAM Eds.
Physica D, 10, pp213-229, North Holland 1984.
- [HIN] G.E.HINTON
Learning in Parallel Networks
Byte pp265-273, april 1985.
- [HO] J.J.HOPPFIELD
Neural networks and physical systems with emergent collective computational abilities
Proc. Nat. Acad. Sciences USA, Vol. 79 pp 2554-2558, 1982.
- [HU] B.A.HUBERMAN, T.HOGG
Adaptation and Self Repair in Parallel Computing Structures
Phys. Rev. Lett. Vol.52 n°12, pp1048-1051, 1984.
- [KA1] S.KAUFFMAN
Metabolic stability and epogenesis in randomly constructed genetic nets
Theoretical Biology, 22, pp437-467.
- [KA2] S. KAUFFMAN
Behaviour of randomly constructed nets
In Towards a theoretical biology Ed. C.H. Waddington, vol3, Edinburg University Press, 1970.
- [KA3] S. KAUFFMAN
The organization of cellular genetic control systems
Lecture on Mathematics in the Life Sciences, Vol 3, American Mathematical society, Providence, pp63-116, 1972.
- [KA4] S. KAUFFMAN
Assessing the probable regulatory structures and dynamics of metazoan genome.
In Lecture notes for Biomathematics. Kinetic logic Ed. R. Thomas, 29, pp30-61. Berlin Springer Verlag, 1979.
- [KO] T.KOIHONEN
Self Organization and Associative Memory
Springer Verlag, series in Information Sciences, Vol.8 1984.

- [LE] M.LEGENDRE
Analyse et simulation de réseaux d'automates
Thèse de Docteur Ing., Grenoble 1982.
- [MA] W.MAC CULLOCH, W. PITTS
A logical calculus of the ideas immanent in nervous activity
Bull. Math. Biophys., 5, pp113-115, 1943.
- [PEL1] D.PELLEGRIN
Dynamics of random boolean networks
Disordered Systems and Biological Organization, Nato ASI Series in
Computer and Systems Sciences, pp 67-71, Springer Verlag 1986.
- [PEL2] D. PELLEGRIN
On learning abilities of random boolean networks
Actes du Colloque COGNITIVA : De l'Intelligence Artificielle aux
Biosciences, Paris, juin 1985.
- [PER] P.PERETTO
Collective properties of neural networks, a statistical approach
Biological Cybernetic 50, pp51-62 , 1984.
- [FRO1] F.ROBERT
Discrete iterations
Springer Verlag 1986.
- [FRO2] F.ROBERT
rappports de recherche IMAG
n° 313 oct. 82, n°536 juin 85, n°556 sept.85
- [YRO] Y.ROBERT
Algorithmique parallèle : Réseaux d'Automates, Architectures Systolique,
machines SIMD & MIMD
Thèse d'Etat Grenoble 1986.
- [SEI] C.L.SEITZ
The Cosmic Cube
Communication ACM, vol. 28, n°1, pp22-33, 1985.
- [SEJ] T.J.SEJNOWSKI, G.E.HINTON
Separating Figures from ground with a Boltzman Machine
Vision ,Brain and Cooperative Computation, S.A.ARBIB, A.R.HANSON
Eds.,MIT Press 1985.

- [SN] E.II.SNOUSSI
Structure et comportement itératif de certains modèles discrets
Thèse de Docteur Ing., Grenoble 1980.
- [TA] G.TARGONSKI
Topics in Iteration theorie
Vandenhoeck and Ruprecht Eds, Zurich : Göttingen, 1981
- [TC] M.TCHUENTE
Contribution à l'étude des méthodes de calcul pour des systèmes de types coopératifs
Thèse d'Etat Grenoble 1982.
- [NE1] J.VON NEUMANN
Theory of Self-Reproducing Automata
Ed. A.W. BURKS, Univ. of Illinois Press, Urbana 1966
- [NE2] J.VON NEUMANN
The Computer and the Brain
Yale Univ. Press 1970.
- [WA] D.L.WALTZ, J.B.POLLACK
Massively Parallel Parsing a Strongly Interactive Model of Natural Language Interpretation
Working Paper n°48, Univ. Illinois, 1984.
- [WE] G.WEISBUCH
Un modèle de l'évolution des espèces à trois niveaux, basé sur les propriétés globales des réseaux booléens
C.R. Acad. Sc. Paris, 298, pp 375-378, 1984.
- [WI] G.Y.WICHINIAC
Cellular automata models of disorder and organization.
Disordered Systems and Biological Organization, Nato ASI Series in Computer and Systems Sciences, pp 3-21, Springer Verlag 1986.

Chapitre 1

TEST ALGORITHMIQUE DE L'ATTRACTION D'UN POINT FIXE D'UNE ITERATION DISCRETE

INTRODUCTION

Dans ce rapport, nous établissons un parallèle entre les démarches mathématique et algorithmique dans l'étude de l'attraction d'un point fixe d'une itération discrète dans son voisinage second.

L'approche mathématique élaborée par F. Robert (1982) exprime des conditions nécessaires et suffisantes d'attraction en utilisant la notion de dérivée discrète.

Notre algorithme de vérification de cette attraction est basé sur la définition de trois ensembles dynamiques et présente deux principaux avantages sur la méthode précédente :

- lorsque le point fixe est effectivement attractif, la vérification de l'attraction "coûte moins cher"; le coût est défini à partir de deux opérations élémentaires
- la généralisation possible de son emploi : il peut vérifier l'attraction d'un cycle, et ce dans un voisinage quelconque de ce cycle.

Nous traiterons successivement au cours de l'étude :

- 1) Principales définitions et propriétés
- 2) Conditions nécessaires et suffisantes épurées
- 3) Algorithme
- 4) Comparaisons, conclusions
- 5) Simulations.
- 6) Un peu de combinatoire.

1 PRINCIPALES DEFINITIONS ET PROPRIETES

Soit une fonction $F : \{0,1\}^n \rightarrow \{0,1\}^n$ et x, y deux points de $\{0,1\}^n$.

On supposera que F admet un point fixe $\xi = F(\xi)$.

On utilisera deux distances sur $\{0,1\}^n$:

la distance de Hamming $d_h : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{N}$

$$d_h(x, y) = \sum_{i=1}^{i=n} |x_i - y_i|$$

et la distance booléenne vectorielle $d : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$

$$(d(x, y))_i = |x_i - y_i|$$

Couronne première de x :

C'est l'ensemble des points de $\{0,1\}^n$ situé à une distance $d_h=1$ de x :

$$C1(x) = \{x^i \in \{0,1\}^n / x^i = x + e_i \text{ pour } i = 1, \dots, n\}$$

où $+$ représente l'addition dans $(\mathbb{Z}/2\mathbb{Z})^n$ et e_i est le $i^{\text{ème}}$ vecteur de la base canonique de $\{0,1\}^n$.

La première couronne de x possède n éléments.

Couronne seconde de x :

C'est l'ensemble des points de $\{0,1\}^n$ situé à une distance $d_h=2$ de x :

$$C2(x) = \{x^{ij} \in \{0,1\}^n / x^{ij} = x + e_i + e_j \text{ pour } i, j = 1, \dots, n \text{ avec } i \neq j\}$$

La seconde couronne de x possède $n(n-1)/2$ éléments.

Voisinages premier et second de x :

$$V1(x) = C1(x) \cup \{x\}$$

$$V2(x) = V1(x) \cup C2(x)$$

Attraction de ξ dans son voisinage premier :

On dira que ξ est attractif dans $V1(\xi)$ si :

- i) $F(V1(\xi)) \subset V1(\xi)$
- ii) $\forall x^0 \in V1(x), F^n(x^0) = \xi$

F. Robert (1982) a énoncé les conditions nécessaires et suffisantes pour que ξ soit attractif dans son voisinage premier :

- a) $F'(\xi)$ a au plus un 1 par colonne.
- b) Λ une permutation près des lignes et des colonnes $F'(\xi)$ est triangulaire stricte.

Attraction de ξ dans son voisinage second :

On dira que ξ est attractif dans $V2(\xi)$ si :

- i) $V2(\xi) \supset F(V2(\xi))$
- ii) $\forall x^0 \in V2(\xi), F^{n(n+1)/2}(x^0) = \xi$

Pour le voisinage premier comme pour le second la première condition exprime la stabilité de F dans le voisinage du point fixe, la seconde la convergence dans ce voisinage.

Dérivée discrète $F'(\xi)$ de F en ξ : cf Robert(1982), (1986)

$F'(\xi)$ est une matrice booléenne (n,n) définie par $|F'(\xi)|^j = d(F(\xi), F(\xi^j))$, où $|F'(\xi)|^j$ est la $j^{\text{ème}}$ colonne de la matrice dérivée.

Si on décompose $F=(f_1, f_2, \dots, f_n)$ où $f_i : \{0,1\}^n \mapsto \{0,1\}$, l'élément (i,j) de la matrice dérivée indique alors (par un 1) si une modification de la $j^{\text{ème}}$ composante de ξ entraîne une modification de $f_i(\xi)$.

Comme dans le cadre continu, la notion de dérivée est employée dans les études de convergence : cf Fogelman(1984a,b) et Robert(1985). C'est sur cette dérivée que F.Robert a basé les conditions nécessaires et suffisantes suivantes d'attraction d'un point fixe dans son voisinage second.

point fixe dans son voisinage second.

Pour que ξ soit attractif dans $V_2(\xi)$ il faut et il suffit que :

- a) $F'(\xi)$ ait au plus deux 1 par colonne.
- b) $\forall i \in \{1, 2, \dots, n\}$ $F'(\xi^i)$ ait sa $i^{\text{ième}}$ colonne qui diffère au plus de deux éléments de chacune de ses autres colonnes.
- c) Pour $H = F, F^2, \dots, F^{n(n+1)/2}$
 - i) $H'(\xi)$ n'ait aucune colonne commune avec la matrice identité
 - ii) $\forall i, j \in \{1, 2, \dots, n\}$ et $(i \neq j)$
 $H'(\xi^i)$ ait sa $(i^{\text{ième}} + j^{\text{jème}})$ colonne $\neq (e_i + e_j)$

remarques :

a- a) et b) sont des conditions nécessaires et suffisantes à la stabilité de F dans $V_2(\xi)$

b- a) et b) étant vérifiées, c) équivaut à : $\forall x^0 \in V_2(\xi)$ on a $F^{n(n+1)/2}(x^0) = \xi$

Notant que ces C.N.S. sont lourdes et parfois redondantes nous allons les "épurer" en utilisant les simplifications suggérées dans Robert(1982).

2 CONDITIONS NECESSAIRES ET SUFFISANTES EPUREES

Il paraît difficile, voire impossible, de "minimiser" le coût des deux conditions a) et b); mais c) peut facilement être améliorée de ce point de vue.

Première simplification.

En utilisant l'équivalence :

"ni F , ni F^2 ,, ni $F^{n(n+1)/2}$ n'ont de point fixe dans $C1(\xi) \cup C2(\xi)$ "

\Leftrightarrow

"ni F^k ,, ni $F^{n(n+1)/2}$ n'ont de point fixe dans $C1(\xi) \cup C2(\xi)$ où k est le plus petit entier supérieur à $n(n+1)/4$ "

Nous pouvons justifier l'implication la moins évidente : si une, au moins, des k premières composées de F avait un point fixe dans $C1(\xi) \cup C2(\xi)$, au moins une des k suivantes serait alors dans le même cas en tant que composée.

Nous pouvons alors réécrire la condition c) en ne faisant varier H que de F^k à $F^{n(n+1)/2}$.

Seconde simplification.

Pour c) il nous faut calculer $H'(\xi)$ donc connaître la valeur de H sur $V2(\xi)$; de plus, vérifiant les conditions a) b) et c) dans l'ordre, lorsque l'on est conduit à vérifier c) on sait déjà que F est stable dans $V2(\xi)$, donc seule la restriction de H à $V2(\xi)$ nous intéresse.

Cette remarque permet d'éviter l'évaluation de F sur $\{0,1\}^n/V2(\xi)$ c'est à dire $2^n - n(n+1)/2$ évaluations de la fonction.

Troisième simplification.

D'après l'égalité $d(F(\xi^{ij}), \xi) = F'(\xi^i) \times (e_i + e_j) = F'(\xi^j) \times (e_i + e_j)^*$, nous savons

également que $H'(\xi^i)$ et $H'(\xi^j)$ ont même $(i^{\text{ième}} + j^{\text{jème}})$ colonne; dès lors, on peut restreindre l'ensemble des indices dans c) ii).

Alors au lieu de $\forall i, j \in \{1, 2, \dots, n\}$ on a $\forall i \in \{1, 2, \dots, n\}$ et $\forall j \in \{i+1, \dots, n\}$.

On économise ainsi $n(n-1)/2$ additions et comparaisons de vecteurs de $\{0, 1\}^n$.

* Cette égalité établie par Robert(1982) est utilisée plusieurs fois dans la démonstration des C.N.S. C'est une généralisation au voisinage second $V_2(\xi)$ de la relation $d(F(\xi^j), \xi) = F'(\xi) e_j$ valable dans le voisinage premier $V_1(\xi)$.

On obtient avec ces trois simplifications un nouvel énoncé des conditions nécessaires et suffisantes.

Pour que ξ soit attractif dans $V_2(\xi)$ il faut et il suffit que :

a) $F'(\xi)$ ait au plus deux 1 par colonne.

b) $\forall i \in \{1, 2, \dots, n\}$ $F'(\xi^i)$ ait sa $i^{\text{ième}}$ colonne qui diffère au plus de deux éléments de chacune de ses autres colonnes.

c) Pour $H = F^k /_{V_2}, \dots, F^{n(n+1)/2} /_{V_2}$ où $k = \lceil n(n+1)/4 \rceil$ et $F /_{V_2}$ représente la restriction de F sur V_2 .

i) $H'(\xi)$ n'ait aucune colonne commune avec la matrice identité

ii) $\forall i \in \{1, 2, \dots, n\}$ et $\forall j \in \{i+1, \dots, n\}$

$H'(\xi^i)$ ait sa $(i^{\text{ième}} + j^{\text{jème}})$ colonne $\neq (e_i + e_j)$.

Avant de définir et de calculer le coût des ces C.N.S. nous donnerons un exemple de fonction $F : \{0, 1\}^3 \rightarrow \{0, 1\}^3$ ayant 000 comme point fixe attractif dans son voisinage second.

Exemple :

x	F(x)	F ² (x)	F ³ (x)	F ⁴ (x)	F ⁵ (x)
a 000	000	000	000	000	000
b 001	011	000	000	000	000
c 010	011	011	000	000	000
d 011	000	000	000	000	000
e 100	010	110	011	000	000
f 101	100	010	110	011	000
g 110	011	000	000	000	000
h 111	NOD	NOD	NOD	NOD	NOD

Remarque :

La fonction en dehors de $V_2(\xi)$ ne nous intéressant pas, il est inutile de la définir pour un exemple (NOD). La vérification de c) se fait pour $H=F^4, F^5$ et F^6 donc pour des applications très simples : F^5/V_2 et F^6/V_2 sont ici identiquement nulles. Dans le cas de points fixes attractifs c'est toujours le cas : $F^{n(n+1)/2}/V_2$ et quelques uns de ses prédecesseurs sont presque constants.

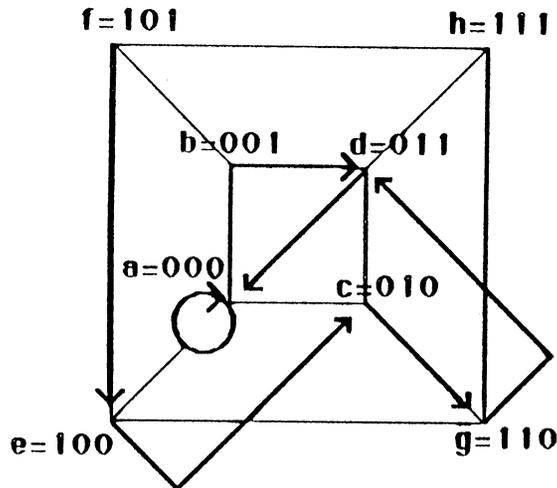


figure 1

Figure représentant le graphe d'itération de la fonction F , dessiné sur le cube $\{0,1\}^3$. C'est la configuration $a = 000$ qui est point fixe et dans le cas présent elle est attractive dans son voisinage second c'est à dire pour (b,c,d,e,f,g) .

Nous avons choisi de définir un coût à partir des deux opérations élémentaires suivantes :

l'évaluation de F en un point de $\{0,1\}^n$ notée éval

la comparaison entre deux vecteurs de $\{0,1\}^n$ notée comp

Evaluation du coût des conditions nécessaires et suffisantes.

1^{ère} Evaluation

a) 1 calcul de $F'(\xi)$ + n comp

b) n calculs de $F'(\xi)$ + n(n-1) comp

c) $n^2(n+1)^2/4$ éval

i) $n(n+1)/4$ calculs de $H'(\xi)$ + $n(n+1)/4$ comp

ii) $n(n+1)(n-1)$ calculs de $H'(\xi^i)$ + $n^2(n+1)(n-1)^2/4$ comp

Le calcul d'une matrice dérivée nécessite n éval et n comp; on peut donc écrire une

2^{ième} Evaluation

a) n éval + 2n comp

b) $n(n$ éval + n comp) + $n(n-1)$ comp

c) $(n(n+1))^2/4$ éval.

i) $[n(n+1) (n$ éval + n comp) + $n(n-1)$ comp] / 4

ii) $[n(n^2-1) (n$ éval + n comp)] / 4 + $[n^2(n^2-1)(n-1)$ comp] / 8

Il est désormais possible d'établir le coût final que nécessite la vérification totale des conditions nécessaires et suffisantes épurées, nous reprendrons plus loin cet aspect de "vérification totale" (associée au coût maximal).

Coût total :

$$1/2 n^4 + 3/4 n^3 + 2 n^2 + n$$

évaluations

$$1/4 n^5 + 5/2 n^2 + 5/4 n$$

comparaisons

Nous disposons, à ce stade d'un outil pour comparer la démarche mathématique et la démarche algorithmique qu'il s'agit maintenant de développer.

3 ALGORITHME

L'idée générale pour un algorithme chargé de vérifier l'attraction de ξ dans son voisinage second est la suivante :

en partant de tout point de $V_2(\xi)$ on itère au plus $n(n+1)/2$ fois la fonction F en vérifiant à chaque pas si k est atteint et si l'itéré n'est pas en dehors de $V_2(\xi)$.

Mais cette idée conduit à des opérations inutiles (parce que répétitives), comme l'atteste la figure suivante :

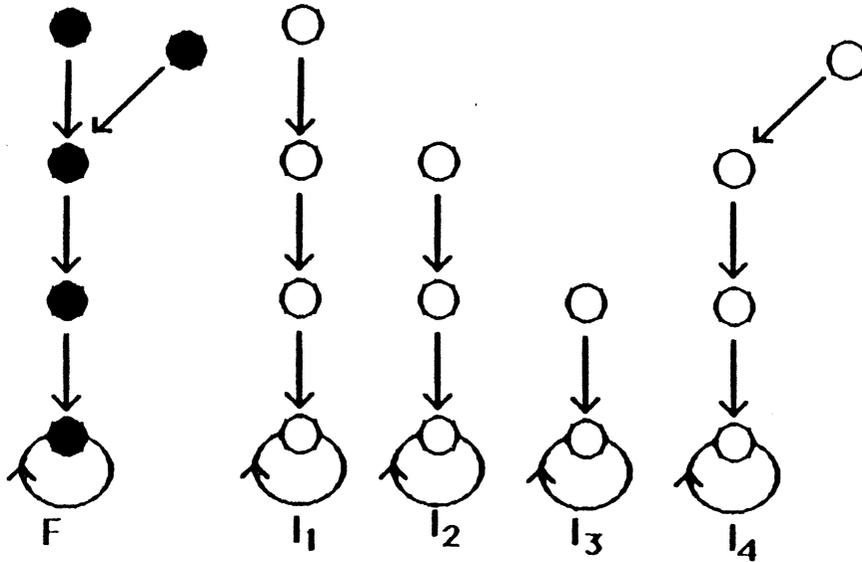


figure 2

En noir on dessine le graphe d'itération d'une fonction F dont on repère 4 points convergeant vers un point fixe.

En blanc on dessine les 4 tentatives de vérifications nécessaires pour répondre à la question de l'attraction : on constate que bien des points sont "visités" plusieurs fois.

Pour éviter cet inconvénient, l'algorithme proposé utilisera la définition de trois ensembles dynamiques.

DONNEES

$$F : \{0,1\}^n \rightarrow \{0,1\}^n \quad n > 1$$

$$\xi \in \{0,1\}^n \text{ tel que } f(\xi) = \xi$$

$V_2(\xi)$ le voisinage second de ξ (voir définition plus haut)

PROBLEME

ξ est-il un point fixe attractif de F dans $V_2(\xi)$?

rappel de la définition :

ξ est attractif dans $V_2(\xi)$ si et seulement si :

$$\forall x^0 \in V_2(\xi) \quad \begin{array}{ll} \text{i) } F(x^0) \in V_2(\xi) & \text{stabilité} \\ \text{ii) } F^{(n+1)/2}(x^0) = \xi & \text{convergence} \end{array}$$

VARIABLES

- ED** ensemble de points de $V_2(\xi)$, à partir desquels on n'a pas encore vérifié l'attraction de ξ . (ensemble de départ)
- EA** ensemble de points $V_2(\xi)$, à partir desquels on a déjà vérifié l'attraction de ξ . (ensemble d'arrêt)
- EC** ensemble de points de $V_2(\xi)$, sur lesquels le travail est en cours. (ensemble critique)
- fin_branche** booléen indiquant la fin de l'inspection d'une branche du bassin d'attraction.
- non_attraction** booléen indiquant la détection de la non attraction.
- x** élément de $\{0,1\}^n$.

ξ est-il un point fixe attractif de F dans $V2(\xi)$?

ALGORITHME (données : ξ et F sur $V2(\xi)$; résultat : réponse)

DEBUT_ACTION_PRINCIPALE

ED := $V2(\xi)/\xi$

EA := $\{\xi\}$

EC := \emptyset

non_attraction := faux

fin_branche := faux

{ (EA + ED = $V2(\xi)$) et (tous les éléments de EA convergent vers ξ sans sortir de $V2(\xi)$) }

REPETER

{ ED $\neq \emptyset$ }

{ (EA + ED = $V2(\xi)$) et (tous les éléments de EA convergent vers ξ sans sortir de $V2(\xi)$) }

Construction_d'un_ensemble_critique(fin_branche,non_attraction,EC)

{ Tous les éléments de l'ensemble EC sont des itérés successifs d'un même point qui appartient à ED }

SI fin_branche ALORS

{ Tous les éléments de l'ensemble EC sont des itérés successifs d'un même point qui appartient à ED et non_attraction est faux donc ils ne cyclent pas et ne sortent pas de $V2(\xi)$ }

{ (EA + ED = $V2(\xi)$) et (tous les éléments de (EA U EC) convergent vers ξ sans sortir de $V2(\xi)$) }

DEBUT

EA := EA+EC

ED := ED/EC

EC := \emptyset

FIN

FIN_SI

{ (EA' + ED' = $V2(\xi)$) et (tous les éléments de EA' convergent vers ξ sans sortir de $V2(\xi)$) }

JUSQU'A (ED = \emptyset) OU (non_attraction)

FIN_REPETER

{ [(EA' + ED' = V2(ξ)) et (tous les éléments de EA' convergent vers ξ sans sortir de V2(ξ))] et [ED = \emptyset]
 $\Rightarrow \xi$ est attractif dans V2(ξ) }

SI (ED= \emptyset) ALORS réponse := oui

{ [(EA' + ED' = V2(ξ)) et (tous les éléments de EA' convergent vers ξ sans sortir de V2(ξ))] et [ED \neq
 \emptyset] $\Rightarrow \xi$ n'est pas attractif dans V2(ξ) }

SINON réponse := non

FIN_SI

FIN_ACTION_PRINCIPALE.

ACTION Construction d'un ensemble critique (donnée : non_attraction,
 résultat: fin_branche, non_attraction : booléens, EC : ensemble)

DEBUT

fin_branche := faux

CHOIX D'UN ELEMENT x DE ED

EC := { x }

{ Tous les éléments de l'ensemble EC sont des itérés successifs d'un même point qui appartient à ED }

TANT QUE [NON(fin_branche) OU NON(non_attraction)] FAIRE
 CAS F(x)

- \in (EC) OU \notin V2(ξ) : non_attraction := vrai

{justification immédiate par la définition}

- \in (EA) : fin_branche := vrai

-AUTREMENT : DEBUT

x := F(x)

{ [Tous les éléments de l'ensemble EC sont des itérés successifs d'un même point qui appartient à ED]
 et x est l'itéré du dernier point placé dans EC}

EC := EC + {x}

{ Tous les éléments de l'ensemble critique EC' (=EC + {x}) sont des itérés successifs d'un même point qui appartient à ED }

FIN_AUTREMENT

FIN_CAS

{ Tous les éléments de l'ensemble EC 'sont des itérés successifs d'un même point qui appartient à ED }

{ le nombre de passages dans cette boucle est borné par le cardinal de V2(ξ) }

FIN_TANTQUE

FIN_ACTION.

Exemple de déroulement de l'algorithme :

On choisit l'exemple de la page 6 : $\xi=a$ et $V2(\xi) = \{a,b,c,d,e,f,g\}$

Initialisation : $ED = \{b,c,d,e,f,g\}$
 $EA = \{a\}$
 $EC = \emptyset$

Choix de **b** comme 1^{er} point de départ :

$EC = \{b,d\}$ arrêt car $F(d)=a \in EA$
 Redéfinition des ensembles
 $ED = \{c,e,f,g\}$
 $EA = \{a,b,d\}$
 $EC = \emptyset$

Choix de **c** comme 1^{er} point de départ :

$EC = \{c,g\}$ arrêt car $F(g)=d \in EA$
 Redéfinition des ensembles
 $ED = \{e,f\}$
 $EA = \{a,b,c,d,g\}$
 $EC = \emptyset$

Choix de **f** comme 1^{er} point de départ :

$EC = \{f,e\}$ arrêt car $F(e)=g \in EA$
 Redéfinition des ensembles
 $ED = \emptyset$
 $EA = \{a,b,c,d,e,f,g\}$
 $EC = \emptyset$

Arrêt car $ED = \emptyset$, on peut conclure à l'attraction de ξ dans son voisinage second.

Se référant à l'exploration du graphe par l'algorithme (cf. *figure 1*), on se convainc de l'importance du choix des points de départ. Il y a diverses stratégies possibles (on les évoquera plus loin) pour ce choix, mais trouver une bonne stratégie nécessiterait de connaître une information préalable que nous ne calculerons pas.

Au niveau de la manipulation des ensembles, il doit être possible d'apporter certaines améliorations.(cf. page 17)

Généralisations de l'utilisation de l'algorithme.

i) Cet algorithme permet de vérifier l'attraction d'un point fixe dans n'importe lequel de ses voisinages discrets. En fait ne se préoccupant pas de la topologie, il est capable de vérifier l'attraction de point fixe dans n'importe quel ensemble de points : il suffit en effet d'initialiser l'ensemble de départ ED avec le voisinage souhaité, ce qui constitue un avantage certain sur la vérification à l'aide des conditions nécessaires et suffisantes qui ne concernent que le voisinage second.

ii) Il permet également de vérifier l'attraction d'un cycle dans un quelconque de ses voisinages discrets. Pour cela il faut initialiser EA avec les éléments du cycle.

iii) En conservant l'idée principale de l'algorithme on peut même déterminer le voisinage d'attraction du point fixe (ou du cycle) considéré. On initialise alors l'ensemble de départ avec l'ensemble des points sur lequel on cherche à déterminer le voisinage d'attraction, et on construit au fur et à mesure ce voisinage.

Evaluation du coût de l'algorithme :

Nous raisonnerons en terme de coût maximal, i.e. le coût du "pire des cas", et nous utiliserons encore les deux opérations de base : éval et comp. Le choix des points de départ est primordial dans le temps d'exécution de l'algorithme mais nous ne pouvons en tenir compte dans l'évaluation de ce type de coût.

Supposons que l'on ait eu besoin de p départs ($p \in \{1, 2, \dots, n(n+1)/2\}$), pour vérifier l'attraction de k , et qu'au $i^{\text{ème}}$ départ nous ayons obtenu une chaîne de k_i itérés.

$$i=p$$

On note que p et k_i sont liés par : $\sum_{i=1} k_i = n(n+1)/2$

Coût du $i^{\text{ème}}$ départ :

i) Nombre d'évaluations de F : k_i éval.

ii) Nombre de comparaisons de deux vecteurs :

Contrôle de l'appartenance de chaque itéré à

$V2(\xi) : k_i \cdot n(n+1)/2$ comp

EA : $[EA]_{i-1}$ c'est à dire $\sum_{j=1}^{j=i-1} k_j$ comp

EC : $k_i \cdot n(n+1)/2$ comp

où $[EA]_i$ est le cardinal de EA à l'étape i.

Parmi les redéfinitions d'ensembles seule la différence d'ensemble ED/EC est coûteuse. La différence de deux ensembles de cardinaux N et P nécessite $N + (N-1) + \dots + (N-P+1)$ comp.

Donc ED/EC coûte $(([ED]_{i-1} - k_i + 1) k_i + k_i(k_i - 1))$ comp.

Le coût du ième départ est finalement de :

$$k_i \quad \text{éval}$$

$$k_i n(n+1)/2 + k_i [EA]_{i-1} + k_i(k_i - 1)/2 + ([ED]_{i-1} - k_i + 1)k_i + k_i(k_i - 1)/2 \text{ comp}$$

En notant $M = n(n+1)/2$ on majore le coût total de l'algorithme par :

M éval

$$M^2 + M(M-1)/2 + M(M-1)/2 + M(M+1)/2 = 5/2 M^2 - 1/2 M \text{ comp}$$

Ce qui s'exprime en fonction de n par :

Coût total de l'algorithme

$n(n+1)/2$ éval

$$5/8 n^4 + 5/4 n^3 + 3/2 n^2 - 1/4 n \text{ comp}$$

4 COMPARAISONS, CONCLUSIONS

Coût total de l'algorithme	
$1/2 n^2 + 1/2 n$	évaluations
$5/8 n^4 + 5/4 n^3 + 3/2 n^2 - 1/4 n$	comparaisons

qu'il est intéressant de comparer avec le

Coût total des C.N.S. épurées	
$1/2 n^4 + 3/4 n^3 + 2 n^2 + n$	évaluations
$1/4 n^5 + 5/2 n^2 + 5/4 n$	comparaisons

Nous pouvons conclure que cet algorithme est optimal pour le nombre d'évaluations de F ; il est en effet impossible d'affirmer que ξ est attractif dans son voisinage second sans considérer au moins $F(x)$ pour tout x dans ce voisinage.

Il est peut-être possible de réduire le nombre de comparaisons en introduisant un ordre sur les éléments des trois ensembles ED, EA, et EC. Cette transformation conduirait à d'autres opérations coûteuses et moins facilement comparables à celles utilisées dans les C.N.S. .

Le coût que nous avons choisi d'étudier est le coût du "pire des cas", c'est évidemment le plus aisé et parfois même le seul que l'on sache évaluer. Mais il convient de l'employer avec la réserve habituelle qui l'oppose au coût moyen : l'algorithme sera-t-il plus efficace sur un jeu de données important ? La réponse est positive chaque fois que l'attraction est effectivement vérifiée (dans ce cas les deux méthodes sont employées dans leur totalité).

Si le point fixe n'est pas attractif la comparaison des deux méthodes exige une

analyse des causes de non-attraction : donnons deux exemples de cas extrêmes :

ex1) Les C.N.S. sont plus efficaces

Le dernier point testé par l'algorithme sort du voisinage.

algo.) Toutes les vérifications précédentes ont été inutiles.

C.N.S.) L'étape a) a conclu immédiatement à la sortie du voisinage.

ex2) L'algorithme est plus efficace

La fonction F cycle dans le voisinage concerné : on suppose que le cycle est de l'ordre du cardinal du voisinage.

algo.) Le point de départ est dans le cycle, donc le premier test de l'algorithme remplit EC et détecte un cycle.

C.N.S.) C'est seulement à la fin de l'étape c) ii) (pour $H = F^{n(n+1)/2}$) que les C.N.S. signalent la présence d'un cycle.

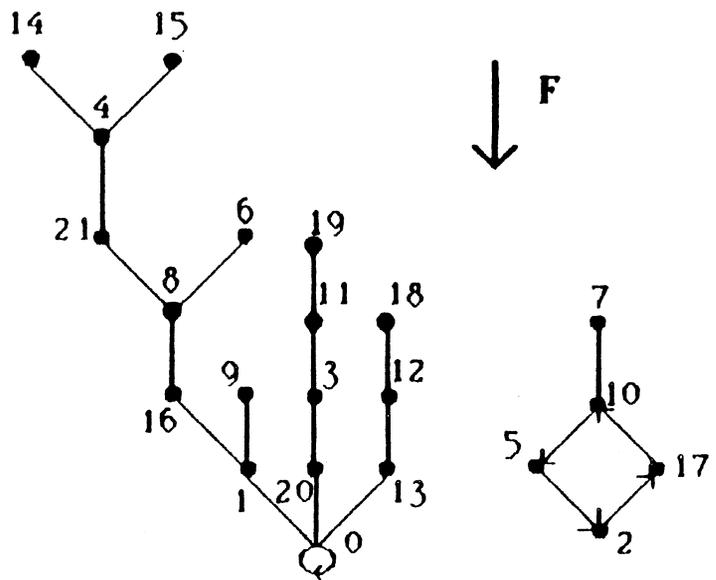
On remarque une fois de plus l'importance du choix des points de départ (il serait intéressant d'établir une heuristique de choix), ainsi que l'importance de la forme du graphe d'itération pour le coût effectif de ces vérifications

5 SIMULATIONS

On trouve dans les pages suivantes deux exemples de simulation de notre algorithme de test de l'attraction d'un point fixe.

