Expérimentation d'automates à seuil pour la connaissance de caractères

Xin An Pan

THESE

présentée à

l'université Scientifique et Médicale de Grenoble

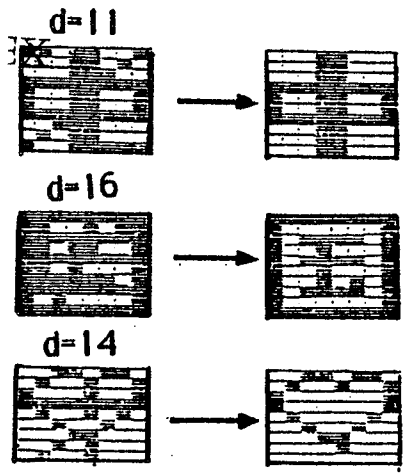
pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

Mathématiques Appliquées

par

Xin an PAN



EXPERIMENTATION D'AUTOMATES A SEUIL POUR LA RECONNAISSANCE DE CARACTERES

Thèse soutenue le 14 juin 1985 devant la Commission d'Examen :

Monsieur P.J. LAURENT : Président

Monsieur M. COSNARD
Madame F. FOGELMAN-SOULIE
Monsieur F. ROBERT
Monsieur M.TCHUENTE
} : Examinateurs

THESE

présentée à

l'université Scientifique et Médicale de Grenoble

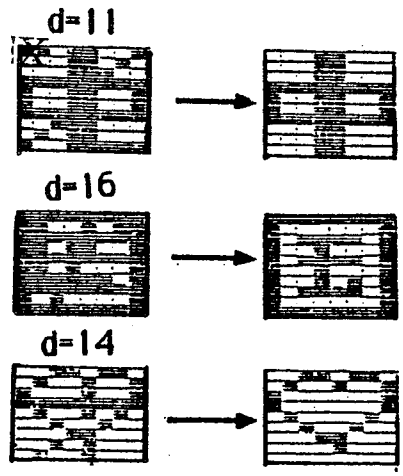
pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

Mathématiques Appliquées

par

Xin an PAN



EXPERIMENTATION D'AUTOMATES A SEUIL POUR LA RECONNAISSANCE DE CARACTERES

Thèse soutenue le 14 juin 1985 devant la Commission d'Examen :

Monsieur P.J. LAURENT : Président

Monsieur M. COSNARD
Madame F. FOGELMAN-SOULIE
Monsieur F. ROBERT
Monsieur M.TCHUENTE
} : Examinateurs

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

Stagiaire chinois n'ayant au départ aucune expérience de la vie et du travail à l'étranger , je suis très reconnaissant à M. F. ROBERT , mon directeur de recherche , de m'avoir chaleureusement accueilli et de m'avoir proposé deux sujets de recherche : 1) calcul de minima de Pareto et 2) la reconnaissance de formes par un automate à seuil, qui m'intéressent réellement. D'ailleurs, au cours de mon travail , j'ai été constamment soutenu par ses conseils et suggestions qui m'étaient indispensables. J'ai beaucoup apprécié durant ces deux années en France, les qualités humaines et scientifiques de M. F. ROBERT dont je conserverai un excellent souvenir.

Je suis très heureux que M. P.J. LAURENT ait accepté d'être le Président de mon jury (*), et que M. M. COSNARD, M^me. F. FOGELMAN-SOULIE , M. F. ROBERT et M. M. TCHUENTE soient membres de ce jury.

Je remercie vivement M. M. COSNARD de m'avoir proposé le sujet de mon second chapitre : Expérimentation numérique d'un modèle itératif de la respiration ; j'ai été aussi très content de travailler avec M. J. DEMONGEOT sur ce sujet.

Je doit remercier beaucoup M^me. F. FOGELMAN-SOULIE et M. D. PELLEGRIN qui m'ont donné des conseils et fait des corrections sur le troisième chapitre de cette thèse.

Je voudrais remercier aussi M. A. EBERHARD , M^elle. C. DICRESCENZO et M. J.M. MULLER de m'avoir donné beaucoup d'assistance au cours de mon apprentissage de l'utilisation des ordinateurs.

Enfin, je voudrais encore remercier tous mes collègues qui m'ont exprimé la gentillesse et l'amitié.

(*) En remplacement de Monsieur M. SOTIF , empêché, à qui je souhaite un prompt rétablissement.

X. PAN

潘新安.

TABLE DES MATIERES

Chapitre Trois

Automate à seuil pour la reconnaissance de formes

Chapitre premier

CALCUL NUMERIQUE DE MINIMA DE PARETO

§ 1.1 Introduction

Dans la pratique, un problème de minimisation multi-critères apparait lorsqu'un unique décideur cherche à tenir compte d'objectifs partiellement contradictoires, ou lorsqu'une décision collective doit être prise par des agents ayant des opinions différentes sur le résultat de la décision. L'interêt d'un état x est alors jugé d'après plusieurs critères $f_1(x)$, $f_2(x)...f_k(x)$ qui peuvent être en partie contradictoires (Il existe probablement x_1 et x_2 tel que $f_1(x_1)<f_1(x_2)$ et $f_2(x_2)<f_2(x_1)$) donc la notion de solution optimale du problème cesse alors d'être évidente. On introduit alors le concept de minima de Pareto.

§ 1.2 Définitions et Propriétés Générales

Dans cette section, on rappele quelques notions et résultats de base qui concernent les minima de Pareto. Pour la plupart des résultats, on peut trouver des démonstrations dans [1], [2].

Soit X un ensemble quelconque (l'espace des états) et f_1, $f_2...f_k$ un nombre fini de fonctions numériques définies sur X (les critéres qu'on cherche à minimiser); on donne alors les deux définitions suivantes:

Définition 1.21 Minima de Pareto

un état $x \in X$ est appelé un minimum de Pareto du probléme $(X, f_1, ... f_k)$ s'il n'existe aucun élément y dans X tel que

$$\forall\ i \in (1, ... k) \qquad f_i(y) \leq f_i(x)$$

l'une au moins de ces inégalités étant stricte.

On note $\mathbf{p}$ l'ensemble des minima de Pareto.

Definition 1.2.2 Minima de Pareto Faibles

Un état $x \in X$ est appelé un minimum de Pareto faible, s'il n'éxiste aucun élément y dans X tel que

$$\forall\, i \in [1, 2, \ldots, k] \qquad f_i(y) < f_i(x)$$

On note $\mathbf{p}_f$ l'ensemble des minima de Pareto faibles

Remarque :

Ces deux définitions sont naturelles et on a toujours :

$$\mathbf{p} \subset \mathbf{p}_f$$

Ainsi la notion de minimum de Pareto faible est moins restrictive que celle de minimum de Pareto.

Des résultats généraux concernant les minima de Pareto ([1], [2]) nous allons tirer un certine nombre d'informations pour notre cas particulier. Nous les regroupons dans les propositions suivantes :

Proposition 1.2.1 (Existence)

Si X est un espace metrique et si les critéres f_i sont continues, alors l'ensemble $\mathbf{p}_f$ des minima de Pareto faibles du problème $[X, f_1 \ldots f_k]$ est fermé eventuellement vide. Si de plus X est compact, $\mathbf{p}_f$ est non vide.

Proposition 1.2.2 (Caractérisation de type algébrique)

On suppose de plus X un espace vectoriel sur $\mathbf{R}$ et les fonctions f_i convexes. Alors pour qu'un élément a de X soit un minimum de Pareto faible du problème $[x, f_1, \ldots f_k]$ il faut et il suffit qu'il existe des constantes $\lambda_1 \ldots \lambda_k \geq 0$ non toutes nulles telles que :

$$\sum_{i=1}^{k} \lambda_i f_i(a) = \inf_{y \in X} \left[\sum_{i=1}^{k} \lambda_i f_i(y) \right] \qquad (1)$$

(De tels λ_i définis à une constante multiplicative positive près sont appelés des multiplicateurs)

Proposition 1.2.3

Si de plus tous les λ_i dans (1) sont strictement positifs, alors a est un minimum de Pareto.

Proposition 1.2.4 (caractèrisation de type topologique):

Soit $(X, f_1, \ldots f_k)$ un problème de minima de Pareto où X est un espace de Hilbert réel et $f_1, \ldots f_k$ des fonctions différentiables et convexes sur X, Alors $a \in X$ est un minimum de Pareto faible de $(X, f_1, \ldots f_k)$ si et seulement si il existe des multiplicateurs

$$\lambda_1 \geq 0, \ldots \lambda_k \geq 0 \qquad \text{non tous nuls}$$

tels que

$$\sum_{i=1}^{k} \lambda_i \operatorname{grad}(f_i(a)) = 0$$

Proposition 1.2.5

Si dans le problème de minima de Pareto $(X, f_1, \ldots f_k)$ les conditions de proposition 1.2.4 sont satisfaites et si de plus, toutes les f_i sont strictement convexes alors les minima de Pareto faibles sont aussi des minima de Pareto, c'est à dire qu'on en fait

$$P_f = P$$

Notre contexte

notre but est de visualiser des ensembles des minima de Pareto. Pour ce faire nous plaçons dans le cas simple suivant:

- $X = R^2$

- les f_i sont des formes quadratiques symétriques définies positives sur R^2 :

$$x \in R^2 : \quad f_i(x) = (1/2)\, x^t A_i x - x^t a_i$$

A_i est une matrice (2,2) symétrique définie positive ($i=1,2 \ldots k$) .

a_i est un vecteur de R^2 ($i= 1,2 \ldots k$).

Ces fonctions f_i sont <u>différentiables</u> (grad $f_i(x) = A_i\, x - a_i$) et <u>strictement convexes</u> Nous sommes dans le contexte des propositions 1.2.4 et 1.2.5 ci-dessus. Donc

1) Il n'y a plus de distinction entre minima de Pareto et minima de Pareto faibles.

2) Tout minimum de Pareto x est alors caractérisé par l'existence de multiplicateurs $\lambda_1, \lambda_2, \ldots \lambda_k \geq 0$ non tous nuls tels que

$$\sum_{i=1}^{k} \lambda_i \operatorname{grad} f_i(x) = 0$$

soit ici :

$$\sum_{i=1}^{k} \lambda_i (A_i x - a_i) = 0$$

Système d'equations linéaires (2 , 2) (paramétré par les λ_i) qui s'écrit encore

$$\underbrace{\left[\sum \lambda_i A_i \right]}_{A(\lambda)} x = \underbrace{\sum \lambda_i a_i}_{b(\lambda)}$$

soit : $\quad A(\lambda)\, x = b(\lambda)$

§ 1.3 Calcul numérique de minima de Pareto quadratiques

1.3.1 k = 1 (Une seule ellipse)

Quand on prend

$$f_1=(1/2)\, x^t A x - x^t a$$

où $x \in R^2$;
A est symétrique definie positive .
Dans ce cas , le problème se simplifie à un problème usuel de minimiser une fonction de deux variables . Evidement, l'ensemble des minima de Pareto est un seul point :

Figure 1.1

En effet x est le centre de cette ellipse (Figure 1.1)

1.3.2 k = 2 (Deux ellipses)

On prend deux ellipses dans le problème de minima de Pareto $[X, f_1, f_2]$

$$f_1(x) = (1/2)x^tAx - x^ta$$

$$f_2(x) = (1/2)x^tBx - x^tb$$

$$x \in R^2$$

où A ,B sont des matrices symétriques definies positives

(2,2)

On va chercher l'ensemble de minima de Pareto D'après les propositions 1.2.4 et 1.2.5 précedentes , l'ensemble de minima de Pareto est caractèrisé par la condition suivante :

$$\exists\ \lambda_1 \geq 0,\ \lambda_2 \geq 0 \quad \text{et} \quad \lambda_1+\lambda_2 = 1$$

tel que

$$(\lambda_1 A + \lambda_2 B)\, x = \lambda_1 a + \lambda_2 b$$

Puisque A, B sont symétriques definies positives , alors $\lambda_1 A+\lambda_2 B$ aussi . Donc

$$x = (\lambda_1 A+\lambda_2 B)^{-1} (\lambda_1 a+\lambda_2 b)$$

(x) est l'ensemble de minima de Pareto

Dans ce cas , on peut calculer explicitement [2] Enfin on obtient :

$$x(\lambda) = \begin{bmatrix} \dfrac{h_1 + \lambda d_1 g_1}{1 + \lambda d_1} \\ \dfrac{h_2 + \lambda d_2 g_2}{1 + \lambda d_2} \end{bmatrix} \qquad 0<\lambda<\infty$$

où $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = A^{-1}a$; $\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = B^{-1}b$; d_1 , d_2 sont deux valeur propres de la matrice D

$$D = (Q^t)^{-1} (R^{-1})^t B R^{-1} Q^{-1}$$

où Q est une matrice unitaire ; $A = R^t R$ (factorisé A en choleski).

On sait que cela représente un morceau d'hyperbole entre les centres de ces deux ellipses.

Interpétation géométrique

selon la définition des minima de Pareto , on cherche les points (x) qui satisfont à la condition suivant :

$$\nexists\ y \in R^2 \text{ tel que } \begin{cases} f_1(y) < f_1(x) \\ f_2(y) < f_2(x) \end{cases}$$

On sait qu'en tout point x de R^2 passent deux ellipses limitant les domaines ouverts :

$$D_1 = \{\ y \in R^2 \qquad f_1(y) < f_1(x)\ \}$$

$$D_2 = \{\ y \in R^2 \qquad f_2(y) < f_2(x)\ \}$$

Deux cas possibles :

1. soit deux ellipses se tangentent en x .
2. soit deux ellipses se coupent en x .

1. Pour le cas où les ellipses se tangentent , on peut diviser en deux cas :

1-a : Deux directions de normale à x sont opposées (figure 1.2) dans ce cas :

$$y \in D_1 \quad \Leftrightarrow \quad f_1(y) < f_1(x) = c_1$$

$$y \in D_2 \quad \Leftrightarrow \quad f_2(y) < f_2(x) = c_2$$

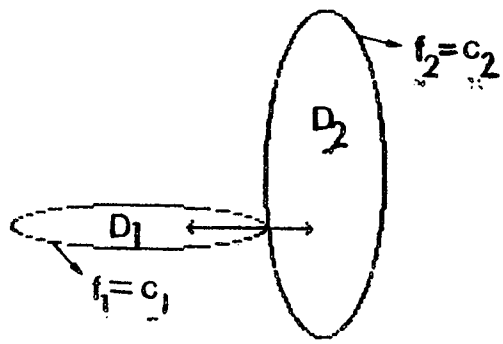


figure 1.2

alors les points y qui satisfont les deux conditions

$$f_1(y) < f_1(x)$$
$$f_2(y) < f_2(x)$$

sont dans l'intersection de D_1 et D_2, mais

$$D_1 \cap D_2 = \emptyset$$

Donc x est un minimum de Pareto.