Exemple de fonction $F : [0,1]^6 \mapsto [0,1]^6$, qui admet (000000) comme point fixe non attractif dans son voisinage second. Dans ce voisinage la fonction admet également un cycle d'ordre 4.

	x	F(x)
1	000001	000000
2	000010	001010
3	000100	000110
4	001000	000011
5	010000	000010
6	100000	100001
7	100001	101000
8	100010	001001
9	100100	000001
10	101000	010000
11	110000	000100
12	010001	010010
13	010010	000000
14	010100	001000
15	011000	001000
16	001001	000001
17	001010	101000
18	001100	010001
19	000101	110000
20	000110	000000
21	000011	100010



DEROULEMENT DE L'ALGORITHME

21
 Choix d'un point de depart
 000001
 Redefinition des ensembles

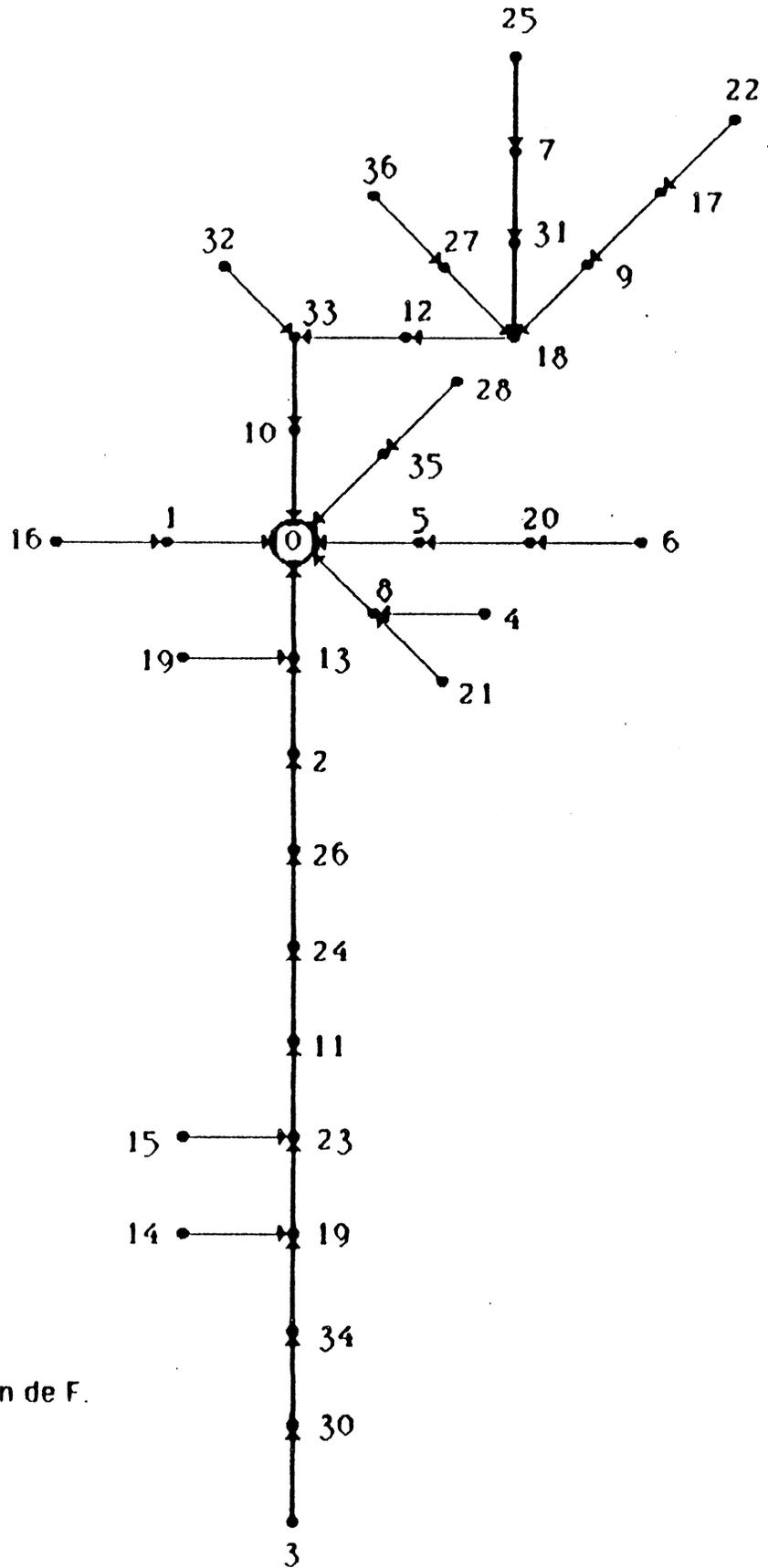
 Choix d'un point de depart
 000010
 CYCLE DETECTE

 000010
 001010
 010000
 101000

- Le nombre de points a tester est :
- Il y a 1 points testes
- Il y a 5 points testes

Exemple de fonction $F : (0,1)^8 \mapsto (0,1)^8$, qui admet (000000) comme point fixe attractif dans son voisinage second. Le graphe d'itération de F dans le voisinage second du point fixe est à la page suivante. Puis dans les deux autres pages on trouve le déroulement de l'algorithme.

	x	F(x)
1	00000001	00000000
2	00000010	10010000
3	00000100	00011000
4	00001000	10000000
5	00010000	00000000
6	00100000	01010000
7	01000000	00001001
8	10000000	00000000
9	10000001	01000100
10	10000010	00000000
11	10000100	00100100
12	10001000	00001100
13	10010000	00000000
14	10100000	01001000
15	11000000	00100010
16	01000001	00000001
17	01000010	10000001
18	01000100	10001000
19	01001000	10010000
20	01010000	00010000
21	01100000	10000000
22	00100001	01000010
23	00100010	10000100
24	00100100	00110000
25	00101000	01000000
26	00110000	00000010
27	00010001	01000100
28	00010010	00000110
29	00010100	00100010
30	00011000	00000101
31	00001001	01000100
32	00001010	00001100
33	00001100	10000010
34	00000101	00010100
35	00000110	00000000
36	00000011	00010001



Graphe d'itération de F.

DEROULEMENT DE L'ALGORITHME

- Le nombre de points a tester est : 36

Choix d'un point de depart
00000001
Redefinition des ensembles
- Il y a 1 points testes

Choix d'un point de depart
00000010
Redefinition des ensembles
- Il y a 3 points testes

Choix d'un point de depart
00000011
Redefinition des ensembles
- Il y a 9 points testes

Choix d'un point de depart
00000100
Redefinition des ensembles
- Il y a 17 points testes

Choix d'un point de depart
00000110
Redefinition des ensembles
- Il y a 18 points testes

Choix d'un point de depart
00001000
Redefinition des ensembles
- Il y a 20 points testes

Choix d'un point de depart
00001001
Redefinition des ensembles
- Il y a 21 points testes

Choix d'un point de depart
00001010
Redefinition des ensembles
- Il y a 22 points testes

Choix d'un point de depart
00010000
Redefinition des ensembles
- Il y a 23 points testes

Choix d'un point de depart
00010010
Redefinition des ensembles
- Il y a 24 points testes

Choix d'un point de depart
00100000
Redefinition des ensembles
- Il y a 26 points testes

Choix d'un point de depart
00100001
Redefinition des ensembles
- Il y a 29 points testes



6 UN PEU DE COMBINATOIRE.

A la suite de cette étude une question se pose :

"Combien de graphes différents l'algorithme est-il amené à traiter totalement, i.e. lorsqu'il répond que ξ est attractif dans un voisinage donné comportant n points ?"

Formulons cette question indépendamment du problème précédent :

"Combien existe-t-il de types différents de graphes d'itérations possédant n sommets n'ayant qu'un bassin et un seul point fixe ?"

Le problème sera, ici aussi, abordé algorithmiquement; l'aspect analyse combinatoire (cf. Cayley(97)) ne sera qu'évoqué au passage. Dans Klein(85) nous avons employé les relations de récurrences et un logiciel multiprécision pour répondre à cette question pour des indices nettement supérieurs à ceux des tables existantes, cf Riordan(58).

Notation : Le nombre cherché est T_n où n représente le nombre de sommets du graphe hormis le point fixe. (par exemple $n = 2^{m-1}$ quand on s'intéresse à des fonctions de $\{0,1\}^m \mapsto \{0,1\}^m$ ou $m+m(m-1)/2$ si on considère seulement le voisinage second)

Plan

- a- Décomposition du problème principal.
- b- Partage d'un entier.
- c- Tirages de k éléments, avec remise, dans un ensemble de cardinal q .
- d- Evaluation effective de T_n pour $n < 27$.
- e- Présentation des résultats.

a) Décomposition du problème principal

L'idée directrice est d'exprimer T_n en fonction d'un ou de plusieurs T_i , $i < n$.
 Décomposons de manière simple les différents graphes répondant à la spécification, suivant leur manière d'atteindre le point fixe. Les sommets étant indifférenciés, ce sont en fait les différents types de graphes qui nous intéressent.

On distingue n cas dans les "arrivées" sur le point fixe. La *figure 1* représente le cas général

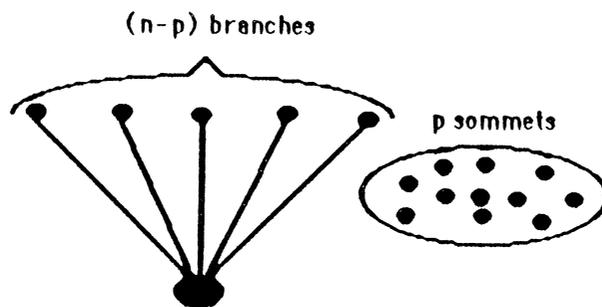


figure 1 : Cas général, noté $C(n-p)$. $(n-p)$ points sont directement reliés au point fixe, et les p restant se répartissent dessus.

Il est clair que le cas $C(1)$ ne définit qu'un graphe et que le cas $C(n)$ engendre T_n graphes.

Combien de graphes constituent le cas général ?

Nous noterons $S(n-p, p)$ le nombre de graphes issus de la répartition de p sommets sur $(n-p)$ branches.

Traisons un exemple pour $n = 10$ et $p = 7$

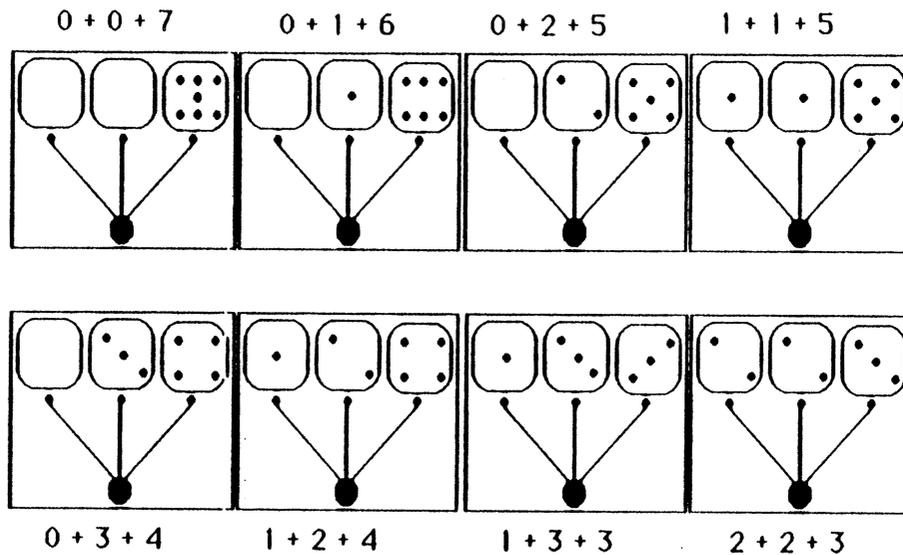


figure 2 : Nous utilisons ce genre de schéma pour montrer l'intérêt de la répartition des 7 sommets sur les 3 branches, de préférence à l'évolution du graphe "au delà du 1^{er} étage".

Ce problème est en fait celui du partage de l'entier p en au plus $(n-p)$ termes. Dans l'exemple ci dessus, les 8 répartitions peuvent se symboliser comme suit :

$$\begin{aligned}
 7 &= 7 \\
 7 &= 6+1 \\
 7 &= 5+2 = 5+1+1 \\
 7 &= 4+3 = 4+2+1 \\
 7 &= 3+3+1 = 3+2+2.
 \end{aligned}$$

Il nous suffit maintenant de savoir combien de graphes sont engendrés par une répartition.

Par exemple $n=7$ et $p=5$, une des répartitions possibles est $5=3+2$ et il est banal de constater que cette répartition donne lieu à $T_2 \cdot T_3$ graphes différents.

Mais malheureusement, l'exemple suivant : $n=6$ et $p=4$ avec la répartition $4=2+2$ montre qu'il ne suffit pas toujours d'effectuer le produit des T_i , en effet ici on trouverait $T_2 \cdot T_2$, c'est à dire 4 or il n'y en a que trois :

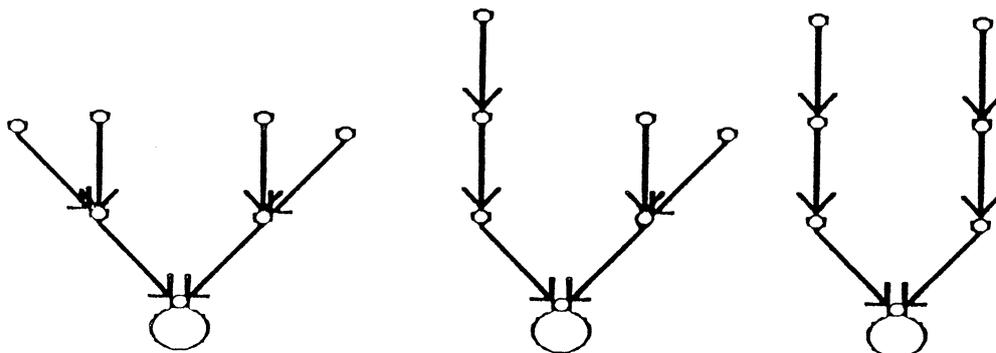


figure 3 : Les trois types de graphes issus de la répartition de 4 branches sur 2 sommets. C'est le graphe du centre qui était compté deux fois dans la multiplication $T_2.T_2$ à cause de sa dissymétrie.

Plus généralement dans le cas de k branches portant chacune l sommets, ce ne sont pas $(T_l)^k$ graphes qui sont générés mais $E(T_l, k)$ où $E(p, q)$ est le nombre de tirages de q éléments, avec remise, dans un ensemble de cardinal p .

Il reste donc à calculer le nombre de partitions d'un entier et $E(p, q)$ pour pouvoir évaluer T_n .

b) Partition d'un entier en somme d'entiers.

Ce problème était déjà abordé par Euler à la fin du 17^{ième} siècle, la bibliographie qui s'y rattache est abondante : Hall, Harary(69), (70), Hardy(70), Riordan(58). Ici c'est la production d'un entier p en somme d'au plus q termes qui nous intéresse. Nous traiterons néanmoins auparavant la question de l'évaluation du nombre $p(n)$ de toutes les partitions de n .

b-1) Evaluation de $p(n)$.

On écrit la décomposition de chaque entier en lignes ayant en commun le plus grand terme de la partition. On note $(n)_i$ la $i^{\text{ième}}$ ligne de la décomposition de n , exemple :

$$(7)_4: \quad 4+3; \quad 4+2+1; \quad 4+1+1+1.$$

Nous avons établi la récurrence suivante :

$(n)_i$ est constituée de toutes les partitions contenues dans les lignes $(n-i)_j$ auxquelles on adjoint l'entier i .

Le calcul effectif de $p(n)$ se fera dans une matrice (n,n) notée PART telle que:

PART[i,j] représente le nombre de décomposition contenues dans la ligne $(i)_j$.

Après avoir initialisé la matrice PART à 0 (sauf la diagonale mise à 1) on calcule PART comme suit :

$$\text{PART}[i,j] = \sum_{k=1}^{k=\min(j,i-j)} \text{PART}[i-j,k]$$

$$\text{d'où l'expression de } p(n) = \sum_{i=1}^n \sum_{k=1}^{\min(n,i-n)} \text{PART}[i-n,k]$$

Dans le tableau 1a on trouve les valeurs de $p(n)$ pour n allant de 1 à 148 ainsi calculé. Dans le tableau 1b on donne les valeurs de $p(n)$ de 1 à 2000 avec un pas de 50 calculé dans Klein(85).

b-2) Production des partitions d'un entier n .

Pour obtenir toutes les partitions d'un entier n en au plus p sommants on reprend l'idée de Deo(76). La production à lieu par ordre lexicographique croissant comme suit :

La 1^{ère} partition est : " $p_1 p_2 \dots p_{k-1} p_k$ " où $p_i = 1$ de $i=1$ à $k-1$ et $p_k = n - (k-1)$

pour le passage d'une partition à la suivante

on recherche le p_i le plus à droite tel que $(p_k - p_i) > 1$

puis on substitue p_j par p_{j+1} pour les j de i à $k-1$ et p_k par $n - \sum_{j=1}^{j=k-1} p_j$

Exemple : partition de 12 en au plus 5 sommants, passage d'une partition à une autre :

$$1 \ 1 \ 3 \ 3 \ 4 \ \text{-----} \> \ 1 \ 2 \ 2 \ 2 \ 5$$

Dans le tableau 2 on fournit la production de toutes les partitions de l'entier 11 en exactement k sommants, k variant de 2 à 11.

c) Dénombrément des tirages avec remise de k éléments parmi q .

On note $E(q,k)$ le nombre cherché. Nous traitons le problème sur un exemple : $E(3,2)$.

Soient a,b,c les trois éléments dont on dispose; énumérons les tirages possibles :

$$\text{groupe } G_a \begin{matrix} [a,a] \\ [a,b] \\ [a,c] \end{matrix} \text{ ---} \> \ 3$$

$$\text{groupe } G_b \begin{matrix} [b,c] \\ [b,c] \end{matrix} \text{ ---} \> \ 2 \qquad E(3,2) = 6$$

$$\text{groupe } G_c [c,c] \text{ ---} \> \ 1$$

Pour la construction de $E(3,3)$, on fait évoluer $E(3,2)$ en mettant

un 'a' en facteur devant $G_a \cup G_b \cup G_c$,

un 'b' devant $G_b \cup G_c$

et un 'c' devant G_c

Ce qui donne pour $E(3,3) = 6 + 3 + 1$.

A partir de ces deux exemples on généralise aisément le calcul de $E(q,k)$: soit (Q_i) une famille de vecteurs à q composantes formés par la relation de récurrence suivante :

$$\begin{aligned} \text{Pour tout } i \text{ de } 1 \text{ à } k \quad & Q_i(1) = 1 \\ \text{pour tout } j \text{ de } 2 \text{ à } q \quad & Q_{i+1}(j) = Q_{i+1}(j-1) + Q_i(j) \end{aligned}$$

cette famille construite on a
$$E(q,k) = \sum_{i=1}^{i=q} Q_k(i).$$

Pour le calcul de $E(q,k)$ il suffit de stocker un vecteur de dimension q , et, si l'on souhaite privilégier la place mémoire vis à vis du calcul nous pouvons substituer au vecteur une fonction : $E(q,k) = C^k_{q+k-1}$. Ceci peut se

déduire de la construction du pseudo triangle de Pascal que l'on vient d'effectuer. On savait déjà que T_n ne pouvait être calculé que pour des valeurs "modestes" de n , nous en avons ici la confirmation.

d) Evaluation de T_n (effective pour $n < 27$)

En reprenant la décomposition du 1^{er} chapitre nous savons maintenant répondre à la question : combien une partition du genre 1,2,3,3,4,5 engendre t-elle de graphes différents ? Il y en a $T_1 \cdot T_2 \cdot E(T_3, 2) \cdot T_4 \cdot T_5$

Le calcul de T_n est alors mené en partant de $T_1 = 1$ puis en employant la formule de récurrence

$$T_n = 1 + \sum_{p=1}^{n-1} S(n-p, p)$$

où la fonction S utilise les partitions p_1, \dots, p_m d'un entier produites par

l'algorithme b2) pour calculer le produit $\prod_{i=1}^{i=n} T_{p_i}$ si tous les p_i sont

distincts, et lorsque il y a k p_i égaux le produit associé est remplacé par $E(T_p, k)$. Il est clair que les T_k intervenant dans $S(n-p, p)$ sont d'indices inférieurs à n .

Nous avons calculé T_n pour n variant de 1 à 26, les valeurs obtenues figurent dans le tableau 3. Les valeurs des tableaux 1 et 3 sont identiques à celles présentées dans Riordan(58) où figure la table des $p(n)$ de 1 à 100, et celle des T_n de 1 à 26. Le tableau 4 donne les valeurs de T_n obtenues dans Klein(85) pour n allant de 2 à 170.

Exemple : Sur le 10.cube ($\{0,1\}^{10}$), le voisinage second d'un point fixe comporte 56 éléments il y a donc T_{55} types de graphes possibles pour l'attraction c'est à dire : 83281891024323882188934 .

e) Présentation des résultats.

A la suite des références figurent les tableaux des nombres annoncés dans ce rapport.

- p. 34 : Nombre de partitions d'un entier n ($n=1 \rightarrow 148$)
- p. 35 : Nombre de partitions d'un entier n ($n=1 \rightarrow 2000$ avec un pas 50)
- p. 36 : Partitions de l'entier 11
- p. 37 : Nombre d'arbres à une racine comportant n sommets ($n=1 \rightarrow 26$)
- p. 38,39 : Idem avec $n=2 \rightarrow 170$

REFERENCES

- A. CAYLEY (1897) :
Collected Papers, Cambridge 13, 26-28
- DEO N., NIEVERGELT J., REINGOLD E.M. (1977) :
Combinatorial Algorithms
Prentice Hall
- F.FOGELMAN-SOULIE (1984) :
Frustration and Stability.
Discrete Applied Mathematics, 9, 1984
- F.FOGELMAN-SOULIE (1984) :
Parallel and Sequential computation on boolean networks.
Soumis à Theoretical Computer Science en 1984
- Marshall HALL :
Combinatorial Theory
- Franck HARARY (69) :
Graph Theory
Addison-Wesley publishing compagny
- Franck HARARY-Edgar PALMER (1970) :
Graphical Enumeration
Academic Press
- G.H. HARDY-E.M.WRIGHT (1968) :
An introduction to the theory of number
Oxford at Clarendon Press
- KLEIN P., MULLER J.M., PELLEGRIN D. (1985) :
Calcul de constantes par un logiciel multiprécision.
Rapport de recherche I.M.A.G. Juillet 1985 n° 544
- RIORDAN J. (1958) :
An Introduction to Combinatorial Analysis
J. Wiley

F.ROBERT (1986) :

Discrete Iterations.

Spinger Verlag.

F.ROBERT (1982) :

Convergence locale d'une itération discrète dans le voisinage
premier ou second d'un point fixe.

Rapport de recherche I.M.A.G. n° 313 oct.1982.

F. ROBERT (1985) :

Convergence locale d'itérations série ou parallèle.

Rapport de recherche I.M.A.G. n° 536 Juin 1985

1 :	1	61 :	1121505	121 :	2056148051
2 :	2	62 :	1300156	122 :	2291320912
3 :	3	63 :	1505499	123 :	2552338241
4 :	5	64 :	1741630	124 :	2841940500
5 :	7	65 :	2012558	125 :	3163127352
6 :	11	66 :	2323520	126 :	3519222692
7 :	15	67 :	2679689	127 :	3913864295
8 :	22	68 :	3087735	128 :	4351078600
9 :	30	69 :	3554345	129 :	4835271870
10 :	42	70 :	4087968	130 :	5371315400
11 :	56	71 :	4697205	131 :	5964539504
12 :	77	72 :	5392783	132 :	6620830889
13 :	101	73 :	6185689	133 :	7346629512
14 :	135	74 :	7089500	134 :	8149040695
15 :	176	75 :	8118264	135 :	9035836076
16 :	231	76 :	9289091	136 :	10015581680
17 :	297	77 :	10619863	137 :	11097645016
18 :	385	78 :	12132164	138 :	12292341831
19 :	490	79 :	13848650	139 :	13610949895
20 :	627	80 :	15796476	140 :	15065878135
21 :	792	81 :	18004327	141 :	16670689208
22 :	1002	82 :	20506255	142 :	18440293320
23 :	1255	83 :	23338469	143 :	20390982757
24 :	1575	84 :	26543660	144 :	22540654445
25 :	1958	85 :	30167357	145 :	24908858009
26 :	2436	86 :	34262962	146 :	27517052599
27 :	3010	87 :	38887673	147 :	30388671978
28 :	3718	88 :	44108109	148 :	33549419497
29 :	4565	89 :	49995925		
30 :	5604	90 :	56634173		
31 :	6842	91 :	64112359		
32 :	8349	92 :	72533807		
33 :	10143	93 :	82010177		
34 :	12310	94 :	92669720		
35 :	14883	95 :	104651419		
36 :	17977	96 :	118114304		
37 :	21637	97 :	133230930		
38 :	26015	98 :	150198136		
39 :	31185	99 :	169229875		
40 :	37338	100 :	190569292		
41 :	44583	101 :	214481126		
42 :	53174	102 :	241265379		
43 :	63261	103 :	271248950		
44 :	75175	104 :	304801365		
45 :	89134	105 :	342325709		
46 :	105558	106 :	384276336		
47 :	124754	107 :	431149389		
48 :	147273	108 :	483502844		
49 :	173525	109 :	541946240		
50 :	204226	110 :	607163746		
51 :	239943	111 :	679903203		
52 :	281589	112 :	761002156		
53 :	329931	113 :	851376628		
54 :	386155	114 :	952050665		
55 :	451276	115 :	1064144451		
56 :	526823	116 :	1188908248		
57 :	614154	117 :	1327710076		
58 :	715220	118 :	1482074143		
59 :	831820	119 :	1653668665		
60 :	966467	120 :	1844349560		

TABLEAU 1 : LES PARTITIONS DE L'ENTIER 1

TABLEAU 1b : Nombre de partitions de l'entier n
 (n allant de 1 à 2000 avec un pas de 50)
 calcul mené a l'aide d'un logiciel multiprécision présenté dans Klein(85).

1	1
50	204226
100	190569292
150	40853235313
200	3972999029388
250	230793554364681
300	9253082936723602
350	279363328483702152
400	6727090051741041926
450	134508188001572923840
500	2300165032574323995027
550	34403115367205050943160
600	456004788008144308553622
650	5503637762499727151307095
700	60378285202834474611028659
750	610450747117966916191771809
800	5733052172321422504456911979
850	50349216918401212177548479675
900	415873681190459054784114365430
950	3246724928206047105940972859506
1000	24061467864032622473692149727991
1050	169918070997619096807349078318498
1100	1147240591519695580043346988281283
1150	7427827738529064471293660118664110
1200	46240102378152881298913555099661657
1250	277437205727159975794000686688315348
1300	1607818855017534550841511230454411672
1350	9017481745765587687202719206979752339
1400	49032194652550394774839040691532998261
1450	258898694145869442049569648660373941152
1500	1329461690763193888825263136701886891117
1550	6648312965925656816271400679772663779731
1600	32417690376154241824102577250721959572183
1650	154308961563716222079735293780517268790662
1700	717802041964941442478681516751205185010007
1750	3266292014712865596629588272103919719684547
1800	14552716211005418005132948684850541312590849
1850	63539872596459336786702846316806859551222764
1900	272089289788583262011466359201428623427767364
1950	1143579458630301665664240006110545368915059329
2000	4720819175619413888601432406799959512200344166

1	---->	1
2	---->	2
3	---->	4
4	---->	9
5	---->	20
6	---->	48
7	---->	115
8	---->	286
9	---->	719
10	---->	1842
11	---->	4766
12	---->	12486
13	---->	32973
14	---->	87811
15	---->	235381
16	---->	634847
17	---->	1721159
18	---->	4688676
19	---->	12826228
20	---->	35221832
21	---->	97055181
22	---->	268282855
23	---->	743724984
24	---->	2067174645
25	---->	5759636510
26	---->	16083734329

TABLEAU 3 : LE NOMBRE D'ARBRES A UNE RACINE AYANT N SOMMETS.

TABLEAU 4 :

Le nombre d'arbres à n sommets.
 n allant de 2 à 170.

2	1
3	2
4	4
5	9
6	20
7	48
8	115
9	286
10	719
11	1842
12	4766
13	12486
14	32973
15	87811
16	235381
17	634847
18	1721159
19	4688676
20	12826228
21	35221832
22	97055181
23	268282855
24	743724984
25	2067174645
26	5759636510
27	16083734329
28	45007066269
29	126186554308
30	354426847597
31	997171512998
32	2809934352700
33	7929819784355
34	22409533673568
35	63411730258053
36	179655930440464
37	509588049810620
38	1447023384581029
39	4113254119923150
40	11703780079612453
41	33333125878283632
42	95020085893954917
43	271097737169671824
44	774088023431472074
45	2212039245722726118
46	6325843306177425928
47	18103111141539779470
48	51842285219378800562
49	148558992149369434381
50	425976989835141038353
51	1222179262369751914558
52	3508609802706585591648
53	10078062032127180323468
54	28963544938490115587690
55	83281891024323882188934
56	239588251950971630070883
57	689586695750027771528858
58	1985698827814122851389544
59	5720475695410470698034352
60	16486885726043465205200778
61	47536435298225838513777689
62	137116646299836640013582158
63	395661426200172120893172166
64	1142146565612503377367247619
65	3298218689025396468807928287
66	9527778769277367435762139714
67	27533018688065675122704256503
68	79590547737981375530085744985
69	230149693903609741713900061706
70	665727478405857651614359412994
71	1926264145029683789201902595481
72	5575255070241458769777331977216
73	16141340753693289446826761342965
74	46745197819192522341803684458539
75	135410903503191503503384705970501
76	392361462360303212219218458046293
77	1137187531620128717310046733793377
78	3296764685809404767411359346950251
79	9559850833672724865816863589301148
80	27728021718947447790064072086211720
81	80443073708065346515907947187548704
82	233430673259657258534788317265918255
83	677523352768124889000758550090536896
84	1966913982276181876912481643993713829
85	5711352604881678383437989565512293584

$V_2(\xi)$ a $m(m+1)/2$ éléments pour
une fonction de $(0,1)^m \mapsto (0,1)^m$.

86 16587575765193708421040000378631758598
87 48185360814676586544379947795827489489
88 140001774817536903942967946008051857304
89 406851804555194134855599602163662891898
90 1182554992433068847214862126968409557746
91 3437850734230350035768412388914235959988
92 9996120786153335036227014784756370016600
93 29070538491469576103323086044968128987841
94 84557105785656201807648937916789795631644
95 245992006962808524149265893171336435342958
96 715754664594286649942853831332902539636094
97 2082946825910124637547671590789256847274970
98 6062636513439329551550570219718718950653609
99 17648704032743316047867540848170516567878799
100 51384328351659326880337136395054298255277970
101 149628319067134258729492635393764561946446898
102 435773555378556621487649875385046928847795784
103 1269318644768626173671280060343924144118929962
104 3697787471843882919503741671571956271724453863
105 10773915272265299096722633935056491599015993526
106 31395278668835569642894597448152288487513654760
107 91498335510499412949944068493812154667201899996
108 266697551736034500974403875918199160381020794943
109 777464909926662274252539710352174256524537146568
110 2266717902172939745356419584149509390193572111090
111 6609491798331308667945916567473795647776350494944
112 19274876718267379183082731606601354770476593956958
113 56216933934440628266170206911485488532804495696011
114 163981098176462436867344900912360872136310147447481
115 478377363488119996199466277447107762158096296204095
116 1395715099260136164597807553243393964797512026534174
117 4072596754913968520526017518234957521901902137340741
118 11884849792225446467164827578705271751426641565787989
119 34686687769075460871528242699923663407376280719194639
120 101246035781088847377087937778977656646151686575253687
121 295555143792188587240218693431569299206317678906752121
122 862866420343537154875548529258938427404377686664564327
123 2519372809120869553578626478943178598917005494455702465
124 7356723287883386705604350855255245130679027499828715557
125 21484182091313353896032663489738498383361689380505730072
126 62747289770322633245796118790042867418654231976047996886
127 183278787565756912981879644306583658286139158264360868095
128 535389528724822411486938974053646103381419434357851114695
129 1564110211480366722758069818159760894039941111859171595236
130 4569871869451983708778255281669612956287808901757579609878
131 13353013498885489592942759523570855123235636856510317581002
132 39020475541213265957998474787559436049343872861180239148852
133 114036337004641853533137427749022172120823880607180497864130
134 333296558812644455558619006213117966372695811470429353512864
135 974214798093723013764874725130574551516668604119508894668187
136 2847832221507880420520548184723186941125101511703828642231628
137 8325481061039310552833247634534922085844974149124193587562801
138 24341033948144688063108681110032370402994477789106884553843327
139 71170980989295279294784719273833589739006988591382370804019485
140 208113683337131426828568510655009953959752138721026381660024142
141 608599501784254797813237611073945468413576947662519911358842496
142 1779898969994874521041554973645120797002074042090639521652323074
143 5205847632349050570394011275497425392992549427924431609781670549
144 15227178705465516721514948412630697076262213358478006662712345209
145 44542940708487341767563190577280957619860211909615015424415202093
146 130307479125598904840333630192678724658382602962549556836386222377
147 381232918388395425149776922776017895077967439104836211313815192673
148 1115428205233465380621886784123969995590622406292378681412440069230
149 3263793112818154774578269983016579269273711933603470184984927188444
150 9550651408538850116424040916940356193332141892140610711711231180087
151 27949390940190093667382857231860071787268010804560840421857520526672
152 81797551187179507035587781099514557633640102155354491478458500568902
153 239406799063525783126549101224527386606953081389948675210572185544275
154 700745817095037391960838562107212671547500606165088807342053138645501
155 2051219060190061892258360712961316201049377085606328736345704208604469
156 600469169972239556201017650193770799695484293486569081354592657407537
157 17579081497593937545146888424324701622177733217707537842079385023766965
158 51466910512819379526867651039207765901128239787789578889240931106579694
159 150690591774064049630017172325499944645213421290424108131364067556492015
160 441235015883543588123335319013488010690511349210208396079341236421864577
161 1292049858632540624995827931124102403122248298572552210016059652703638588
162 3783674174889366497149526575580637689237319753713179238035512420596655339
163 11080848932950313204520480607878821622368070688719159448392642476898572694
164 32453151625318245113052241411475169279456647163175178234778031657360858698
165 95052811368305002383779036012429067137142055372634872454643924349336979183
166 278417804119243218614168906446435758330808103386317314191004723681940564053
167 81555396446012238274747342432806381835149311202346932899774828929897344523
168 2389085991492006032190423931302317063090110134015476845615010072983508704783
169 6998967261207301746138303741473831302593357057506135598276457602765079009986
170 20504962113985965815884696788979202172292976747246429719427480947194635784968

Chapitre 2

RACINE CARREE DE MATRICES BOOLEENNES

INTRODUCTION

La notion de racine de matrices a été introduite dans divers cadres avec des objectifs différents, tant du point de vue de l'étude que de celui de l'utilisation des résultats. Parmi les principales références bibliographiques il faut retenir :

Matrices Booléennes :

- ◇ C. Reischer et D.A. Simovici ([Rei.82], [Rei.83]) proposent un algorithme de calcul des racines $n^{\text{ièmes}}$ de matrices de permutation.
- ◇ K.H. Kim [Kim.82] place le problème de la racine carrée de matrices booléennes dans le cadre plus général des équations quadratiques de telles matrices.
- ◇ G.Targonski ([Tar.78], [Tar.81], [Tar.82], [Tar.85]) s'intéresse à l'aspect théorique; il considère l'équation fonctionnelle suivante :

$f : S \rightarrow S$ où S est une ensemble fini de cardinal n ,
 $p \in \mathbb{N}, p \geq 2$

Trouver $g : S \rightarrow S$ telle que $g^p = f$. (produit de composition)

Matrices Réelles :

- ◇ E.D.Denman et A.E.Beavers [Bea.76] ont résolu pour la première fois le problème du calcul de la racine carrée d'une matrice réelle définie positive.
- ◇ W.D.Hoskin et D.J.Walton [Hos.79] étendent le résultat précédent au calcul de la racine $p^{\text{ième}}$ d'une matrice réelle définie positive.
- ◇ E.D. Denman ([Den.81a], [Den.81b]) propose finalement un algorithme pour des matrices réelles quelconques et des matrices complexes.

Le problème restant ouvert pour les matrices booléennes quelconques, notre première tentative a porté sur la possibilité d'adapter au contexte discret les algorithmes fonctionnant dans le cadre réel. Ce type de passage avait été réalisé avec

succès pour les méthodes de Newton et de Gauss-Seidel [Rob.86].

Nous exposons (§1) l'idée des algorithmes dans \mathbb{R}^n ainsi que l'impossibilité de les "traduire" dans $\{0,1\}^n$.

Après avoir rappelé les principaux résultats pour les matrices booléennes (§2) nous proposons un algorithme d'extraction de racines carrées de matrices booléennes quelconques (§3).

Dans un dernier paragraphe nous présentons des résultats obtenus avec cet algorithme qui englobent les résultats obtenus par C. Reischer et D.A. Simovici.

1 RACINES $p^{\text{ième}}$ DE MATRICES REELLES.

Dans le contexte réel, A.N.Beavers et E.D.Denman ont été les premiers à proposer des algorithmes de calcul de racine carrée de matrices [Bea.76]. Ils situent ces travaux dans l'utilisation de la fonction signe d'une matrice pour résoudre le problème général d'équations de Riccati sous forme matricielle.

W.D.Hoskin et D.J.Walton utilisent puis généralisent [Hos.79] cette méthode pour calculer les racines $p^{\text{ième}}$ d'une matrice définie positive. Leur algorithme est, en fait, une accélération de la méthode de Newton.

Soit A une matrice (n,n) définie positive,
 p un entier ($p \geq 2$)

On cherche X , matrice (n,n) définie positive telle que :

$$X^p = A \quad (E1)$$

On considère une matrice (n,n) définie positive Y telle que :

$$\begin{aligned} X^{p-1}Y &= \text{Id} \quad (\text{matrice identité } (n,n)) \\ \text{et} \\ X &= AY \end{aligned}$$

$$\text{d'où} \quad Y^p = A^{1-p} \quad (E2)$$

L'itération de Newton pour résoudre les deux équations (E1, E2) se formule comme suit :

$$X_{n+1} = 1/p [(p-1) X_n + A X_n^{1-p}] \quad (N)$$

$$Y_{n+1} = 1/p [(p-1) Y_n + A^{1-p} Y_n^{1-p}]$$

Hoskin et Walton proposent le schéma suivant :

$$\begin{aligned} X_{n+1} &= \alpha_n X_n + \beta_n X_n^{2-p} Y_n^{-1} \\ Y_{n+1} &= \alpha_n Y_n + \beta_n X_n^{1-p} \end{aligned} \quad (HW)$$

avec $X_0 = A$ et $Y_0 = Id$.

Les paramètres α_n et β_n sont les coefficients d'accélération qui se calculent à partir d'une approximation de $\rho(A)$.

E.D. Denman et J. Leyva-Ramos ont étendu à trois reprises le domaine d'application de cette méthode ([Den.81a], [Den.81b]). Ils ont finalement mis au point un algorithme de calcul des racines $p^{\text{ièmes}}$ de matrices réelles quelconques et de matrices complexes, grâce à une décomposition de matrice qui leur permet d'employer l'algorithme (HW).