1-b: Deux directions de normale à x sont coincidentes (figure 1.3)

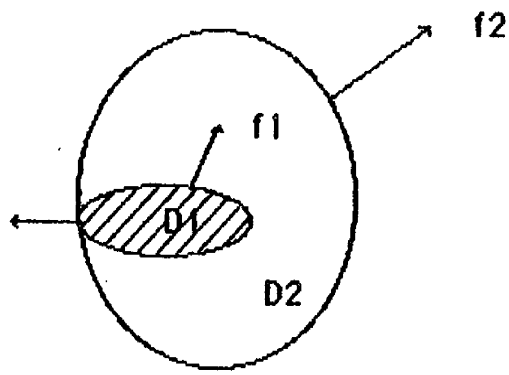


Figure 1.3

Dans ce cas :

$$y \in D_1 \quad \Leftrightarrow \quad \begin{cases} f_1(y) < f_1(x) \\ f_2(y) < f_2(x) \end{cases}$$

donc x n'est pas minimum de Pareto.

2. Pour le cas où les ellipses se coupent. (Figure 1.4)

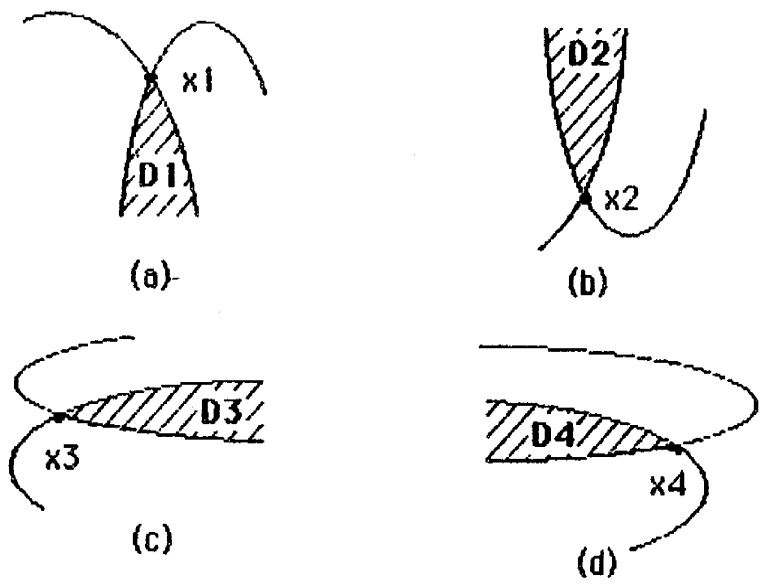


Figure 1.4

mais dans tous les cas (a), (b), (c), (d) ; on a

$$y_i \in Di \Rightarrow \begin{matrix} f_1(y_i) < f_1(x_i) \\ f_2(y_i) < f_2(x_i) \end{matrix} \qquad i = 1,2,3,4$$

donc selon la définition 1.2.1 les points x_i (i=1,2,3,4) ne sont pas des minima de Pareto.

En tous cas : l'ensemble des minima de Pareto est l'ensemble des points qui se tangentent par ces deux ellipses et ont des directions normales opposées

Un exemple graphique calculé sur ordinateur est illustré dans la Figure [1].

1.3.3 k = 3 (Trois ellipses)

On prend

$$f_1(x) = (1/2)\, x^t A\, x - x^t a$$
$$f_2(x) = (1/2)\, x^t B\, x - x^t b$$
$$f_3(x) = (1/2)\, x^t C\, x - x^t c$$

où $x \in R^2$ A, B ,C sont des matrices symétriques definies positives (2,2).

On va chercher l'ensemble des minima de Pareto , D'après les propositions 1.2.4 et 1.2.5 , l'ensemble des minima de Pareto est caractèrisé par la condition suivante :

$$x = (\lambda_1 A+\lambda_2 B+\lambda_3 C)^{-1}(\lambda_1 a+\lambda_2 b+\lambda_3 c) \qquad (2)$$

où $\lambda_1, \lambda_2, \lambda_3 > 0$ et $\lambda_1+\lambda_2+\lambda_3 = 1$

Maintenant , on veut connaitre le domaine des minima de Pareto (2) . Puisque on ne peut pas utiliser la méthode géométrique comme dans le cas précédent , alors on utilise la méthode numérique pour chercher l'ensemble des x vérifiant (2)

1) choisir trois matrices A, B , C (symétriques définies positives) et trois vecteurs a,b,c quelconques .

2) faire varier les nombres $\lambda_1, \lambda_2, \lambda_3 \geq 0$ sous la relation

$$\lambda_1+\lambda_2+\lambda_3 = 1$$

et calculer x de (2)

3) utiliser le programe graphique pour tracer tous les points x . Enfin on obtient des résultats numériques qui sont illustrés dans les Figures [2] , [3] , [4] , [5] , [6] , [9]

En fait , on a la proposition suivante :

Propositon 1.3.1

Dans le probléme de minima de Pareto (X , f_1 , f_2 , f_3) où

f_1, f_2, f_3 sont trois ellipses , si on note :

P_1 : l'ensemble des minima de Pareto de (X, f_1, f_2)

P_2 : l'ensemble des minima de Pareto de (X, f_2, f_3)

P_3 : l'ensemble des minima de Pareto de (X, f_3, f_1)

et $$H = P_1 \cup P_2 \cup P_3$$

alors l'ensemble des minima de Pareto est un domaine P délimité par H

Démonstration

D'abord on sait que $P \supset P_i \quad i = 1,2,3$.

On peut éliminer λ_3 dans (2) avec $\lambda_3 = 1-\lambda_1-\lambda_2$

$$\begin{aligned} x &= (\lambda_1 A+\lambda_2 B+\lambda_3)^{-1}(\lambda_1 a+\lambda_2 b+\lambda_3 c) \\ &= [\lambda_1(A-C)+\lambda_2(B-C)+C]^{-1}(\lambda_1(a-c)+\lambda_2(b-c)+c] \\ &= G(\lambda_1, \lambda_2) \end{aligned}$$

D :

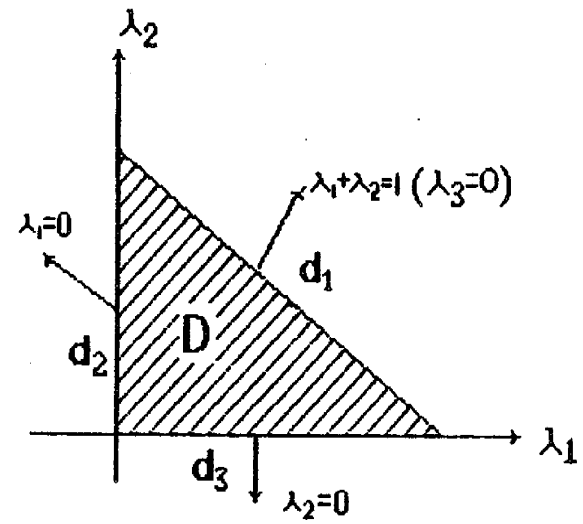


Figure 1.5

G est une fonction continue des deux variables λ_1, λ_2 où

λ_1 , λ_2 sont dans le domaine D defini suivant (Figure 1.5) donc l'ensemble des minima de Pareto de $[X f_1, f_2, f_3]$ est l'image de D transformé par $G(\lambda_1, \lambda_2)$; alors

$$d_1 \xrightarrow{G(\lambda_1, \lambda_2)} P_1(A, B)$$

$$d_2 \xrightarrow{G(\lambda_1, \lambda_2)} P_2(B, C)$$

$$d_3 \xrightarrow{G(\lambda_1, \lambda_2)} P_3(C, A)$$

Avec cette transformation, on arrive aux deux cas suivants :

—— P_i (i= 1,2,3) ne se coupent pas (Figure 1.6)

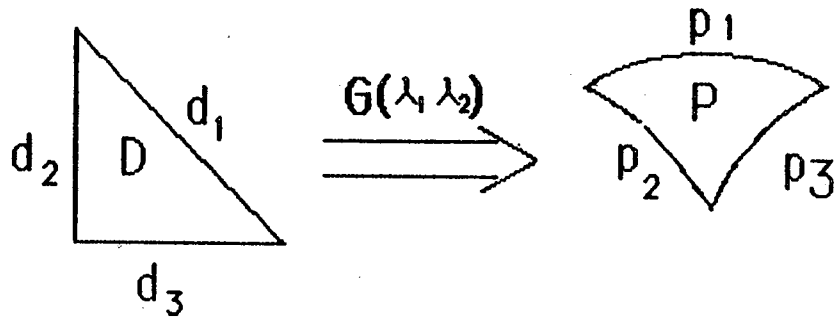


Figure 1.6

Dans ce cas : $G(\lambda_1, \lambda_2)$ est une transformation <u>bijective</u> en effet , pour un point $x_0 \in P$

1) tout point de D est envoyé dans P par G.

2) Il n'y a qu'un seul point $y \in D$ qui correspond à $x_0 \in D$; car s'il y a deux points $(\lambda_1^*, \lambda_2^*)$ et $(\lambda_1^{**}, \lambda_2^{**})$ dans D qui correspondent à $x_0 \in P$; alors par le point $x_0 \in P$, on a

$$[\lambda_1 (A-C) + \lambda_2 (B-C) + C]\, x_0 = \lambda_1 (a-c) + \lambda_2 (b-c) + c \qquad (3)$$

de plus , on peut écrire (3) comme un système linéaire

$$\begin{pmatrix} d_{11}(x_0) & d_{12}(x_0) \\ d_{21}(x_0) & d_{22}(x_0) \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} d_1(x_0) \\ d_2(x_0) \end{pmatrix} \quad (4)$$

donc on a deux solutions $(\lambda_1^*, \lambda_2^*)$ et $(\lambda_1^{**}, \lambda_2^{**})$ dans ce système ; mais pour le système linaire, s'il y a deux solutions, alors il y a des solutions infinies. En effet dans (4) il y a des solutions :

$$a\lambda_1 + b\lambda_2 = c$$

Si on prend

$$\lambda_1 = 0 \quad \text{alors} \quad \lambda_2 = c / b$$

$$\lambda_2 = 0 \quad \text{alors} \quad \lambda_1 = c / a$$

donc, les points $(0, c/b)$ et $(c/a, 0)$ aussi correspondent au point x_0, c'est à dire dans les droites d_2 et d_3, il y a deux points qui correspondent au même point $x_0 \in D$. cela est contradictoire avec le fait que P_i ne se coupent pas. Illustration numérique : Figures [3], [4].

—— P_i se coupent en au moins un point. c'est à dire il y a au moins un point dans P qui correspond à deux points dans D. D'après l'analyse précédente, s'il y a deux points dans D correspondant à un point dans P, alors il y a une droite

$$a\lambda_1 + \lambda_2 b = c$$

dans D qui correspond à ce point, cette droite coupe D en deux parties (Figure 1.7)

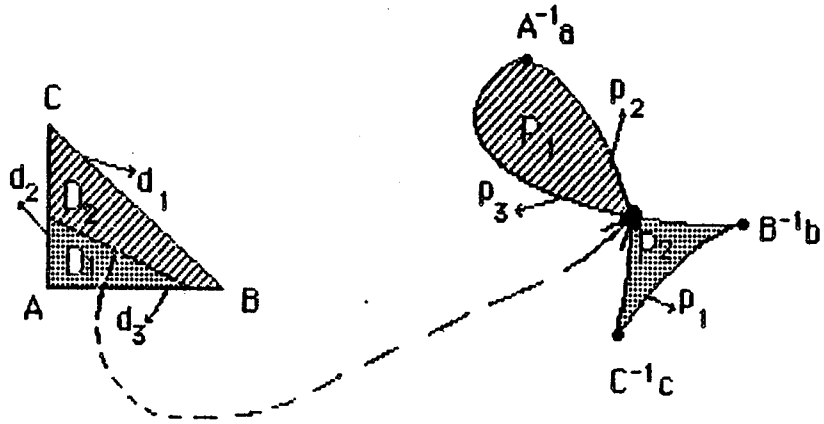


Figure 1.7

Ces deux parties dans D se transforment bijectivement en deux parties dans P (sauf la droite qui correspond à un point).

Si trois hyperboles se coupent en deux points , alors il y a deux droites dans D qui correspondent respectivement à ces deux points et ces deux droites couperont D en trois parties (Figure 1.8)

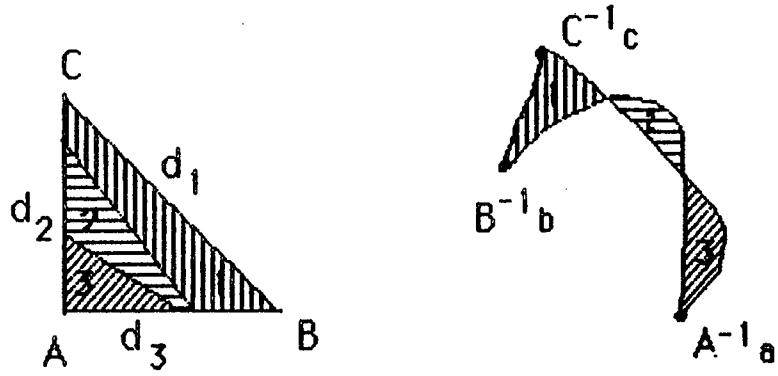


Figure 1.8

alors ces trois parties dans D se transforment bijectivement en trois parties dans P (sauf les deux droites qui correspondent chacune à un point).

Illustations numériques : Figure [5] , Figure [6] et Figure [9].

1.3.4 k = 4 (Quatre ellipses)

On prend quatre ellipses comme multi-critères dans le problème de minima de Pareto de $[X, f_1, f_2, f_3, f_4]$

$$f_1(x) = (1/2)\, x^t A x - x^t a$$
$$f_2(x) = (1/2)\, x^t B x - x^t b$$
$$f_3(x) = (1/2)\, x^t C x - x^t c$$
$$f_4(x) = (1/2)\, x^t D x - x^t d$$

où $x \in R^2$; A , B , C , D sont des matrices symétriques definies positives (2 , 2).

Comme dans les cas précédents , l'ensemble des minima de Pareto est l'ensemble de x qui satisfont la formule ci-dessous :

$$x = (\lambda_1 A + \lambda_2 B + \lambda_3 C + \lambda_4)^{-1} (\lambda_1 a + \lambda_2 b + \lambda_3 c + \lambda_4 d)$$

où $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ et $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.

On calcule x numériquement en variant des λ_i (i = 1, 2, 3 ,4). Le résultat graphique est illustré dans la Figure [12].

Dans ce cas : on trouve une chose intéressante :

L'ensemble des minima de Pareto pour quatre ellipses est la réunion des ensembles de minima de Pareto pris trois à trois entre ces quatre ellipses.

1) Lorsqu'on passe d'une ellipse à deux

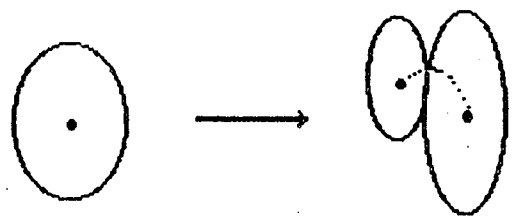


Figure 1.9

il s' introduit des nouveaux points (Figure 1.9) . On passe d'un point à une courbe .

2) Lorsqu'on passe de deux ellipses à trois

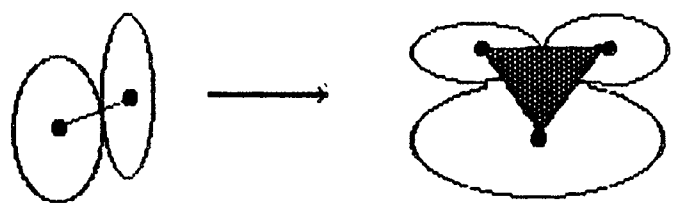


Figure 1.10

il s'introduit aussi des nouveaux points (Figure 1.10) ; l'ensemble des minima de Pareto se transforme d'une courbe en un domaine. Mais lorsqu'on passe de trois ellipses à quatre , il n'y a pas de nouveaux points qui sont introduits. Pourquoi ? pour répondre à cette question , on introduit le théorème de Carathéodory (que m'a suggéré P.J. Laurent)

Théorème (Carathéodory)

Soit A est un sous - ensemble d'un espace vectoriel linaire de dimension-n, alors tous les points dans la fermeture convexe de A sont exprimables comme une combinaison convexe de (au plus) n + 1 éléments de A

En utilisant ce théorème , on a la proposition suivante :

Proposition 1.3.2

Si on note :

H : l'ensemble des minima de Pareto de quatre ellipses ;

G : La réunion des ensembles des minima de Pareto pris trois à trois entre ces quatre ellipses.

alors $H = G$

Démonstration :

Si $x_0 \in H$ on a

$$\sum_{i=1}^{4} \lambda_i \text{ grad } f_i(x_0) = 0$$

C'est à dire

$x_0 \in H \Leftrightarrow 0 \in$ convexe (grad $f_i(x_0)$) (i = 1 , 2 , 3 , 4).

On prend [grad $f_i(x_0)$] (i = 1, 2 ,3 ,4) comme A dans le théorème de Carathéodory (pour le cas : n = 2).

D'après le théorème de Carathéodory : $\exists \ \lambda_1^*, \lambda_2^*, \lambda_3^* \geq 0$ et $\lambda_1^* + \lambda_2^* + \lambda_3^* = 1$

tel que

$$0 = \sum_{k=1}^{3} \lambda_k^* \text{ grad } f_{i_k}(x_0) \qquad i_k \in \{1,2,3,4\}$$

donc $x_0 \in G$ C'est à dire on a

$$x_0 \in H \Rightarrow x_0 \in G \qquad (H \subset G)$$

évidement , on a toujours :

$$x_0 \in G \Rightarrow x_0 \in H \qquad (G \subset H)$$

Donc : $G = H$

§ 1.4 Conclusion

Dans le probléme des minima de Pareto $(X, f_1, \ldots, f_k)$ où f_i sont des fonctions quadratiques definies positives sur R^2 on a conclusion suivante

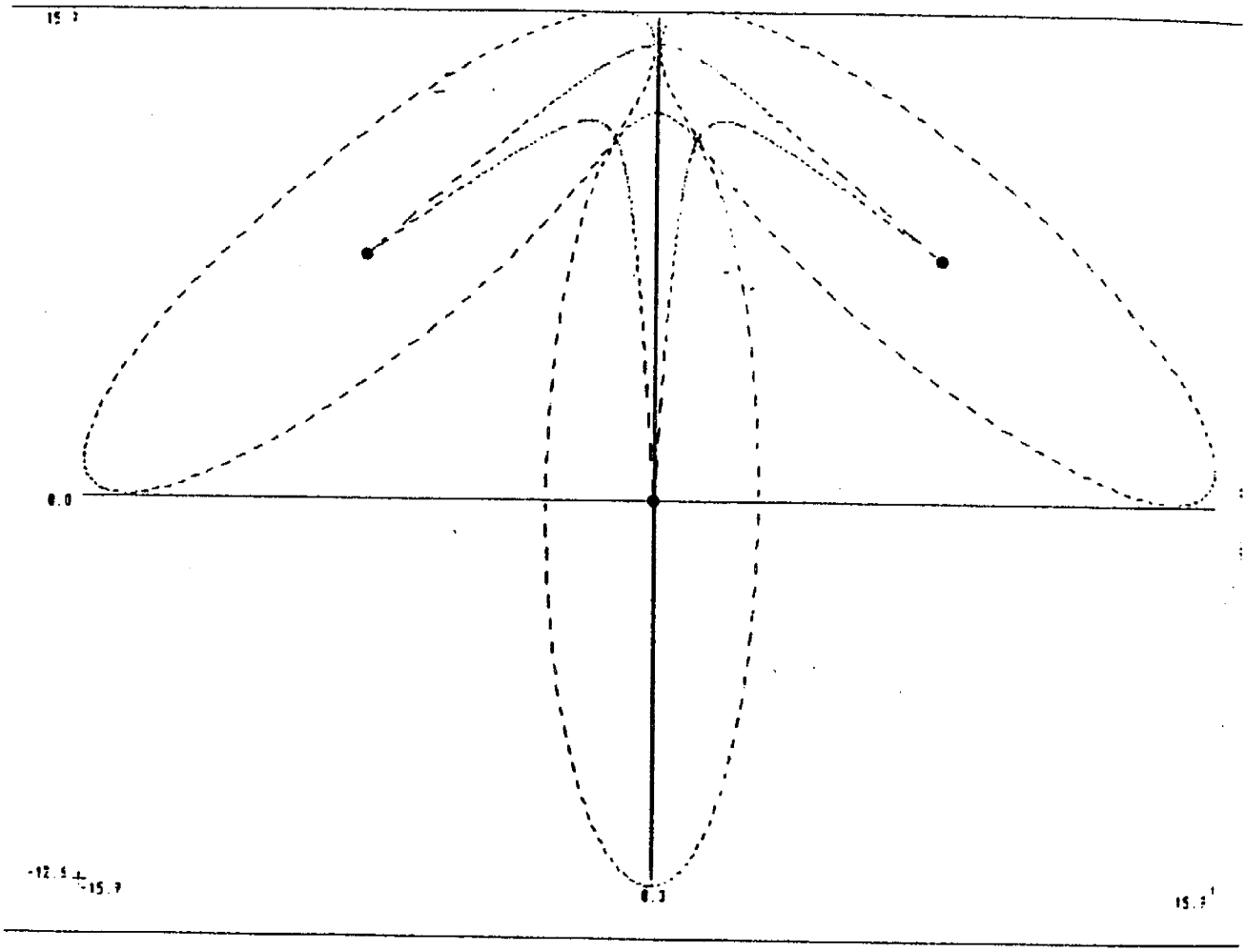
Pour k = 1 : L'ensemble des minima de Pareto est un seul point . (le centre de cette ellipse)

k = 2: L'ensemble des minima de Pareto est un morceau de courbe (une hyperbole reliant les centres de ces deux ellipses)

k = 3 : L'ensembles des minima de Pareto est un domaine délimité par les trois hyperboles (ces trois hyperboles sont les ensembles des minima de Pareto de ces trois ellipses prises deux à deux).

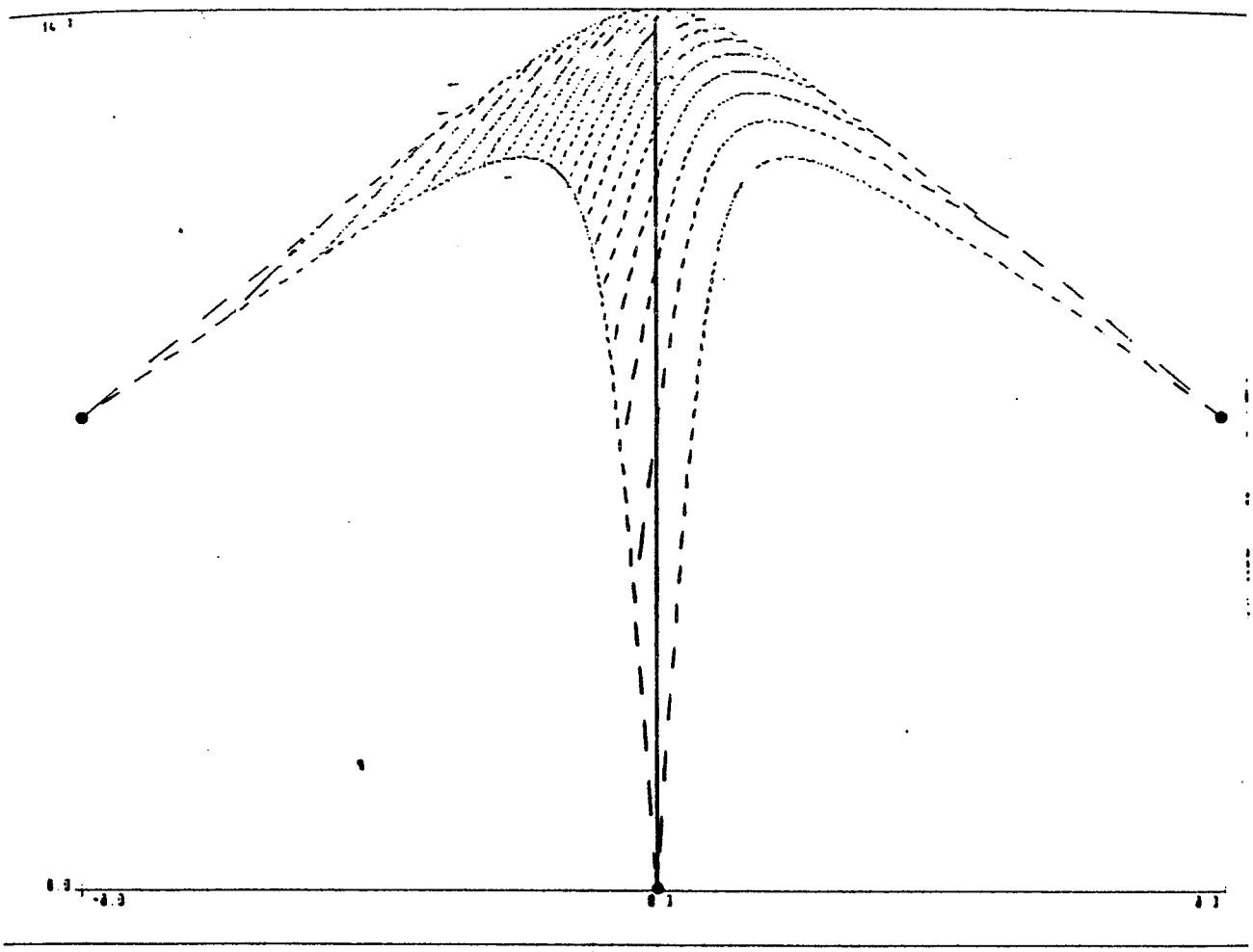
k > 3 : L'ensemble de minima de Pareto est la réunion des minima de Pareto de ces k ellipses prises trois à trois .

la généralisation à k formes quadratiques sur R^2 est évidente : pour k > n+1 , il suffit de prendre la réunion des minima de Pareto des k formes prises n+1 à n+1 .



L'ensemble des minima de Pareto pour deux ellipses est l'ensemble des points qui se tangentent par ces deux ellipses et ont des directions normales opposées.