Le passage de résultats d'algorithmique numérique dans \mathbb{R}^n à $\{0,1\}^n$ s'est effectué plusieurs fois avec succès. F. Robert a introduit à cette occasion la notion de dérivée discrète; on trouvera dans ([Rob.76], [Rob.79], [Rob.86]) des exemples de transposition de résultats ou d'algorithmes du domaine continu au domaine discret.

Nous sommes tentés, à priori, d'utiliser ce type de passage pour obtenir à partir des méthodes précédentes un algorithme de calcul de racines pour les matrices booléennes. Mais, que l'on souhaite employer l'itération de Newton (N), ou le processus accéléré (HW), il nous faut disposer l'inverse de la matrice A. Or l'inverse usuel d'une matrice booléenne n'existe que pour les permutations [Kim.82].

La méthode de Newton discrète [Rob.86] peut donc s'appliquer à l'extraction de racines de matrices booléennes de permutation. Pour les matrices booléennes quelconques, les itérations (N) et (HW) ne sont pas exploitables car la décomposition de matrices réalisée dans [Den.81a] n'est pas possible.

Le cas des matrices de permutation est traité directement par ailleurs [Rei.82] et [Rei.85]; nous n'avons pas "transcrit" l'itération de Newton dans $\{0,1\}^n$.

Si l'on désire poursuivre dans cette perspective ($\mathbb{R}^n \mapsto \{0,1\}^n$), il conviendrait de considérer les différentes définitions d'inverse de matrices booléennes énumérées dans [Kim.82].

2 RESULTATS POUR DES MATRICES BOOLEENNES (n,n).

Dans la littérature, on dénombre trois principaux axes de références : G. Targonski, C. Reischer et D.A. Simovici, K.H. Kim, correspondant à trois centres d'intérêt distincts.

a) G. TARGONSKI : *Une approche théorique : l'équation fonctionnelle.*

Soit f une application d'un ensemble fini S dans lui-même, on considère l'équation :

$$g^p = f \quad p \in \mathbb{N},$$

où l'inconnue est $g : S \rightarrow S$, appelée racine itérative $p^{\text{ième}}$ de f .

Il précise [Tar.82] que cette question, d'énoncé simple, n'a été formulée pour la première fois qu'en 1968.

Le problème de l'existence d'une solution d'une telle équation fonctionnelle a été résolu dans [Rig.76]. La caractérisation de cette existence, exposée dans [Zim.78] et reprise dans [Tar.81], présente deux inconvénients :

- i) la complexité de l'énoncé la rend très difficilement exploitable,
- ii) le théorème n'est pas constructif : s'il permet de prouver l'existence d'une solution il ne fournit pas d'élément de construction d'une racine.

G. Targonski a reformulé cette question sous forme matricielle [Tar.82] en exposant l'énoncé suivant comme un problème ouvert de la théorie des itérations :

Soit A une matrice d'application, booléenne (n,n) (elle admet exactement un seul 1 par ligne),

trouver un algorithme qui, pour tout $p \in \mathbb{N}$ ($p \geq 2$) :

1) répond à la question : Existe-t-il une matrice B d'application booléenne (n,n) telle que

$$B^p = A \quad ?$$

2) calcule B si la réponse à 1 est : oui.

On notera que les matrices en question ne sont pas quelconques.

b) C. Reischer et D.A. Simovici : Multivalued Switching Functions

Un premier article sur les structures algébriques associatives dans l'ensemble des fonctions booléennes [Rei.71] a permis à S. Rudéanu de résoudre le problème de la racine carrée d'une fonction booléenne [Rud.76].

Mais ce n'est qu'à partir de 1982 que l'existence et le calcul de racines $n^{\text{ièmes}}$ de matrices de permutation sont traités avec succès [Rei.82], [Rei.83] et [Rei.85].

Théorème d'existence

Soit f une permutation de $\{1,2,\dots,m\}$ et r un nombre premier,

on définit la fonction spectre de f : $\text{Spec}(f) = (t_1, t_2, \dots, t_i, \dots)$

où t_i précise le nombre de cycle d'ordre i de la fonction f .

f admet une racine d'ordre r si et seulement si :

$$i \equiv 0 \pmod{r} \Rightarrow t_i \equiv 0 \pmod{r}$$

Algorithme de construction d'une racine $r^{\text{ième}}$ de f (lorsqu'elle existe).

Soit une permutation g de $\{1,2,\dots,m\}$ telle que $g^r = f$.

1) Détermination du spectre de g : $\text{Spec}(g) = (s_1, s_2, \dots, s_i, \dots)$

si $i \equiv 0 \pmod{r^2}$ alors $s_i = t_{i/r}/r$

si $i \not\equiv 0 \pmod{r^2}$ et $i \equiv 0 \pmod{r}$ choisir $s_i \leq \lceil t_{i/r}/r \rceil$

et dans ce cas : $s_{i/r} = t_{i/r} - r s_i$

2) Agglomération des cycles de f de même ordre pour constituer les cycles de g . Pour les détails voir [Rei.83]

Remarque:

Cet algorithme produit toutes les racines à condition d'effectuer tous les choix possibles pour les s_i lorsque $i \not\equiv 0 \pmod{r^2}$ et $i \equiv 0 \pmod{r}$; on obtient alors tous les

spectres possibles pour g , puis il faut créer toutes les agglomérations afin d'obtenir tous les cycles d'un même spectre.

c) K.H. KIM; *Matrices booléennes : un résultat asymptotique.*

C'est en s'intéressant à quelques équations quadratiques particulières de matrices booléennes que K.H. Kim se pose le problème de la racine carrée de matrices [Kim.82].

Il y démontre le théorème suivant :

théorème

La probabilité qu'une matrice booléenne aléatoire (n,n) admette une racine carrée est en $O(n (3/4)^n)$.

Pour fixer les idées $n = 50 \quad n (3/4)^n \approx 2.83 \cdot 10^{-5}$

$n = 100 \quad n (3/4)^n \approx 3.21 \cdot 10^{-11}$.

Parmi les 30 problèmes ouverts exposés à la fin de son ouvrage on trouve le suivant

"Calculer une racine carrée d'une matrice booléenne quelconque"

est-il un problème NP-complet ?

Après ce tour d'horizon des résultats existants nous présenterons un algorithme de calcul de racines carrées de matrices booléennes quelconques.

3 ALGORITHME DE CALCUL DE LA RACINE CARREE D'UNE MATRICE BOOLEENNE QUELCONQUE.

L'exposé de notre méthode exige que l'on conduise en permanence un parallèle entre matrice et graphe.

Soit un graphe orienté $G = (S, V)$,
 où S est l'ensemble des sommets ($|S| = n$)
 et V l'ensemble des arêtes ($V \subseteq S \times S, |V| = p$)

on établit une correspondance entre le **graphe G** et sa **matrice d'incidence A** qui est une matrice booléenne (n, n) telle que :

$$A = ((a_{ij})) \quad \text{où} \quad \begin{cases} a_{ij} = 1 & \text{si } (i, j) \in V \\ a_{ij} = 0 & \text{sinon} \end{cases}$$

Le problème :

Soit A une matrice booléenne (n, n) ,

◇ existe-t-il une matrice booléenne (n, n) B telle que $B^2 = A$?

(le carré est à considérer au sens du produit booléen de matrice.)

◇ si la réponse est positive, comment calculer B ?

Notations et correspondances :

Matrice donnée A

Une racine carrée de A , $B = \sqrt{A}$.

Graphe donné $G = (S, V)$

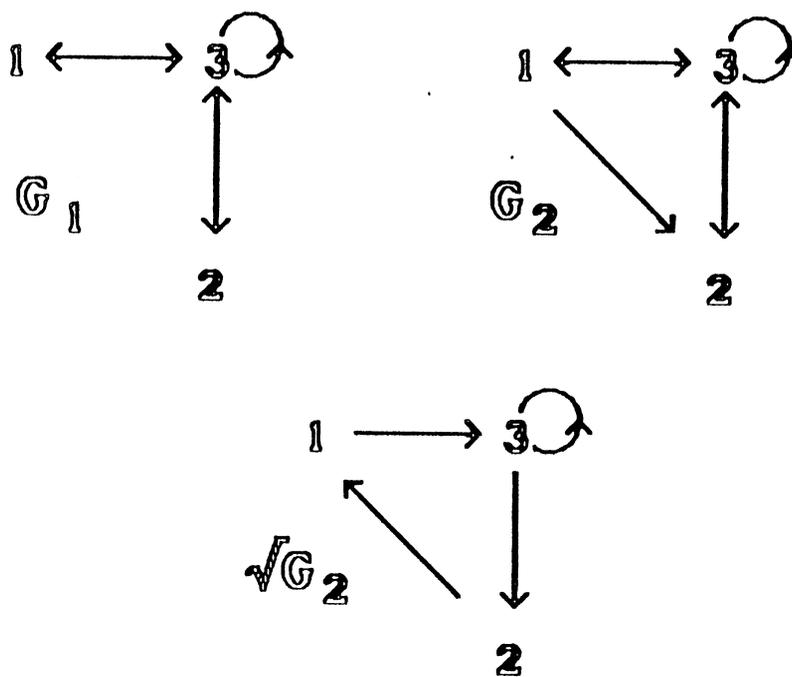
Une racine carrée de G , $H = \sqrt{G} = (S, U)$.

A est la matrice d'incidence de G et B celle de H (par définition) Nous pouvons maintenant parler de *racine carrée de graphe*.

Exemple :

Les deux graphes suivants G_1 et G_2 sont "voisins" ; une seule arête les

différence (1,2). Pourtant G_1 n'admet pas de racine carrée, ce qui n'est pas le cas pour G_2 .



La racine de G_2 que l'on exhibe est sa seule racine. Mais l'unicité, comme l'existence, ne répond pas à des critères simples.

Idée de l'algorithme : partant d'un graphe G nous allons construire une racine carrée $\sqrt{G}=H$. (si elle existe.) Nous connaissons l'ensemble des sommets S , il nous faut construire son ensemble d'arêtes U .

Chacune des p arêtes de G peut être engendrée par n paires d'arêtes de H , en effet :
 (on numérote de 1 à n les sommets de G)

- $(i,j) \in V$ provient de la paire $((i,1), (1,j)) \in U^2$
- ou de la paire $((i,2), (2,j)) \in U^2$
-
- ou de la paire $((i,n), (n,j)) \in U^2$

nous obtenons alors le schéma suivant :

1^{ère} arête de G n paires d'arêtes de H

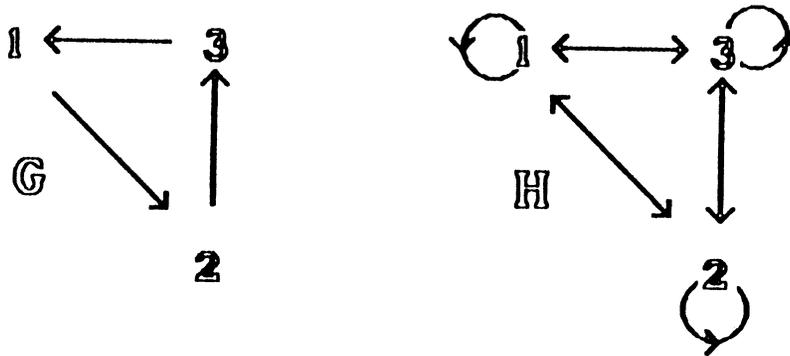
2^{ème} arête de G n paires d'arêtes de H

.....
 p^{ème} arête de G n paires d'arêtes de H

Il faut donc choisir parmi chacun des p groupes de n arêtes de H une paire afin d'obtenir G par l'opération de mise au carré.

Mais il est évident que si ce choix est quelconque, cette opération va créer un graphe contenant strictement G, c'est à dire qu'elle va **générer des arêtes supplémentaires.**

Exemple :



On cherche la racine carrée du graphe G. On désire donc créer les trois arêtes (1,2), (2,3) et (3,1).

	(1,1)	(1,2) ←
(1,2)	(1,2)	(2,2)
	(1,3)	(3,2)
	(2,1)	(1,3)
(2,3)	(2,2)	(2,3) ←
	(2,3)	(3,3)
	(3,1)	(1,1)
(3,1)	(3,2)	(2,1)
	(3,3)	(3,1) ←

II- 12

Si on effectue le choix marqué par les flèches, on obtient comme "racine carrée" le graphe H. Ce dernier est le graphe complet : il ne peut pas être la racine d'un graphe non-complet.

On remarque que le choix de chaque arête que l'on souhaite placer dans U introduit un ensemble d'arêtes impossibles pour la construction du reste de U ; ceci éviter que $G \not\subseteq (\sqrt{G})^2$.

Cet ensemble se construit à chaque choix comme suit :

si on désire placer (i,j) dans U alors

pour tout sommet k de S : $(k,j) \notin V \Rightarrow (k,i) \in U$
et
 $(i,k) \notin V \Rightarrow (j,k) \in U$

On construit l'ensemble d'arêtes U comme suit :

Choisir, si possible, p paires d'arêtes qui engendrent les p arêtes de G sans s'exclure mutuellement, au sens des ensembles impossibles que l'on vient de définir.

Si cette construction est impossible, nous avons montré la non-existence d'une racine carrée de G.

Algorithme

DONNEES

G = (S, V) un graphe orienté,

S l'ensemble des sommets comportant n éléments (n>1),

V l'ensemble des arêtes comportant p éléments (p>1).

PROBLEME

On cherche le graphe $H = (S, U)$ tel que $H^2 = G$

VARIABLES

k entier indiquant l'arête de G que l'on est en train de créer. (il y a p arêtes)

ind [k] entier indiquant la paire d'arêtes choisie pour appartenir à U afin de créer k dans V . Cette paire d'arêtes s'appuie sur un des sommets du graphe. (il y a n sommets)

impos [k] ensemble d'arêtes impossibles générées par le choix de $\text{ind}[k]$.

précédent booléen précisant si l'on régresse ou si l'on progresse dans la création des arêtes.

FONCTIONS - ACTIONS

fonction **impossible**(k : entier) : booléenne

indiquant l'appartenance de l'une des 2 arêtes de $\text{ind}[k]$ à $\bigcup_{j < k} \text{impos}[j]$.

action **calcul_de** ($\text{impos}[k]$: ensemble)

calculant l'ensemble d'arêtes impossibles généré par le choix de $\text{ind}[k]$.

action **créer** (k : entier; précédent : booléen)

calculant la paire d'arêtes pouvant créer k , si le calcul est impossible le booléen précédent est dans la valeur "vrai";

ALGORITHME (donnée : $G = (S, V)$ un graphe orienté; résultat : $H = (S, U)$)

DEBUT ACTION_PRINCIPALE

POUR $k := 1$ A p FAIRE $\text{ind}[k] := 0$ FIN_FAIRE

$k := 1$

{ (on a pu créer (k-1) arêtes) }

REPETER

{ ($0 < k < p+1$) et (précédent est quelconque) }

Créer (k , précédent)

{ ($0 < k < p+1$) et (précédent = "vrai" ssi k n'a pas pu être créée) }
 { si (précédent = "faux") on a pu créer k arêtes }

SI précédent ALORS $k:=k-1$ SINON $k:=k+1$ FIN_SI

{ (on a pu créer (k-1) arêtes) }

JUSQU'A ($k=1$ ET $\text{ind}[k]=n$) OU ($k=p+1$)

FIN_REPETER

{ (on a pu créer (k-1) arêtes) }

{ (on a pu créer (k-1) arêtes) et ($k = 1$) donc il n'y a pas de racine }

SI ($k=1$ ET $\text{ind}[k]=n$) ALORS \rightarrow " PAS DE RACINE "

{ (on a pu créer (k-1) arêtes) et ($k = p+1$) donc il y a au moins une de racine }

SINON \rightarrow " RACINE "

FIN_SI

FIN_ACTION_PRINCIPALE

ACTION Créer (donnée k : entier; résultat précédent : booléen)

{ (on a pu créer (k-1) arêtes)

DEBUT

ind [k] := ind [k] + 1

TANT QUE (impossible(k) ET (ind [k] < n))

FAIRE k:=k+1

FIN_TANTQUE

SI impossible(k) **ALORS**

DEBUT

impos [k-1] := \emptyset

ind [k] := 0

precedent := vrai

FIN_ ALORS

(on n'a pas réussi à créer k, on va essayer de recréer (k-1) avec une paire d'arêtes d'indice supérieur strictement à ind[k-1], précédent = "vrai")

SINON

DEBUT

calcul_de (impos [k])

precedent := faux

FIN_SINON

{ (on a pu créer (k-1) arêtes) et (on a essayé ind[k] possibilités pour créer l'arête n° k) , précédent = "faux"}

FIN_SI

FIN_ACTION_CREER

Visualisation du fonctionnement de l' algorithme

<u>Arêtes de G</u>		<u>Paires d'arêtes de \sqrt{G}</u>
		-
		-
1 ^{ère}	→	-→ ind[1] → impos(k) -→ <u>condition d'arrêt</u> : on ne parvient pas à créer la première paire d'arêtes ind[1]=n donc pas de racine !
.....
		-
(k-1) ^{ième}	→	- → ind[k-1] → impos(k-1) - - - On cherche la première paire d'arêtes possibles <u>si elle existe</u> : on remonte au niveau (k-1) après
k ^{ième}	→	avoir fait : ind[k]:= ∅ et impos(k-1):=∅ <u>sinon</u> : on calcule impos(k) et on descend au niveau (k+1) - -
.....
		-
		-
p ^{ième}	→	- <u>Condition d'arrêt</u> : on est arrivé au niveau p,toutes les arêtes sont créés,on vient de calculer une racine.

Dans une seconde version on calcule éventuellement plusieurs racines, l'algorithme utilise encore l'action créer(k,précédent), seul le test d'arrêt est modifié.

ALGORITHME version2

DEBUT

POUR k := 1 A p FAIRE ind [k] := 0 FIN_FAIRE
plus_de_racines := faux

TANT QUE NON (plus_de_racines) FAIRE

DEBUT

k := 1

REPETER

Creer (k, précédent)

SI précédent ALORS k:=k-1 SINON k:=k+1 FIN_SI

JUSQU'A (k=1 ET ind[k]=n) OU (k=p+1)

FIN_REPETER

SI (k=1 ET ind[k]=n) ALORS

DEBUT

→ " PAS DE RACINE "

plus_de_racines := vrai

FIN

SINON → "RACINE "

FIN_SI

FIN_TANT QUE

FIN

Attention, cet algorithme ne calcule pas toutes les racines, mais les racines qui possèdent certaines arêtes distinctes ; les arêtes qui sont les nœuds de l'arbre de construction.

Au niveau de la programmation nous avons admis la possibilité d'introduire des contraintes sur les racines potentielles. Cela permet

- i) de retrouver les résultats de [Rei.83] sur les matrices de permutation,
- ii) de mettre en évidence les arêtes "vitales" de la racine carrée : celles qui appartiennent à toutes les racines.

Remarques-Améliorations possibles.

La gestion de la place mémoire nécessaire à l'exécution de l'algorithme se conduit comme suit :

pour éviter de stocker les ensembles impossibles, dont on ignore la taille (elle varie de 0 à $2n$), nous travaillons sur une matrice "Mrac" qui respecte les conventions suivantes :

les 0 représentent les arêtes encore possibles pour le graphe racine,

les l représentent les arêtes appartenant déjà à la racine, k indique le dernier niveau où l'arête a été placé dans H,

les -k représentent les ensembles impossibles, k indique le premier niveau où l'arête a été déclarée impossible ; i.e. les -k repèrent les éléments de : $\text{impos}(k) \cup_{j < k} \text{impos}(j)$.

Le travail s'effectue donc uniquement sur cette matrice, ce qui simplifie largement le calcul et la mise à jour des ensembles impossibles. Ces deux opérations se résument alors à des recopies de lignes ou de colonnes.

Sur la figure n°1 on voit l'évolution de cette matrice de stockage dans l'exemple de la recherche de la racine carrée d'un cycle d'ordre 7.

Afin d'accroître la convergence de l'algorithme il serait intéressant de mettre au point une stratégie pour ordonner les arêtes à traiter. Par exemple, choisir de considérer en premier lieux les arêtes qui semblent les plus difficiles à créer. Le nombre d'arêtes qui aboutissent sur un même sommet est un des critères (simple) de cette difficulté.

Sur la figure n°2, on indique les différents temps d'exécution (programme Pascal sous DPS 8 / Multics) pour un même graphe avec 3 numérotations différentes.

Une heuristique s'avère également utile pour ordonner les paires d'arêtes dans les groupes, autrement que par l'ordre naturel. A ce propos le cardinal de ensembles impossibles fournit un moyen de numéroté ces paires certainement très efficace. Par contre le calcul de ces cardinaux entraîne des manipulations complexes: un compromis est à rechercher.

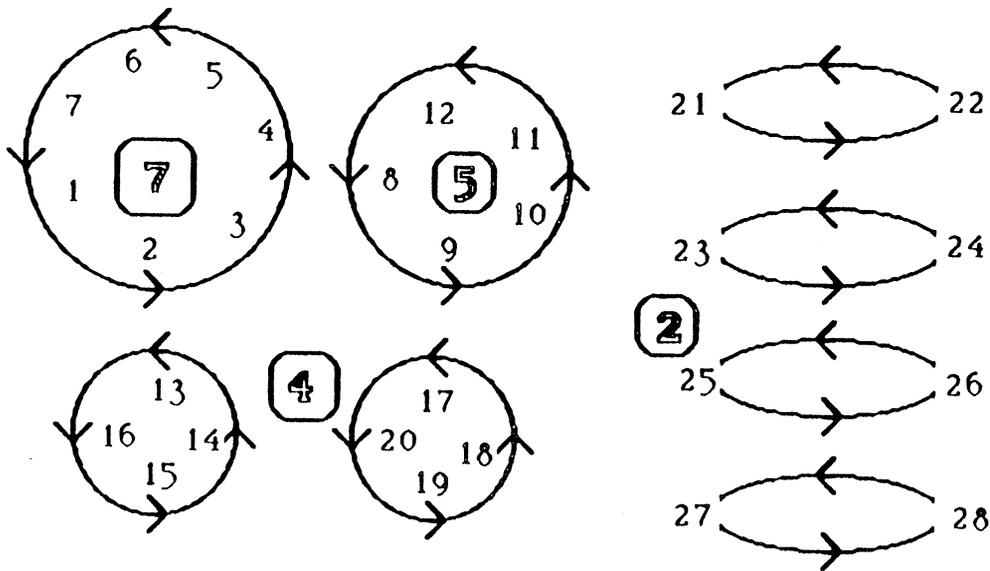
4 PRESENTATION DES RESULTATS

Nous distinguons dans ce paragraphe trois types de matrices : les matrices de permutation, celles d'application et les matrices quelconques.

a) Les Matrices de Permutation.

Notre algorithme nous permet de retrouver les résultats figurant dans [Rei.83]. Nous donnons ici un exemple de racine carrée d'un graphe de permutation composée de :

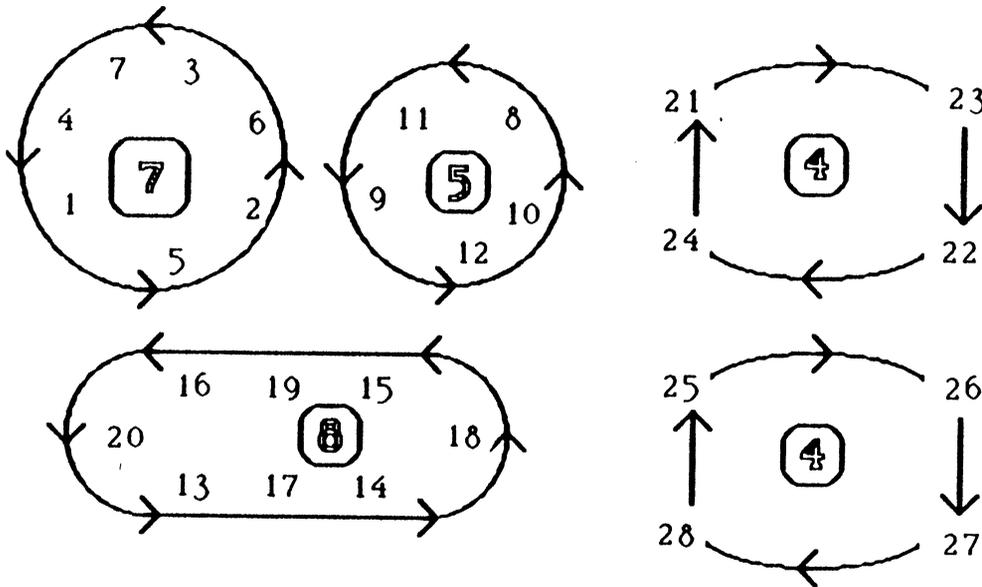
1 cycle d'ordre 7,
1 cycle d'ordre 5,
2 cycles d'ordre 4,
4 cycles d'ordre 2.



Exemple de graphe de permutation admettant une racine carrée.

Le théorème d'existence nous assure de l'existence d'au moins une racine carrée. Notre algorithme en a extrait plusieurs : dans celle que nous présentons (cf. dessin) nous observons les deux agglomérations suivantes :

2 cycles d'ordre 4 - 1 cycle d'ordre 8,
4 cycles d'ordre 2 - 2 cycles d'ordre 4.



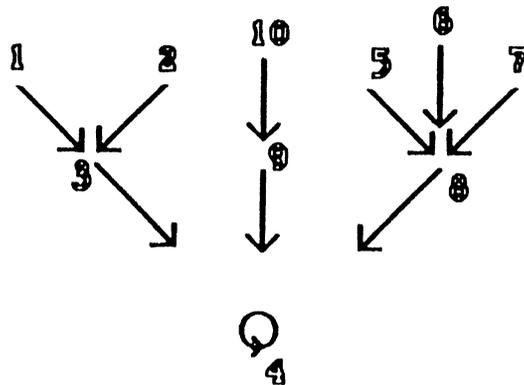
Racine carrée du graphe précédent.

Sur la figure 3 ce résultat apparaît sous forme matricielle.

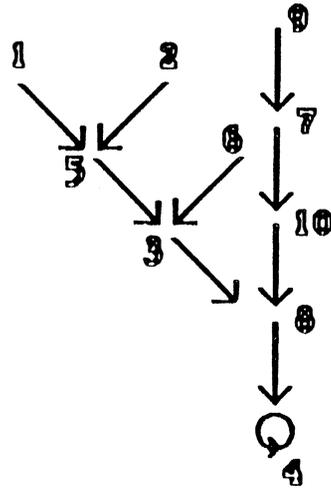
b) Les Matrices d'Application.

Ces matrices ont exactement un 1 par ligne. Il n'existe pas encore de caractérisation d'existence de racine carrée de ces matrices. Néanmoins nous travaillons actuellement dans cette direction et les quelques exemples que nous présentons ci-dessous sont encourageants.

Exemple d'application avec un point fixe :

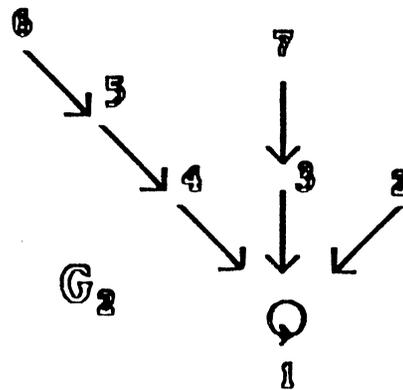
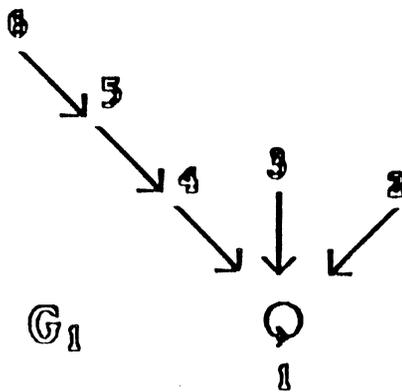


Ce graphe admet comme racine carrée le graphe suivant. La construction de la racine paraît simple ; on y "lit" instantanément le carrée. (les chemins de longueur 2)



Mais, même sur des graphes simples comme les deux suivants (G_1 et G_2) l'existence d'une racine n'apparaît pas immédiatement.

Un seul de ces deux graphes admet une racine carrée : lequel ?



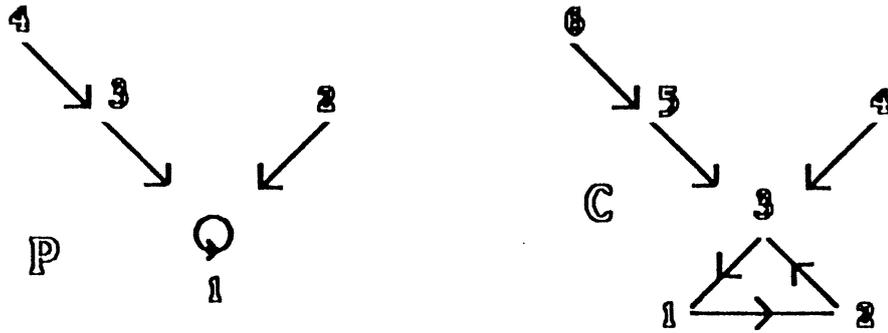
Réponse : G_1 G_2

C'est l'arête (7,3) de G_2 qui permet de construire (6,5), cette arête ne pourrait être créée dans une éventuelle racine de G_1 .

Un dernier exemple pour montrer l'impossibilité de généraliser des résultats valables pour des applications avec 1 point fixe à des applications admettant des

cycles d'ordres supérieurs.

P a une racine carrée, C n'en a pas, alors que seule la taille de leur cycle les différencie.

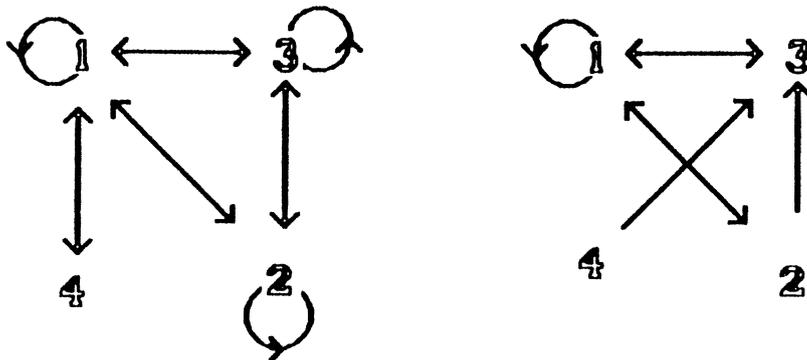


c) Les matrices quelconques.

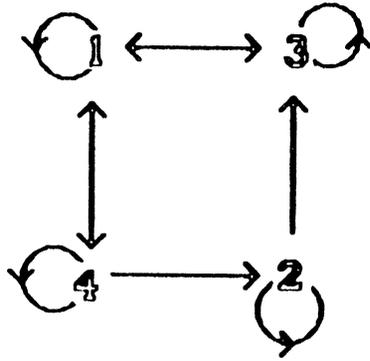
K.H. Kim, avec le résultat asymptotique et le problème ouvert (cf. §2), cerne bien la difficulté de l'étude à ce niveau.

La diversité des cas de figures possibles impose aux éventuelles caractérisations de l'existence d'une racine carrée l'obligation d'être constructives.

Ci dessous, pour deux "petits" graphes (4 sommets), un exemple d'existence et d'absence de racine carrée.



Graphe quelconque G Sa racine carrée \sqrt{G}



Grphe quelconque n'admettant pas de racine carrée.

5 CONCLUSION

Nous avons apporté à une question déjà bien cernée une réponse de type algorithmique. Celle-ci présente les avantages suivants :

- ◇ retrouver des résultats acquis dans le cas particulier de matrices de permutation,
- ◇ obtenir des résultats probants pour les racines des matrices d'application,
- ◇ calculer les éventuelles racines carrées de matrices booléennes quelconques, avec une restriction due à la complexité du problème : le temps de calcul peut devenir "vertigineux" suivant la taille des matrices.
- ◇ contrôler la place mémoire nécessaire à l'exécution de l'algorithme.

Ce travail effectué, quatre directions possibles apparaissent pour une poursuite de l'étude :

i) Chercher une définition de l'inverse d'une matrice booléenne permettant la transcription des algorithmes de \mathbb{R}^n .

Cette voie mène vers une étude radicalement différente du présent travail.

ii) Enoncer une caractérisation "pratique" de l'existence d'une racine carrée pour une matrice d'application.

C'est le but de notre travail actuel.

iii) Mettre au point des heuristiques pour la numérotation des arêtes afin d'accélérer la convergence de l'algorithme.

iv) Répondre à K.H. Kim : "Oui ce problème est NP-Complet !"

(et le démontrer....)

Des éléments de la théorie des graphes sont très certainement utiles pour aborder les points iii) et iv).

(nous reconnaissons avoir un tendre penchant pour le dernier point).

REFERENCES

- [Ale.82] G.ALEFED, N.SCHNEIDER : " On Square Roots of M -Matrices."
Linear Algebra and Appl. 42:119-132 (1982).
- [Bea.76] A.N.BEAVERS, E.D.DENMAN : "The Matrix Sign Function and Computations in Systems."
Applied Mathematics and Computation 2:63-94 (1976).
- [Den.81a] E.D.DENMAN, J.LEYVA-RAMOS : "Spectral Decomposition of a Matrix Using the Generalized Sign Matrix."
Applied Mathematics and Computation 8:237-250 (1981).
- [Den.81b] E.D.DENMAN : "Roots of Real Matrices."
Linear Algebra and Appl. 36, pp 133-139 (1981).
- [Hos.79] W.D.HOSKIN, D.J.WALTON : "A Faster, More Stable Method for Computing the p th Roots of Positive Definite Matrices."
Linear Algebra and Appl. 26:139-163 (1979).
- [Kim.82] K.H.KIM : "Boolean Matrix Theory and Applications."
Marcel Dekker, Inc. New York and Basel (1982).
- [Rei.71] C.REISCHER, D.A.SIMOVICI : "Associative Algebraic Structures in the Set of Boolean Functions and some Applications in Automata Theory."
IEEE Trans. on Comp. Vol. C-20, N°3, Mars 1971, pp298-303.
- [Rei.82] C.REISCHER, D.A.SIMOVICI : "Several Remarks on Iteration Properties of Switching Functions."
Proceedings of the Twelfth International Symposium on Multiplied-Valued Logic. Mai 1982, CNAM Paris, pp224-247.
- [Rei.83] C.REISCHER, D.A.SIMOVICI : "Roots of N-Valued Switching Functions."
Proceedings of the Thirteenth International Symposium on Multiplied-Valued Logic. Mai 1983, Holiday Inn Kyoto, pp183-188.

- [Rei.85] C.REISCHER, D.A.SIMOVICI: " Iterative Properties of Transformations of Finite Sets with Application to Multivalued Logic."
Dynamical Systems and Cellular Automata. J. Demongeot, M.Tchunte et E. Golès (Eds.) pp. 369-374. Academic Press 1985.
- [Rig.76] G.RIGGERT : "Note on the n-th iterative roots of mappings of a finite set into itself."
Lecture at the 1976 International Symposium on Functional Equations. Abstract in : Aequ. Math. 15 (1977), 288.
- [Rud.76] S.RUDEANU : "Square Roots and Functional Decomposition of Boolean Functions."
IEEE Trans. on Comp. Vol. C-25, (1976) pp 528-532.
- [Rob.76] F.ROBERT : " Théorème de Perron Frobénius et Stein-Rosenberg booléens. "
Linear Algebra and its App. 19, pp237-250. (1976)
- [Rob.79] F.ROBERT : " Dérivée discrète et convergence locale d'une itération booléenne."
Linear Algebra and its App. 19, pp237-250. (1976)
- [Rob.86] F.ROBERT : " Discrete Iterations."
Springer Verlag (1986).
- [Tar.78] G.TARGONSKI : "Problem posed at the 1978 International Symposium on Functional Equations (Matrix form of the problem of iterative roots of finite mappings)."
Aequ. Math. 19 (1979), 285.
- [Tar.81] G.TARGONSKI : "Topics in Iteration Theory."
Vandenhoeck & Ruprecht Gottingen (1981).
- [Tar.82] G.TARGONSKI : "Unsolved problems in Iteration Theory."
Proceeding of the International Colloquium on Iteration Theory and its Applications. Toulouse, May 1982, pp17-22.
- [Tar.85] G.TARGONSKI : "Embedding of Discrete Dynamical Systems."
Dynamical Systems and Cellular Automata. J. Demongeot, M.Tchunte et E. Golès (Eds.) pp. 65-74.
Academic Press 1985.

- [ZIM.78] G. ZIMMERMANN : " Über die Existenz iterativer Wurzeln von Abbildungen."
Doctoral dissertation, University of Marburg 1978.

```

0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
indice de 1 1
AU SUIVANT
1 1 -1 -1 -1 -1 -1
-1 0 -1 -1 -1 -1 -1
-1 0 0 0 0 0 0
-1 0 0 0 0 0 0
-1 0 0 0 0 0 0
-1 0 0 0 0 0 0
-1 0 0 0 0 0 0
indice de 2 7
AU PRECEDENT
indice de 1 2
AU SUIVANT
0 1 0 0 0 0 0
-1 1 -1 -1 -1 -1 -1
-1 -1 0 0 0 0 0
-1 -1 0 0 0 0 0
-1 -1 0 0 0 0 0
-1 -1 0 0 0 0 0
-1 -1 0 0 0 0 0
indice de 2 7
AU PRECEDENT
indice de 1 3
AU SUIVANT
-1 0 1 0 0 0 0
-1 -1 -1 0 -1 -1 -1
-1 1 -1 -1 -1 -1 -1
-1 0 -1 0 0 0 0
-1 0 -1 0 0 0 0
-1 0 -1 0 0 0 0
-1 0 -1 0 0 0 0
indice de 2 7
AU PRECEDENT
indice de 1 4
AU SUIVANT
-1 0 0 1 0 0 0
-1 -1 -1 -1 0 -1 -1
0 0 0 -1 0 0 0
-1 1 -1 -1 -1 -1 -1
-1 0 0 -1 0 0 0
-1 0 0 -1 0 0 0
-1 0 0 -1 0 0 0
indice de 2 5
AU SUIVANT
-1 -2 0 1 -2 0 0
-1 -1 -1 -1 1 -1 -1
-2 -2 -2 -1 -2 0 -2
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -1 -2 -2 -2
-1 -2 0 -1 -2 0 0
-1 -2 0 -1 -2 0 0
indice de 3 7
AU PRECEDENT
indice de 2 7
AU PRECEDENT
indice de 1 5
AU SUIVANT
-1 0 0 0 1 0 0
-1 -1 -1 -1 -1 0 -1
-1 0 0 0 -1 0 0
0 0 0 0 -1 0 0
-1 1 -1 -1 -1 -1 -1
-1 0 0 0 -1 0 0
-1 0 0 0 -1 0 0
indice de 2 6
AU SUIVANT
-1 -2 0 0 1 -2 0
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 0
0 -2 0 0 -1 -2 0
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 0 0 -1 -2 0

```

```

indice de 3 7
AU SUIVANT
-1 -2 -3 0 1 -2 -3
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 1
0 -2 -3 -3 -1 -2 -3
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 -3 1 -1 -2 -3
indice de 4 1
AU SUIVANT
-1 -2 -3 -4 1 -2 -3
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 1
1 -2 -3 -3 -1 -2 -3
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 -3 1 -1 -2 -3
indice de 5 2
AU SUIVANT
-1 -2 -3 -4 1 -2 -3
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 1
1 -2 -3 -3 -1 -2 -3
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 -3 1 -1 -2 -3
indice de 6 3
AU SUIVANT
-1 -2 -3 -4 1 -2 -3
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 1
1 -2 -3 -3 -1 -2 -3
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 -3 1 -1 -2 -3
indice de 7 4
AU SUIVANT
-1 -2 -3 -4 1 -2 -3
-1 -1 -1 -1 -1 1 -1
-1 -2 -2 -2 -1 -2 1
1 -2 -3 -3 -1 -2 -3
-1 1 -1 -1 -1 -1 -1
-1 -2 1 -2 -1 -2 -2
-1 -2 -3 1 -1 -2 -3
ET UNE RACINE, ET UNE I
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0

```

Figure 1

Exemple d'évolution de la matrice "Mrac" contenant les ensembles impossibles.

```

NOMBRE DE SOMMETS
78
NOMBRE D ARETES
78
RENTRE LES DONNEES
73 5
75 7.
77 2
72 8
78 3
71 5
74 7
76 2
  0 0 0 0 1 0 0 0
  0 0 0 0 0 0 0 1
  0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0
  0 0 0 0 0 0 1 0
  0 1 0 0 0 0 0 0
  0 1 0 0 0 0 0 0
  0 0 1 0 0 0 0 0
ET UNE RACINE, ET UNE!
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
r 14:20 0.816 7
    
```

```

NOMBRE DE SOMMETS
78
NOMBRE D ARETES
78
RENTRE LES DONNEES
71 2
72 3
73 4
74 5
75 1
76 2
77 3
78 4
  0 1 0 0 0 0 0 0
  0 0 1 0 0 0 0 0
  0 0 0 1 0 0 0 0
  0 0 0 0 1 0 0 0
  1 0 0 0 0 0 0 0
  0 1 0 0 0 0 0 0
  0 0 1 0 0 0 0 0
  0 0 0 1 0 0 0 0
  0 0 0 0 1 0 0 0
ET UNE RACINE, ET UNE!
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0
r 14:22 0.790 14
    
```

Figure 2

Effet d'une renumérotation des arêtes du graphe de départ. Le temps d'exécution du programme est souligné. Le meilleur temps 0.790s CPU s'obtient en opérant une permutation sur la matrice initiale.

```

NOMBRE DE SOMMETS
78
NOMBRE D ARETES
78
RENTRE LES DONNEES
71 5
72 8
73 5
74 7
75 7
76 2
77 2
78 3
  0 0 0 0 1 0 0 0
  0 0 0 0 0 0 0 1
  0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0
  0 0 0 0 0 0 1 0
  0 1 0 0 0 0 0 0
  0 1 0 0 0 0 0 0
  0 0 1 0 0 0 0 0
ET UNE RACINE, ET UNE!
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
r 14:21 1.652 7
    
```




Chapitre 3

UTILISATION DE FONCTIONS D'ENERGIE COMME OUTIL D'ETUDE DE RESEAUX A SEUIL

"Decreasing Energy Functions as a Toll for studying Threshold Networks"
en collaboration avec Eric GOLES-CHACC, F. FOGELMAN-SOULIE
Discrete Applied Maths 12 (1985) 261-277

0 INTRODUCTION

Let F be a function from $\{0,1\}^n$ into itself whose components are symmetric threshold functions. We introduce a monotonic operator, analogous to the spin glass energy [1], [9], to compare the different iteration modes on F : block-sequential, sequential and parallel. We give a general bound on the transient length and general conditions to characterize the structure of steady-states : fixed points, cycle lengths...

We apply these methods in particular networks : namely networks of generalized majority functions wich include as a particular case spin glasses. Conditions are there given wich ensure the stability of a given spin in any limit cycle.

In section 1, we give definitions, in section 2 provide sufficient conditions for the convergence of-block-sequential iterations towards fixed points. In section 3 we study parallel iterations of threshold functions and in section 4 sequential iterations of majority functions, with the particular case of the spin glasses in section 5.

1 DEFINITIONS

Let $F : \{0,1\}^n \rightarrow \{0,1\}^n$ be a mapping whose components f_1, \dots, f_n are threshold functions :

$$\forall i \in (1, \dots, n), \forall x \in \{0,1\}^n, \quad f_i(x) = \mathbb{1} \left[\sum_{j=1}^n a_{ij} x_j - b_i \right]$$

$$\text{where } \mathbb{1}[u] = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

and $A=(a_{ij})$ is a real $n \times n$ matrix and (b_1, \dots, b_n) is the real threshold vector.

It is easy to see that, for any f_i , there exists a threshold function g_i such that :

$$\diamond \text{ for all } x \in \{0,1\}^n, f_i(x) = g_i(x)$$

$$\diamond g_i(x) = \begin{cases} 1 & \text{if } \sum_{j=1}^n a_{ij}x_j - b_i < 0 \\ 0 & \text{if } \sum_{j=1}^n a_{ij}x_j - b_i > 0 \end{cases}$$

We shall say that g_i is a strict threshold function. In the following, we always assume that functions f_i have strict threshold.

An ordered partition of the set $\{1, \dots, n\}$ is a partition $(I_k)_{k=1, \dots, p}$ such that :

$$\forall x \in I_i, \forall y \in I_j, \quad i < j \Rightarrow x < y.$$

The block sequential iteration on F associated to the ordered partition $(I_k)_k$ is defined by :

$$\forall k \in (1, \dots, p), \forall i \in I_k, \quad x_i(t+1) = f_i(y^k(t))$$

$$\text{where } y^1(t) = x(t)$$

$$\forall k \in (2, \dots, p), y_j^k(t) = \begin{cases} x_j(t+1) & \text{if } j \in I_1 \cup \dots \cup I_{k-1} \\ x_j(t) & \text{otherwise} \end{cases}$$

Particular cases of block sequential iterations correspond to particular choices of the partition :

- when the partition is $(\{k\})_{k=1, \dots, n}$, the iteration is called a sequential iteration on F .
- when the partition is trivially reduced to the unique set $\{1, \dots, n\}$, the iteration is called a parallel iteration on F .

Clearly, since $\{0, 1\}^n$ is a finite set, all the trajectories, $(x(t))_{t \geq 0}$, of any

block sequential iteration are ultimately periodic.

Hence, for every $x \in \{0,1\}^n$, there exist numbers $p(x)$, $t(x)$ such that :

$$x(t+p(x))=x(t) \text{ for any } t \geq t(x)$$

and $x(t+q) \neq x(t)$ for any $t < t(x), q < p(x)$.

We denote $T(\Lambda, b) = \text{Max} \{t(x)/x \in \{0,1\}^n\}$

$$p(\Lambda, b) = \text{Max} \{p(x)/x \in \{0,1\}^n\}$$

and call them the transient and cycle lengths of the mapping F defined by $\Lambda = (a_{ij})$ and $b = (b_1, \dots, b_n)$.

It is clear that the different block sequential iterations on F , for the different ordered partitions of $\{1, \dots, n\}$, have the same fixed points. But they may have limit cycles (of length larger than 1) which are different.

In this paper, we will give a characterization of the limit cycles structure and transient length of block sequential iterations on F , dependig on specific assumptions on F .

2 CONDITIONS FOR CONVERGENCE TO FIXED POINTS

This formalism has mainly been introduced in models of neural networks and spin glasses. The various techniques used to study the dynamics of this model range from classical algebra [5] to statistical mechanics [1]. One feature of the model which proved to be particularly useful is the existence of a monotonic operator (or Lyapunov function) defined on the trajectories : in the case of the spin glass problem [1], this operator is the spin glass interaction energy.

In our general framework, we define the energy associated to F as :

$$\forall x \in \{0,1\}^n, E(x) = -1/2 \sum_{i=1}^n x_i \sum_{j=1}^n a_{ij}x_j + \sum_{i=1}^n b_i x_i$$

Lemma 1 . Let F be a threshold network whose matrix A is symmetric .

Then, for any ordered partition $(I_k)_{k=1,\dots,p}$, if $A_k = (a_{ij})_{i,j \in I_k}$

is non negative definite on the set $\{-1,0,1\}$, for all $k \in (1,\dots,p)$, the block sequential iteration on F associated to $(I_k)_k$

is such that :

$$\forall t, x(t+1) \neq x(t) \Rightarrow E(x(t+1)) < E(x(t))$$

Proof

Let us decompose the block sequential iteration on F at time t into p successive steps, where at time $t+k/p$, $k=1,\dots,p$, elements in block I_k only change states :

$$x(t) = (x_1(t), \dots, x_n(t))$$

$$\forall k \in (1,\dots,p), x_j(t+k/p) = \begin{cases} f_j(x(t+(k-1/p))) & \forall j \in I_k \\ x_j(t+(k-1/p)) & \forall j \notin I_k \end{cases}$$

$$\text{Let } \Delta_t E = E[x(t+1)] - E[x(t)]$$

$$= \sum_{k=1}^p E[x(t+k/p)] - E[x(t+(k-1/p))]$$

If Λ is symmetric, we have :

$$\begin{aligned} \Delta_t E = & - \sum_{k=1}^p \sum_{i \in I_k} [x_i(t+1) - x_i(t)] \sum_{\substack{j_k' \in \cup I_{k'} \\ k' < k}} a_{ij} x_j(t+1) + \\ & \sum_{\substack{j_k' \in \cup I_{k'} \\ k' > k}} a_{ij} x_j(t+1) - b_i + 1/2 \sum_{i \in I_k} x_i(t+1) \sum_{j \in I_k} a_{ij} x_j(t+1) \\ & - 1/2 \sum_{i \in I_k} x_i(t) \sum_{j \in I_k} a_{ij} x_j(t+1) \end{aligned}$$

Let us denote :

$$1 \quad \partial_k E = - \sum_{i \in I_k} [x_i(t+1) - x_i(t)] \left[\sum_{\substack{j_k' \in \cup I_{k'} \\ k' < k}} a_{ij} x_j(t+1) + \sum_{j \in I_k} a_{ij} x_j(t+1) \right] +$$

$$\sum_{\substack{j_k' \in \cup I_{k'} \\ k' > k}} a_{ij} x_j(t+1) - b_i]$$

$$2 \quad \partial_k E = -1/2 \sum_{i \in I_k} [x_i(t+1) - x_i(t)] \sum_{j \in I_k} a_{ij} [x_j(t+1) - x_j(t)]$$

$$\text{Then : } \Delta_t E = \sum_{k=1}^p [\partial_k^1 E + \partial_k^2 E]$$

As each f_i is a strict threshold function, it follows that :

$$x(t+1) \neq x(t) \Rightarrow \partial_k^1 E \leq 0$$

and there exists at least one $k \in \{1, \dots, p\}$ such that $\partial_k^1 E < 0$

Moreover, since $\{x_i(t+1) - x_i(t)\} \in \{-1, 0, 1\}$ the assumption on each block A_k implies that : $\partial_k^2 E \leq 0$.

Hence $\Delta_t E < 0$.

Remark : The diagonal dominance assumption on A_k :

$$\forall i \in I_k, \quad a_{ii} \geq \sum_{j \in I_k, j \neq i} |a_{ij}|$$

is a sufficient condition to ensure the non negative definite condition. Thus this condition generalizes results (cf. Goles[6]) based on the diagonal dominance assumption. It is not a necessary condition.

Proposition 1

Let $F=(A,b)$ be a threshold network and $(I_k)_{k=1, \dots, p}$ be an ordered partition of $\{1, \dots, n\}$ such that :

◇ A is a symmetric matrix.

◇ $\forall k \in (1, \dots, p)$,

$\Lambda_k = (a_{ij})_{i,j \in I_k}$ is a non negative definite matrix on the set $\{-1, 0, 1\}$.

Then the block sequential iteration on F associated to $(I_k)_k$ is such that :

(i) $P(A,b) = 1$: the iteration only has fixed points.

$$(ii) T(A,b) \leq \frac{1}{\mathcal{E}(A,b)} \left(\frac{1}{2} \sum_{i,j=1}^n |a_{ij}| + \sum_{i=1}^n b_i \right)$$

where $\mathcal{E}(\Lambda, b) = \text{Min} \{ |E(x(t)) - E(x(t+1))| / x(0) \in \{0, 1\}^n, x(t) \neq x(t+1) \}$

Proof

Suppose the iteration has a limit cycle of period $T > 1 : x(0), x(1), \dots, x(T-1)$.

Then : $\forall t \in (0, \dots, T-1), x(t) \neq x(t+1)$

Hence lemma 1 implies that :

$$E(x(0)) > E(x(1)) > \dots > E(x(T-1)) > E(x(0)) \quad \text{which is impossible.}$$

Thus $T=1$, which proves (i).

It is easy to see that, for any $x \in \{0, 1\}^n$:

$$\sum_{i: b_i < 0} b_i - 1/2 \sum_i \sum_{j: a_{ij} > 0} a_{ij} \leq E(x) \leq \sum_{i: b_i > 0} b_i - 1/2 \sum_i \sum_{j: a_{ij} < 0} a_{ij}$$

hence :

$$\begin{aligned} T(\Lambda, b) \mathcal{E}(\Lambda, b) &\leq [-1/2 \sum_i \sum_{j: a_{ij} < 0} a_{ij} + \sum_{i: b_i > 0} b_i] - \\ &\quad [-1/2 \sum_i \sum_{j: a_{ij} > 0} a_{ij} + \sum_{i: b_i < 0} b_i] \end{aligned}$$

which proves (ii).

Remark :

In the particular case of a sequential iteration, proposition 1 applies and provides a bound for $T(\Lambda, b)$. This result had been directly proven in [4].

In the case where $a_{ij} \in \mathbb{Z}$, it is possible to find a uniform bound on $\mathcal{E}(\Lambda, b)$: suppose that $b_i \in [k_i, k_{i+1}[$, with $k_i \in \mathbb{Z}$, for all $i \in (1, \dots, n)$. Then for each function f_i , there exists a function g_i with strict threshold $(k_i + 1/2)$ such that $f_i = g_i$. Hence, we can assume that $(b_i - 1/2) \in \mathbb{Z}$.

We then have :

Corollary 1

If A is symmetric integer matrix with non negative definite blocks A_k , then the block sequential iteration on F associated to the ordered partition $(I_k)_k$ is such that :

$$T(A,b) \leq \sum_{i,j=1}^n |a_{ij}| + 2 \sum_{i=1}^n |b_i|$$

Proof

In lemma 1, we have shown that :

$$\Delta_t E = \sum_{k=1}^p \left[\partial_k^1 E + \partial_k^2 E \right]$$

and that : $\partial_k^2 E \leq 0$, $\partial_k^1 E \leq 0 \forall k \in (1, \dots, p)$

and $\partial_k^1 E < 0$ for one k at least, if $x(t+1) \neq x(t)$.

Hence $\Delta_t E \leq \partial_k^1 E$ where $\partial_k^1 E < 0$

From the strict threshold assumption, it is easy to show that :

$$a_{ij} \in \mathbb{Z} \Rightarrow \left[\sum_{\substack{j \in \cup I_{k'} \\ k' < k}} a_{ij} x_j(t+1) + \sum_{\substack{j \in \cup I_{k'} \\ k' \geq k}} a_{ij} x_j(t) - (b_i - 1/2) \right] \in \mathbb{Z}^*$$

$$\Rightarrow \partial_k^1 E \leq -1/2$$

$$\Rightarrow |\Delta_t E| \geq 1/2 = \varepsilon(A,b) \geq 1/2$$

The result then follows from proposition 1.

An interesting case corresponds to $a_{ij} \in \{-1,0,1\}$: there are classically the interaction coefficients in spin glass problems.

Corollary 2

If A is symmetric matrix, with elements in $\{-1,0,1\}$ and non negative definite blocks A_k , then the block sequential iteration on F associated to the ordered partition $(I_k)_k$ is such that : $T(A,b) < 3n^2$.

(the bound on the transient is quadratic).

Furthermore, if $|V(i)| = \text{card} \{ j \in (1, \dots, n) : a_{ij} \neq 0 \}$ is uniformly (in i) bounded by some constant v , then the bound is linear : $T(A,b) \leq 3nv$.

Proof

It is easy to see that : $\forall i \in (1, \dots, n) \quad -|V(i)| \leq \sum_j a_{ij} x_j \leq |V(i)|$

hence :

◇ if $b_i \in [-|V(i)|, |V(i)|]$ the i -th cell is fixed by the iteration into one value (0 if $b_i \geq |V(i)|$, 1 if $b_i \leq -|V(i)|$) in one step.

◇ otherwise, corollary 1 leads to :

$$T(A,b) \leq \sum_{i,j} |a_{ij}| + 2 \sum_i |b_i| \leq 3 \sum_i |V(i)|$$

the bounds follows immediatly.

We have see that there were different block sequential iterations corresponding to the different ordered partitions. We now introduced an order on the partitions wich will allow to compare the corresponding block sequential iterations.

Let $P = (I_1, \dots, I_p)$ and $P' = (I'_1, \dots, I'_q)$ be two ordered partitions of the set $\{1, \dots, n\}$. Then P' is finer than P , which we denote $P' \leq P$, iff any block I_k , $k=1, \dots, p$ is a reunion of some I'_j .

Remark

Endowed with this relation, the set of ordered partitions of $\{1, \dots, n\}$ is a partially ordered set which is a lattice.

Lemma 2

Let P be an ordered partition. Then if F satisfies the assumptions of lemma 1, any partition P' finer than P also satisfies them.

This means that, as soon as one partition P satisfies the required assumptions, any P' a P has only fixed points (those of P) and has a bounded transient length.

The proof is obvious.

In the example shown figure 1, we represent the ordered partition lattice and iteration graphs of the different block sequential iterations on a threshold networks F of size $n=4$.

3 PARALLEL ITERATIONS

In this paragraph, we will study the parallel iterations of strict threshold mappings. This is a particular case of paragraph 2, which results therefore apply in this section. However, the non negative definite assumption on matrix A is very restrictive and results are known [5] when A is simply symmetric : the parallel iteration has only limit cycles of period 2 or fixed points. In [5], GOLES has used an algebraic invariant to produce this result.

We will here make use of an "energy" which will allow us to give another proof of the previous result and moreover provide a bound on the transient length.

Proposition 2

Let $F = (A, b)$ be a threshold network such that A is symmetric matrix. Then the parallel iteration on F is such that :

i) $P(A, b) \leq 2$

ii)
$$T(A, b) \leq \frac{1}{\epsilon(A, b)} \left[\frac{1}{2} \sum_{i, j} |a_{ij}| + \sum_j |b_j| \right]$$

Proof

Let $E(x(t)) = f(x(t), x(t-1))$

with
$$f(u, v) = - \sum_{i=1}^n u_i \sum_{j=1}^n a_{ij} v_j + \sum_{i=1}^n b_i (u_i + v_i)$$

then $\Delta_t E = E[x(t+1)] - E[x(t)]$

$$\Delta_t E = - \sum_{i=1}^n x_i(t+1) \sum_{j=1}^n a_{ij} x_j(t) + \sum_{i=1}^n b_i [x_i(t+1) + x_i(t)] +$$

$$\sum_{i=1}^n x_i(t) \sum_{j=1}^n a_{ij} x_j(t-1) + \sum_{i=1}^n b_i [x_i(t) + x_i(t-1)]$$

$$\Delta_t E = - \sum_{i=1}^n [x_i(t+1) - x_i(t-1)] [\sum_{j=1}^n a_{ij} x_j(t) - b_i]$$

because A is symmetric.

And thus, either $x(t) = x(t+1)$ and $\Delta_t E = 0$
 or $x(t) \neq x(t+1)$ and $\Delta_t E < 0$

which proves that P(A,b) is either 2 or 1 (on any limit cycle $x(t+1)$ must be equal to $x(t-1)$, for any t).

$$f(u,v) = - \sum_{i=1}^n u_i [\sum_{j:a_{ij}>0} a_{ij} v_j + \sum_{j:a_{ij}<0} a_{ij} v_j] + \sum_{i:b_i>0} b_i [u_i + v_i] + \sum_{i:b_i<0} b_i [u_i + v_i]$$

if $(u,v) \in \{0,1\}^2$:

$$- \sum_{i=1}^n \sum_{j:a_{ij}>0} a_{ij} + 2 \sum_{i:b_i<0} b_i \leq f(u,v) \leq - \sum_{i=1}^n \sum_{j:a_{ij}<0} a_{ij} + 2 \sum_{i:b_i>0} b_i$$

$$= T(A,b) \leq \frac{1}{\mathcal{E}(A,b)} \sum_{i,j} |a_{ij}| + 2 \sum_i |b_i|$$

with $\mathcal{E}(A,b) = \text{Min} \{ |f(F^2(x),F(x)) - f(F(x),x)| \mid x \in \{0,1\}^n, x \neq F^2(x) \}$

As in corollary 1, it follows that :

Corollary 3

If A is a symmetric integer matrix $(a_{ij} \in \mathbb{Z})$ then :

i) the parallel iteration on F is such that : $T(\Lambda, b) \leq \sum |a_{ij}| + 2 \sum |b_i|$

ii) If, in addition, $a_{ij} \in \{-1, 0, 1\}$, $\forall i, j$ then $T(\Lambda, b) \leq 3n^2$

furthermore, if $|V(i)| = v$ then $T(\Lambda, b) \leq 3nv$.

Proof : as in corollary 2.

Corollary 3 shows that for parallel iterations on threshold networks with symmetric matrix of elements -1, 0, or 1 and regular neighbourhood, the transient length is linear in n ($O(n)$). When the a_{ij} are arbitrary, the transient length may be much larger : see GOLES-OLIVOS [7] for an example of convergence in $O(2^{n/3})$.

Corollary 4.

Let $F = (\Lambda, b)$ be a threshold network. Then there exists a uniform bound on the transient lengths of different iterations on F, T_{iter} , provided Λ fulfills the corresponding assumptions :

$$\exists \epsilon > 0 : T_{iter}(\Lambda, b) \leq 1/\epsilon \left[1/2 \sum_{i,j} |a_{ij}| + \sum_i |b_i| \right].$$

Proof : there only exists a finite number of iterations on F : parallel, sequential, and block sequential. Hence proposition 1, corollary 1 and proposition 2 prove the existence, under the suitable assumptions, of a finite number of coefficients $\epsilon^k(\Lambda, b)$.

The corollary follows by taking $\epsilon = \min_k \epsilon^k(\Lambda, b)$.

Remark : $e \neq e_0 = \min \{ |E(x) - E(y)| \mid x, y \in \{0, 1\}^n, x \neq y \}$.

e_0 will often be 0, which corresponds to the high degeneracy usually observed in the physical systems (see spin glasses [1]).

4 SEQUENTIAL ITERATIONS OF MAJORITY FUNCTIONS.

In this paragraph, we will study sequential iterations of a particular class of threshold functions, namely majority functions, which have been introduced in the framework of social systems ([5], [8]). In the following section, we apply our results to a model of spin glasses.

Let $n \in \mathbb{N}$ be given and I_0, I_1 be a partition of $\{1, \dots, n\}$.

A majority function is defined as follows :

$$\forall i \in I_0, \forall x \in \{0,1\}^n,$$

$$f_i(x) = \begin{cases} 1 & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i > 0 \\ x_i & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i = 0 \\ 0 & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i < 0 \end{cases}$$

$$\forall i \in I_1, \forall x \in \{0,1\}^n,$$

$$f_i(x) = \begin{cases} 1 & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i > 0 \\ 1-x_i & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i = 0 \\ 0 & \text{if } \sum_{j \neq i} a_{ij} x_j - b_i < 0 \end{cases}$$

The two different formulations for $i \in I_0$ and $i \in I_1$ model the two possible reactions in case of a tie : if $\sum_{j \neq i} a_{ij} x_j - b_i = 0$, element i can either its state (I_0) or reverse it (I_1).

It is easy to see that for any majority function f_i there exist a threshold function g_i such that :

$$\diamond \forall x \in \{0,1\}^n, f_i(x) = g_i(x)$$

$$\diamond g_i(x) = \mathbb{1} \left[\sum_{j=1}^n a_{ij} x_j - b_i \right]$$

Proof :

It is sufficient to take :

$$\text{for } i \in I_0 \quad b_i = b_i + 1/2 a_{ii}, \quad a_{ii} \in [0, m_+]$$

$$\text{for } i \in I_1 \quad b_i = b_i, \quad a_{ii} \in]-m_+, 0[$$

$$\text{with } m_+ = \text{Min} \left\{ \sum_{j \neq i} a_{ij} x_j - b_i / x \in \{0,1\}^n : \sum_{j \neq i} a_{ij} x_j - b_i > 0 \right\}$$

Remarks

1- Note that : $\forall i \in I_0, a_{ii} \geq 0$ and $\forall i \in I_1, a_{ii} < 0$

2- According to this last result, majority functions are a subclass of threshold functions, namely those with diagonal elements bounded by some given constants m_+^i .

In the following, we always assume that these conditions are satisfied.

Let $F : \{0,1\}^n \mapsto \{0,1\}^n$ be a mapping whose components f_1, \dots, f_n are majority functions.

We define the energy associated to the sequential iteration on F as follows :

$$\forall x \in \{0,1\}^n, E(x) = -1/2 \sum_{i=1}^n x_i \sum_{j \neq i} a_{ij} x_j + \sum_{i=1}^n b_i x_i$$

Proposition 3

Let $F = (A, b)$ be a mapping whose components are majority functions. If A is a symmetric matrix, then the sequential iteration on F is such that :

$$(i) \quad \forall k \in I_0, \\ x_k(t+k/n) \neq x_k(t+(k-1)/n) = E(x(t+k/n)) - E(x(t+(k-1)/n)) < 0$$

$$(ii) \quad \forall k \in I_1, \\ x_k(t+k/n) \neq x_k(t+(k-1)/n) = E(x(t+k/n)) - E(x(t+(k-1)/n)) \leq 0$$

Proof

Let decompose the sequential iteration on F at time t into n successive sets, where at time $t + k/n$, $k=1, \dots, n$, element k only changes states (we assume, for sake of simplicity, that the sequential iteration corresponds to the identical permutation).

$$\text{Let } \Delta_t E = E(x(t+1)) - E(x(t)) \quad \Delta_t E = \sum \delta_t^k E$$

$$\text{with } \delta_t^k E = E(x(t+k/n)) - E(x(t+(k-1)/n))$$

(i) if $k \in I_0$, then :

$$\delta_t^k E = - [x_k(t+k/n) - x_k(t+(k-1)/n)] [\sum_{j \neq k} a_{kj} x_j(t+(k-1)/n) - b_k]$$

then it is easy to see, from the definition of f_k that $\delta_t^k E < 0$

(ii) if $k \in I_1$, then :

$$\delta_t^k E = - [x_k(t+k/n) - x_k(t+(k-1)/n)] [\sum_{j \neq k} a_{kj} x_j(t+(k-1)/n) - b_k]$$

we then have : $\delta_t^k E \leq 0$ and

$$\delta_t^k E = 0 \quad \text{if and only if} \quad \sum_{j \neq k} a_{kj} x_j(t+(k-1)/n) - b_k = 0$$

which ends the proof.

Corollary 4

If A is symmetric, then :

(i) $\forall t, x(t+1) \neq x(t) \Rightarrow \Delta_t E \leq 0.$

(ii) Furthermore, if $\exists k \in I_0 : x(t+1) \neq x(t)$, then : $\Delta_t E < 0.$

(iii) If $\exists k \in I_1 : x(t+1) \neq x(t)$ and there is no tie for k , then : $\Delta_k E < 0.$

(iv) In any limit cycle, all elements i with $i \in I_0$ are stable, all elements i with $i \in I_1$ changes state only when there is a tie.

Proof

obvious from proposition 3.

Remark

When $I_1 = \emptyset$, this result is a particular case of proposition 1.

5 MODELLING SPIN GLASSES BY MAJORITY FUNCTIONS.

We will now address ourselves to a particular block sequential iteration of threshold networks which has been introduced in solid states physics.

Some magnetic alloys are obtained by diluting magnetic impurities in a non magnetic metal. These systems are called spin glasses. Their magnetic properties can be deduced from the study of an Hamiltonian :

$$H = - \sum_{i,j} J_{ij} S_i \cdot S_j - F \cdot \sum_i S_i$$

where S_i is the spin (or magnetic moment) of impurity i , J_{ij} is the interaction between impurities i and j (J_{ij} is chosen at random), F is a magnetic field.

The ground states of a spin glass are associated to a minimum of energy H . Physicists have introduced many methods to study these problems [1]. One model is the 2D Ising models : the impurities are supposed to be distributed regularly at the nodes of a grid, normally toric. Each spin may only have 2 values : -1 (down) or +1 (up). In the absence of magnetic field, the energy is then :

$$H(\omega) = - \sum_{i,j} J_{ij} S_i S_j$$

where ω is the state vector (S_i) and the sum runs on couples (i,j) which are neighbours on the grid.

It has been shown that, in the 2D problem, ground states could be determined in polynomial time, whereas this problem in 2D with magnetic field or 3D is NP-complete [2]. In the cases, heuristic methods have been designed to find approximate solutions [3,10] : these methods usually rely on Monte Carlo iterations.

Formally, these methods consist in updating a given spins configuration by choosing one spin at random and eventually changing its state with a probability depending on a local majority rule.

We will study here sequential iterations of deterministic such rules, which corresponds in the spin glass context to a zero temperature : the choice of a spin which is to change states is not random, but fixed by a given permutation of $\{1, \dots, n\}$ which be assumed in the following to be identity.

We suppose in the following that all spins are located at the nodes of an $n \times n$

grid drawn on a torus : lines 1 and n are connected, columns 1 and n also. Each spin i may interact with its four neighbours on the grid with the interaction coefficients $a_{ij} \in \{-1, 0, +1\}$.

Let $n \in \mathbb{N}$ be given and I_0, I_1 be a partition of $\{1, \dots, n\}$.

A generalized spin glass is a mapping $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ whose components f_1, \dots, f_n satisfy :

$$\forall i \in I_0, \forall x \in \{0, 1\}^n,$$

$$f_i(x) = \begin{cases} 1 & \text{if } \sum_{j \neq i} c(x_j) - 2 > 0 \\ x_i & \text{if } \sum_{j \neq i} c(x_j) - 2 = 0 \\ 0 & \text{if } \sum_{j \neq i} c(x_j) - 2 < 0 \end{cases}$$

$$\forall i \in I_1, \forall x \in \{0, 1\}^n,$$

$$f_i(x) = \begin{cases} 1 & \text{if } \sum_{j \neq i} c(x_j) - 2 > 0 \\ 1 - x_i & \text{if } \sum_{j \neq i} c(x_j) - 2 = 0 \\ 0 & \text{if } \sum_{j \neq i} c(x_j) - 2 < 0 \end{cases}$$

$$\text{with } c(x_j) = \begin{cases} x_j & \text{if } a_{ij} = +1 \\ 1 - x_j & \text{if } a_{ij} = -1 \end{cases}$$

It is clear that f_i is a majority function as defined in the previous section, with coefficients $b_i = \sum_{j \neq i : a_{ij} = -1} a_{ij} + 2$

Lemma 3

A generalized spin glass F with interaction coefficients $(a_{ij})_{i \neq j}$ is a threshold network with a matrix $\Lambda = (a_{ij})$ iff :

$$\forall i \in \{1, \dots, n\}, a_{ii} \in]-1, +1[.$$

Proof

as here all coefficients a_{ij} are in $\{-1, 0, +1\}$, $m_+^i = 1$, which ends the proof, using the previous section.

In the following, we will always assume that the coefficients a_{ij} are in $] -1, +1[$.

In a spin glass, each spin intends to get parallel to the neighbours with which it is positively connected ($a_{ij} = +1$) and antiparallel otherwise. Of course, these conditions may be contradictory and it is usually assumed that the spin takes on the majority direction : in case of a tie, the spin takes may either keep its direction ($i \in I_0$) or reverse ($i \in I_1$). In this last case, it is called a crazy spin.

Corollary 5

For any sequential iteration of a generalized spin glass $F = (A, b)$ with diagonal elements a_{ij} in $] -1, +1[$ and symmetric matrix A ,

(i) $\forall i \in (1, \dots, n)$, $a_{ii} \geq 0 = x_i(t)$ is constant.

(ii) All oscillating elements i have their coefficients $a_{ij} < 0$ and change state only in case of a tie.

Proof : obvious from corollary 4.

Remarks

1- Oscillating regions in the limit cycle are made up from connected elements $i \in I_1$ (see figure 2).

2- It is clear that elements i in I_0 help to stabilize the network : neighbours of such "stable" elements tend to be stable also.

Furthermore, oscillating elements i in I_1 , are scarce, since the combinatorics needed for this situation makes it occur very rarely : a tie must happen at each instant when the element is supposed to change states. Simulations usually exhibited less than 20% of the elements i in I_1 oscillating.

3- If condition $a_{ij} \in]-1,+1[$ is violated, some elements $i \in I_0$ may oscillate (see figure 3).

6 CONCLUSION

In this paper, we have shown that the general concept of energy, first introduced in the spin glasses, allows a very good understanding of different iterations of threshold networks.

In particular, it has been shown that block sequential iterations with symmetric matrix A and non negative definite blocks could only lead to fixed points.

Furthermore, the transient length of the different iterations is uniformly bounded, by a linear bound (in the size of the network) in spin glasses.

Finally, application of same ideas in a spin glass model allows to predict the stability of individual spins in sequential iterations : these results may provide an insight into the problem of rigid clusters [3].

Simulations were run on a PDP 11/23, with programs in Fortran and Pascal.