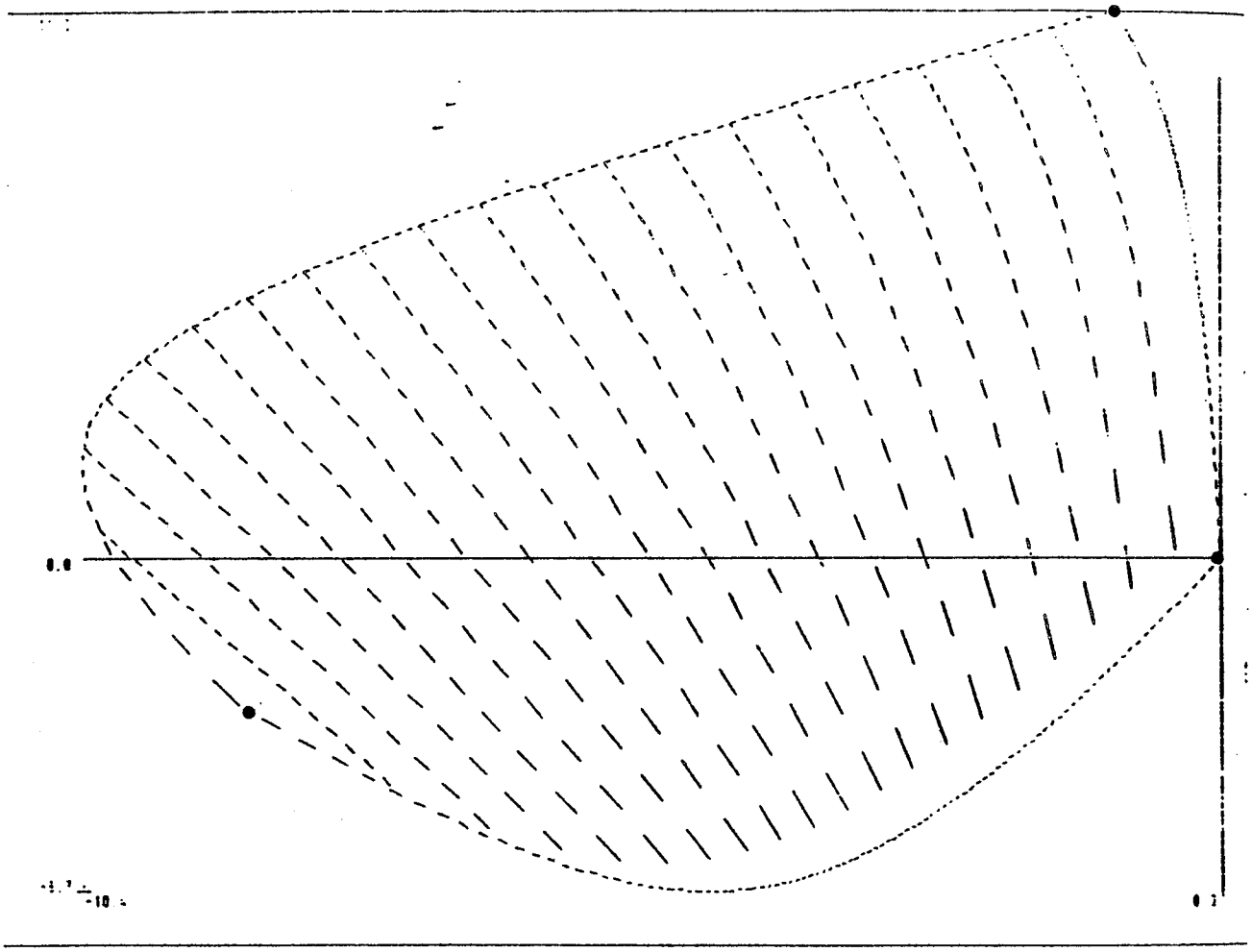
Figure [1]



Les trois hyperboles ne se recoupent pas :

L'ensemble des minima de Pareto pour trois ellipses est un domaine délimité par les trois hyperboles . (Ces trois hyperboles sont les ensembles des minima de Pareto des trois ellipses prises deux à deux).

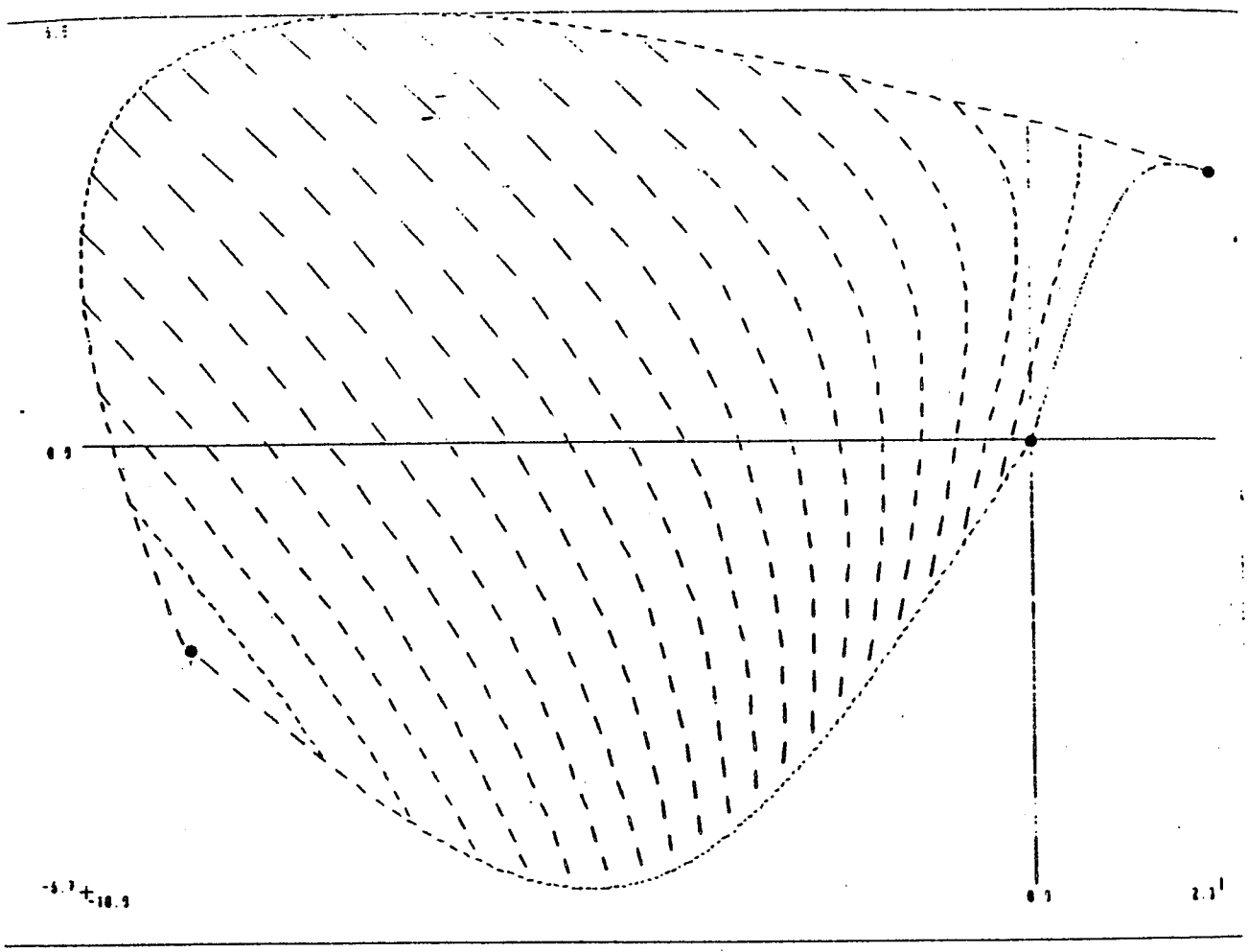
Figure [2]



Les trois hyperboles ne se coupent pas.

L'ensemble des minima de Pareto est un domaine délimité par trois hypreboles.

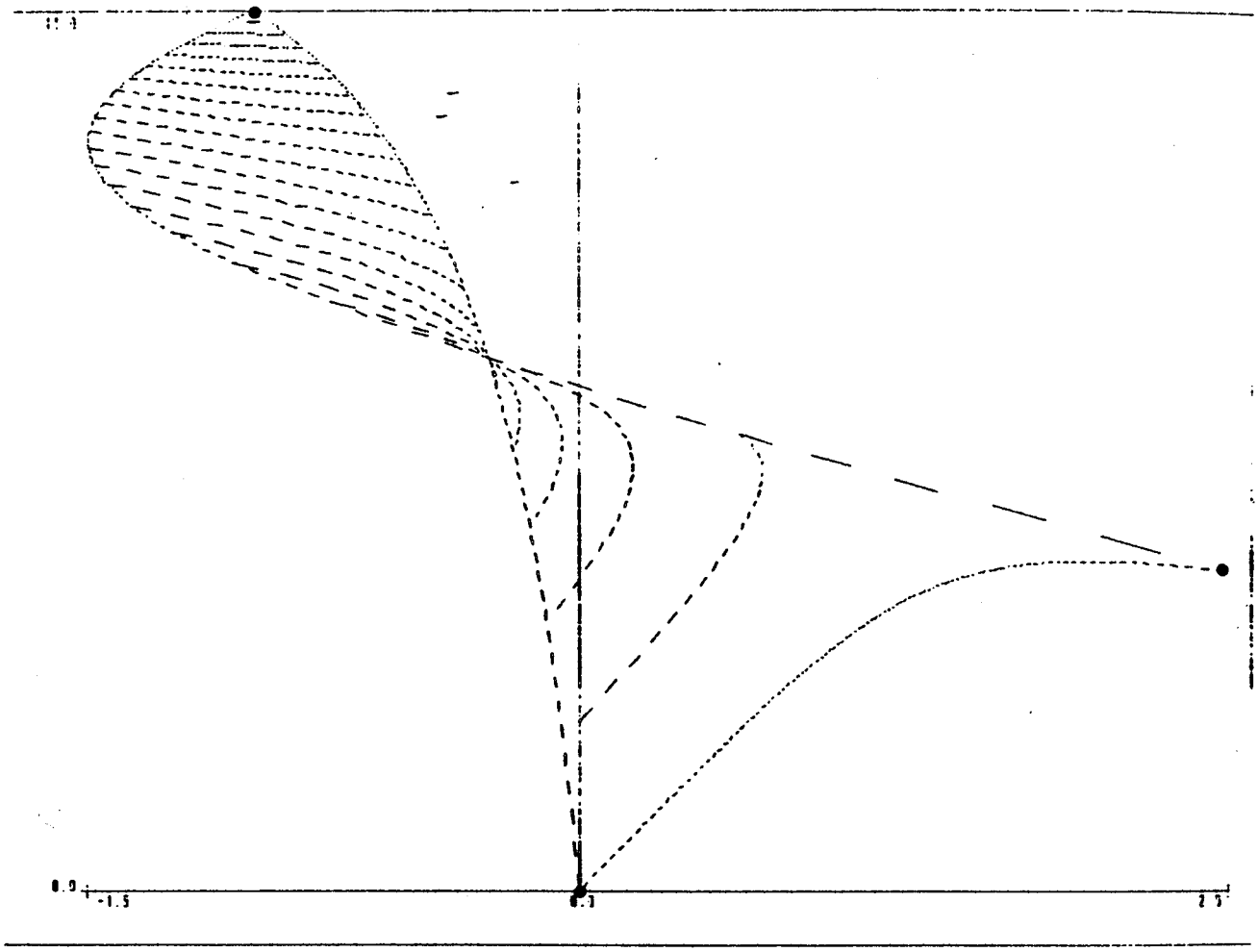
Figure [3]



Le cas : trois hyperboles ne se coupent pas.

L'ensemble des minima de Pareto pour trois ellipses est un domaine délimité par trois hyperboles.

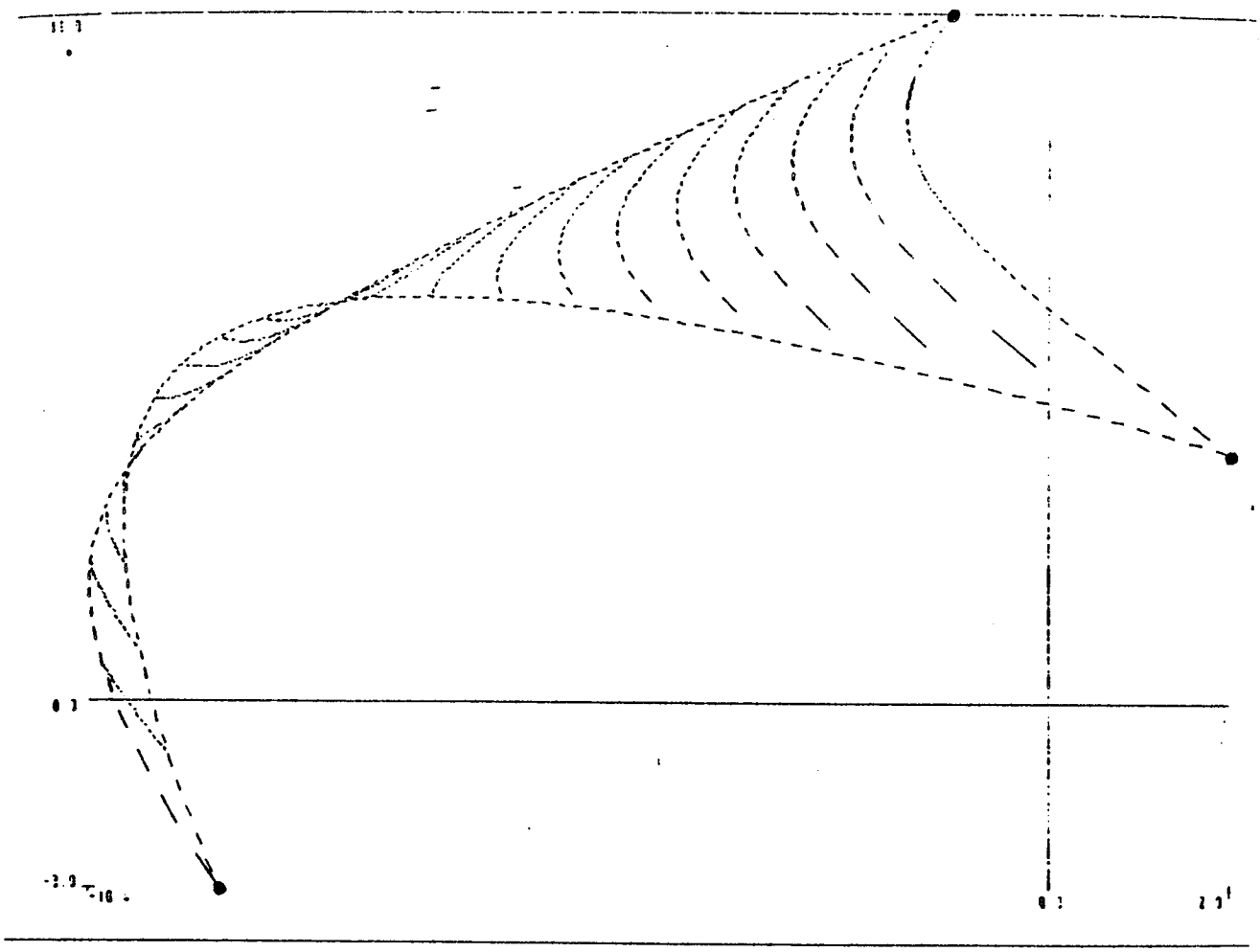
Figure [4]



Le cas : trois hyperboles se coupent en un points.

Avec les Figures [10] et [11] , il indique que l'ensemble des minima de Pareto pour trois ellipses est un domaine délimité par trois hyperboles.

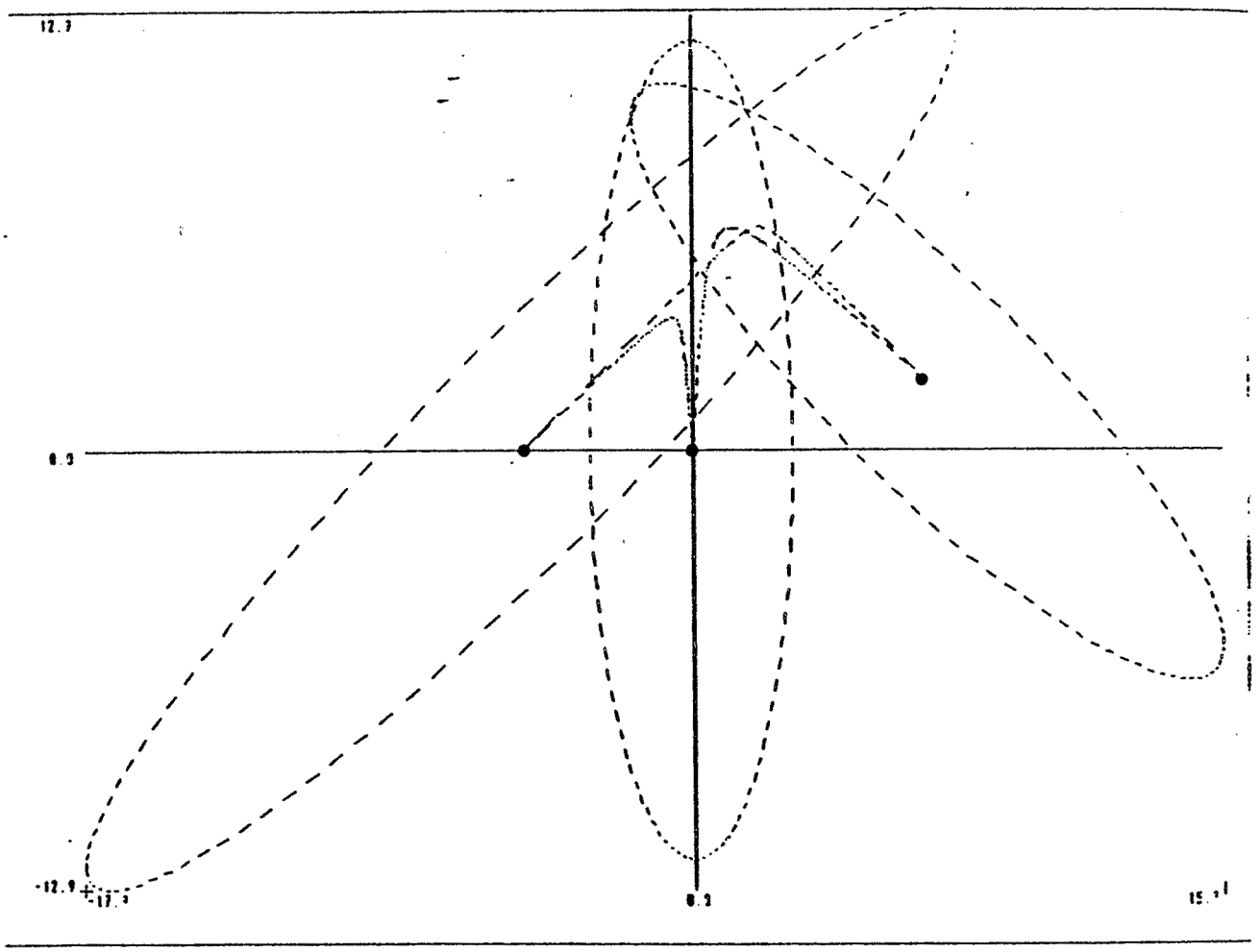
Figure [5]



Le cas : trois hyperboles se coupent en deux points ;

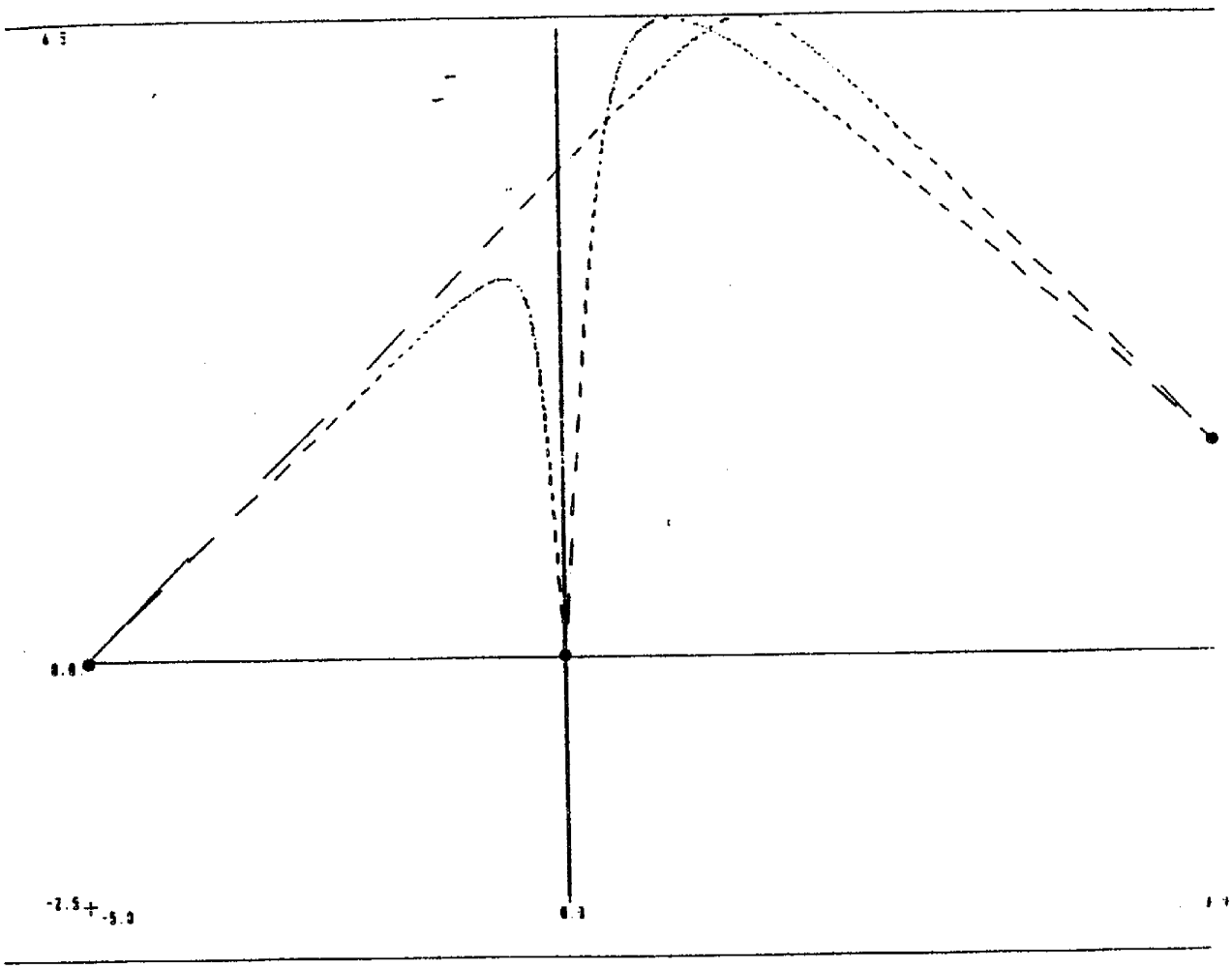
Avec les Figures [10] et [11], il indique que l'ensemble des minima de Pareto pour trois ellipses est un domaine délimité par ces trois hyperboles.

Figure [6]



Les trois hyperboles se coupent en deux points.

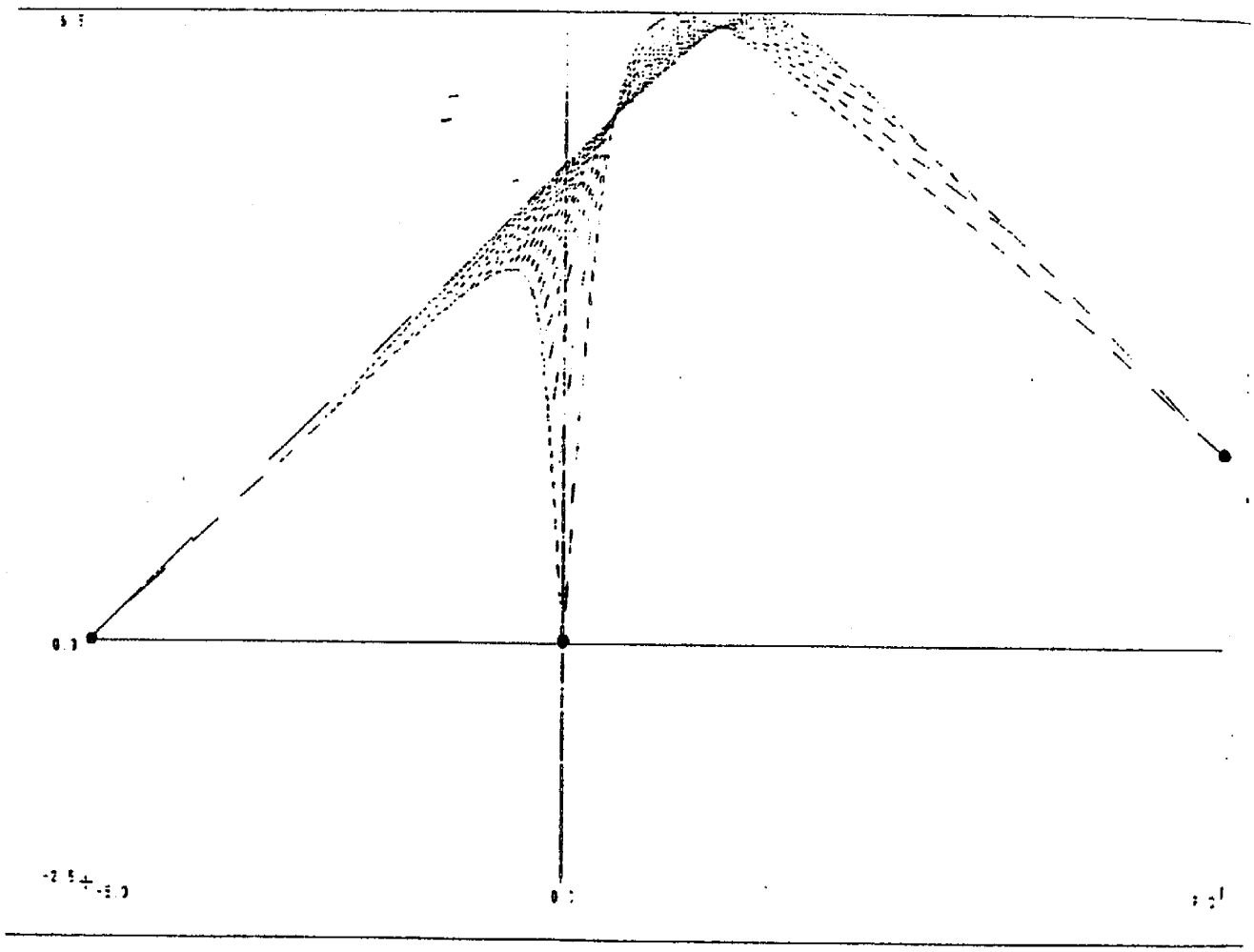
Figure [7]



Le cas : trois hyperboles se coupent .

Avec Figure [9] , il indique que l'ensemble des minima de Pareto pour trois ellipses est un domaine délimité par trois hyperboles .

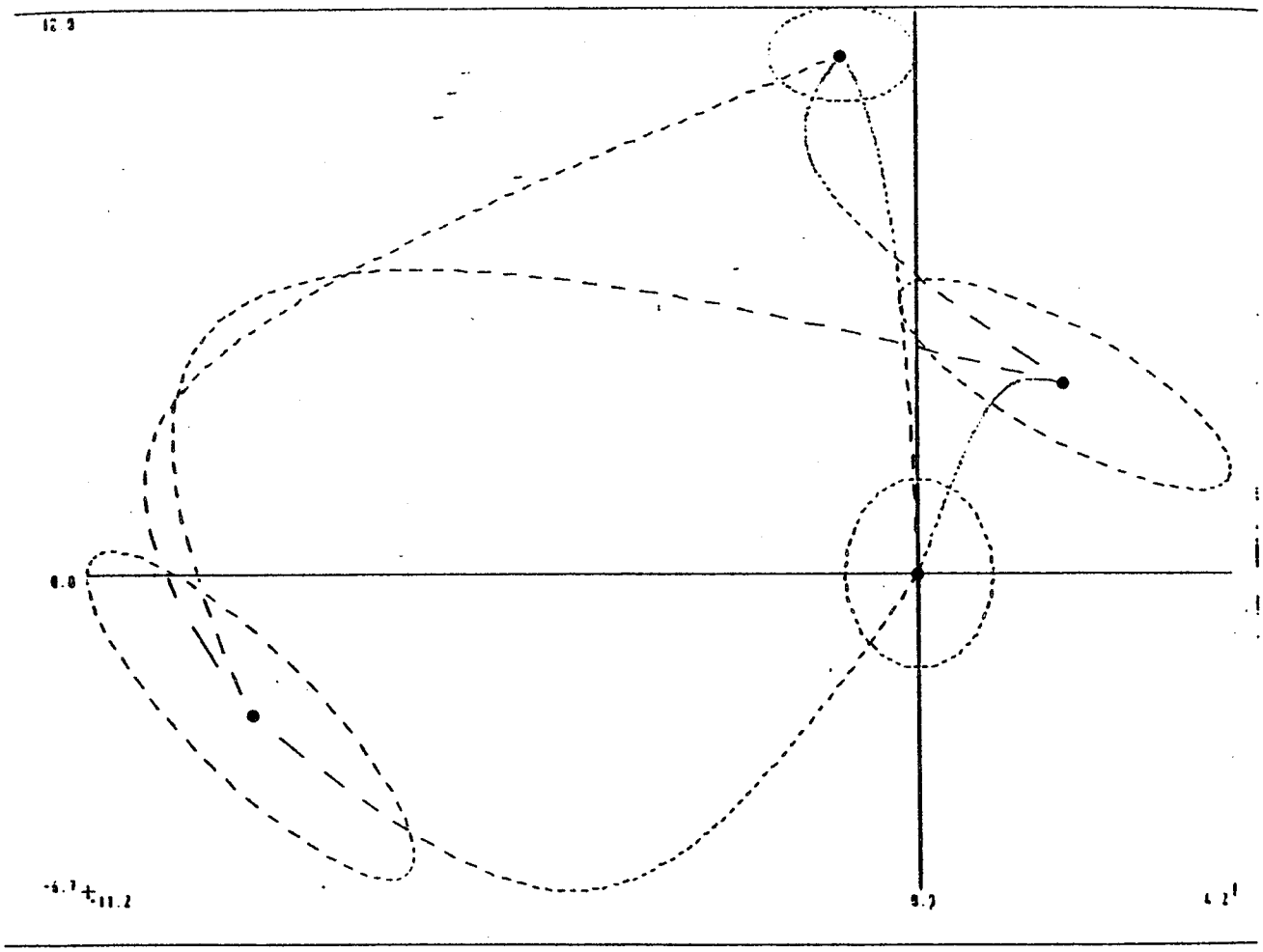
Figure [8]



Le cas : trois hyperboles se coupent en deux point .