REFERENCES

- 1- BALIAN R., MAYNARD R., TOULOUSE G. (Eds) : (1979)
III Condensed Matter. Pceedings of Les Houches 1978 Summer School. North Holland.
- 2- BARAHONA F. : (1980)
Application de l'optimisation combinatoire à certains modèles de verres de spins. Thèse IMAG - GRENOBLE .
- 3- BIECHE I., MAYNARD R., RAMMAL R., UHRY J.P. : (1980)
On the ground states of the frustration model of a spin glass by a matching method of graph theory. J. Phys.A 13, pp 2553-2576 .
- 4- FOGELMAN F., GOLES E. , WEISBUCH G. : (1983)
Transient length in sequential iteration of threshold functions. Discrete Applied Math. 6, pp 95-98.
- 5- GOLES-CHACC E. : (1980)
Comportement oscillatoire d'une famille d'automates cellulaires non uniformes. Thèse IMAG, GRENOBLE, 1980.
- 6- GOLES-CHACC E. :
Fixed point behaviour of threshold functions on a finite set. SIAM J. on Alg. and Appl. Maths, 3, n°4.
- 7- GOLES E., OLIVOS J. : (1983)
Iterative behaviur of generalized majority functions. Math. Soc. Sci. 4, pp197-204.
- 8- HOPFIELD J.: (1982)
Neural networks and physical systems with emergent computational abilities. Proc. Nat. Acad. Sci. USA, 79, pp 2554-58.
- 9- KIRKPATRICK S., GELATT C.D., VECCHI M.P. : (1983)
Optimization by simulated annealing. Science, 220, n°4598, pp671-680.
- 10- PERETTO P. : (1984)
Collective properties of neural networks, a statistical physics approach. Biol. Cyb., 50, pp51-62.

Figure 1:

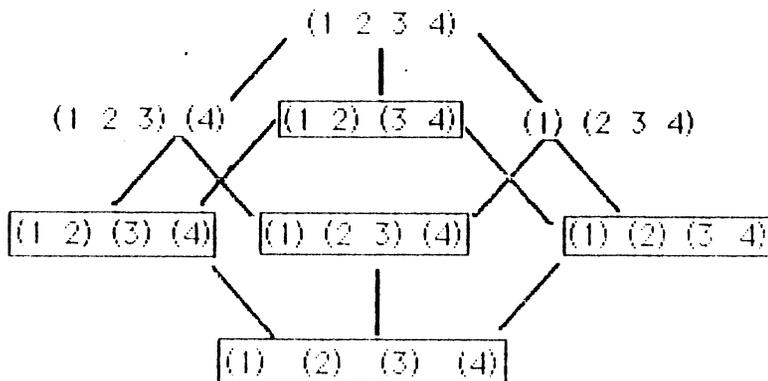
This figure shows the ordered partitions lattice of a threshold network $F=(A,b)$ with $n=4$ and:

$$A = \begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & 2 & 1 & -2 \\ -1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The ordered partition $P=(I_1, \dots, I_p)$ is represented as: (elements in I_1) ... (elements in I_p). For example in partition (1 2) (3) (4), elements 1 and 2 are run in parallel, then element 3 and finally element 4.

A partition P' finer than a partition P is represented as a descendant of P on the lattice. For example, partition (1 2) (3) (4) is finer than partition (1 2 3) (4) and is thus one of its descendants; an other one is (1) (2 3) (4) and of course all their descendants.

Partitions which are circled satisfy the assumptions of lemma 1. According to lemma 2, all their descendants also satisfy these assumptions, and are thus also circled.



The iteration graphs of the different block sequential iterations on F are represented below. The 16 states (0000...1111) are shown by their decimal representation. For each state, its successor under the corresponding iteration is indicated: for example, in both iterations (1 2) (3 4) and (1 2) (3) (4), (bottom left), state 4 (0100) leads to state 12 (1100), which is a fixed point.

Note that all iterations have the same fixed points, the parallel iteration -alone- also has a limit cycle of period 2 (with states 5 and 10).

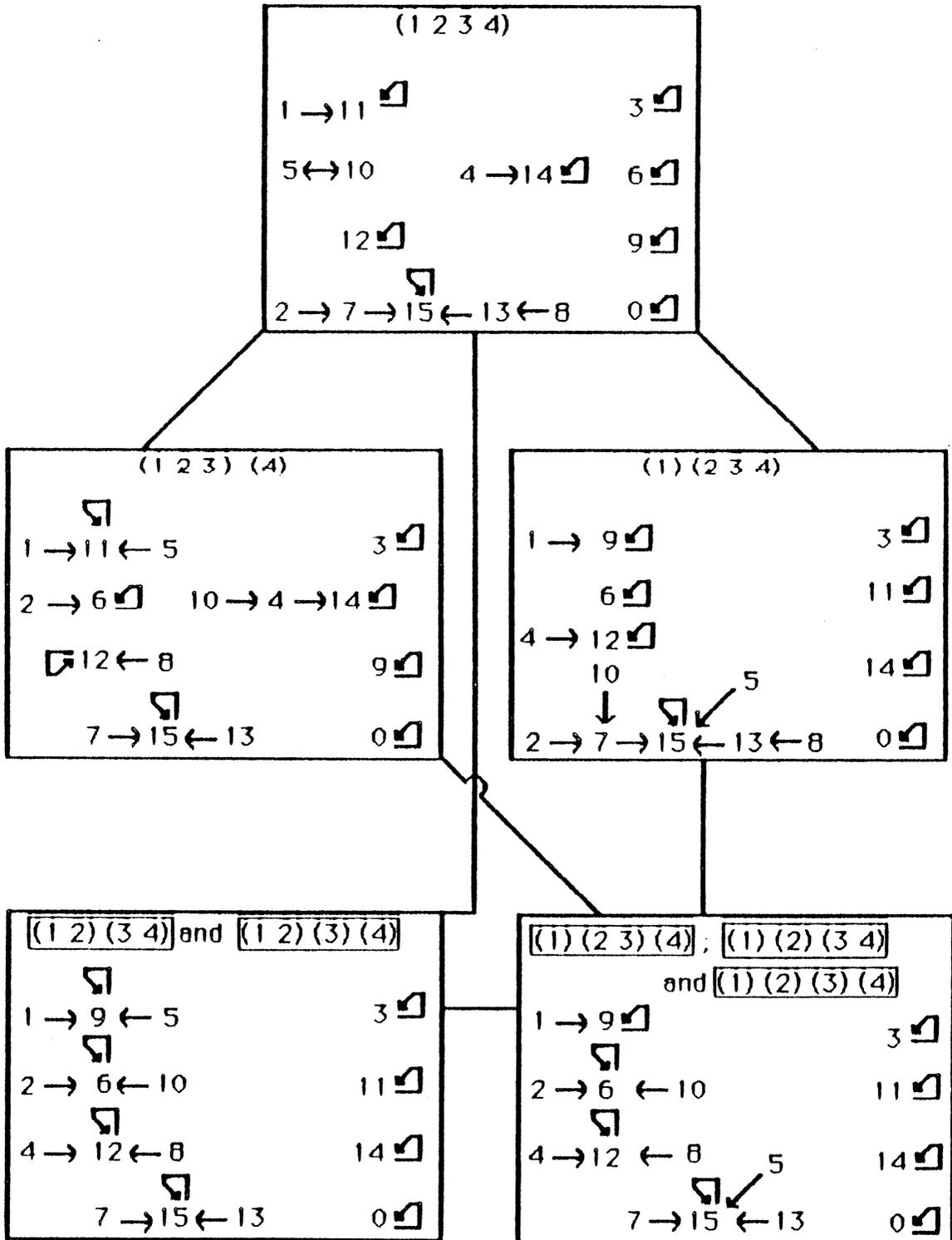


Figure 2

This figure shows a limit cycle for a given generalized spin glass.

The network is supposed to be drawn on a torus, its size is $n = 8 \times 8$.

The interactions are shown below (right): "+" indicates a coefficient $a_{ij} = +1$ and "-" a coefficient $a_{ij} = -1$, between elements i and j (marked *).

The D-matrix gives the diagonal coefficients a_{ii} .

For a given initial condition, the limit cycle has been computed. The period is 12. Elements in I_0 were all found stable, either in 0 or in 1 (marked so on the picture). Some elements in I_1 were stable (marked 0 or 1 as before), but 14 of them were oscillating (marked 3 -left- or M -right- on the pictures).

$$D = \begin{pmatrix} -0.5 & 0.9 & 0.1 & -0.2 & -0.2 & 0.2 & 0.6 & 1.0 \\ -1.0 & -0.8 & -0.1 & -0.9 & 0.1 & -1.0 & 0.8 & 0.0 \\ 0.7 & -0.4 & -0.9 & -0.4 & -0.2 & 0.8 & 0.6 & -0.8 \\ -0.8 & 0.0 & -0.9 & 0.3 & -0.5 & 0.3 & 0.9 & 0.1 \\ 0.5 & -0.3 & -0.1 & 0.0 & -0.5 & 0.8 & -0.4 & 0.9 \\ 0.4 & -0.5 & 0.4 & 0.4 & -0.6 & 0.8 & 0.7 & 0.4 \\ 1.0 & 0.0 & 0.3 & -1.0 & -0.7 & 0.1 & 0.0 & -0.2 \\ 0.9 & 0.3 & -0.2 & -1.0 & 1.0 & -0.6 & 0.1 & -0.9 \end{pmatrix}$$

1 0 0 3 0 0 0 1	*-*+*-M+*+*+*-*+
0 0 0 3 0 1 1 3	- + + - + - - -
1 0 0 3 3 1 1 1	*+*+*+M+*-+*+M-
0 1 1 1 1 1 0 1	- + + - + + + -
1 3 1 1 1 0 1 0	*+*+*+M-M+*-+*+*
1 3 1 1 1 0 1 0	+ - - - - + + +
1 1 1 3 0 1 3 3	*-*+*+*+*+*-*-*-
0 0 3 3 1 3 1 3	- - + + + - + +
	+M++*+*-*-*-*
	+ - + - - + + +
	-M++*+*-*-*-*
	- + + - - - + -
	++*+M+*-*-M+M-
	- - - - - + +
	+-M-M+*+M+*-M+
	- + - - - + + +

Figure 3

This figure shows a generalized spin glass with diagonal elements not in]-1,+1[.
 The connection structure and notations are similar to those in figure 1.

This simulation shows that some elements i in I_0 may oscillate if some coefficients a_{ii} lie outside]-1,+1[.

These elements are marked P -right- on the figure.

$$D = \begin{pmatrix} 1.3 & 0.7 & -1.1 & -1.0 & 0.3 & -1.1 & 0.4 & -0.5 \\ -1.5 & -1.2 & -0.3 & -0.1 & 0.5 & 1.2 & 1.6 & 0.3 \\ 0.5 & -1.8 & -0.4 & -1.3 & -1.0 & 0.2 & -1.9 & 0.6 \\ -1.2 & 1.9 & 1.8 & -1.8 & -0.2 & 0.7 & 0.2 & -1.3 \\ 1.6 & -0.9 & 2.0 & -0.2 & 1.0 & -0.3 & 0.0 & 0.1 \\ -0.4 & 1.3 & -0.3 & 0.8 & 0.5 & -1.8 & -1.5 & 0.8 \\ 0.9 & -1.3 & -1.9 & -0.1 & -0.9 & -1.7 & -0.1 & 1.8 \\ 0.3 & 1.3 & -1.0 & 0.4 & 1.5 & 0.1 & 1.6 & 1.7 \end{pmatrix}$$

0 3 3 3 1 3 1 1	*+P-M-M+*-M+**+*-
	- - - - - - - +
3 3 3 3 0 1 0 1	M-M+M+M+*-*-+**+
	+ + + + + + + +
0 3 3 3 0 1 3 1	*-M-M+M+*-+*M-*-
	- - + - - - - -
3 0 1 3 1 0 0 3	M-*-*+M-*-*+*-M+
	- - - + + - - -
1 3 0 3 1 1 1 1	*-M+*-M-+**+**+
	+ - + - + - - -
3 0 0 0 1 3 3 0	M+**+**+*-+M-M-*-
	+ - + - + - - +
0 3 3 1 1 3 3 0	*+M-M-+**+M-M+**+
	+ - - - - + + +
0 0 1 0 0 3 0 1	**+*-*-*-P+*-*-
	+ - - - - + - +

En annexe à ce chapitre nous donnons un exemple de cycle limite, pour un verre de spin généralisé, qui entraîne une remarque sur la taille possible des périodes que ces réseaux peuvent admettre.

I1	I1	I0	I1	I0	I0	I0	I1
I0	I1	I1	I0	I0	I1	I1	I0
I0	I0	I1	I0	I0	I1	I1	I0
I1	I1	I1	I1	I1	I0	I1	I1
I1	I0	I1	I1	I0	I0	I1	I1
I0	I0	I0	I0	I1	I0	I0	I0
I0	I0	I0	I1	I0	I1	I0	I1
I1	I0	I0	I0	I1	I1	I1	I0

Cette figure indique la partition I0/I1, des cellules du réseau, utilisée dans la définition de la fonction F.

1	-	0	+	0	+	0	-	0	+	0	-	1	+	1	+
+	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-
0	-	3	-	3	+	0	+	0	+	1	-	0	+	0	+
-	-	-	+	+	-	+	+	-	-	-	-	-	-	-	-
1	+	1	-	0	-	1	-	0	+	0	-	1	-	0	-
-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+
0	+	0	+	1	+	1	+	1	+	1	-	1	-	0	+
+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-
0	+	0	+	1	-	0	-	0	+	0	+	0	+	1	-
-	+	+	+	+	+	+	-	-	-	+	-	-	+	-	-
1	-	0	-	1	-	0	+	0	-	1	-	1	+	1	-
-	+	+	-	-	-	+	+	-	-	-	-	-	-	-	-
0	-	0	+	0	+	3	+	1	+	1	+	1	+	1	-
-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
1	+	0	+	0	+	0	+	3	-	3	+	3	-	1	+
+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+

Il est démontré dans le paragraphe 5 que seules les cellules appartenant à I1 peuvent osciller dans le cycle limite. Sur la figure précédente nous précisons le signe des connexions et l'état du réseau dans un cycle limite : les 3 représentants des cellules oscillantes. La période de ce cycle limite est 12 et elle est issue des deux sous périodes (3 et 4) avec lesquelles oscillent les deux groupes de cellules marquées 3. La cellule 3 isolée admet naturellement une sous période 2.

La répartition des connexions (+/-) et le voisinage stable permettent

d'étudier en détail les 2 "sous-cycles".



Les 2 sont la contribution stable que le réseau fournit aux deux cellules oscillantes. Voici le cycle que parcourt ces deux cellules :

1	-	1
0	-	1
1	-	0

On montre que pour toutes fonctions à seuils on ne peut obtenir de cycle de période 4 avec 2 cellules. L'étude exhaustive des cycles possibles conduit dans tous les cas à une contradiction dans les inégalités construites avec les éléments de la matrice des coefficients et les composantes du vecteur des seuils.

Par contre avec 3 cellules le même réseau propose un exemple de cycle de période 4 :



agrandissement
du réseau autour
des 3 cellules oscillantes.



Cycle suivant
lequel elles
oscillent.

0	-	1	+	0
0	-	0	+	0
1	-	0	+	0
0	-	1	+	1

Il faudrait donc établir un résultat sur la période maximale des cycles et la taille d'un réseau. L'exemple précédent montre qu'en couplant plusieurs "sous-cycles" un réseau (type verres de spins généralisé) peut posséder des cycles très longs dans lesquels, seules les cellules de II oscillent.



Chapitre 4

DYNAMIQUE DE RESEAUX BOOLEENS ALEATOIRES ET APPLICATIONS

Articles sources :

1) *Dynamics of random boolean networks*

Disordered Systems and Biological Organization, Nato ASI Series in Computer and Systems Sciences, pp 67-71, Springer Verlag 1986.

2) *On learning abilities of random boolean networks*

Actes du Colloque COGNITIVA : De l'Intelligence Artificielle aux Biosciences, Paris, juin 1985.

3) *Emergence of classification procedures in automata networks as a model for functional self-organization.*

en collaboration avec

H.ATLAN, E.BEN-EZRA, F.FOGELMAN-SOULIE, et G.WEISBUCH
Theoretical Biology 1986, à paraître .

I INTRODUCTION

En 1969, S.Kauffman emploie pour la première fois des réseaux booléens aléatoires comme modèle de réseaux génétiques afin d'étudier le phénomène de différenciation cellulaire. Il a depuis approfondi son étude et les nombreux articles [Kau.69,70,72,79] qu'il a publiés sur ce sujet font référence comme base indispensable à tout travail sur les réseaux booléens aléatoires.

En toute généralité un réseau aléatoire à n éléments peut-être vu comme un ensemble d'automates booléens interagissant suivant un graphe de connexion qui fournit à chaque loi booléenne régissant un automate ses entrées et ses sorties. Nous précisons au second paragraphe les principales définitions utilisées dans ce rapport.

Il est clair que si on laisse au réseau la "liberté de choisir tous ses paramètres au hasard", la complexité des dynamiques qui en résultent interdit toutes interprétations. Il est donc nécessaire de fixer quelques unes des caractéristiques des réseaux : interconnectivité, graphe de connexion, choix parmi les différentes lois booléennes Nous donnerons au paragraphe III les résultats sur les liens entre les types de réseaux et leurs dynamiques ainsi que les spécifications des réseaux que nous étudions.

La répartition spatiale des éléments stables (et oscillants) pour tous les attracteurs d'un réseau donné constitue la caractéristique prépondérante de la dynamique de ce réseau. L'existence de ces coeurs stables et oscillants fait l'objet de nombreuses études principalement celle de F. Fogelman [Fog.84b] qui a construit une approximation théorique du coeur stable : le domaine forçant. Dans le paragraphe IV nous montrons les limites des différents moyens d'approximations du coeur stable puis nous proposons une méthode d'étude des dynamiques comportant :

i) une modification de la définition du domaine forçant,

ii) la mise en évidence d'un réseau partiel aux entrées stables et l'étude dans ce sous-réseau du coeur oscillant et des cellules qui n'appartiennent à aucun des deux coeurs (nous introduisons la notion de paliers pour regrouper ces cellules)

Déjà, dans ses premiers articles sur les réseaux booléens aléatoires, S. Kauffman s'intéresse aux perturbations que l'on peut faire subir à la dynamique de ses modèles : il remarquait que celle-ci était en général très robuste au bruit. H.

IV - 3

Atlan, F. Fogelman et G. Weisbuch reprennent ce problème dans [Atl.81]; ils ont également donné une définition de reconnaissance de séquences booléennes [Atl.85]: c'est dans cette optique que nous exposons dans le paragraphe V des moyens pour prévoir les qualités de reconnaissance que possède un réseau donné.

II DEFINITIONS

Un réseau booléen à n éléments :

une fonction $F : \{0,1\}^n \mapsto \{0,1\}^n$

où F est un n-uplet $(f_1, f_2, \dots, f_i, \dots, f_n)$ chaque f_i étant une fonction booléenne à n variables $f_i : \{0,1\}^n \mapsto \{0,1\}$.

Cette fonction peut être considérée comme un réseau de n automates chacun d'entre eux étant régi par une fonction f_i .

Le graphe de connexion du réseau:

le graphe $G = (X, U)$

où $X = \{1, 2, \dots, n\}$ est l'ensemble des cellules du réseau et

$X \times X \supset U$, $(i, j) \in U$ si f_j dépend de l'état de la cellule i c'est-à-dire qu'il existe un état du réseau $(x_1, x_2, \dots, x_i, \dots, x_n)$ tel que :

$$f_j(x_1, x_2, \dots, \bar{x}_i, \dots, x_n) \neq f_j(x_1, x_2, \dots, x_i, \dots, x_n)$$

C'est ce graphe qui précise à chaque loi f_i ses entrées et ses sorties. Il s'agit bien évidemment des entrées qui jouent un rôle effectif vis à vis de la loi f_i .

Si par exemple ce graphe est complet chaque automate devra connaître l'état de tout le réseau pour pouvoir calculer son nouvel état.

L'interconnectivité du réseau :

Un réseau à n éléments est d'interconnectivité k si :

$$\forall j \in X \quad \text{il existe au plus k éléments } i_1, \dots, i_k \text{ de } X \text{ tels que} \\ (i_l, j) \in U \text{ pour tout } l \text{ de } 1 \text{ à } k.$$

Dans un tel réseau, l'état de chaque automate ne dépend que de l'état d'au plus k autres automates. Nous développerons au paragraphe suivant le rôle important de l'interconnectivité vis à vis de la dynamique du réseau.

Choix des lois booléennes

Dans un réseau d'interconnectivité k le "tirage aléatoire des lois" s'effectue parmi les 2^{2^k} lois booléennes à k entrées.

Par exemple, il y a 16 fonctions booléennes à 2 variables, et il y en a 4294967296 à 5 variables !

Malgré "l'explosion" du nombre de lois celles-ci se classent toujours en 4 catégories :

1) les lois constantes :

il y en a 2 : la tautologie, fonction constante à 1 et la contradiction fonction constante à 0.

Ces deux lois rendent leurs automates entièrement indépendants du reste du réseau, elles jouent donc un rôle faible dans la dynamique. (elles ne seront pas choisies lors des simulations)

2) les lois de transfert :

elles ne dépendent que d'une seule de leurs entrées qu'elles transmettent soit directement soit en l'inversant.

Il y en a $2n$ que l'on note pour i de 1 à n :

$$\forall (x_1, \dots, x_n) \in \{0,1\}^n \quad t_i(x_1, \dots, x_n) = x_i \quad \text{et}$$

$$\forall (x_1, \dots, x_n) \in \{0,1\}^n \quad \overline{t_i}(x_1, \dots, x_n) = \overline{x_i}$$

3) les lois forçantes :

une de leurs entrées (par exemple i), au moins, possède une valeur particulière (valeur forçante associée à cette entrée, notée x_i) qui contraint la fonction à prendre une valeur forçée (unique, notée v_i) quelques soient les valeurs des autres entrées. On peut considérer les lois de transfert et même les lois constantes comme des lois forçantes, mais nous avons préféré les distinguer.

Une fonction $f : \{0,1\}^n \rightarrow \{0,1\}$ est forçante en son $i^{\text{ième}}$ entrée si :

$$\exists x_i \in \{0,1\} \text{ tel que } \forall y \in \{0,1\}^n \text{ on ait } (y_i = x_i) \implies f(y) = v_i$$

La proportion p_n des lois forçantes parmi les fonctions à n variables vaut :

pour $n=2$ $p_2 = 14/16$

pour $n=3$ $p_3 = 120/256$, cette proportion décroît très rapidement

Walker et Gelfand [Gel.77] montrent que :

$$p_n \text{ tend vers } 0 \text{ comme } 4n \cdot 2^{-2^{n-1}}$$

4) *lois non forçantes :*

Il n'existe pas de valeurs particulières pour une entrée qui peut forcer la sortie de la loi.

Nous donnerons des exemples de ces 4 types de lois dans le cas d'une interconnectivité 2.

Dynamique

Plusieurs modes d'itération permettent de calculer la dynamique du réseau partant d'une condition initiale donnée.

Soient $F : \{0,1\}^n \mapsto \{0,1\}^n$ un réseau booléen, $x^0 \in \{0,1\}^n$ une condition initiale, $X = \{1,2,\dots,n\}$ l'ensemble des cellules,

itération parallèle : pour tout $t \geq 0$ $x^{t+1} = F(x^t)$; dans ce mode toutes les cellules calculent leurs états en même temps .

itération série :

$$x_1^{t+1} = f_1(x_1^t, \dots, x_n^t)$$

$$x_2^{t+1} = f_2(x_1^{t+1}, x_2^t, \dots, x_n^t)$$

.....

$$x_n^{t+1} = f_n(x_1^{t+1}, x_2^{t+1}, \dots, x_n^t)$$

On obtient toutes les autres itérations séries en renumérotant l'ensemble X (ce qui donne n! itérations).

F.Robert a comparé les différents modes (cf. [Rob.86]) : les itérations séries et parallèles ont mêmes points fixes mais admettent des cycles limites distincts. Il existe également une itération par blocs : mode hybride où l'on itère certains blocs de cellules en série, et, à l'intérieur des blocs on conduit l'itération de manière parallèle. On se référera pour une analyse plus détaillée à [Rob.76].

Transitoire et Cycle limite.

L'ensemble des états que peut prendre un réseau à n éléments étant fini (2^n états possibles), l'itération étant déterministe, la dynamique aboutit obligatoirement à un cycle limite (de période P), après un transitoire (de taille T)

$\forall x^0 \in \{0,1\}^n$ condition initiale,

$\exists T(x^0) \in \mathbb{N}$ et $\exists P(x^0) \in \mathbb{N}$ tels que

$$\forall t \geq T(x^0) \quad x^{t+P(x)} = x^t$$

et $\forall t \leq T(x^0)$ ou $0 < q < p \quad x^{t+q} \neq x^t$

III Rappels de résultats sur la dynamique de certains réseaux. Spécifications des réseaux étudiés.

Le rôle de l'interconnectivité sur la dynamique du réseau est capital. On trouve à ce propos dans [Kau.70] un commentaire sur les effets de différentes interconnectivités dont nous rappelons les principales observations produites à ce titre.

interconnectivité $k=1$: le réseau n'est composé que de lois de transfert. Sa dynamique est trop simple pour être génératrice de modèles intéressants.

interconnectivité $k=2$: c'est le cas le plus approfondi par S.Kauffman qui le trouve satisfaisant comme modèle de réseau génétique : "The behaviour of randomly interconnected, deterministic nets in which each element received just two inputs from other elements is biologically reasonable." [Kau70a].

C'est également pour cette interconnectivité qu'il a établi de surprenantes lois d'échelle : pour un réseau de taille n il y a en moyenne $\sqrt{n}/2$ cycles limites différents pour ce genre de réseau et ils admettent des périodes moyennes de $\sqrt{n}/2$. Si l'on rappelle que le réseau peut prendre 2^n états, ces résultats révèlent la grande simplification qui s'opère au niveau des dynamiques (i.e. un faible nombre d'attracteurs). La forte proportion de lois forçantes (14/16) pour cette interconnectivité expliquerait cette stabilité des attracteurs. Il est en effet aisé de comprendre qu'une présence importante de lois forçantes tend à simplifier la dynamique du réseau en ne faisant pas intervenir toutes les entrées de chaque automate.

interconnectivité $k=3$: la complexité des dynamiques commence à croître mais ces réseaux ont encore fait l'objet de nombreuses études [Fog.85]

interconnectivité $k=3n$: le graphe de connexion du réseau est complet et le choix des fonctions s'effectue parmi $2^{2^{*n}}$ lois possibles ! Dans ce cas un résultat donne une idée des dynamiques obtenues : la période moyenne des cycles limites est de $2^{n/2}$ cf [Rub.54].

On trouve dans [Fog.82] une étude des dynamiques des réseaux homogènes (ce qui sort du cadre des réseaux aléatoires à proprement parler) qui explicite le rôle des différents types de lois sur la dynamique : lois de transfert et lois non forçantes en particulier.

F.Fogelman [Fog.85] fait varier le pourcentage de fonctions forçantes dans des réseaux à faible interconnectivité (2 ou 3) pour montrer que l'on peut simuler la complexité des dynamiques de réseau plus fortement interconnectés.

Spécifications des réseaux étudiés.

Nous nous intéresserons aux réseaux d'interconnectivité 2; le tirage aléatoire des lois se fera donc parmi les 16 fonctions booléennes à deux variables. Ces 16 lois se trouvent dans la table de la figure (III-1) avec leurs valeurs forcées et forçantes.

Par exemple : la loi = (loi n° 13) admet 1 pour valeur forcée avec 0 comme valeur forçante pour la première entrée et 1 pour la seconde. En effet :

$$\forall b \in \{0,1\} \quad (0 = b) \Rightarrow 1 \quad \text{et} \quad (b = 1) \Rightarrow 1.$$

Sauf spécification contraire le graphe de connexion employé sera régulier : il est formé de circuits élémentaires constitués de 4 cellules voisines et les connexions des bords lui donnent une structure de tore : voir figure (III-2). Ce type de graphe autorise une meilleure interprétation (lisibilité) des résultats sans nuire particulièrement à la généralité de ceux-ci.

Nos simulations porteront sur des réseaux à 256 éléments présentés sous forme de tableaux 16 x 16, où les lois sont figurées par leur code de la table. On peut voir dans la figure (III-3) un exemple de réseau aléatoire ainsi présenté.

Parties Stables et Oscillantes.

Durant un cycle limite un automate peut rester stable ou osciller; on peut voir sur la figure (III-4) ces deux cas : y sont représentés les états des cellules stables durant un cycle limite, et symbolisées les cellules qui ont changé au moins une fois d'état (durant ce cycle) par une *.

L'ensemble des cellules stables (resp. oscillantes) pendant un cycle limite C s'appelle la partie stable du cycle : PS(C) (resp. la partie oscillante du cycle : PO(C)).

Coeurs Stable et Oscillant.

Depuis longtemps on s'est aperçu de l'existence d'un noyau commun aux parties stables (resp. oscillantes) : ce lieu du réseau composé de cellules stables (resp. oscillantes) dans tous les cycles limites a reçu le nom de coeur stable (resp. oscillant) du réseau.

Si on note \mathbb{C} l'ensemble des cycles limites d'un réseau donné, on définit les coeurs de ce réseau comme suit :

$$\text{le Coeur Stable} \quad CS = \bigcap_{C \in \mathbb{C}} PS(C)$$

$$\text{le Coeur Oscillant } CO = \bigcap_{C \in \mathcal{C}} PS(C)$$

Ces définitions ne sont évidemment **pas constructives** : comment imaginer que l'on puisse déterminer l'ensemble des cycles limites d'un réseau ? Il faudrait alors simuler les dynamiques issues des 2^n conditions initiales possibles : par exemple, pour une taille $n=256$ (celle des réseaux dont nous avons simulé les dynamiques) cela donne 2^{256} dynamiques à tester !!

Pour donner un ordre de grandeur, si le calcul d'un cycle limite nécessite 5 millisecondes (ce qui est loin d'être le cas) la simulation de toutes les conditions initiales demanderait (à quelques millénaires près) $1.836 \cdot 10^{67}$ années !!!

IV DYNAMIQUES DES RESEAUX BOOLEENS D'INTERCONNECTIVITE 2.

Les coeurs (particulièrement le coeur stable) constituent l'élément de base dans l'étude des dynamiques, pour deux raisons :

- i) ce sont les points à comportement commun pour tous les attracteurs d'un réseau, on peut donc les considérer caractéristiques d'un réseau.
- ii) c'est à l'extérieur de ces deux coeurs que l'on peut considérer l'effet des conditions initiales donc différencier les attracteurs du réseau.

Comme nous l'avons observé au paragraphe précédent la définition des coeurs ne permet malheureusement pas de les calculer. C'est donc par approximation que nous procéderons.

La méthode la plus naturelle est directement issue de la définition des coeurs: on simule un "petit" nombre de dynamiques en fournissant au réseau des conditions initiales aléatoires, et on mise sur la stabilité du calcul de l'intersection (vis à vis du nombre de cycles limites sur laquelle elle est prise). Ce type d'opération pose immédiatement deux questions :

◇ qu'est ce qu'un nombre "petit" (80 et 10^3 sont tous deux "petits" devant $2^{256} \approx 1.157 \cdot 10^{77}$) ?

◇ quelle confiance peut-on avoir en la stabilité de l'intersection ?

F.Fogelman a construit un outil, le **domaine forçant**, permettant d'évaluer le coeur stable en évitant la simulation de dynamiques.

Nous décrivons tout d'abord les avantages et les inconvénients de ces deux approximations.

1) Approximations des coeurs d'un réseau par simulations.

On a longtemps utilisé ce seul moyen en pensant que la robustesse du coeur stable vis à vis du changement de conditions initiales testées était suffisante pour garantir une bonne approximation. Sans remettre en cause cette robustesse nous insisterons sur deux exemples significatifs des problèmes soulevés par cette méthode.

Sur un réseau aléatoire à 256 éléments (figure IV-1a) nous calculons une approximation des coeurs par simulations de dynamiques.

Nous fournissons au réseau 100 conditions initiales aléatoires et calculons les cycles limites des dynamiques partant des ces 100 états. Il faut souligner que, bien évidemment, nous n'avons aucune idée, à priori, du nombre de cycles limites ainsi atteints.

Pour chacun des cycles limites nous déterminons la partie oscillante et nous notons pour chaque cellule le nombre de fois qu'elle a appartenu aux parties oscillantes (figure IV-1b).

Donc une cellule notée 0 (resp. 100) est restée stable dans tous les cycles limites calculés : elle fait partie de l'approximation du coeur stable (resp. oscillant) du réseau.

Dans ce cas précis, nous pouvons effectuer les remarques suivantes sur ces approximations :

i) le coeur stable représente 64 % du réseau (164 éléments) et il n'est formé que d'une composante connexe.

ii) le coeur oscillant couvre un quart du réseau (62 éléments) et est constitué de 5 composantes connexes.

L'aspect compact de ces composantes est du à la régularité du graphe de connexion mais nous verrons plus loin que celle-ci n'interdit absolument pas l'existence de coeurs beaucoup "plus éclatés". Elle autorise, par contre, une lecture plus rapide des résultats.

Dans cet exemple, i.e. pour ce réseau et les 100 conditions initiales fournies, il reste 30 cellules qui ont oscillé dans certains cycles limites et qui sont restées stables dans d'autres.

Ces 30 cellules se divisent en 2 groupes :

- 8 qui ont appartenu 49 fois aux parties oscillantes,
- 22 qui ont appartenu 59 fois aux parties oscillantes.

Nous appellerons degré d'oscillation le rapport :

"nombre d'appartenances aux parties oscillantes / nombre de conditions initiales aléatoires fournies"

Remarquons ici qu'avec ce genre de tableau (appartenance aux parties oscillantes) nous ne distinguons dans ce réseau que 4 types de cellules.

Nous avons reproduit la même expérience sur le même réseau, mais en testant 250 conditions initiales (figure IV-1c), puis 1000 (figure IV-1d). Les trois tableaux IV-1b,c et d permettent d'étudier l'évolution de l'approximation des coeurs.

Sur cet exemple on ne remarque en fait aucune évolution ; c'est à dire que les cellules marquées d'un 0 sur le tableau IV-1c le sont également sur les deux tableaux suivants et celles marquées d'un 100 sur le premier tableau "se transforment naturellement" en 250 puis 1000 sur les deux autres.

Ce premier exemple confirme ainsi la robustesse des coeurs stables et oscillants des réseaux aléatoires : il représente en réalité la majorité de ces réseaux. Cette stabilité des approximations vis à vis du nombre de conditions initiales testées nous autorise à ne simuler qu'une centaine de dynamiques (pour des réseaux de cette taille) pour obtenir les deux ensembles déterminant de toutes les dynamiques, donc du réseau : le coeur stable et le coeur oscillant.

Mais la confiance jusqu'alors admise en cette robustesse est mise en défaut par l'existence de réseaux réfractaires à ce type de conclusion. Nous en donnons un exemple en effectuant sur le réseau de la figure IV-2a les mêmes expériences que précédemment. Il faut comparer les tableaux IV-2b,c,d et IV-1b,c,d. Il ressort de cette comparaison deux faits majeurs: le premier portant sur les coeurs, le second sur le reste du réseau.

1) L'évolution des approximations des coeurs :

l'approximation du coeur stable passe de 76 éléments (IV-2b) pour 100 conditions initiales testées à 71 éléments (IV-2d) pour 1000 conditions initiales. De même l'approximation du coeur oscillant évolue de 16 à 9 éléments. Pour ce réseau particulier et au vu de cette évolution nous avons simulé 10^4 dynamiques et les approximations donnent alors un coeur stable de 70 éléments (27.34 % du réseau) et un coeur oscillant de 3 éléments (1.17 %) !

2) La complexité du comportement du reste du réseau :

Alors que sur l'exemple précédent on ne trouvait que 4 degrés d'oscillation distincts (ce nombre ne variant pas avec le nombre de dynamiques testées) ici on a jusqu'à 61 degrés d'oscillation différents (ici ce nombre évolue comme suit : 41 pour 100 conditions initiales, 50 pour 250 C.I., 61 pour 100 C.I. et 66 pour 10000).

Ce type de réseau est assez peu fréquent mais pose néanmoins deux questions :

◇ peut-on faire confiance aux approximations des coeurs ainsi calculées ?

◇ qu'est-ce qui distingue un réseau à dynamique "simple" (le précédent) d'un réseau à dynamique complexe ? En effet, rien sur les deux tableaux IV-1a et IV-2a (représentant les lois du réseau) ne différencie à priori ces deux réseaux.

Nous tenterons de répondre à ces deux questions dans les paragraphes suivants.

2) Approximation du coeur stable par le domaine forçant.

Depuis les premiers travaux sur les réseaux booléens une notion s'est imposée comme outil d'étude privilégié des dynamiques. : les **liaisons forçantes** cf [Kau.72] . Nous avons déjà souligné l'influence de ces liaisons dans la complexité des dynamiques.

F.Fogelman, pour calculer une approximation du coeur stable, rapidement et en évitant de simuler des dynamiques, étend l'utilisation de cette notion en définissant le domaine forçant. cf [Fog.84b].

Définition du **graphe forçant** d'un réseau booléen.

Soit $F : \{0,1\}^n \rightarrow \{0,1\}^n$ un réseau booléen d'interconnectivité $k = 2$

et $G = (X,U)$ son graphe de connexion.

Le **graphe forçant** $h = (Y,V)$ du réseau est construit à l'aide de l'étiquetage des sommets $l : X \rightarrow \{0,1\}$:

soit $j \in X$, les fonctions constantes étant exclues du choix, il peut se présenter les 3 cas suivants :

1) f_j n'est ni une loi forçante, ni une loi de transfert : i.e. $f_j = \text{XOR}$ ou bien

$f_j = \Leftrightarrow$. Dans ce cas $l(j)$ n'est pas défini.

2) f_j est une loi de transfert : elle transfère l'une de ses entrées (+) ou son inverse (-).



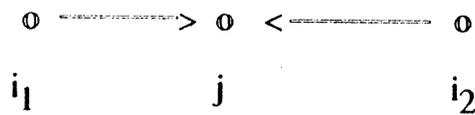
Si l'entrée qu'elle transfère admet une valeur forcée $vf(i)$ alors :

◇ j admettra une valeur forcée $vf(j) = vf(i)$ (ds le cas +)
 $= vf(i)$ (ds le cas -)

◇ on placera j dans Y et (i,j) dans V.

◇ on étiquettera : $l(j) = vf(j)$.

3) f_j est une loi forçante (1^{ère} entrée i_1 , 2^{ème} entrée i_2).



Dans ce cas :

◇ on étiquette j, $l(j) = vf(j)$,

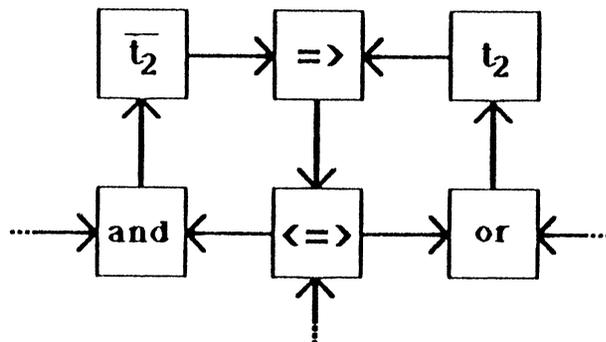
◇ on place j dans Y,

◇ si i_1 (resp. i_2) admet une valeur forcée $vf(i_1)$ (resp. $vf(i_2)$) et que celle-ci est la valeur forçante de f_j pour l'entrée correspondante i.e.

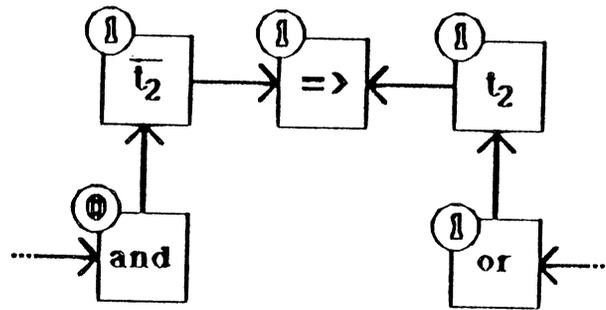
$vf(i_1)=vf_1(j)$ (resp. $vf(i_2)=vf_2(j)$) alors on place (i_1,j) (resp. (i_2,j)) dans V.

Pour obtenir le graphe forçant on itère les procédures d'étiquetage et de construction de Y et V qui ont lieu dans les cas 2) et 3) jusqu'à stabilité.

Exemple : Réseau partiel avec son graphe de connexion



Le même réseau partiel avec son graphe forçant et ses valeurs forcées.



Définition du **Domaine Forçant** d'un réseau :

Le domaine forçant D , sous-ensemble de X est construit comme extension des circuits forçants, i.e. des circuits de H . Cette construction s'accompagne du calcul de la valeur stable de chaque point du domaine D notée $v^* : D \rightarrow \{0,1\}$.

La définition est basée sur les 4 règles suivantes :

R1) $\forall i \in Y$, si i appartient à un circuit forçant alors :

$$i \in D \text{ et } v^*(i) = l(i)$$

R2) $\forall i \in D, \forall j \in D$ si $(i,j) \in V$ et $v^*(i) = vf_i(j)$ alors

$$j \in D \text{ et } v^*(j) = l(j)$$

commentaire : les éléments concernés par cette règle sont les successeurs dans le graphe forçant des éléments du domaine forçant.

R3) $\forall j \in D$ avec (i_1,j) et (i_2,j) appartenant à U , si i_1 et i_2 appartiennent à D alors

$$j \in D \text{ et } v^*(j) = f_j(vf(i_1),vf(i_2)).$$

commentaire : cette dernière valeur $f_j (vf(i_1), vf(i_2))$ est forcément égale à $l(j)$, sinon l'une au moins des deux valeurs $vf(i_1)$ ou $vf(i_2)$ aurait été forçante pour f_j et c'est la règle R2) qui s'appliquait. La règle R3) intervient pour les automates dont les deux entrées appartiennent déjà au domaine D.

R4) $\forall i \in D \quad \forall j \in D$ si $(i,j) \in U$ et $v^*(i) = vf_i(j)$ alors

$$j \in d \text{ et } v^*(j) = vf(j).$$

commentaire : ce cas résulte de la création d'un arc forçant n'appartenant pas au graphe forçant mais apparaissant lors de la construction de D lorsque qu'une valeur stable $v^*(i)$ du domaine est différente de la valeur forcée $vf(i)$. Cette règle pourrait aisément fusionner avec R2 mais elles correspondent à deux niveaux dans la construction de D.

Comme pour la définition du graphe forçant les trois règles R2, R3 et R4 sont appliquées jusqu'à stabilisation du domaine D.

Exemple. La figure IV-3 représente un réseau avec son graphe forçant et le domaine forçant y apparait en gras. Pour illustrer les 4 règles de construction on note :

1) les 10 circuits forçants dont un qui a la particularité d'occuper "les 4 coins du tore".

2) le prolongement de ces circuits suivant le graphe forçant.

3) Dans la portion agrandie, on remarque que l'automate régi par la loi n°13 appartient au domaine car ses deux entrées (lois n°5 et n°14) sont elles mêmes dans D. Cet automate admet comme valeur stable 0 et non pas sa valeur forcée 1. (cf remarque de R3)

4) Ce qui précède entraine l'automate régi par la loi n°10 dans D avec comme valeur stable 0, et non pas 1 qui est la valeur donnée par l'étiquetage l. Ce phénomène crée l'arc forçant de 10 à 8 et fait donc basculer l'automate régi par la loi n°8 dans D avec pour valeur stable sa valeur forcée.

F.Fogelman a mis en évidence les qualités du domaine forçant comme approximation du coeur stable, en calculant la probabilité de stabilité d'un circuit forçant cf [Fog.84b]. Cette probabilité s'exprime sous la forme d'une série dont la

complexité des termes rend difficile le calcul effectif pour un réseau donné. Elle a également énoncé la condition d'oscillation d'un circuit forçant [Fog.84a].

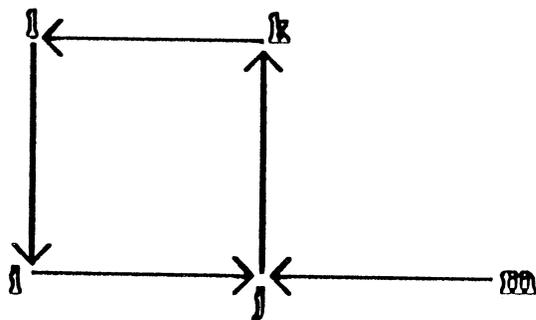
Un circuit forçant C peut osciller (sans prendre évidemment sa valeur stable) et alors pour tout temps t du cycle limite on a :

$$\forall i, j \in C \text{ et } (i, j) \in U$$

$$\text{si } v_i^* \neq x_i(t)$$

$$\text{alors } \exists k \in X, k \neq i \text{ tel que : } (k, j) \in U = x_k(t) \neq v(k, j).$$

Pour visualiser cette condition plaçons nous dans le cas d'un graphe de connexion régulier :



A chaque temps du cycle, si "i" (par exemple) n'est pas dans sa valeur stable il faut que l'autre entrée de "j" i.e. "m" ne puisse forcer "j" à prendre sa valeur stable : c'est donc une sorte de synchronisation entre "i" et m qui permet l'oscillation de "j".

Il est clair que, ce cas devant se reproduire pour tous les éléments du cycle et non pas seulement pour "j", cette condition est très restrictive. Nous analyserons par la suite ce qui peut favoriser un tel évènement.

Nous donnons à la figure IV-4 un exemple montrant les qualités d'approximation du coeur stable par le domaine forçant. C'est le réseau de la figure IV-1a dont on avait simulé 100, 250 et 1000 dynamiques (IV-1b,c,d). On peut comparer les tableaux IV-1d et IV-4 : on remarque que les 14 circuits forçants du réseau sont stables et que le domaine D est donc inclus dans le coeur stable. On note quand même que certains points du coeur stable n'appartiennent pas à D.

Cependant sur certains réseaux cette approximation soulève des problèmes plus importants : la figure IV-5 représente le réseau IV-2a avec son graphe et son domaine forçant. Ici la comparaison des tableaux IV-5 et IV-2d met en évidence les lacunes suivantes :

a) comme dans l'exemple précédent, l'extension des circuits forçants ne parait pas suffisante d'où l'existence de points appartenant au coeur stable sans être dans D.

b) ici 4 circuits forçants oscillent !

c) il existe des points isolés dans le coeur stable que le domaine forçant ne peut contenir par construction.

Même si dans ce contexte ce réseau n'est pas représentatif (c'est un cas extrême parmi de très nombreux tirages aléatoires) il a l'avantage de regrouper les trois principales lacunes du domaine forçant que nous allons maintenant détailler.

A) Eléments du coeur stable n'appartenant pas au domaine forçant.

1) *à la frontière du domaine forçant.*

Sont en fait transformées en lois de transfert les lois de tous les automates ayant une entrée dans le domaine forçant sans y appartenir eux-mêmes. En effet, dans le cas contraire, l'automate prendrait une valeur stable : il appartiendrait donc au domaine forçant. A ce sujet on peut voir sur la table des lois (figure III-1) un signe (+ ou -) dans les cases réservées aux valeurs forçantes : ce signe est celui de la loi de transfert créée par une entrée constante dans la valeur non forcée.

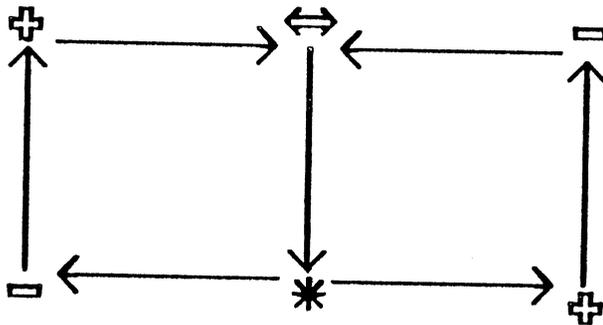
Aux frontières de D apparaissent donc des lois de tranfert qui peuvent favoriser une stabilisation durant le cycle limite de deux manières :

i) en recréant des circuits forçants . (fig. IV-6a)

ii) en imposant des valeurs stables à des points isolés.
(voir plus loin le détail de tels mécanismes)

2) *existence de points isolés dans le coeur stable.*

Il est clair que si l'on montre cette existence, les points en question ne pourront être repérés dans le domaine forçant. On voit déjà apparaître ce type de points dans les exemples de F. Fogelman. Pour se convaincre de cette existence fournissons l'exemple le plus simple possible :



Quelque soit la loi $*$, la loi \neq aura toujours ses deux entrées identiques : l'automate sera donc dans l'état stable 1. Son appartenance au coeur stable ne doit rien à un circuit forçant.

De plus, comme pour les circuits forçants, tous les successeurs sur le graphe forçant de tels points appartiendront également au coeur stable. Cette remarque souligne que l'erreur par défaut (provenant de ces points) d'approximation du coeur stable par le domaine forçant peut être importante.

Sur cet exemple le rôle des lois de transfert paraît capital : c'est en effet souvent le cas, car, en simplifiant le graphe de connexion, elles ont tendance à lier les entrées d'un automate.

Aux figures IV6a et IV6b nous fournissons deux exemples de points toujours stables qui ne sont en rien liés à des circuits forçants.

Dans le premier cas c'est un environnement stable qui en créant des lois de transfert fait apparaître un schéma stabilisateur.

Le second cas est beaucoup plus complexe : il met en évidence un type de stabilisation pratiquement indétectable. On suppose en effet qu'un automate prenne sa valeur non forcée puis on "remonte dans le temps" le graphe de connexion jusqu'à l'obtention d'une impossibilité : i.e. qu'à un top donné un automate devrait fournir au réseau deux sorties différentes. Ce cas se distingue d'une stabilisation originaire d'un circuit forçant par plusieurs points:

i) la difficulté d'automatiser une telle recherche. Avec un graphe de connexion aléatoire le "circuit" qui conduit à l'impossibilité peut être très long et dans tous les cas l'existence d'un tel circuit est imprévisible. De plus les arguments qui permettent de remonter le graphe de connexion sont très variés et il faudrait certainement employer des primitives logiques pour automatiser cette construction.

ii) la certitude d'appartenance au coeur stable. Contrairement aux circuits forçants qui possèdent une forte probabilité d'être toujours stable ici on montre qu'un automate ne peut pas prendre une certaine valeur : on démontre donc l'appartenance de ce dernier au coeur stable.

B) Eléments du domaine oscillant n'appartenant pas au coeur stable.

De la définition du domaine forçant, on retient que la seule cause possible d'oscillation d'un des ses points est l'oscillation d'un circuit forçant. En effet les règles d'extension R1, R2 et R3 ne peuvent adjoindre au domaine des éléments oscillants si leurs origines sont stables.

Une fois cette cause identifiée, une remarque et une question s'imposent :

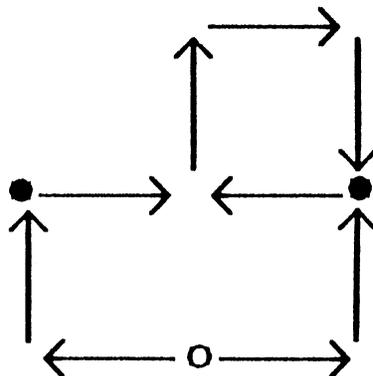
i) il est impossible de majorer l'erreur d'approximation du coeur stable par le domaine forçant : si un circuit oscille, toute la composante du domaine qui lui est rattachée peut osciller et on ne peut pas évaluer la taille de ces composantes.

ii) Qu'est ce qui peut favoriser la condition d'oscillation d'un circuit forçant?

Au regard de la condition Co on note deux sources d'oscillation possibles :

1) la synchronisation entre les automates qui constituent le circuit et ceux qui sont les entrées de ce circuit. Cet évènement (synchronisation pendant toute la durée du cycle) possède une très faible probabilité de se produire cf [Fog.84b].

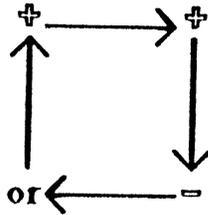
Il est néanmoins certain qu'un graphe de connexion régulier la favorise comme le montre la figure suivante :



Les deux entrées (cellules noires) d'une cellule du circuit forçant reçoivent avec un même retard un message provenant de la cellule blanche. Ce proche lien peut fort jouer le rôle de synchronisateur.

Cette remarque faite, on signale que les erreurs par excès figurant dans les exemples précédents ne sont jamais dues à ce phénomène.

2) l'importance du nombre de lois de transfert dans un circuit.
Soit le circuit forçant suivant :



La condition d'oscillation d'un circuit forçant :

pour tout temps t du cycle limite on a :

$$\forall i, j \in C \text{ et } (i, j) \in U$$

$$\text{si } v_i^* \neq x_i(t)$$

$$\text{alors } \exists k \in X, k \neq i \text{ tel que : } (k, j) \in U \Rightarrow x_k(t) \neq v(k, j)$$

s'écrit pour ce circuit :

pour tout temps t du cycle limite on a :

$$\forall i \in C \text{ et } (i, j) \in U$$

$$\text{si } v_i^* \neq x_i(t)$$

$$\text{alors } \exists k \in X, k \neq i \text{ tel que : } (k, j) \in U \Rightarrow x_k(t) \neq v(k, j)$$

la condition qui était restrictive quand elle s'appliquait à toutes les cellules du circuit n'est plus ici une contrainte car elle ne joue que sur une cellule. Si, de plus, celle-ci n'a pas d'entrée forçante, rien ne donne l'impulsion nécessaire pour mettre le circuit dans sa valeur stable.

Par exemple, les 4 circuits forçants qui n'appartiennent pas au coeur stable dans le réseau IV-2a sont formés de 3 lois de transfert et d'une loi n'ayant pas d'entrée forçante hors du circuit.

3) Définition d'un nouveau domaine approchant le coeur stable

La définition que nous proposons est une modification de celle du domaine forçant qui tient compte des constatations précédentes. La définition est constructive :

a) on recherche dans le graphe forçant du réseau les circuits qui

possèdent au moins une loi avec une entrée forçante
ou
deux lois qui ne soient pas de transfert

b) On étend ces circuits à l'aide des 3 règles R1, R2 et R3 de la définition du domaine forçant.

c) On effectue la transformation (en lois de transfert) des lois en contact avec le domaine déjà calculé en lois.

On itère ces 3 étapes jusqu'à stabilisation du domaine.

Remarques

◇ Les différences avec le domaine forçant sont :

- l'élimination des circuits à risques (restriction)
- la transformation des lois (extension)

◇ On peut observer sur les exemples précédents les améliorations apportées par ces modifications.

◇ Avec cette définition il est toujours impossible de détecter des points isolés qui appartiendraient au coeur stable (la structure de base est encore un circuit).

Pour étudier la dynamique du reste du réseau (les éléments n'appartenant pas à la partie du coeur stable "issue" de circuits forçants) nous disposons maintenant d'une information importante qui nous permet d'étudier un réseau partiel dont les entrées sont des constantes.

4) Etude d'un réseau partiel aux entrées constantes

La première composante de ce réseau que l'on cherchera à calculer est bien

évidemment le coeur oscillant en employant une technique analogue à celle du domaine forçant.

a) le coeur oscillant.

Comme pour le coeur stable il convient trouver la structure de base de cet ensemble de cellules qui oscillent dans tous les cycles limites. Ici cette structure est le circuit frustré.

Définition : Un **circuit frustré** est un circuit du graphe de connexion composé uniquement de lois de transfert avec imparité du nombre de lois d'inversion.

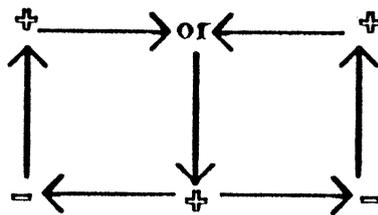
Quelques soient les conditions initiales, toutes les cellules appartenant à de tels circuits ne peuvent qu'osciller dans le cycle limite. cf [Fog.85].

Dans un réseau aléatoire, l'apparition de ces circuits est exceptionnelle mais, du fait des transformations se produisant au bord du coeur stable nous observons dans tous les réseaux où le coeur stable occupe au moins 40% des cellules la création de plusieurs circuits frustrés. (voir figures.IV-7a,b).

Partant de tels circuits le coeur oscillant se prolonge par les lois de transfert, qu'elles proviennent du tirage aléatoire ou de transformation.

Mais comme pour le coeur stable il n'y a pas unicité de cause à l'appartenance au coeur oscillant.

Donnons un exemple simple de cellule toujours oscillante qui n'a aucun lien avec un circuit frustré :



La cellule régie par la loi or ne peut prendre une valeur stable.

Un exemple plus complet est fourni dans la figure IV-7c. Pour détecter ce genre d'appartenance on emploie une technique d'itération formelle, i.e. on n'itère plus des valeurs booléennes mais des variables symboliques. Après quelques itérations, on étudie la possibilité que possède chaque cellule d'être stable. On est dès lors conduit à résoudre un système d'équations booléennes si celui-ci n'a pas de solution

les cellules en question oscillent. Cette méthode ne peut être employée que sur une petite composante, sinon rien ne garantit la possibilité de simplifier les expressions booléennes ou de résoudre le système. Pour une utilisation automatique il faudrait disposer d'un langage de calcul formel d'expressions booléennes.

b) le reste du réseau : les paliers.

Tout d'abord nous rappelons que dans ce reste du réseau il se trouve des points du coeur stable (points isolés à détecter par la technique exposée dans l'exemple IV-8) et des points du coeur oscillant (à détecter par itération formelle).

À l'exclusion de ces deux cas particuliers, le réseau partiel que nous considérons maintenant est constitué de cellules qui oscillent dans certains cycles limites et sont stables dans d'autres.

Nous avons défini précédemment le degré d'oscillation d'un automate : cette définition uniquement basée sur un résultat de simulations correspond en fait à une probabilité d'osciller propre à une cellule ou à un groupe de cellules.

Cette probabilité est impossible à calculer théoriquement, sauf dans des cas très simples; nous sommes donc contraints à calculer des degrés d'oscillation pour étudier le comportement dynamique du réseau.

Les cas très simples évoqués ci-dessus comprennent naturellement les probabilités 0 (coeur oscillant) et 1 (coeur stable).

Un autre cas simple, les circuits non frustrés (circuit composé de lois de transfert dont un nombre pair de lois inversante), permet d'observer la correspondance degré d'oscillation/probabilité d'oscillation.

Un circuit non frustré composé de n cellules a une probabilité de $1/2^{(n-1)}$ d'être stable (on notera qu'avec une itération série cette probabilité est de 1). Sur toutes les simulations effectuées qui comportent un circuit frustré de 4 éléments les degrés d'oscillation prennent des valeurs voisines de $7/8$ (0.871, 0.880, 0.879, 0.875). L'écart avec la valeur théorique s'explique de la manière suivante : la génération de hasard sur les conditions aléatoires ne permet pas d'atteindre "rapidement" tous les cycles limites (l'uniformité du hasard est biaisée par l'évolution du réseau durant le transitoire).

Hors ces cas simples c'est donc sur les simulations qu'il faut s'appuyer. En dehors des coeurs le réseau est structuré en paliers :

ensemble de cellules ayant un même degré d'oscillation calculé sur toutes les conditions initiales possibles.

Comme pour les coeurs, cette définition n'autorise que des approximations, qui seront ici beaucoup plus complexes à calculer. Ces paliers sont néanmoins, au même

titre que les coeurs, caractéristiques du réseau.

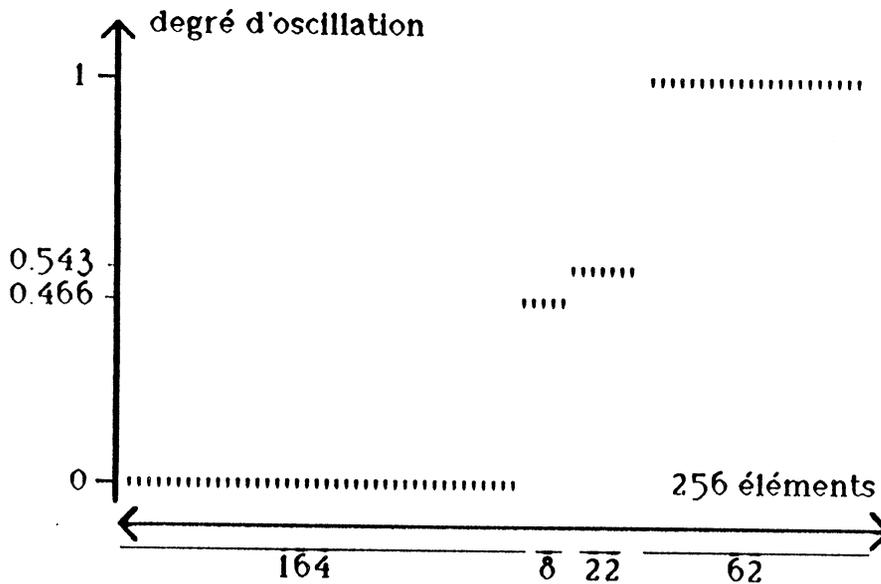
Ces différentes composantes de la dynamique des réseaux aléatoires permettent d'établir la partition suivante dans l'ensemble des réseaux.

a) *Les réseaux à dynamique simple*

Caractéristiques :

- 1) Un coeur stable important (occupant plus de 50% des cellules).
- 2) Un petit nombre de paliers (moins de 20).
- 3) Une détermination des paliers rapides (100 simulations).

Sur un exemple de ce type de réseau (IV-1a) on peut visualiser les différentes composantes de la dynamique à l'aide du schéma suivant :



b) *les réseaux à dynamique complexe :*

Caractéristiques :

- 1) Un coeur stable peu important.

- 2) Un grand nombre de paliers.
- 3) Une détection de nouveaux paliers possibles même après plusieurs milliers de simulations.

Il n'est plus question ici de tracer le schéma précédent; nous établissons par contre un histogramme des paliers : i.e. nous précisons pour un réseau donné le nombre N de simulations effectuées et le nombre de cellules qui ont oscillé n fois. (figure IV-10 et IV-11a,b,c,d)

On s'aperçoit que pour le réseau à dynamique complexe qu'un palier peut se scinder pour former deux nouveaux paliers possédant des degrés d'oscillation voisins. Ce phénomène se poursuit même après la simulation de 5000 dynamiques (réseau à 256 éléments). Le nombre de paliers restreint à une seule cellule est important (une trentaine). Ces paliers différencient, en fait, les cycles limites du réseau : le graphe d'itération du réseau (que l'on se garde de tracer) a nécessairement des bassins d'attraction proches, i.e. le comportement d'une cellule peut distinguer un cycle limite d'un autre.

Le nombre de paliers Nb_p d'un réseau est trivialement une borne inférieure du nombre de cycles limites. En assimilant deux cycles si ils possèdent les mêmes parties stables et oscillantes (on ne s'intéresse pas à la manière d'osciller), on a comme borne supérieure à $2^{Nb_p} + Nb_p$. Cette dernière borne est évidemment très large car elle sous-entend que toutes les combinaisons de paliers sont possibles alors que les degrés d'oscillations interdisent en pratique certaines associations.

Dans ce paragraphe nous avons souvent fait référence à 2 réseaux particuliers (IV-1a et IV-2a) il est bien évident que ce sont deux représentants des deux classes de réseaux. Pour mieux dégager les caractéristiques de ces deux classes les exemples ont été choisis parmi de très nombreuses simulations en raison de leur représentativité.

V APPLICATION : CLASSIFICATION ET RECONNAISSANCE DE SEQUENCES BOOLEENNES.

Dès les premiers travaux sur les réseaux booléens on a étudié la robustesse de leurs dynamiques vis-à-vis de certaines perturbations. S. Kauffman [Kau.70] choisit, par exemple, de bruite la dynamique de ses réseaux en inversant l'état d'une cellule pendant 1 temps d'itération : il note que ce type de perturbation n'entraîne aucune modification dans l'attracteur atteint.

H. Atlan et al se sont également intéressés au bruit que l'on peut envoyer sur un réseau [Atl.81,85]. Pour que ce bruit ait un effet mesurable il doit être appliqué de façon durable. La définition utilisée ici est celle qu'ils ont adoptée :

On dira que l'on bruite un réseau lorsque l'on impose à une cellule une suite périodique d'états.

Par exemple, si on bruite un réseau sur une cellule avec la séquence 0100, cette cellule prendra successivement les états 0100 0100 0100 au lieu de prendre à chaque itération du réseau l'état obtenu en faisant fonctionner sa loi sur ces deux entrées.

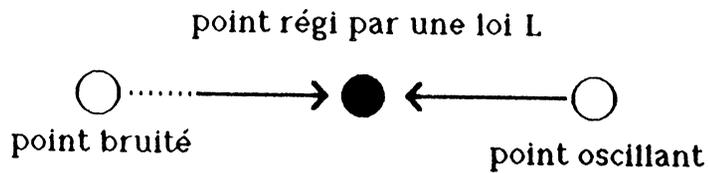
L'application d'un tel bruit sur un réseau peut se faire à différents temps de la dynamique : soit au début, i.e. le bruit a déjà un effet dans le transitoire, soit une fois que le réseau a atteint son cycle. Dans tous les cas la taille de la séquence est choisie comme diviseur de la période du cycle limite afin de pouvoir atteindre de nouveaux cycles.

En comparant les 2 cycles (présence et absence de perturbation) on observe que le partage du réseau en partie stable et partie oscillantes est distinct. L'envoi de la séquence a donc stabilisé certains points et en a déstabilisé d'autres.

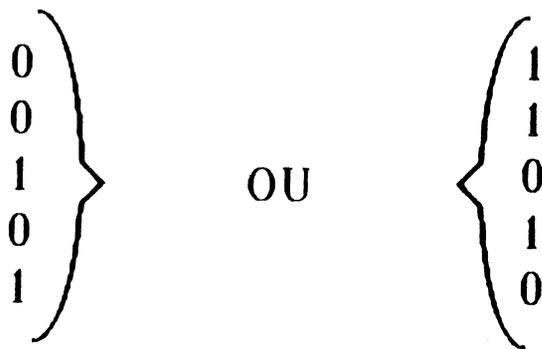
Afin d'étudier les effets d'un tel bruit nous avons établi une **classification des stabilisations**.

1) *Stabilisation indirecte ou par résonance.*

Pour que la cellule bruite se stabilise il faut que le bruit (ou ce qu'il en reste après son trajet dans le réseau) entre en "résonance" avec l'oscillation de l'autre entrée de la cellule. Cette résonance doit être comprise au sens de la loi qui régit l'automate : les 2 entrées simultanées (bruit et oscillation) doivent maintenir une sortie constante à la loi L.



Exemple : Sur le schéma suivant la loi OU avec 2 entrées oscillantes a sur les 5 temps considérés, une sortie constante de valeur 1.



Cette stabilisation a souvent comme origine une alternance dans le rôle forçant des entrées. (c'est le cas dans le schéma précédent)

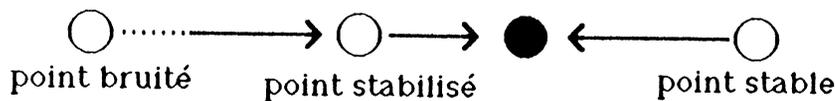
Il y a néanmoins d'autres sources de stabilisation par résonance qui occasionnent des stabilisations dans des valeurs non forcées ou pour des lois non forçantes (xor, ≠).

La nécessaire synchronisation qui doit s'opérer dans ces stabilisations les rend invisibles sur un graphe de connexion ou forçant.

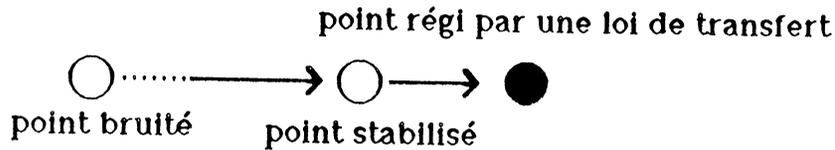
2) *Stabilisation directe.*

Une fois qu'une cellule est stabilisée (par résonance ou par une des deux séquences constantes) la stabilisation se propage directement en suivant un des trois schéma suivants :

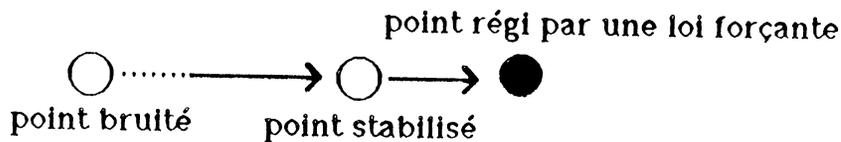
a) **Environnement stable :**



b) Loi de transfert :



c) Loi forçante :



Sur les figures V-1a,1b on observe les effets d'un bruit sur un cycle limite du réseau IV-2a : on note que les stabilisations peuvent se situer loin du point perturbé.

Remarque : Si la première cellule stabilisée par l'envoi d'une séquence l'est obligatoirement par résonance, les suivantes le sont par une combinaison des 2 types : direct et indirect.

Partant de cet effet stabilisateur on a défini la reconnaissance de séquences booléennes [Atl.85] comme suit :

Un séquence de taille k est reconnue par un réseau si l'ensemble des cellules qu'elle stabilise la différence des autres séquences de tailles k .

Si par exemple, (0010) et (0011) (et elles seules) stabilisent le même ensemble de cellules on dira que (01*0) est reconnue par le réseau.

La constatation principale qui s'impose alors intéresse les ensemble de cellules qui reconnaissent les séquences :

ces ensembles sont constitués d'une union de paliers.

Cette observation issue de toutes les simulations effectuées s'explique aisément par la définition des paliers : ceux-ci regroupent des cellules qui ont un comportement commun vis-à-vis des entrées que le réseau leur fournit.

On visualise très aisément cette remarque sur les deux exemples de réseaux précédemment étudiés.

le réseau IV-1a (dynamique simple, 2 paliers) offre une très faible capacité de reconnaissance. figure V-2a,b.

le réseau IV-2a (dynamique complexe, plus de 70 paliers) a une très grande capacité de reconnaissance et possède de plus une importante diversité de points que l' on peut perturber . figure V-3.

On note en outre que l'application du bruit dévie l'itération vers un nouveau bassin d'attraction : i.e. si on ne perturbe plus la dynamique après l'obtention d'un cycle limite celle-ci se maintient dans le même cycle limite qui est un attracteur du réseau non bruité. Cette observation (à moduler pour les points voisins immédiats du point bruité) est délicate à justifier entièrement.

Si la reconnaissance ne pose plus de problème de compréhension, la prévision achoppe toujours sur plusieurs points. Provenant de la différenciation des réceptions à une perturbation extérieure et donc principalement de la stabilisation par résonance, la reconnaissance a une partie de son origine dans le transitoire : époque de la dynamique la plus difficile à étudier.

Afin d'améliorer les capacités de reconnaissances de ces réseaux nous sommes alors tentés de réduire les facteurs de stabilisations directes : lois de transfert par exemple. Mais celles-ci en intervenant dans le transport et le retardement du signal favorise la résonance "plus loin". C'est la difficulté majeure de cette étude : comment séparer dans la pratique ces deux types de stabilisation ?

Nous avons quand même simulé ces opérations pour des réseaux avec des pourcentages de lois non forçantes élevés : le seul résultat probant est l'augmentation de la taille des périodes et donc de la difficulté d'observer les phénomènes de reconnaissance.

L'étude précédente fournit néanmoins de précieuses indications sur les capacités de reconnaissance des réseaux ainsi que sur les lieux du réseau favorables à la différenciation de réception des messages.

Quant aux possibilités d'apprentissage, largement étudié dans le cadre des réseaux d'automates à seuil ([Fuk.78], [Hin.81], [Hop.82], [Hub.84]), elles sont ici plus difficiles à mettre en évidence. Rappelons en effet, que toutes les fonctions forçantes sont des fonctions à seuil; nous pourrions donc leur appliquer une règle de type Little-Hopfield (cf. [Fog.85b]). Mais les fonctions non forçantes ne peuvent être mise sous la forme de fonctions à seuil et nous avons souligné l'importance de leur rôle dans la reconnaissance des séquences.

VI CONCLUSION

Nous avons montré la difficulté de prévoir la dynamique des réseaux booléens aléatoires d'interconnectivité 2. Avec une interconnectivité supérieure l'analyse du cœur stable se complique très rapidement.

L'extraordinaire complexité des réseaux booléens produite par une combinatoire explosive rend, à notre sens, impossible la mise au point mathématique permettant, sans recours aux simulations, une étude précise de ces réseaux.

Il conviendrait, pour mettre au point une méthode d'analyse pertinente, de combiner les moyens disponibles suivant la démarche suivante :

Commencer par *simuler* une centaine de dynamiques pour dégager l'essentiel des composantes du réseau.

Puis, *confirmer ou infirmer*, à l'aide des outils décrits au paragraphe IV, les conclusions relatives au cœur stable.

Enfin *étudier le reste du réseau* après transformation en lois de transfert des lois situées en périphérie du cœur stable. Au besoin, l'on utilisera des techniques d'itérations formelles sur de petites composantes.

L'utilisation de réseaux booléens aléatoires comme moyen de reconnaissance est loin d'être efficace si on la compare, par exemple au techniques d'automates à seuils. Il ne faudrait pas pour autant négliger cette approche car on ne retrouve pas dans les réseaux à seuils la possibilité d'employer des fonctions booléennes non forçantes. Actuellement cette utilisation rentre plutôt dans le cadre d'une modélisation de phénomènes biologiques.