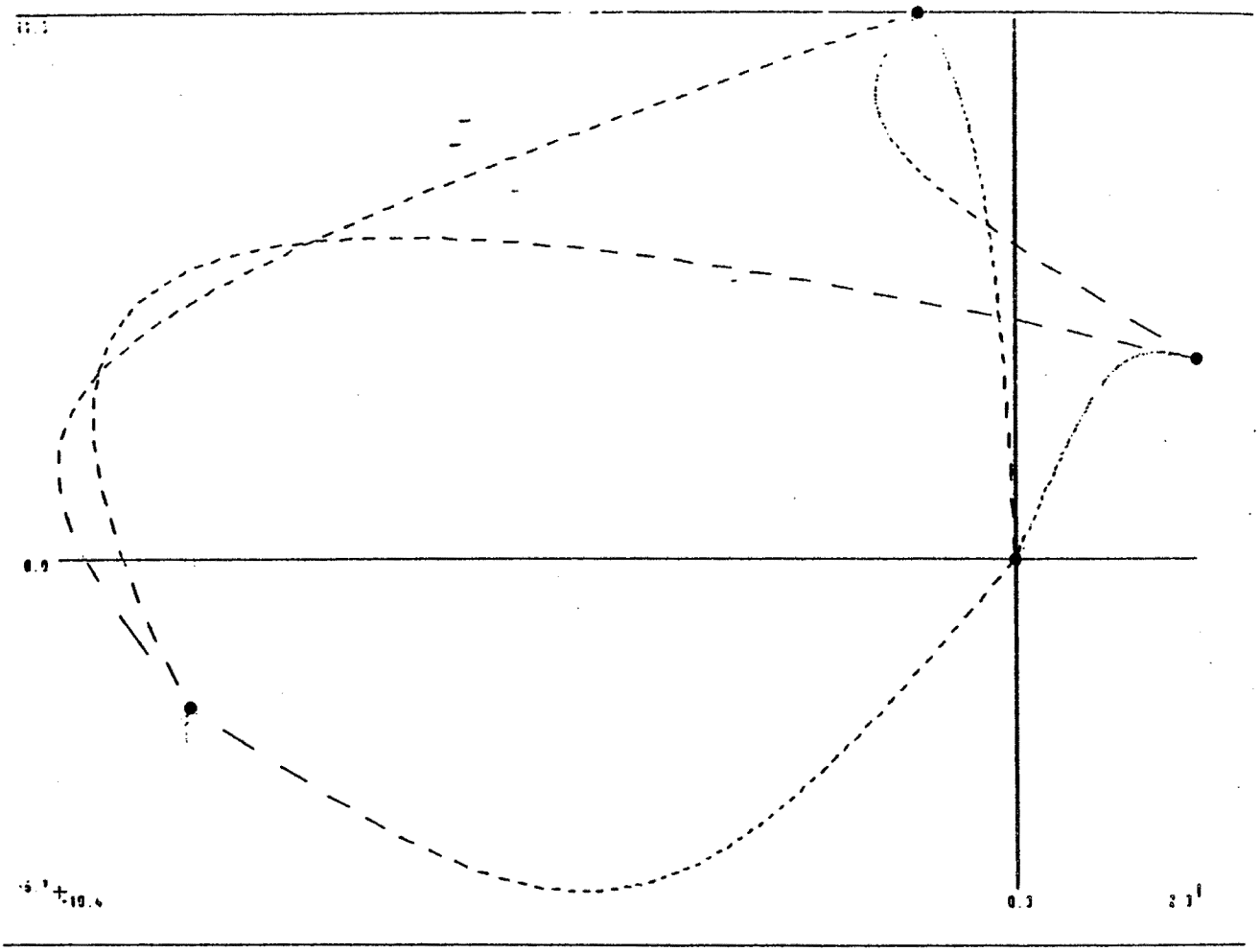
Avec les Figures [7] et [8], il indique que l'ensemble des minima de Preto pour trois ellipses est un domaine délimité par trois hyperboles

Figure [9]



Avec les Figures [3] ,[4] , [5] , [6] , [11] , il indique que l'ensemble des minima de Pareto pour quatre ellipses est la réunion des minima de Pareto de ces quatre ellipses prises trois à trois.

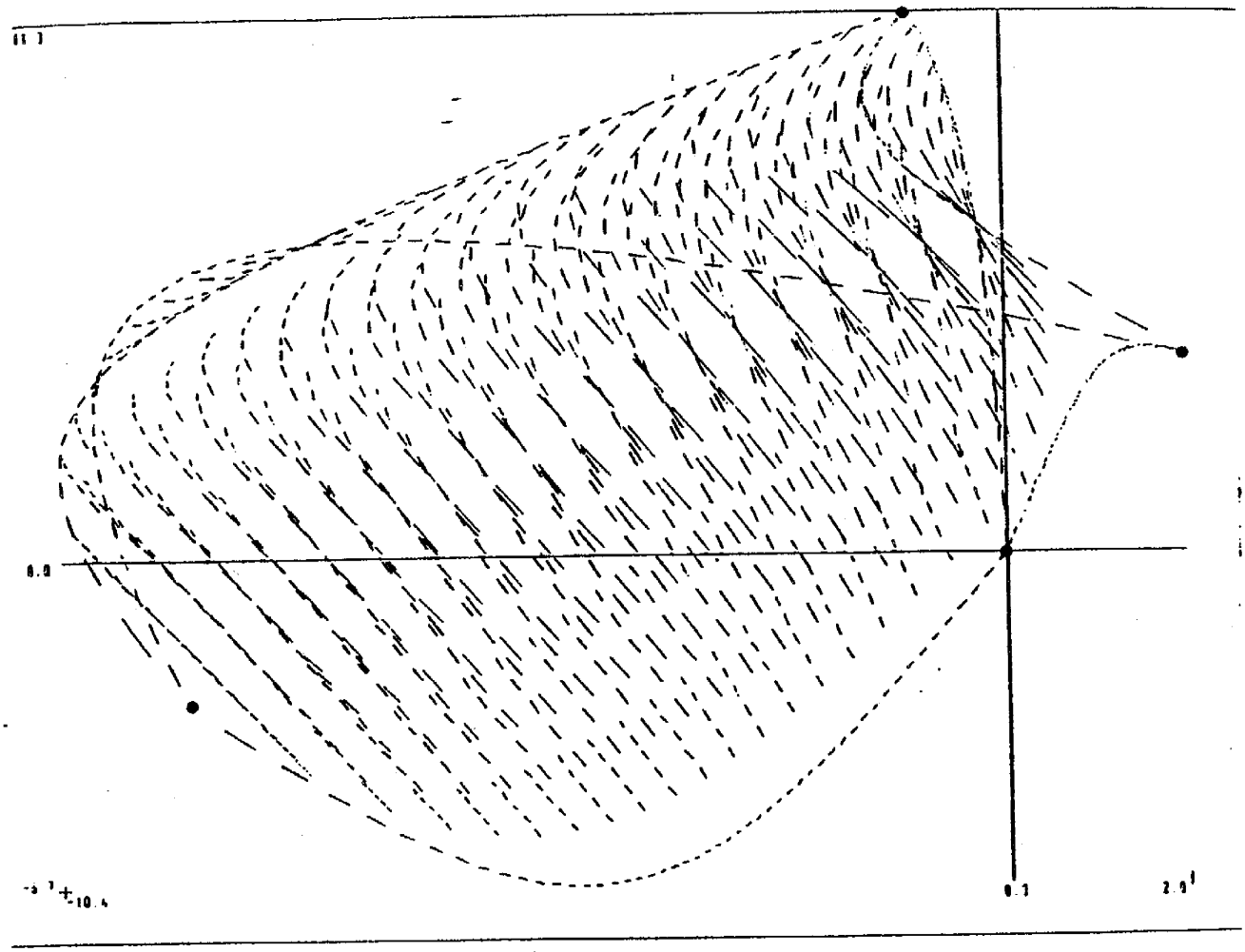
Figure [10]



Les quatre ellipses :

Avec les Figures [3] , [4] , [5] , [6] , [10] , [12] , il indique que l'ensemble de minima de Pareto pour quatre ellipses est la réunion des minima de Pareto de ces quatre ellipses prises trois à trois.

Figure [11]



Quatre ellipses :

Avec les Figures [3], [4], ,[5], ,[6], [10], [11], il indique que l'ensemble des minima de Pareto pour quatre ellipses est la réunion des minima de Pareto de ces quatre ellipses prises trois à trois.

Figure [12]

Références

[1]. H. MOULIN , F. FOGELMAN-SOULIE : La convexité dans les mathématiques de la décision.
(Hermann)

[2]. F . ROBERT : Meilleure approximation en norme vectorielle et minima de Pareto.
Rapport de Recherche N^0 392 IMAG (octobre 1983) et MAN^2 (ex;RAIRO) Vol 19 N^0 1 1985 pp 89-110.

Chapitre Deux

Entraînement du Rythme Respiratoire : Simulation par un modele explicite

§ 2.1 Introduction :

Dans le problème de l'entraînement du système respiratoire par une simulation périodique, plusieurs modèles expliquant ce phénomène ont été élaborés (1 à 4). [10] . Dans ce chapitre , Nous nous proposons de réduire l'un de ces modèles (fondé sur des donneés physiologiques expérimentale [1],[2],[3]) à son expression minimale , contenant les éléments essentiels du mode d'action de la stimulation. Cette réduction nous permettra de dégager de manière explicite la fonction de réponse du système à la stimulation et ce , dans le cas de plusieurs modes d'action différents , prenant en compte les effets de stimulations répétées , soit en les cumulant , soit en ne retenant que l'effet maximum.

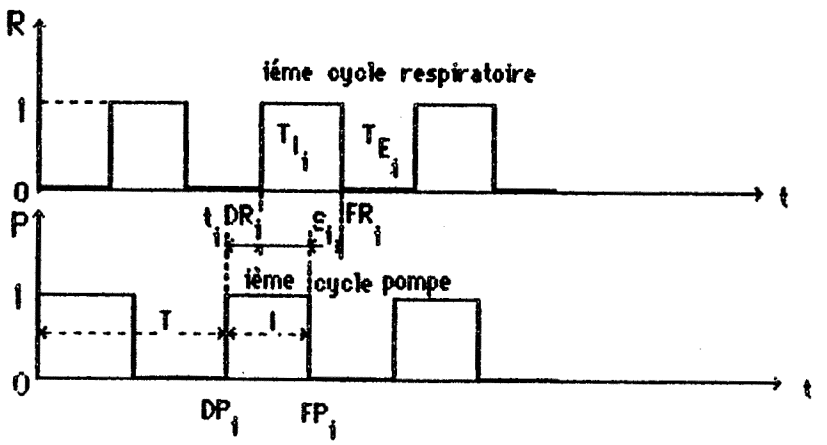
Dans chacun de ces cas , le diagramme de bifurcation correspondant aux entraînements harmonique et fractionnaires sera analysé. La confrontation aux données expérimentales, la généralisation à des modèles plus complexes (utilisant les notion développées dans [10]) et la comparaison à des approches voisine (cf [4], [5] et l'article de Belair et Glass dans [11]) seront enfin abordées.

§ 2.2. Modèle Explicite Minimum.

Le modèle booléen introduit dans les article [1] et [3] utilise deux variables essentielles R et P , représentant respectivement l'état respiratoire de l'animal entraîné et celui de la pompe réalisant l'entraînement : la variable R prend valeur 1 , lorsqur l'animal est en " inspiration " , c'est à dire lorsque le nerf phrénique commandant le diaphragme manifeste une activité et prend valeur 0 lorsque l'animale est en " expiration " , c'est à dire durant le silence phrénique , ces inspirations et expirations étant virtuelles, puisque le nerf phrénique a été coupé et donc ne conduit plus les ordres respiratoires centraux au diaphragme.

La variable P , quant à elle , prend la valeur 1 , lorsque la pompe ou respirateur externe insuffle les poumons de l'animal , c'est à dire en phase d'"inflation" et prend la valeur 0 , lorsque la pompe ne fonctionne pas , c'est à dire en phase de "déflation" passive de poumons.

Le schéma ci - dessous résume les rapports existant entre les variables R et P :



Définition des variables " retard " t_i et s_i

Figure 2.1

Nous utiliserons , pour décrire ces rapports , deux

variables "retard" auxiliaires t_i et s_i, égales respectivement à l'intervalle de temps séparant le début DP_i du ième cycle pompe du début DR_i du ième cycle respiratoire :

$$t_i = DR_i - DP_i$$

et à l'intervalle de temps séparant la fin FP_i du ième cycle pompe de la fin FR_i du ième cycle respiratoire :

$$S_i = FR_i - FP_i$$

Si nous désignons par ${}^{T}I_i$ et ${}^{T}E_i$ respectivement les durées de la iéme inspiration et de la iéme expiration et par T et I respectivement les durées du cycle pompe et de l'inflation supposées constantes dans une expérience d'entraînement, alors les relations (1) sont vérifiées par les variables ci-dessous :

$$\text{et} \begin{cases} t_i + {}^{T}I_i - s_i - I = 0 \\ \\ s_i + {}^{T}E_i - t_{i+1} - (T-I) = 0 \end{cases} \qquad (1)$$

Le problème de l'entraînement consiste à étudier le comportement asymptotique du système dynamique :

$$\begin{bmatrix} s_i \\ t_i \end{bmatrix} = F \begin{bmatrix} s_{i-1} \\ s_{i-1} \end{bmatrix} ,$$

où F est issue de la modélisation.