REFERENCES

- [Atl.81] H. ATLAN, F. FOGELMAN, J. SALOMON, G. WEISBUCH : "Random Boolean Networks." *Cybernetics and Systems* 12, 103-121.
- [Atl.85] H. ATLAN, E. BEN-EZRA, F. FOGELMAN-SOULIE, D. PELLEGRIN, G. WEISBUCH : "Recognition of sequences by random networks." (en préparation)
- [Fog.82] F. FOGELMAN-SOULIE, E. GOLES, G. WEISBUCH : " Specific roles of the different boolean mappings in random networks." *Bull. Math. Biology* 44, 715-730.
- [Fog.84a] F. FOGELMAN-SOULIE : "Frustration and stability in random boolean networks." *Discrete Applied Mathematics*, 9, 139-156.
- [Fog.84b] F. FOGELMAN-SOULIE : " Parallel and sequential computation on boolean networks." *Theoretical Computer Science*.
- [Fog.85] F. FOGELMAN-SOULIE : " Contributions à une théorie du calcul sur réseaux." Thèse d'état, Grenoble 1985.
- [Fuk.78] K. FUKUSHIMA, S. MIKAYE : "A self organizing neural network with a function of association memory feed back type cognitron." *Biological Cybernetics*, 28, 4, pp201-208.
- [Gel.77] A. E. GELFAND, C. C. WALKER : "The Distribution of Cycle Lengths in a Class of Abstract Systems." *Int. J. Gen. Systems* 4, 39-45.
- [Hin.81] G. E. HINTON, J. A. ANDERSON (Eds) : "Parallel models of associative memory." Hillsdale NJ : Erlbaum.
- [Hop.82] J. J. HOPFIELD : "Neural networks and physical systems with emergent collective computational abilities." *Proc. Nat. Acad. Sci. USA*, vol 79, pp2554-2558.
- [Hog.84] T. HOG, B. A. HUBERMAN : "Understanding biological computation : reliable learning and recognition. *Proc. Nat. Acad. Sci. USA*, Nov. 84
- [Kau.69] S. KAUFFMAN : "Metabolic stability and epogenesis in randomly constructed genetic nets." *Theoretical Biology*, 22, pp437-467.

- [Kau.70] S. KAUFFMAN : " Behaviour of randomly constructed nets".
In Towards a theoretical biology Ed. C.H. Waddington, vol3,
Edinburg University Press
- [Kau.72] S. KAUFFMAN : "The organization of cellular genetic control
systems". Lecture on Mathematics in the Life Sciences, Vol 3, American
Mathematical society, Providence, RI, 63_116.
- [Kau.79] S. KAUFFMAN : " Assessing the probable regulatory structures and
dynamics of metazoan genome. Kinetic logic ". In Lecture notes for
Biomathematics. ED. R. Thomas, 29, 30-61. Berlin Springer Verlag.
- [Mac.43] W.MACCULLOCH, W. PITTS : "A logical calculus of the ideas
immanent in nervous activity". Bull. Math. Biophys., 5:113-115.
- [Pel.85] D.PELLEGRIN : "Dynamics of random boolean networks."
Nato advanced workshop on disordered systems and biological
organization, Les Houches, Fév. 1985, à paraître chez Springer Verlag.
- [Pel.85] D. PELLEGRIN : "On learning abilities of random boolean networks.
Actes du Colloque COGNITIVA : De l'Intelligence Artificielle aux
Biosciences, Paris, juin 1985.
- [Rob.86] F.ROBERT : "Discrete iterations" Springer Verlag, à paraître.
- [Rob.76] F.ROBERT : " Comparaison des modes opératoires d'un automate
cellulaire fini." Rapport de recherche IMAG n°31, Fév. 76.
- [Rub.54] H.RUBINAND, R.SITEGREAVE : "Probability Distributions Related to
Random Transformation on a Finite Set."
Tech. Report 19a, Appl. Math. and Stats. Stanford University.
- [Slo.54] N.J.H.SLONE : "Length of Cycle Time in Random Neural Networks."
Cornell University Press, Ithaca.
- [Tho.79] R.THOMAS Ed (1979) : " Kinetic logic, a boolean approach to the
analysis of complex regulatory systems."
Lecture Notes in Biomathematics, Vol. 29, Springer Verlag.

CODE	NOM	TABLE	EXPRE.	VaFo	VF n°1	VF n°2
0	CONT	0 0 0 0	0			
1	NOR	1 0 0 0	$\overline{x_1 + x_2}$	0	1 ₋	1 ₋
2	\Rightarrow	0 0 1 0	$x_1 \overline{x_2}$	0	0 ₋	1 ₊
3	t_2	1 0 1 0	$\overline{x_2}$			
4	\Leftarrow	0 1 0 0	$\overline{x_1} x_2$	0	1 ₊	0 ₋
5	t_1	1 1 0 0	$\overline{x_1}$			
6	XOR	0 1 1 0	$(x_1 + x_2) \overline{x_1 x_2}$			
7	NAND	1 1 1 0	$\overline{x_1 x_2}$	1	0 ₋	0 ₋
8	AND	0 0 0 1	$x_1 x_2$	0	0 ₊	0 ₊
9	\Leftrightarrow	1 0 0 1	$x_1 + x_2 + x_1 x_2$			
10	t_1	0 0 1 1	x_1			
11	\Leftarrow	1 0 1 1	$\overline{x_2} + x_1 x_2$	1	1 ₋	0 ₊
12	t_2	0 1 0 1	x_2			
13	\Rightarrow	1 1 0 1	$\overline{x_1} + x_1 x_2$	1	0 ₊	1 ₋
14	OR	0 1 1 1	$x_1 + x_2$	1	1 ₊	1 ₊
15	TAUT	1 1 1 1	1			

numéro de l'entrée :

1	2	0	1
0			
1			

Table d'une loi.

signe de la loi de transfert associée.

Figure III-1 : Table des lois booléennes à deux variables.

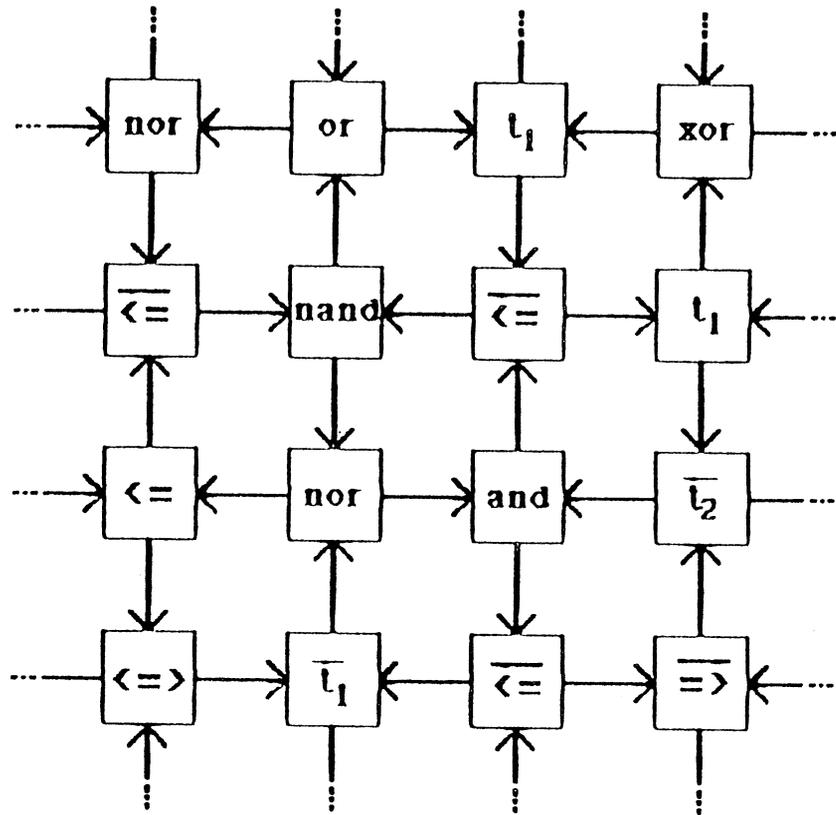


figure III-2

Exemple de réseau partiel avec le graphe de connexion régulier.

figure III-3 : Réseau à 256 éléments où les lois apparaissent sous le code de la table des lois.

3	8	11	4	13	3	3	2	6	2	14	2	14	9	6	14
10	8	9	9	6	4	12	9	7	12	11	5	11	1	2	11
8	6	2	9	3	13	2	10	14	14	1	9	4	4	2	1
10	9	1	7	14	3	13	7	5	1	3	7	5	3	8	6
3	8	9	11	4	4	9	6	6	13	13	10	5	8	9	4
12	1	8	2	5	13	6	6	8	10	13	1	9	3	3	8
3	9	9	1	2	9	12	11	5	3	13	10	2	2	1	4
13	2	2	5	4	10	3	9	7	1	7	6	5	1	6	4
1	10	5	1	14	8	10	2	2	13	7	10	5	3	7	1
8	4	11	14	13	13	7	4	5	7	2	11	11	10	9	2
10	3	12	7	5	10	8	11	5	13	11	5	8	9	2	2
8	3	3	1	10	11	5	5	5	13	2	13	1	4	6	1
7	10	1	4	13	8	5	12	7	12	13	14	14	8	6	5
6	14	14	14	1	10	4	11	11	2	4	5	1	11	5	5
6	4	11	7	9	13	11	10	1	4	11	12	7	14	11	1
1	14	8	14	14	3	8	5	12	12	9	2	5	6	4	12

figure III-4a :
conditions initiales.

0	1	1	0	1	0	0	1	1	0	1	1	1	0	0	1
1	0	1	1	0	1	1	1	0	0	1	1	0	0	1	0
0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	0
1	0	1	0	0	1	1	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0	1	1	1	0	0	1	1
1	0	1	0	1	1	0	1	0	0	0	0	1	1	0	1
0	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1
0	1	0	0	0	1	1	1	0	0	0	0	1	1	1	0
0	1	0	1	1	0	1	1	1	1	1	1	1	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1
0	1	1	1	0	1	1	1	0	0	0	0	1	0	1	0
1	1	1	0	1	1	1	1	1	0	1	0	1	1	0	0
0	1	0	1	1	1	1	0	1	0	1	0	0	0	0	1
0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1

figure III-4b :
"etat" du réseau dans le cycle.

0	0	0	1	*	*	0	0	1	1	0	1	1	0	0	1
*	*	*	0	*	*	*	*	1	1	1	1	0	*	*	0
*	*	*	*	*	*	*	0	0	0	*	*	*	*	*	1
*	*	*	*	*	*	*	1	0	0	*	*	*	*	*	*
*	1	*	*	*	*	*	*	0	1	*	*	0	0	*	*
*	*	0	1	*	*	*	*	*	0	*	*	*	0	*	*
*	*	*	0	*	*	*	*	0	*	*	*	1	0	0	*
*	1	*	0	1	*	*	*	*	*	*	*	1	1	*	*
*	*	*	1	0	*	*	*	*	*	*	*	1	1	*	*
1	*	*	*	*	*	0	1	*	*	*	0	1	1	*	*
0	0	*	*	*	*	*	*	*	*	0	0	0	0	1	1
1	0	*	*	*	0	*	*	*	1	*	0	0	0	0	0
1	0	0	0	1	*	*	1	1	*	*	*	1	1	0	1
0	1	1	1	1	*	*	0	*	*	*	1	1	0	0	0
1	1	0	1	0	*	1	0	0	1	*	*	1	1	1	1
1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1

Les cellules stables sont représentées par leurs valeurs (0 ou 1), celles qui oscillent sont marquées d'une croix.

figure IV-1a : Réseau à 256 éléments.

1	11	1	13	6	8	14	14	3	4	6	13	6	9	13	9
13	2	10	4	8	11	6	8	11	3	7	9	11	12	2	13
6	8	4	9	10	6	8	1	10	13	11	2	5	6	12	2
8	10	3	4	4	9	6	5	11	3	9	1	12	6	8	7
9	7	13	12	1	4	5	2	9	8	5	1	7	1	5	13
4	2	1	8	11	11	2	12	12	8	14	12	8	4	5	3
5	9	8	12	9	4	6	11	1	4	3	10	14	13	5	12
12	6	2	8	8	5	6	13	3	1	8	13	7	6	9	13
3	7	12	10	3	1	10	10	9	6	10	5	5	14	9	1
3	3	1	8	2	11	10	3	5	11	10	8	5	6	10	8
11	6	2	8	4	6	9	1	5	2	12	11	4	3	7	13
4	13	10	1	11	14	14	2	7	12	4	9	6	1	13	5
11	2	12	8	5	11	5	11	13	12	12	5	11	6	14	1
1	5	7	5	7	2	1	10	4	7	8	6	1	11	1	11
1	2	4	11	9	9	11	9	8	4	7	12	2	3	11	5
9	14	1	12	2	10	5	11	13	1	4	10	14	12	11	9

Les simulations effectuées sur ce réseau apparaissent dans les figures IV1b,c,d

figure IV-2a : Réseau à 256 éléments.

3	4	5	5	6	6	2	8	7	10	4	8	5	12	4	4
10	9	2	1	9	9	9	12	5	12	11	14	1	10	6	2
8	8	1	1	6	4	4	6	2	3	6	7	5	11	7	11
2	3	13	12	12	3	12	3	3	3	5	14	6	1	14	4
13	13	5	12	13	6	9	1	4	8	3	5	9	8	4	3
3	12	4	11	12	3	12	5	9	12	1	9	6	1	12	10
9	13	14	1	14	10	6	4	6	2	14	9	13	1	8	11
14	14	8	4	7	4	10	3	9	14	10	3	8	14	10	9
13	6	8	11	1	1	3	7	6	5	5	14	14	10	4	2
5	5	12	8	5	4	2	3	5	9	3	4	6	13	2	3
10	8	2	9	2	9	2	6	7	7	2	4	10	9	3	7
14	2	11	7	6	1	14	1	7	11	9	8	9	8	10	8
13	13	8	4	11	9	7	9	14	7	3	9	14	7	3	5
4	11	13	10	5	9	12	3	10	1	4	2	12	14	13	12
14	5	5	12	2	6	5	10	9	7	10	6	10	7	12	2
7	3	5	7	3	3	6	6	3	2	8	2	11	2	2	3

Les simulations effectuées sur ce réseau apparaissent dans les figures IV-2b,c,d

IV- 40

figure IV-2b : Nombre d'appartenance aux parties oscillantes. (100 dynamiques)

0	0	5	5	82	82	0	0	0	61	49	49	0	0	0	0
6	30	30	5	80	82	41	41	61	61	26	26	0	85	85	0
6	6	30	30	44	41	41	0	61	61	84	90	85	85	85	0
54	54	30	30	44	44	41	41	61	61	61	84	89	58	85	54
54	0	30	30	30	44	41	41	33	61	61	90	90	0	0	54
54	54	26	26	30	30	41	35	35	35	0	90	90	90	0	54
45	54	69	26	73	61	75	35	20	33	84	84	0	0	0	11
9	0	17	0	0	61	61	61	99	94	83	83	0	0	24	24
15	24	24	0	0	61	61	61	99	99	83	83	0	0	24	24
0	24	24	18	61	61	0	0	100	99	83	14	23	0	9	9
0	0	100	100	61	61	76	32	100	99	0	14	23	23	23	22
3	3	100	100	61	0	32	32	100	79	100	23	23	23	23	22
3	3	0	0	0	32	32	0	0	100	100	100	0	23	23	23
0	0	0	0	0	32	32	32	0	100	100	100	0	0	23	23
0	0	0	0	0	32	0	0	0	0	100	100	0	0	0	23
0	0	5	5	0	0	0	0	0	0	49	49	0	0	0	0

figure IV-2c : Nombre d'appartenance aux parties oscillantes. (250 dynamiques)

0	0	10	10	201	206	0	0	0	142	136	136	0	0	0	0
25	84	84	10	194	206	121	121	142	142	70	70	0	193	193	0
25	25	84	84	125	121	121	0	142	142	226	233	193	193	193	0
138	138	84	84	125	125	121	121	142	142	142	222	229	135	193	138
138	0	84	84	84	125	118	120	104	142	142	218	218	0	0	138
138	138	65	65	84	84	118	109	109	109	0	218	218	218	0	138
111	138	176	65	171	142	190	106	43	101	190	187	0	0	0	40
36	0	59	0	0	142	142	142	245	235	215	215	0	0	64	64
58	70	70	0	0	142	142	142	245	248	215	215	0	0	64	64
0	70	70	59	142	142	0	0	248	248	215	35	69	0	36	36
0	0	250	250	142	142	180	79	248	248	0	35	69	69	69	66
3	3	250	250	142	0	79	79	248	172	250	69	69	69	69	66
3	3	0	0	0	79	79	0	0	250	250	250	0	69	69	69
0	0	0	0	0	79	79	79	0	250	250	250	0	0	69	69
0	0	0	0	0	79	0	0	0	0	250	250	0	0	0	69
0	0	10	10	0	0	0	0	0	0	136	136	0	0	0	0

figure IV-2d : Nombre d'appartenance aux parties oscillantes. (1000 dynamiques)

0	0	49	49	826	839	0	0	0	575	567	567	0	0	0	0
120	382	381	49	774	839	467	467	575	575	320	320	0	784	784	0
120	120	381	382	495	467	467	0	575	575	849	895	784	784	784	0
587	587	381	381	495	495	467	467	575	575	574	838	875	531	784	587
587	0	381	381	382	495	476	465	384	574	574	876	876	0	0	587
587	587	294	294	382	382	476	402	402	402	1	876	876	876	0	587
483	587	708	294	649	518	694	392	180	374	758	756	1	1	0	159
139	0	247	0	0	518	518	518	994	934	857	857	1	1	284	284
233	278	278	0	0	518	518	518	993	992	857	857	0	0	284	284
0	278	278	229	517	517	0	0	987	992	857	161	248	0	139	139
0	0	999	999	517	517	713	350	987	992	0	161	248	248	248	240
10	10	999	999	517	0	350	350	987	741	1000	248	248	248	248	240
10	10	0	0	0	350	350	0	0	1000	1000	1000	0	248	248	248
0	0	0	0	0	350	350	350	0	1000	1000	1000	0	0	248	248
0	0	0	0	0	350	0	0	0	0	1000	1000	0	0	0	248
0	0	49	49	0	0	0	0	0	0	567	567	0	0	0	0

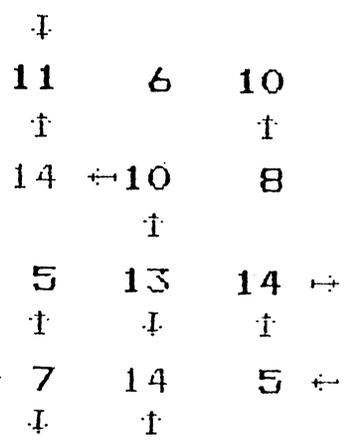
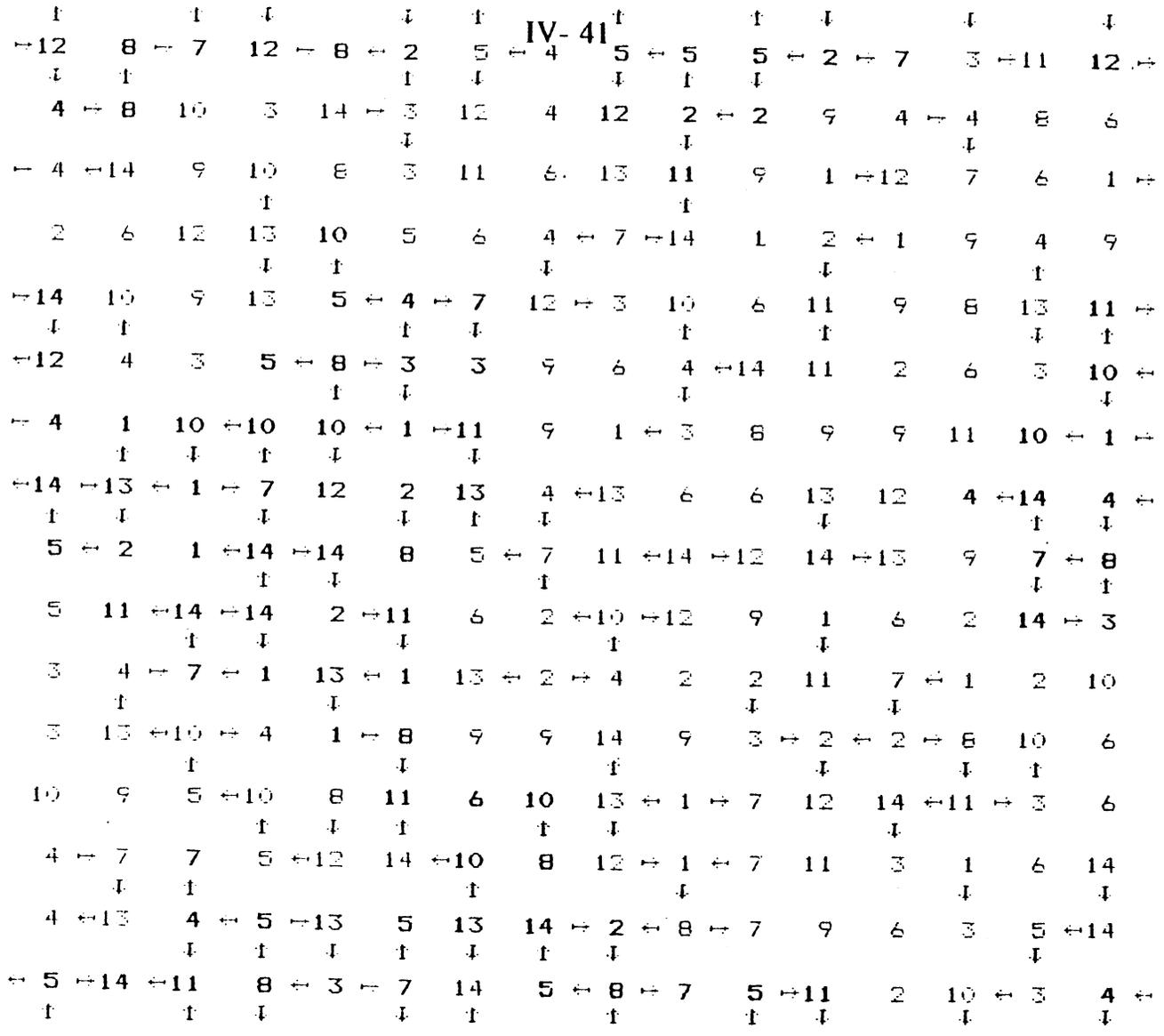


Figure IV-3

Réseau aléatoire avec son graphe forçant et son domaine forçant (en gras).
 On note 10 circuits forçants.
 Sur "l'agrandissement" on peut observer un exemple d'application des règles R1, R2 et R3 de la définition du domaine forçant.

IV- 42

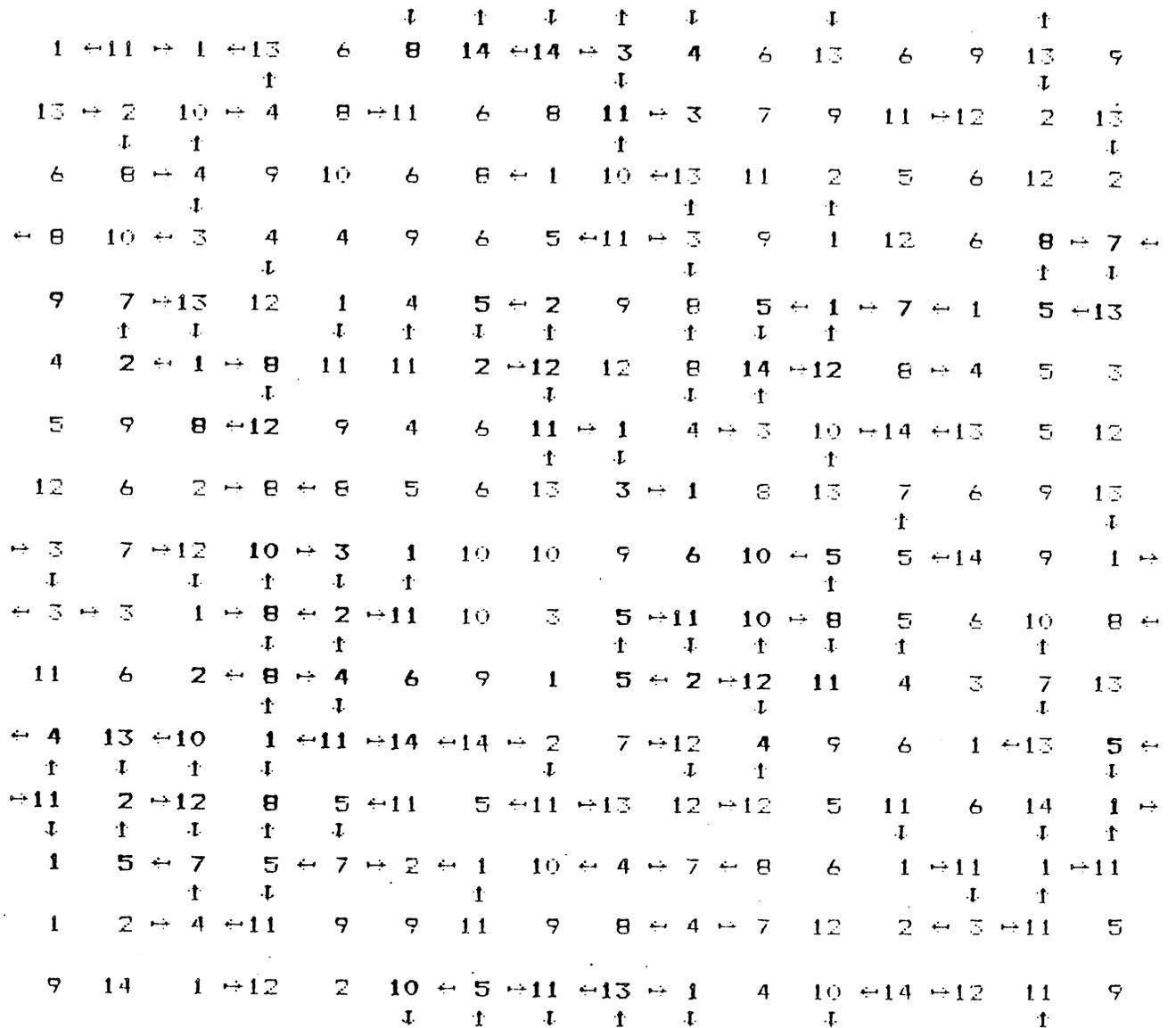


Figure IV-4

Réseau de la figure IV-1a avec son graphe forçant et son domaine forçant que l'on compare avec le coeur stable de la figure IV-1d

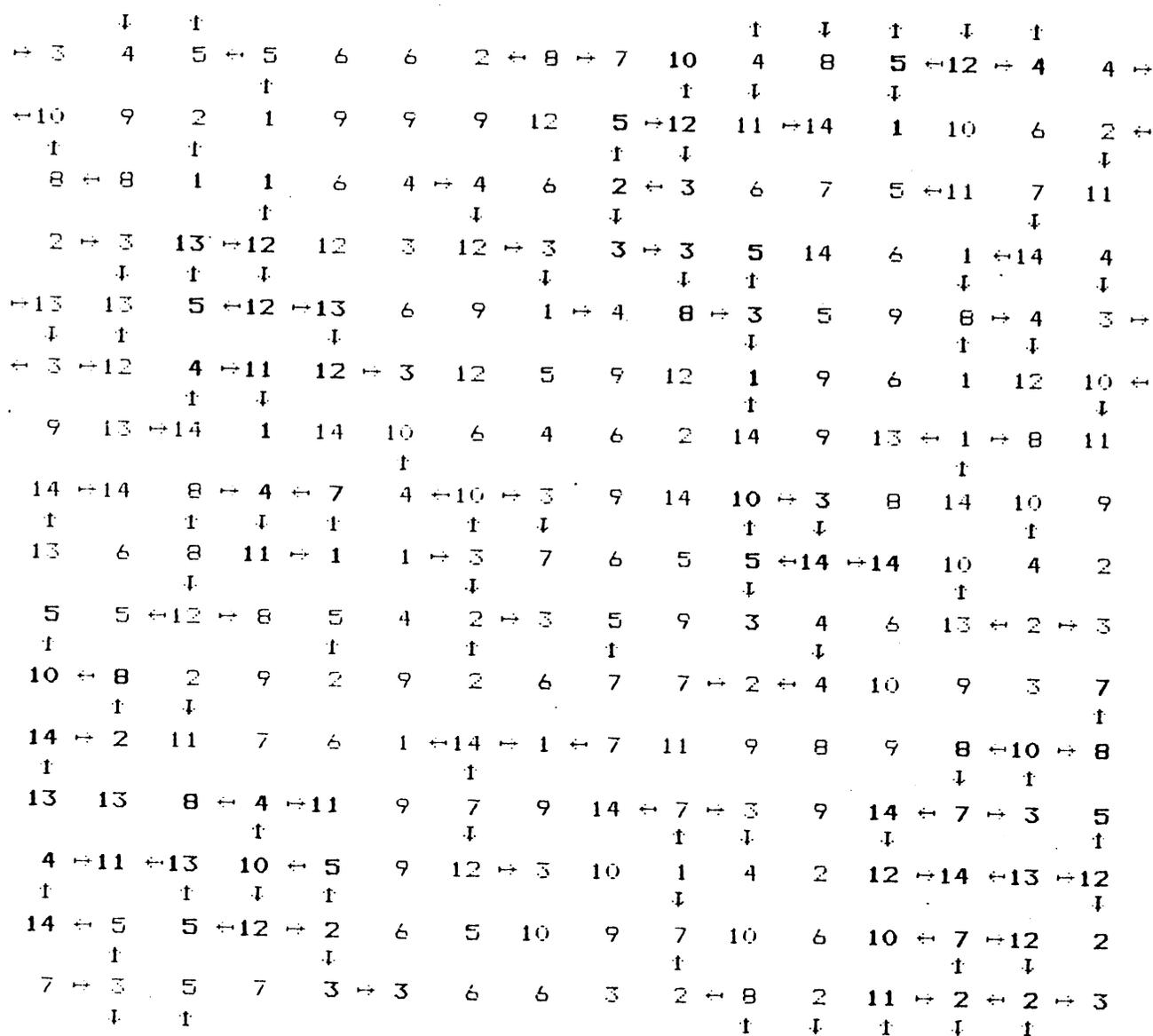
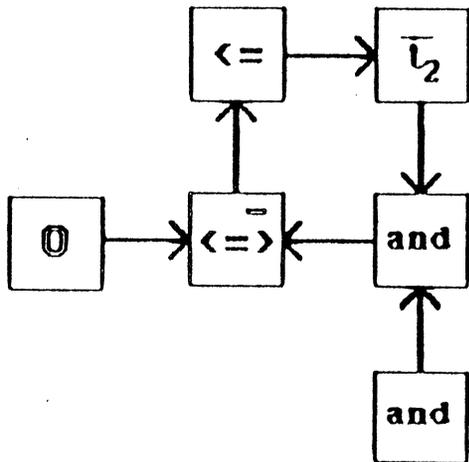


Figure IV-5

Réseau de la figure IV-2a avec son graphe forçant et son domaine forçant que l'on compare avec le coeur stable de la figure IV-2d. Les circuits marqués d'une croix ne font pas partie du coeur stable.

Effets de la transformation de lois en lois de transfert.

Figure IV-6a

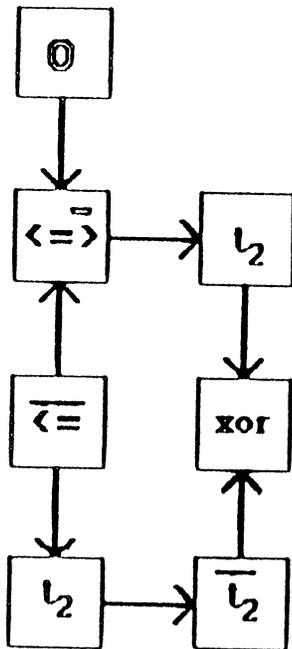


Exemple de création d'un circuit forçant par transformation des lois en contact avec le domaine forçant.

La loi $\langle = \rangle$ est transformée en loi de transfert (-) de par son entrée constante à 0, et on voit apparaître le circuit forçant.

(exemple tiré du réseau IV-1a)

Figure IV-6b



Une autre conséquence de la transformation: l'existence de points isolés dans le coeur stable.

La loi $\langle = \rangle$ est transformée en loi de transfert (-), la loi XOR reçoit deux entrées identiques, l'automate régi par cette loi reste donc stable dans l'état 0.

(exemple tiré du réseau IV-2a)

Les trois figures suivantes (IV-7a,b,c) montrent l'appartenance effective au coeur oscillant du réseau IV-1a des trois plus petites composantes de l'approximation de celui-ci qui apparaissent en IV-1b,c,d.

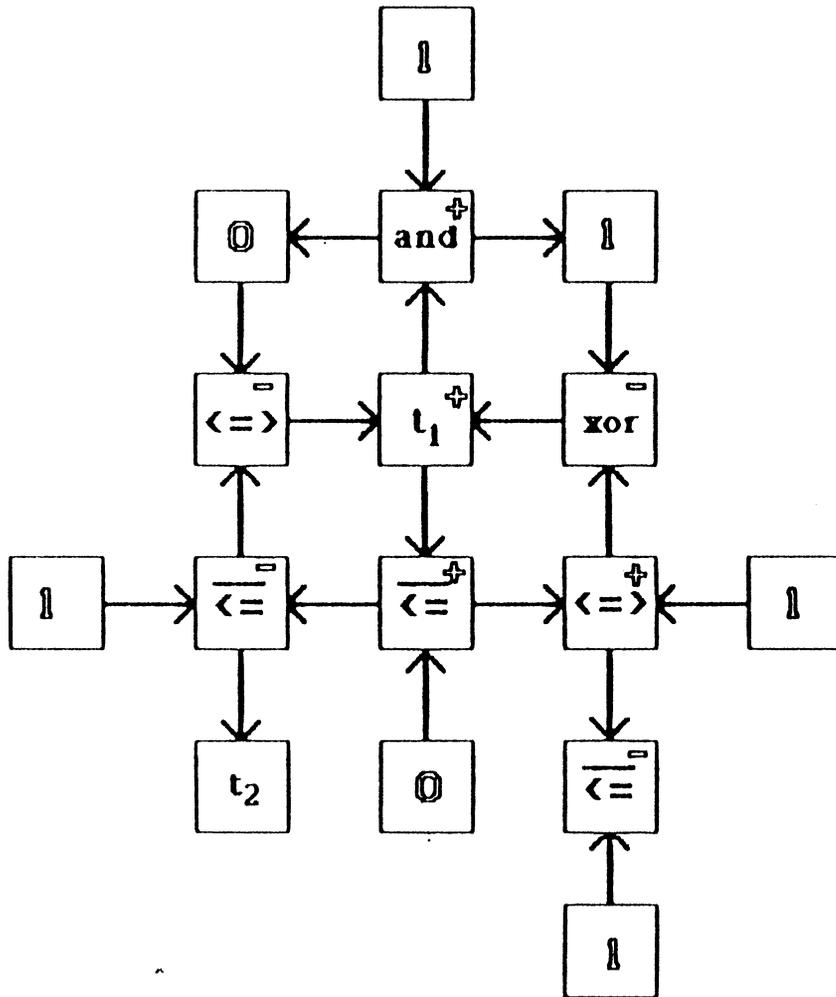


figure IV-7a

Création d'un circuit frustré du à l'environnement stable qui fait apparaitre des lois de transfert. (ici 2 circuits frustrés)

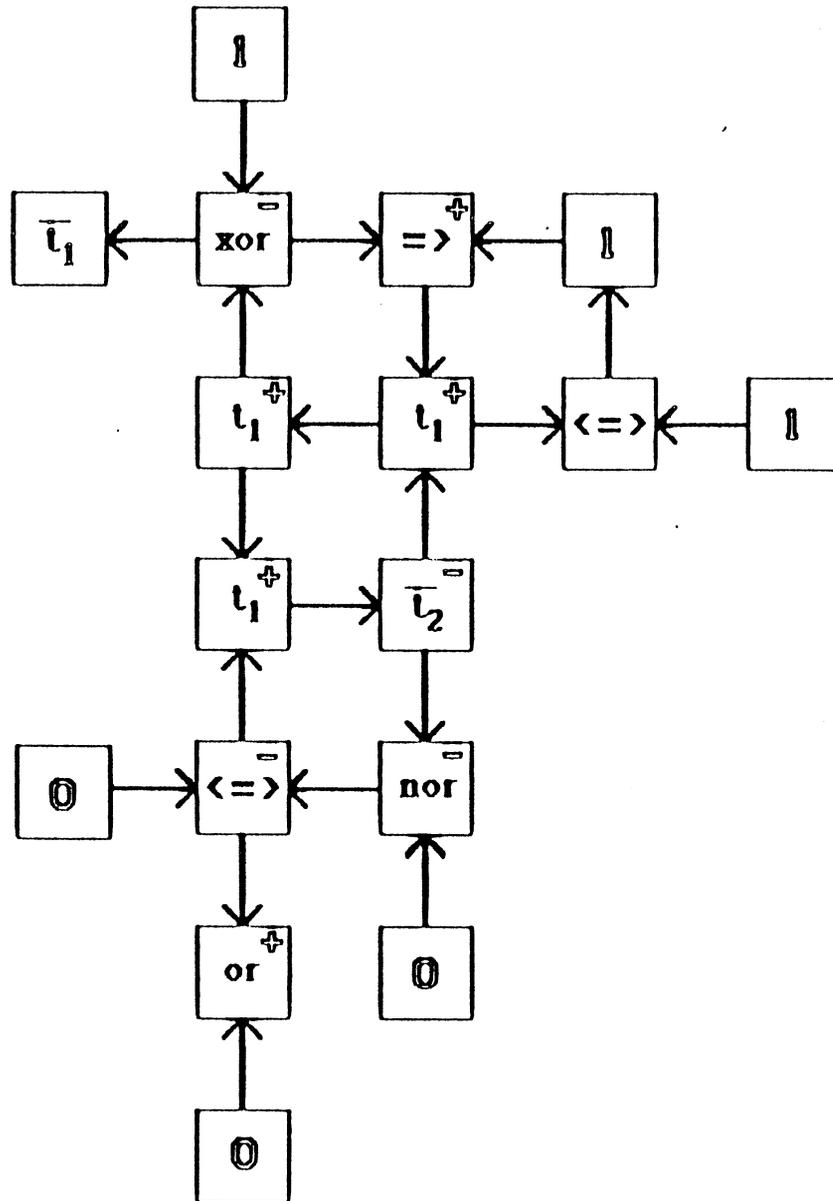


figure IV-7b

Ici il y a 1 circuit frustré "d'origine" et 2 nouveaux.

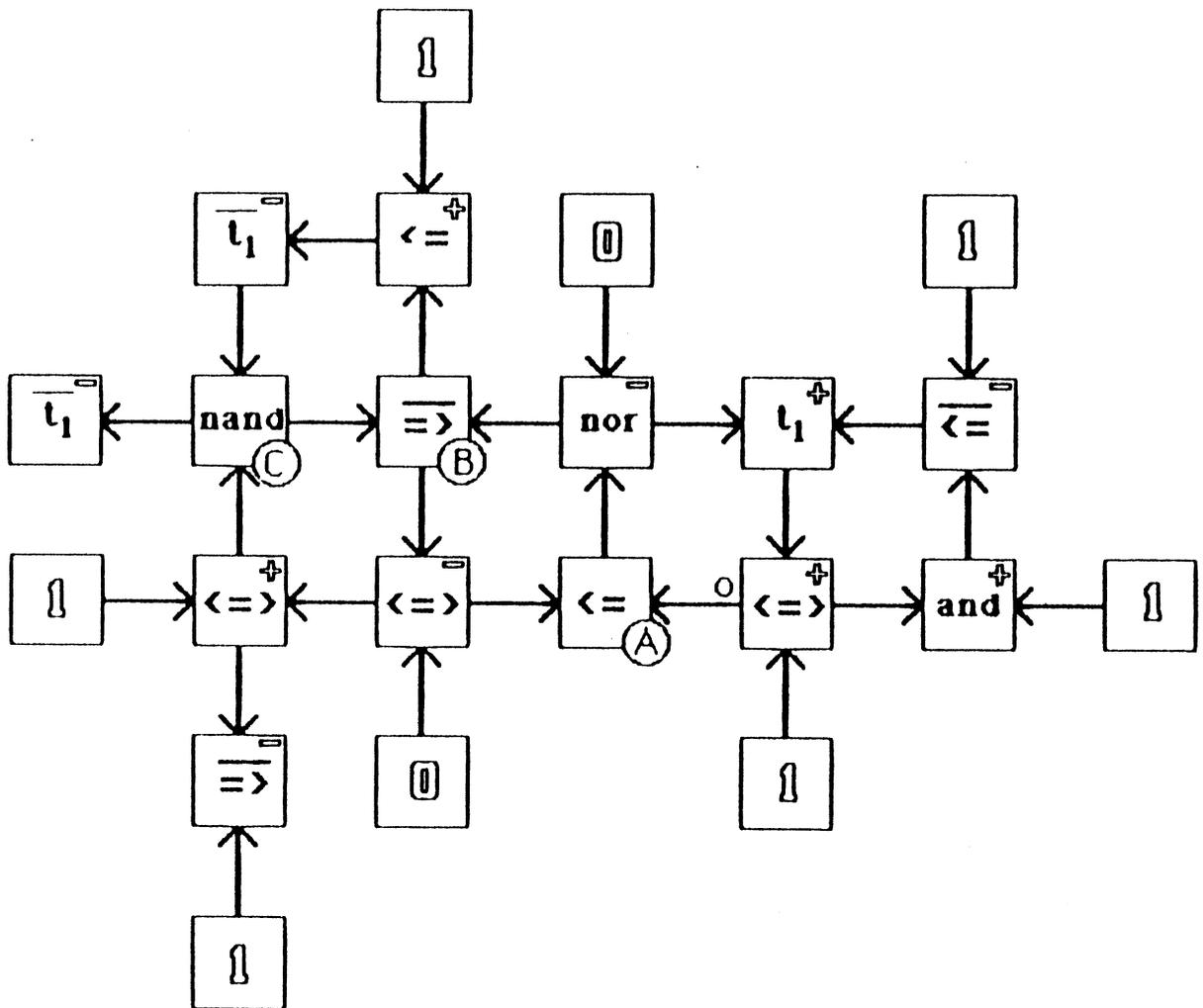


Figure IV-7c

Cas plus complexe : dans cette composante on observe la création d'un circuit frustré qui n'explique pas l'appartenance de toutes les cellules au coeur oscillant.

On note o la valeur de l'entrée de la cellule A qui provient du circuit frustré, et on suppose que les cellules A, B et C ont des états constants a, b, c . L'expression booléenne des lois de A, B et C donnent le système suivant : $a = ob + b$

$$b = ac$$

$$c = b$$

on vérifie qu'il est impossible de trouver 3 valeurs constantes pour a, b et c .

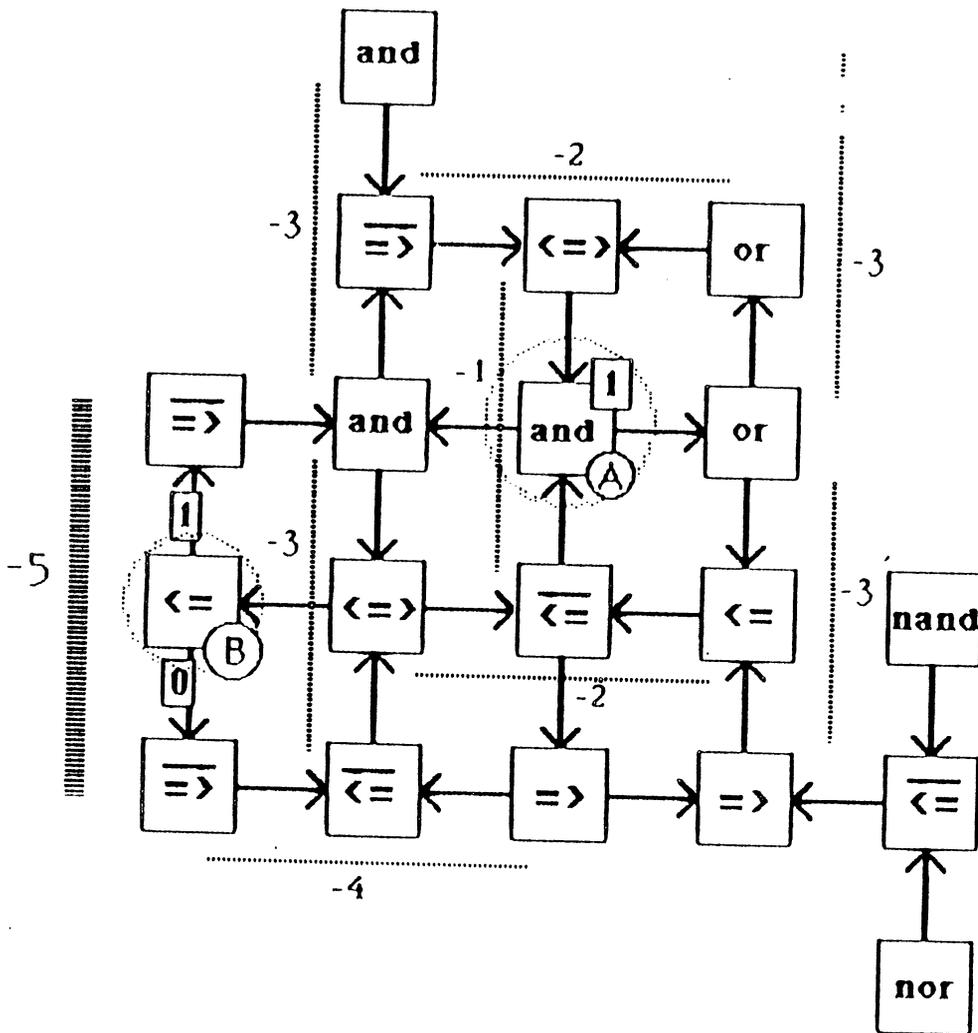


Figure IV-8 : point isolé du coeur stable (la cellule A).

Supposons que l'automate A prenne sa valeur non forcée : 1. Nous remontons alors dans le temps à l'aide du graphe de connexion pour savoir ce qu'il a fallu pour que A soit dans l'état 1. En pointillé et avec des indices négatifs figurent les temps (itérations). Nous arrivons à la contradiction suivante : l'automate B aurait dû fournir au réseau 2 sorties distinctes 5 temps auparavant.

A ne peut pas prendre la valeur 1 : il est stable en l'état 0.

Commentaires sur la "construction" des paliers

Les 2 figures suivantes montrent la construction des paliers des 2 réseaux IV-1a et IV-2a pour 20 conditions initiales.

Figure IV-9a

Sur les 20 tableaux on peut décompter la présence de 4 cycles limites distincts : A, B, C, D.

Cet exemple montre que la borne supérieure $2^{N_b}_p$ peut être atteinte.

cycle A : Seuls les éléments du coeur stable restent stables.

cycle B : Seuls les éléments du coeur oscillant oscillent.

cycle C : Le coeur oscillant et 1 palier ($d^\circ 0.543$) oscillent.

cycle D : Le coeur oscillant et l'autre palier ($d^\circ 0.466$) oscillent.

Figure IV-9b

Sur ces 20 tableaux on dénombre 20 cycles limites distincts : A,.....,T.

Hormis les 3 tableaux L,M et T, on décèle immédiatement la présence d'un nouveau cycle car il y a autant de nombres différents dans les tableaux que de conditions initiales testées.

Pour T et M on observe dans les passages $L \rightarrow M$ et $S \rightarrow T$ la conversion de 0 en 1 donc la présence d'un nouveau cycle.

Quant à L, il faut considérer d'un peu plus près l'évolution $K \rightarrow L$ pour s'apercevoir que la partition (partie stable/partie oscillante) ne se retrouve pas dans les 19 autres tableaux.

IV- 53

0 0 0 0 10 10 0 0 0 7 5 5 0 0 0 0
2 4 4 0 9 10 6 6 7 7 2 2 0 7 7 0
2 2 4 4 6 6 6 0 7 7 7 9 7 7 7 0
7 7 4 4 6 6 6 6 7 7 7 9 9 4 7 7
7 0 4 4 4 8 5 6 3 7 7 10 10 0 0 7
7 7 3 3 4 4 5 3 3 3 0 10 10 0 0 7
6 7 9 3 7 7 7 3 1 3 10 10 0 0 1
1 0 2 0 0 7 7 7 11 11 10 10 0 0 4 4
2 3 3 0 0 7 7 7 11 11 10 10 0 0 4 4
0 3 3 1 7 7 0 0 11 11 10 0 4 0 1 1
0 0 11 11 7 7 10 7 11 11 0 0 4 4 4 3
1 1 11 11 7 0 7 7 11 7 11 4 4 4 4 3
1 1 0 0 0 7 7 0 0 11 11 11 0 4 4 4
0 0 0 0 0 7 7 0 0 11 11 11 0 0 4 4
0 0 0 0 0 7 0 0 0 11 11 0 0 0 4
0 0 0 0 0 0 0 0 0 0 5 5 0 0 0 0

0 0 0 0 13 13 0 0 0 9 10 10 0 0 0 0
3 7 7 0 12 13 9 9 9 9 4 4 0 12 12 0
3 3 7 7 9 9 9 0 9 9 11 13 12 12 12 0
10 10 7 7 9 9 9 9 9 9 13 13 6 12 10
10 0 7 7 7 9 8 9 8 9 9 15 15 0 0 10
10 10 6 6 7 7 8 6 6 6 0 15 15 0 0 10
9 10 14 6 12 10 12 6 4 6 15 15 0 0 0 1
1 0 2 0 0 10 10 10 16 16 14 14 0 0 5 5
2 4 4 0 0 10 10 10 16 16 14 14 0 0 5 5
0 4 4 2 10 10 0 16 16 14 1 5 0 1 1
0 0 16 16 10 10 14 9 16 16 0 1 5 5 5 3
2 2 16 16 10 0 9 9 16 11 16 5 5 5 5 3
2 2 0 0 0 9 9 0 0 16 16 16 0 5 5 5
0 0 0 0 0 9 9 9 0 16 16 16 0 0 5 5
0 0 0 0 0 9 0 0 0 16 16 16 0 0 5 5
0 0 0 0 0 0 0 0 0 10 10 0 0 0 0

0 0 0 0 11 11 0 0 0 8 6 6 0 0 0 0
3 5 5 0 10 11 7 7 8 8 2 2 0 8 8 0
3 3 5 5 7 7 7 0 8 8 8 10 8 8 8 0
8 8 5 5 7 7 7 7 8 8 8 10 4 8 8
8 0 5 5 5 5 6 7 4 8 8 11 11 0 0 8
8 8 4 4 5 5 6 4 4 0 11 11 0 0 8
7 8 10 4 8 7 8 4 2 4 11 11 0 0 1
1 0 2 0 0 7 7 7 12 12 11 11 0 0 4 4
2 3 3 0 0 7 7 7 12 12 11 11 0 0 4 4
0 3 3 1 7 7 0 0 12 12 11 0 4 0 1 1
0 0 12 12 7 7 10 7 12 12 0 0 4 4 4 3
1 1 12 12 7 0 7 7 12 8 12 4 4 4 4 3
1 1 0 0 0 7 7 0 0 12 12 12 0 4 4 4
0 0 0 0 0 7 7 0 0 12 12 12 0 0 4 4
0 0 0 0 0 7 0 0 0 12 12 0 0 0 4
0 0 0 0 0 0 0 0 0 6 6 0 0 0 0

0 0 0 0 14 14 0 0 0 9 10 10 0 0 0 0
3 7 7 0 13 14 9 9 9 9 4 4 0 13 13 0
3 3 7 7 9 9 9 0 9 9 12 14 13 13 13 0
11 11 7 7 9 9 9 9 9 9 14 14 6 13 11
11 0 7 7 7 9 8 9 6 9 9 16 16 0 0 11
11 11 6 6 7 7 8 6 6 6 0 16 16 16 0 11
10 15 8 13 11 13 6 4 6 15 15 0 0 0 1
1 0 2 0 0 11 11 11 17 17 15 15 0 0 5 5
2 4 4 0 0 11 11 11 17 17 15 15 0 0 5 5
0 4 4 2 11 11 0 0 17 17 15 1 5 0 1 1
0 0 17 17 11 11 15 10 17 17 0 1 5 5 5 3
2 2 17 17 11 0 10 10 17 17 5 5 5 5 3
2 2 0 0 0 10 10 0 0 17 17 17 0 5 5 5
0 0 0 0 0 10 10 10 0 17 17 17 0 0 5 5
0 0 0 0 0 10 0 0 0 17 17 17 0 0 5 5
0 0 0 0 0 0 0 0 0 10 10 0 0 0 0

0 0 0 0 12 12 0 0 0 8 7 7 0 0 0 0
3 6 6 0 11 12 8 8 8 8 3 3 0 9 9 0
3 3 6 6 8 8 8 0 8 8 8 11 9 9 9 0
8 8 6 6 8 8 8 8 8 8 11 11 4 9 8
8 0 6 6 6 6 8 7 8 5 8 8 12 12 0 0 8
8 8 5 5 6 6 7 5 5 5 0 12 12 0 0 8
7 8 11 5 9 8 9 5 3 5 12 12 0 0 1
1 0 2 0 0 8 8 8 13 13 11 11 0 0 5 5
2 4 4 0 0 8 8 8 13 13 11 11 0 0 5 5
0 4 4 2 8 8 0 0 13 13 1 1 5 0 1 1
0 0 13 13 8 8 11 8 13 13 0 1 5 5 5 3
2 2 13 13 8 0 8 8 13 8 13 5 5 5 5 3
2 2 0 0 0 8 8 0 0 13 13 13 0 5 5 5
0 0 0 0 0 8 8 8 0 13 13 13 0 0 5 5
0 0 0 0 0 8 0 0 0 13 13 0 0 0 5
0 0 0 0 0 0 0 0 0 7 7 0 0 0 0

0 0 0 0 15 15 0 0 0 10 10 10 0 0 0 0
3 8 8 0 13 15 10 10 10 10 4 4 0 14 14 0
3 3 8 8 10 10 10 0 10 10 13 15 14 14 0
12 12 8 8 10 10 10 10 10 10 15 15 6 14 12
12 0 8 8 8 10 9 10 6 10 10 17 17 0 0 12
12 12 7 7 8 8 9 6 6 6 0 17 17 17 0 12
11 12 16 7 14 12 14 6 4 6 18 16 0 0 0 1
1 0 2 0 0 12 12 12 18 18 16 16 0 0 6 6
2 5 5 0 0 12 12 12 18 18 16 0 0 6 6
0 5 5 2 12 12 0 0 18 18 16 2 6 0 1 1
0 0 18 18 12 12 16 10 18 18 0 2 6 6 4
2 2 18 18 12 0 10 10 18 12 18 6 6 6 6 4
2 2 0 0 0 10 10 0 0 18 18 18 0 6 6 6
0 0 0 0 0 10 10 10 0 18 18 18 0 0 6 6
0 0 0 0 0 10 0 0 0 18 18 0 0 0 6
0 0 0 0 0 0 0 0 0 10 10 0 0 0 0

0 0 0 0 12 12 0 0 0 8 8 8 0 0 0 0
3 6 6 0 11 12 8 8 8 8 3 3 0 10 10 0
3 3 6 6 8 8 8 0 8 8 10 12 10 10 10 0
9 9 8 6 8 8 8 8 8 8 12 12 5 10 9
9 0 6 6 6 8 7 8 5 8 8 13 13 0 0 9
9 9 5 5 6 6 7 5 5 5 0 13 13 0 0 9
8 9 12 5 10 9 10 5 3 5 13 13 0 0 1
1 0 2 0 0 9 9 9 14 14 12 12 0 0 5 5
2 4 4 0 0 9 9 9 14 14 12 12 0 0 5 5
0 4 4 2 9 9 0 0 14 14 1 1 5 0 1 1
0 0 14 14 9 9 12 8 14 14 0 1 5 5 5 3
2 2 14 14 9 0 8 8 14 9 14 5 5 5 5 3
2 2 0 0 0 8 8 0 0 14 14 14 0 5 5 5
0 0 0 0 0 8 8 8 0 14 14 14 0 0 5 5
0 0 0 0 0 8 0 0 0 14 14 0 0 0 5
0 0 0 0 0 0 0 0 0 14 14 0 0 0 0

0 0 0 0 16 16 0 0 0 11 11 11 0 0 0 0
3 8 8 0 14 16 10 10 11 11 5 5 0 15 15 0
3 3 8 8 10 10 10 0 11 11 14 16 15 15 0
13 13 8 8 10 10 10 10 11 11 16 16 7 15 13
13 0 8 8 8 10 9 10 6 11 11 18 18 0 0 13
13 13 7 7 8 8 9 6 6 6 0 18 18 18 0 13
11 13 17 7 14 12 14 6 4 6 17 17 0 0 0 2
1 0 2 0 0 12 12 12 19 19 17 17 0 0 7 7
2 5 5 0 0 12 12 12 19 19 17 17 0 0 7 7
0 5 5 2 12 12 0 0 19 19 17 2 7 0 1 1
0 0 19 19 12 12 16 10 19 19 0 2 7 7 5
2 2 19 19 12 0 10 10 19 13 19 7 7 7 7 5
2 2 0 0 0 10 10 0 0 19 19 19 0 7 7 7
0 0 0 0 0 10 10 10 0 19 19 19 0 0 7 7
0 0 0 0 0 10 0 0 0 19 19 0 0 0 7
0 0 0 0 0 0 0 0 0 11 11 0 0 0 0

0 0 0 0 13 13 0 0 0 8 9 9 0 0 0 0
3 7 7 0 12 13 9 9 8 8 3 3 0 11 11 0
3 3 7 7 9 9 9 0 8 8 10 12 11 11 11 0
9 9 7 7 9 9 9 9 8 8 12 12 6 11 9
9 0 7 7 7 9 8 9 6 8 8 14 14 0 0 9
9 9 6 6 7 7 8 6 6 6 0 14 14 0 0 9
8 9 13 6 11 9 11 6 4 6 14 14 0 0 1
1 0 2 0 0 9 9 9 15 15 13 13 0 0 5 5
2 4 4 0 0 9 9 9 15 15 13 13 0 0 5 5
0 4 4 2 9 9 0 0 15 15 1 1 5 0 1 1
0 0 15 15 9 9 13 9 15 15 0 1 5 5 5 3
2 2 15 15 9 0 9 9 15 10 15 5 5 5 5 3
2 2 0 0 0 9 9 0 0 15 15 15 0 5 5 5
0 0 0 0 0 9 9 9 0 15 15 15 0 0 5 5
0 0 0 0 0 9 0 0 0 15 15 0 0 0 5
0 0 0 0 0 0 0 0 0 15 15 0 0 0 5

0 0 1 1 17 17 0 0 0 12 12 12 0 0 0 0
4 9 9 1 15 17 11 11 12 12 5 5 0 16 16 0
4 4 9 9 11 11 11 0 12 12 15 17 16 16 0
14 14 9 9 11 11 11 11 12 12 17 17 7 16 14
14 0 9 9 9 11 10 11 7 12 12 19 19 0 0 14
14 14 7 7 9 9 10 7 7 7 0 19 19 19 0 14
12 14 16 7 15 13 15 7 5 7 18 18 0 0 0 2
1 0 2 0 0 13 13 13 20 20 18 18 0 0 7 7
2 5 5 0 0 13 13 13 20 20 18 18 0 0 7 7
0 5 5 2 13 13 0 0 20 20 18 2 7 0 1 1
0 0 20 20 13 13 17 11 20 20 0 2 7 7 5
2 2 20 20 13 0 11 11 20 14 20 7 7 7 7 5
2 2 0 0 0 11 11 0 0 20 20 20 0 7 7 7
0 0 0 0 0 11 11 11 0 20 20 20 0 0 7 7
0 0 0 0 0 11 0 0 0 20 20 0 0 0 7
0 0 1 1 0 0 0 0 0 0 12 12 0 0 0 0

HISTOGRAMME DES PALIERS 100 C. I.

Coeur stable : 164 éléments
Coeur oscillant : 62 éléments

49 *****
59 *****

Nombre de paliers : 2

HISTOGRAMME DES PALIERS 250 C. I.

Coeur stable : 164 éléments
Coeur oscillant : 62 éléments

104 *****
140 *****

Nombre de paliers : 2

HISTOGRAMME DES PALIERS 1000 C. I.

Coeur stable : 164 éléments
Coeur oscillant : 62 éléments

466 *****
543 *****

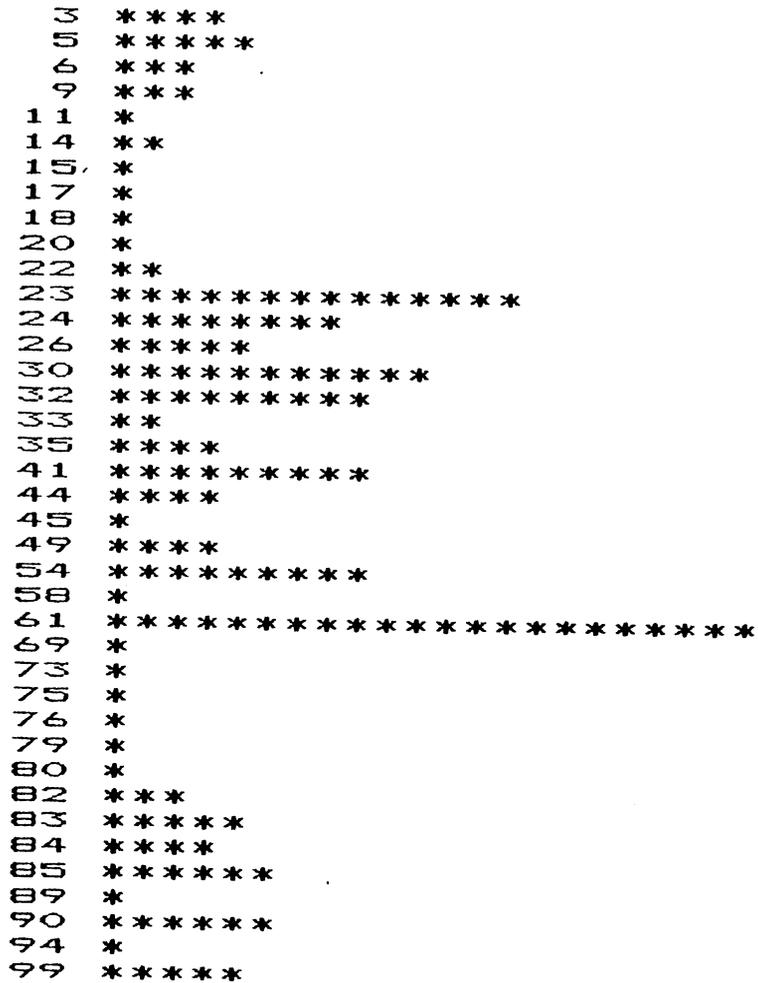
Nombre de paliers : 2

Figure IV-10

Histogrammes des paliers du réseau IV-1a calculés pour 100, 250 et 1000 conditions initiales aléatoires : on observe une grande simplicité de la dynamique.

HISTOGRAMME DES PALIERS 100 C. I.

Coeur stable : 76 éléments
 Coeur oscillant : 16 éléments



Nombre de paliers : 39

Figures IV-11 a,b,c,d

Histogrammes des paliers du réseau IV-2a pour 100(a), 250(b), 1000(c) et 9999(d) conditions initiales aléatoires.

On observe une grande complexité et instabilité de la dynamique.

HISTOGRAMME DES PALIERS 250 C.I.

Coeur stable : 76 éléments
Coeur oscillant : 13 éléments

```
3 ****
10 *****
25 ***
35 **
36 ***
40 *
43 *
58 *
59 **
64 ****
65 ***
66 **
69 *****
70 *****
79 *****
84 *****
101 *
104 *
106 *
109 ***
111 *
118 **
120 *
121 *****
125 ***
135 *
136 ***
138 *****
142 *****
171 *
172 *
176 *
180 *
187 *
190 **
193 *****
194 *
201 *
206 **
215 *****
218 *****
222 *
226 *
229 *
233 *
235 *
245 **
248 *****
```

Nombre de paliers : 48

HISTOGRAMME DES PALIERS 1000 C. I.

Coeur stable : 71 éléments
 Coeur oscillant : 9 éléments

```

1 *****
10 *****
49 *****
120 *****
139 *****
159 *****
161 *****
180 *****
229 *****
233 *****
240 *****
247 *****
248 *****
278 *****
284 *****
294 *****
320 *****
350 *****
374 *****
381 *****
382 *****
384 *****
392 *****
402 *****
465 *****
467 *****
476 *****
483 *****
495 *****
517 *****
518 *****
531 *****
567 *****
574 *****
575 *****
587 *****
649 *****
694 *****
708 *****
713 *****
741 *****
756 *****
758 *****
774 *****
784 *****
826 *****
838 *****
839 *****
849 *****
857 *****
875 *****
876 *****
895 *****
934 *****
987 *****
992 *****
993 *****
994 *****
999 *****
    
```

Nombre de paliers : 59

HISTOGRAMME DES PALIERS 9999 C. I.

Coeur stable : 70 éléments
 Coeur oscillant : 3 éléments

```

    6 *
    8 *****
   133 *****
   402 *****
  1184 *****
  1392 *****
  1630 *****
  1661 *
  1996 *
  2249 *
  2259 *
  2408 *
  2506 **
  2618 *****
  2619 *
  2764 *****
  2841 *****
  2845 *****
  3078 *****
  3445 *****
  3774 *****
  3967 *
  4046 *
  4149 *
  4228 *****
  4720 *
  4812 *
  4846 *****
  4907 *****
  5096 *****
  5295 *
  5366 *****
  5378 *****
  5576 *****
  5758 *****
  5830 *****
  5841 *****
  6631 *
  7083 *
  7197 *
  7206 *
  7236 *
  7520 *
  7566 *
  7732 *
  7796 *****
  8199 *
  8318 *****
  8496 *
  8581 *****
  8585 *
  8735 *****
  8736 *****
  8762 *
  9091 *
  9410 *
  9914 *****
  9917 *
  9921 *
  9941 *****
  9976 *
  9981 *****
  9991 *
  9992 *****

```

Nombre de paliers : 64

FIGURE V-2a : Exemple de cycle limite du réseau IV-1a avec ses valeurs stables.

```

. 0 0 1 1 0 1 1 0 0 0 0 0 0 . .
. 0 0 0 . 1 1 1 1 0 1 1 1 1 . .
. 0 0 . . . 0 0 1 1 0 0 . . . .
. 1 1 . . . 1 0 1 0 0 . . . 0 1
1 1 1 . 0 . 1 0 1 0 1 0 1 . 0 1
. 0 0 0 1 0 0 0 1 1 1 1 . . . .
. . 0 0 1 0 1 1 0 1 0 . . . . .
. . . . 1 . . . 1 0 0 . . . . .
. . . 0 1 0 . . . 1 1 1 . . . .
. . . 0 0 1 . . . 0 1 0 0 . . 1 .
1 . 0 0 0 0 . . 1 0 0 1 . . 1 .
0 1 0 0 1 1 . 0 0 0 0 . . 0 1 1
1 0 0 0 . . 0 1 1 0 0 1 1 1 1 0
0 0 1 . . . . . 1 0 0 0 1 0 1
0 1 0 1 . . . . . 1 1 0 0 0 1 1
. 1 1 1 . 0 0 1 1 0 1 0 0 0 . 0
    
```

FIGURE V-2b : Exemple de faible reconnaissance au point B, sequence (1000). Les cellules marquées d sont déstabilisées par rapport au cycle limite non bruité. Les cellules marquées 0 ou 1 sont stabilisées dans cet état. On note que les cellules stabilisées constituent une union de paliers.

```

. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . . 0 0 . .
. . . . . 0 0 B . .
. . 1 1 . . . . . 0 1 1 0 .
. 1 1 . . . . . . 0 1 . .
. . 0 . . . . . . 0 1 . .
. . . . . . . 1 0 . .
. . . . . . . 1 0 . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
    
```

00....001.011001
11..110110
000....1011
01.....01.
 .1.....00....00.
 ..01...0110...0.
 ...0...000..1001
 11001.....1111
 10010.....1110
 1100..01...01101
 00.....1..000011
 10...010.1.00000
 100010111...1101
 011111100...1100
 1101011001..1111
 10..101010011001

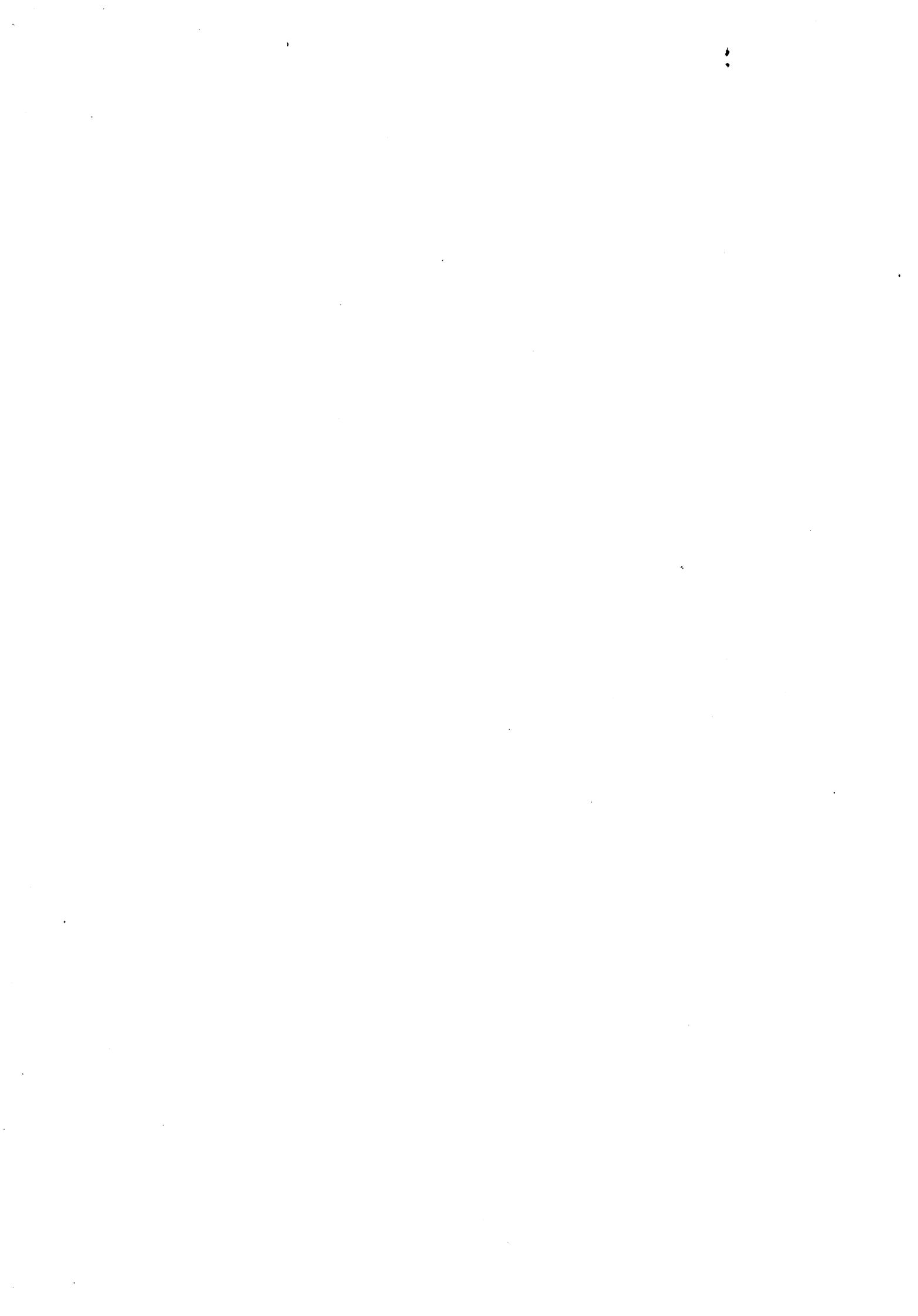
FIGURE V-3 : Exemple de bonne reconnaissance.

Le premier tableau représente un cycle limite, sans bruit, du réseau IV-2a

Les suivants indiquent les séquences reconnues et les ensembles de cellules les reconnaissant.

(000*)	(1*0*)	(011*)
..0101.....0.....	..01.....
0B10.1dd....dd.	.B....dd....dd.	.B.0.....
0010.dd....0ddd.dd....0ddd.	..00.....
..11..dd...1..d.dd...1..d.	..1101.....
..011..dd.....dd.....	..01110.....
....10.ddd.....ddd.....100.....
.....d.d.....d.d.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..10.....10.....

(010*)	(001*)
..0101.....	..01.....
.B.0.1dd....dd.	0B10..dd.....
...0.dd....0ddd.	0010.dd.....
..11..dd...1..d.	..11..dd.....
..011..dd.....	..011.0d.....
....10.ddd.....100.....
.....d.d.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..10.....	..10.....



AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USMG

Vu les dispositions de l'arrêté du 16 avril 1974,

Vu les dispositions de l'arrêté du 5 juillet 1984,

Vu les rapports de M.^r Gérard WEISBUCH.....

M.^r Yves CHIARAMELLA.....

M.^r Didier PELLEGRIN..... est autorisé
à présenter une thèse en vue de l'obtention du ... DOCTORAT... de.....
..l'Université... Scientifique... Technologique... et Médicale... de Grenoble....

Grenoble, le 18 mai 84

Le Président de l'Université Scientifique
et Médicale



M. Tanche

M. TANCHE