Supposons, dans un premier temps, que cette étude soit faite et désignons par ${t^T}_i$ et ${s^T}_i$ les variables comprises respectivement dans les intervalles]I-T,I] et]-I,T-I] et égales respectivement à t_i et s_i modulo T :

$$t_i = t^T_i + d_i T \quad , d_i \in \mathbb{N}$$
$$s_i = s^T_i + f_i T \quad , f_i \in \mathbb{N}$$

Alors , l'entraînement fractionnaire p/q (p cycles respiratoires pour q cycles pompe) peut être défini de la manière suivante :(t^T_i , s^t_i) est une suite admettant p points d'accumulation et cyclant asymptotiquement sur ces p points et les suites $(d_{i+p}-d_i)$ et $(f_{i+p} - f_i)$ tendent vers p-q , lorsque i tend vers plus l'infini .

Nous disposons donc d'un critère simple permettant de caractériser les entraîement p/q , pour p , q entiers quelconques

Nous allons présenter maintenant un mécanisme caricatural d'action de la stimulation des récepteurs d'étirement ; en fin d'inflation , la différence ΔV entre volume attendu (correspondant aux ordres de respiration virtuelle données par le phrénique) et volume effectif (dû à l'inflation par la pompe du respirateur externe) provoque deux types d'action , en fonction de son moment de survenue (mécanisme de Hering-Breuer , effets Knox et Clarke - von Euler) :

- un arrêt de l'inspiration virtuelle attendue , car le volume attendu ne doit pas dépasser une courbe d'arrêt représentée ici par un segment de droite dans le temps local de la respiration virtuelle (cf Fig. 2.2) ; de plus , la durée de l'inspiration attendue doit rester comprise entre deux bornes m et M .

- un allongement de l'expiration virtuelle attendue , si cette différence ΔV apparaît avant une zone finale (les trois dernières dixièmes) de l'expiration considérée comme zone réfractaire (cf Fig. 2.3) .La durée de l'allongement dépend de ΔV et de l'instant de survenue dans le temps locale de la respiration virtuelle .

Si plusieurs fins d'inflation , c'est à dire plusieurs

stimulations d'étirement, surviennent au cours de la même expiration virtuelle, on peut prendre en compte leur action conjointe de quatre manières :

1) en retenant l'action du ΔV maximum

2) en retenant l'action du stimulus qui provoque l'allongement expiratoire maximum

3) en sommant les allongements provoqués par les stimuli intervenant au cours de cette même expiration virtuelle

4) en " composant" ces allongements, c'est à dire en prenant l'allongement du deuxième stimulus calculé à partir du temps local de l'expiration allongée par le premier stimulus, puis l'allongement du troisième stimulus calculé dans le temps local de l'expiration allongée pour la deuxième fois ,... jusqu'à l'allongement du dernier stimulus calculé dans le temps local de l'avant-dernière expiration allongée.

La durée de l'expiration virtuelle attendue est, en l'absence de stimulation, donnée par la loi de Clarke - von Euler :

$$T_{E_i} = n\, T_{I_i} + p \qquad (2)$$

c'est donc une fonction affine de la durée inspiratoire ; elle est augmentée, en cas de stimulation avant la zone réfractaire, d'un allongement égal à :

$$\frac{\Delta V}{V(T_{I_i})} \times \left(e^{\frac{t_s}{-T_{E_i}"}} + f \right) ,$$

où t_s est le temps local de la stimulation dans l'expiration de durée " T_{E_i}" (éventuellement allongée par des stimulations antérieurs, dans le mécanisme 4).

Tous calculs faits , la fonction de réponse F du système donnant $\begin{bmatrix} t^T_i \\ s^T_i \end{bmatrix}$ en fonction de $\begin{bmatrix} t^T_{i-1} \\ s_{i-}{}^T_1 \end{bmatrix}$ se ramène, grâce à l'utilisation des équations (I), à:

$$\begin{bmatrix} t_i^T \\ s_i^T \end{bmatrix} = F \begin{bmatrix} t^T_{i-1} \\ s^T_{i-1} \end{bmatrix} = \begin{bmatrix} t^T_{i-1} + {}^T I_i + {}^T E_i \\ s^T_{i-1} - ({}^T I_i + {}^T E_i) \end{bmatrix} \pmod T \quad (3)$$

Comme la quantité ${}^T I_i + {}^T E_i$ est fonction de t_i^T seulement (grâce à l'équation (2)), l'étude de (3) se ramène à celle de :

$$t_i^T = f(t_i^T) = t_i^T + {}^t I_i + {}^T E_i \pmod T \quad (4)$$

Tous calculs faits, la prise en compte des mécanismes 1, 2, 3, 4 conduit aux équations ci-dessous, où $x_i = t_i^T$:

- $x_0 = 0$

- $z_i = {}^t I_i = \inf (\sup (m , \alpha x_i + \beta), M)$, où, si Q désigne la pente de l'inflation et V_M le volume atteint en M sur la courbe d'arrêt de pente a : $\alpha = Q/(a-Q)$ et $\beta = (V_M - aM)/Q - a$

- si $(I - x_i - z_i) / (nz_i + p) \in [0 , 1]$, $x_{i+1} = x_i + (n-1)z_i + p \pmod T$ (ceci correspond au cas où aucune stimulation ne survient au cours de l'expiration attendue).

- si s désigne le plus petit entier tel que :
$(I - x_i - z_i + sT) / (nz_i + p) \in [0 , 1]$, ${}^x i_M = x_i + (n+1)z_i + p + A \pmod T$

$$\text{avec } A = \left(\sup_{j=i}^{i+s} (\sup (0 , B_j)) \right) \times c_j \text{ dans le mécanisme 1}$$

$$A = \sup_{j=i}^{i+s} \left(\sup(0, B_j) \times c_j \right) \quad \text{dans le mécanisme 2}$$

$$= \sum_{j=i}^{i+s} \sup(0, B_j) \times c_j \quad \text{dans le mécanisme 3,}$$

où l'on a posé :

$$B = QD_j + \left((a(z_i - M) + V_M) / (nz_i + p) \right) (D_j - x_i + T(j-i) - z_i(n+1) - p),$$

avec $D_j = \inf(1, z_i + 0.7(nz_i + p) + x_i - (j-i)T)$

et $C_j = \left(e\left((D_j - x_i + (j-i)T - z_i) / (nz_i + p) \right) + f \right) / (a(z_i - M) + V_M)$

- dans le mécanisme (4), on pose :

$z_i = \inf(\sup(m, \alpha x_i + \beta), M)$ et si $(1 - x_i - z_i^{(1)})/(nz_i^{(1)} + p \in [0, 1]$, on calcule $A^{(1)} = B^{(1)} \cdot C^{(1)}$ par la formule ci-dessus correspondant à un seul stimulus.

Ensuite, on considère $z_i^{(2)} = \inf(\sup(m, (nz_i^{(1)} + A^{(1)})/n, M)$ c'est à dire la durée de l'inspiration qui aurait donné naissance à l'expiration allongée par $A^{(1)}$, à l'aide de la formule (2).

Si $(1 - x_i - z_i^{(2)}) / (nz_i^{(2)} + p) \in [0, 1]$, on répète le calcul, et on itère ce raisonnement jusqu'à ce que $(1 - x_i - z_i^{(k)}) / (n z_i^{(k)} + p)$ n'appartienne plus à $[0, 1]$; on obtient alors :

$$x_{i+1} \equiv x_i + (n+1) z_i^{(k)} + A^{(k)} \pmod{T}$$

Nous donnons , dans les Figures 2.4 , 2.5 , 2.6 , 2.7 , 2.8 et 2.9 des exemples de la fonction f à itérer dans les quatre mécanismes décrits ci-dessus : l'allure générale est celle d'une fonction de [0,T] vers [0,T] bimodale avec discontinuité (cf .-[1] pour la démonstration du caractère bimodal et discontinu).

§ 2.3 La Programation :

- program 1 (PGRESP) :

Il permet de visualiser le graphe de la fonction f et d'itérer avec les quatre systèmes . (Figure 2.4 , 2.5 , 2.6 , 2.7 ,2.8 et 2.9)

- program 2 (EXPLOR) :

Il permet d'itérer automatiquement la fonction f (dans les quatre systèmes) c'est à dire , on donne les valeurs T et I , Il commence à itérer à partir de (0,0) jusqu'à (T,I) avec un pas choisi . Il nous donne le schéma de bifurcations (Figure 2.6)

- Programme 1 (PGRESP)

```
program  ITER-de-FONCTION ;
label    1,2,3,4,5 ;
const    MN = 0 ; N = 430 ;
         VD = 13 ; PK = 11 ; PA = 46.9 ; GM = 1.42 ; VM = 20.4 ;
type     INDMAX = -maxint . . maxint ;
         IVECT = array [MN. .N] of real ;
var      I , IXC , IYC , IY1 , IY2 , J , K , KT , L , NI , S , Q : integer ;
         GT , GI , GQ , AL , BE , A , B ,X ,YI,YMIN ,YMAX,ECHX : real ;
         DEP : char ;
         PRFOIS : boolean ;
         CHX , CHXO : string ;
         Y : IVECT ;
```

```
external  procedure    DECURHOR (IYC,IXMIN,IXMAX : integer ;
          XO,ECHX :real ; var IXC : integer) ;
external  procedure TRACECOURBE(BI,BS,IX1,IY1,IX2,IY2 :integer ;
          POINTILLE,EFFACE:boolean; str : string ; var C :  array [IB
          ..IH: INDMAX] of real ;
external  procedure MOVETO ( X , Y :  integer ) ;
external  procedure DRAW ( X , Y :integer ) ;
external  procedure DRAWOFF ( X , Y :integer ) ;
external  procedure CLRGR ;
external  procedure CLRAN ;
external  procedure MODANGR ( F : byte ) ;
external  procedure WRITEGR ( var  S  :  string ) ;
external  procedure CURSGR ( B : byte )  ;
external  procedure CODPFX (x :real ;L,D :integer ; var CHX :sting) ;
external  function   F (GT,GI,XIE :real ; var S,Q : integer) : real ;

begin
1 :       write(' ') ;
          CLRGR ; CLRAN ;
          writeln ; writeln ;
          write('choisissez " T " : ') ;
          read(GT) ;
          write('   et       " I " : ') ;
          read(GI) ;
          GQ:= (PK*GT + VD) / GI ;
          AL:= GQ / (PA - GQ) ;
          BE:= (VM -PA*GM) / (GQ - PA) ;
          writeln ; writeln ; writeln ;
          YMIN := F(GT,GI,0.0,S,Q) ;
          YMAX:= YMIN ;
          Y[0] := YMIN ;
          for I:=1 to N do
          begin
                X :=I*GT/N ;
                YI := F(GT,GI,X,S,Q) ;
                if YI < YMIN
                then  YMIN := YI
                else    if YI > YMAX
                        then YMAX := YI ;
                Y[I] := YI
          end ;
```

```
        writeln ( ' YMIN = ' , YMIN , '     YMAX = ' , YMAX ) ;
        readln ;
2 :     CLRGR ; CLRAN ;
        K := 0 ;
        PRFOIS := true ;
        MOVETO (104 , 1 ) ;
        DRAW(104,431) ; DRAW(534,431) ; DRAW(534,1) ;
        DRAW(104,1) ; DRAW(534,431) ;
        IY1 := -24 + round(YMIN*N/GT) ;
        IY2 :=  1  + round(YMAX*N/GT) ;
        TRACECOURBE(0,N,79,IY1,534,IY2,false,false , ' ',Y) ;
        writeln( chr(27) , chr(102) , chr(32) , chr(32) ) ;
        writeln ; writeln ; writeln  ; writeln ;
        CODPFX (GT, 8 , 5 , CHX ) ;
        writeln( ' T = ' ,CHX) ;
        CODPFX (GI ,8 , 5 , CHX ) ;
        writeln( ' I := ' , CHX) ;
        ECHX := GT/N  ;
        IXC := 104  ;
        DECURHOR( 1,104 ,534, 0.0,ECHX,IXC) ;
        writeln( ' ') ;
3 :     writeln( '                                        ') ;
        X := (IXC - 104)*ECHX  ;
        CODPFX (X , 7 , 4 , CHXO) ;
        writeln( ' X[0] = ' , CHXO ) ;
        write ( chr(27),chr(102) , chr(52) , chr(56) ,
                ' nombre d''iterations ? ' ) ;
        readln (NI) ;
        MOVETO (IXC , IYC)  ;
        YI := F (GT , GI , X , S , Q )  ;
        IYC := 1 + round (N*YI/GT)  ;
        if  PRFOIS
        then  DRAW ( IXC , IYC )  ;
        PRFOIS := false ;
        L := 1 ;
        for  I :=1  to  NI  do
        begin
              CURSGR(0) ;
              K := K + 1 ;
              X := YI ;
              CODPFX ( X , 7 , 4 , CHX )  ;
```

```
                if  k :=10
                then    l := 2
                else    if  k := 100
                        then  L := 3 ;
                write(chr(27), chr(102), chr(52), chr(56),
                   -    ' X[ ', K:L , ' ] = ', CHX) ;
                write( ' S= ', s:2 , ' Q= ' Q:2 ,' frappez "R-C" ') ;
                IXC := 104 + round (N*X/GT) ;
                DRAW (IXC , IYC)  ;
                CURSGR (1)  ;
                readln
            end ;
4:          writeln ( '                                          ') ;
            write( ' 1 Suite ; 2 Ch. X[0] ; 3  Ch. "T" et "I" ; 4 Fin ') ;
            write ( ' Votre  Choix ? ' )  ;  read (DEP) ;
            write ( '  ' ) ;
            case  DEP  of
                        '1'  :   begin
                                        CURSGR(0) ;
                                        goto   3
                                 end ;
                        '2'  :   begin
                                        CURSGR(0) ;
                                        goto 2
                                 end ;
                        '3'  :   begin
                                        CURSGR(0) ;
                                        goto 1
                                 end ;
                        '4'  :   begin
                                        CURSGR(0) ;
                                        goto 5
                                 end ;
                        else     goto   4
            end ;
5:          write ( '  ' ) ;
            CLRAN ; CLRGR
end.
```

Program 2 (EXPLOR)

```
program  EXPLOR-RESP
const    VD = 13 ; PK = 11 ; PA = -46.9 ; GM = 1.42 ; VM = 20.4 ;
type     RVECT = array[0..1000] of real ;
var      I , J , K , NI , NJ , NK , NK1 , LC , S , Q , SQ : integer ;
         YL , GT , BIGT , BSGT , GI , BIGI , BSGIO , BSGI , PAS , GQ,
         AL , BE , X , EPS : real ;
         CNT : boolean ;

external function F (GT , GI , XIE :real ; var S , Q :integer ) : real ;
begin
         write ( ' ') ;
         writeln ; writeln ;
         writeln ( ' Choix de l''intervalle de variation de "T" ) ;
         write ( '                    Borne inferieure ? ' ) ;
         readln ( BIGT ) ;
         write ( '                    Borne Superieure ? ' ) ;
         readln (BSGT) ;
         write ( '                          Pas ? ' ) ;
         readln (PAS) ;
         writeln ;
         writeln (' Choix de l''intervalle de variation de "I" ') ;
         write ( '                    Borne inferieure ? ' ) ;
         readln (BIGI) ;
         writeln ( '                  Borne superieure ? ' ) ;
         write ( ' (pour T/2, entrez une valeur négative) ? ') ;
         readln(BSGI) ;
         if  BSGI <= 0.0  then  BSGI := BSGT  ;
         writeln  ;
         write ( ' Nombre d''itérations systematiques ? ' ) ;
         readln(NK1) ;
         writeln ;
         write ( ' Nombre d''itérations pour la recherche d''un
                cycle ? ' ) ;
         readln(NK) ;
         write ( ' Epsilon d''arrêt pour la recherche d''un cycle ?) ;
         readln(EPS) ;
         writeln ; writeln ; writeln ;
         NI := round ( (BSGT - BIGT)/PAS) ;
```

```
GT := BIGT ;
for  i:=0  to  NI  d
begin
      writeln ;
      BSGIO := GT/2 ;
      if  BSGIO > BSGI  then  BSGIO := BSGI ;
      NJ := round ( BSGIO - BIGI)/PAS ) ;
      GI := BIGI ;
      for  J := 0 to  NJ  do
      begin
            GQ := (PK*GT+VD)/GT ;
            AL := GQ / (PA - GQ) ;
            BE := (VM-PA*GM) / (GQ-PA) ;
            X := 0.0 ;
            CNT := true ;
            for  K:=1  to  NK1  do
            X := F(GT , GI , X , S , Q ) ;
            L :=NK1 ;
            YL := X ;
            K := NK1 ;
            while ( (K<NK) and CND )  do
            begin
                  K := K+1 ;
                  X := F (GT,GI,X,S,Q) ;
                  if  abs(YL-X) < EPS
                  then  begin
                        CNT := false ;
                        LC := K-NK1 ;
                        SQ := 0 ;
                        X := YL ;
                        for  L:=1  to  LC  do
                        begin
                              X:=F(GT,GI,X,S,Q) ;
                              SQ := SQ + Q
                        end ;
                        writeln(GT:5:2,GI:7:2,LC:6,
                              SQ:6 )
                  end
            end ;
            if CNT then
            write (GT:5:2,GI:7:2,'Pas de convergence') ;
```

```
                GI := GI + PAS ;
            end { J } ;
            GT := GT + PAS ;
        end { I }
end.
```

§ 2.4 Etude du schéma des bifurcations.

Sur la figure 2.6, on donne le schéma des bifurcations sur et sous-harmoniques, dans le cas du mécanisme 1. Les deux paramètres bifurquants retenus sont la période T de la pompe et la durée I de l'inflation. On a exploré dans le plan (T, I) les zones correspondant aux expériences: I<T/2 et 0<T<5 s. Les valeurs des paramètres m, M, V_M, a, Q, n, p, e, f ont des valeurs physiologiques (m=0.58 s, M=1.42 s, V_M=20.4 ml, a=-46.9, Q=(kT+V_D)/I, avec V_D=13 ml et k=11 ml/s, n=0.86, p=0.36, e=2/7 et f=0.2) (Fig. 2.6).

Nous constatons la présence d'importantes zones d'entraînement harmonique dans la plage 2-4 s, correspondant à la plage d'entraînement expérimental. Nous constatons d'autre part la présence de "langues d'Arnold" d'entraînements fractionnaire sur et sous-harmoniques p/q vérifiant : entre tout couple de langues p/q et p'/q', s'intercale une langue (p+p')/(q+q') cf [9].

L'explication théorique d'une telle observation numérique exige la connaissance, dans la classe de conjugaison topologique de f, de comportements de variables de codage que nous devrons expliciter (cf [7] et l'article de Cosnard et Goles dans [11]). La nature bimodale-unidiscontinue de f est certainement responsable de cette configuration du schéma de bifurcation.

De plus, la confrontation actuellement menée entre schémas théoriques et schéma expérimental (chez le lapin) permettra de choisir entre les quatre mécanismes d'action des stimulations proposées ci-dessus : il existe

d'importantes différences qualitatives entre les zones des quatre modèles (cf . par exemple la réduction de la zone 1/1 par passage du mécanisme 1 au mécanisme 2 , sur la Figure 2.6) et l'adéquation utilisera comme critère de choix ces différences qualitatives .

§ 2.5 Conclusion .

Cette étude , en simplifiant les modèles proposés antérieurement en [1] , [2] , [3] , permet , sur un modèle caricatural mais archétypique , d'étudier les bifurcations des entraînements du système respiratoire . Comme l'ont fait L.Glass et al. dans [4] , [5] , la considération d'un modèle très simple rendant compte des phénomènes d'arrêt inspiratoire et d'allongement expiratoire permet d'obtenir des schémas de bifurcation dont l'étude théorique ultérieure et la confrontation aux schémas expérimentaux pourront dégager le degré éventuel d'universalité postulé par L. Glass [8] , [9] , en comparant les schémas obtenus en entraînement des rythmes respiratoire et cardiaque .

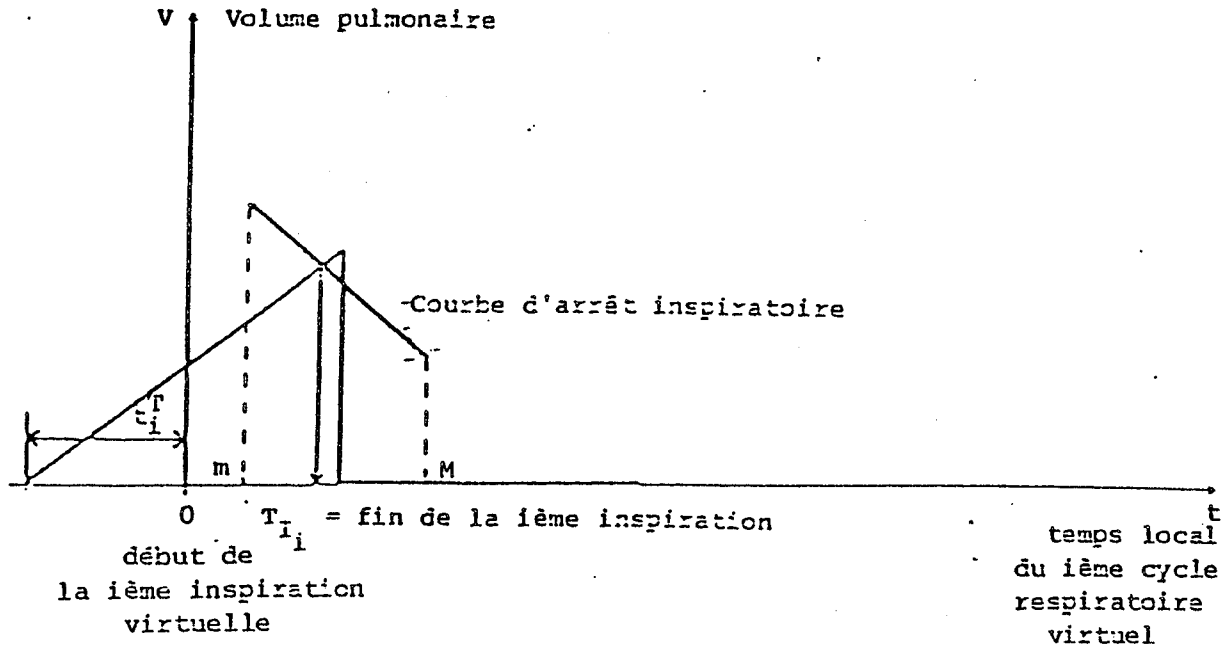


Figure 2.2 Mécanisme d'arrêt inspiratoire.

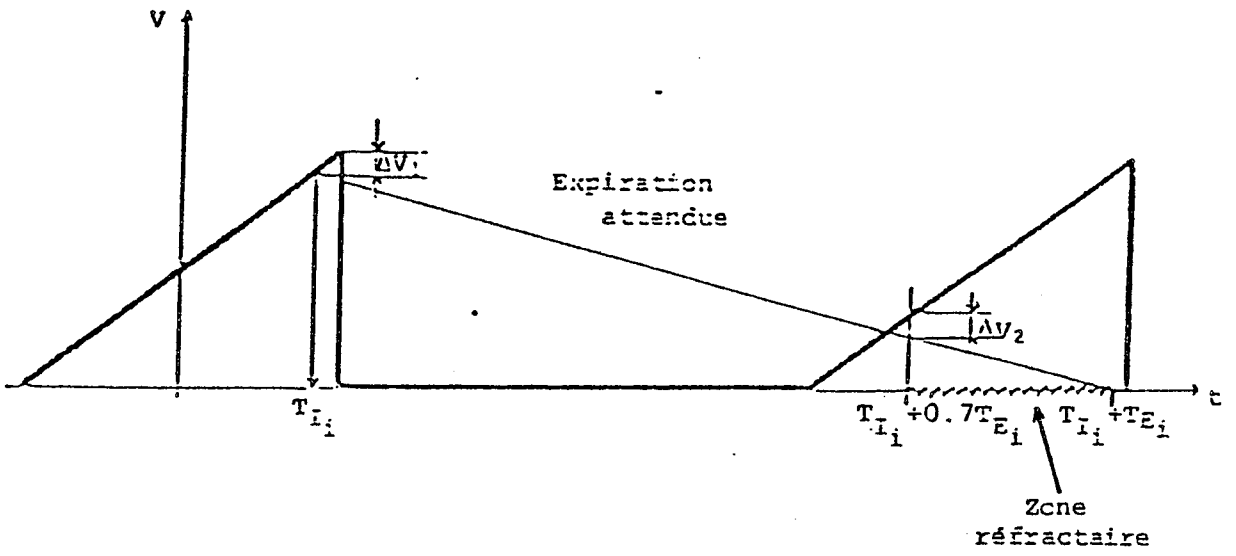
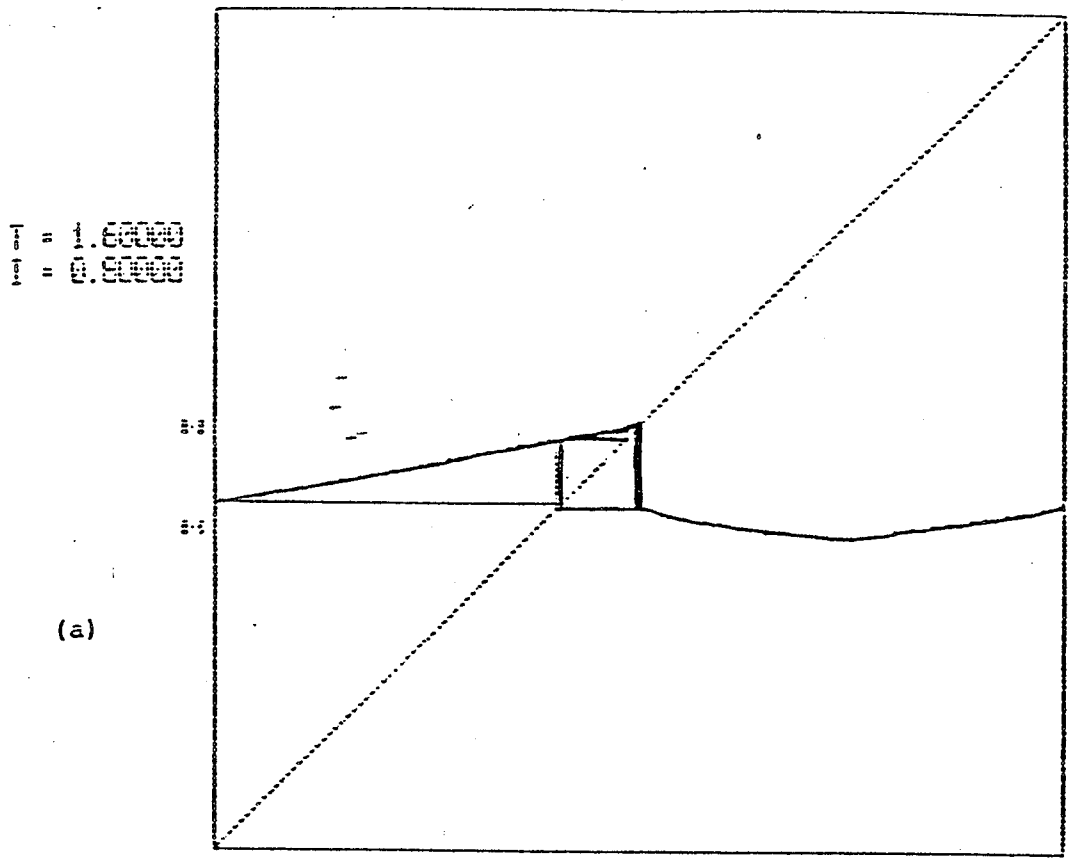


Figure 2.3 Mécanisme d'allongement expiratoire.



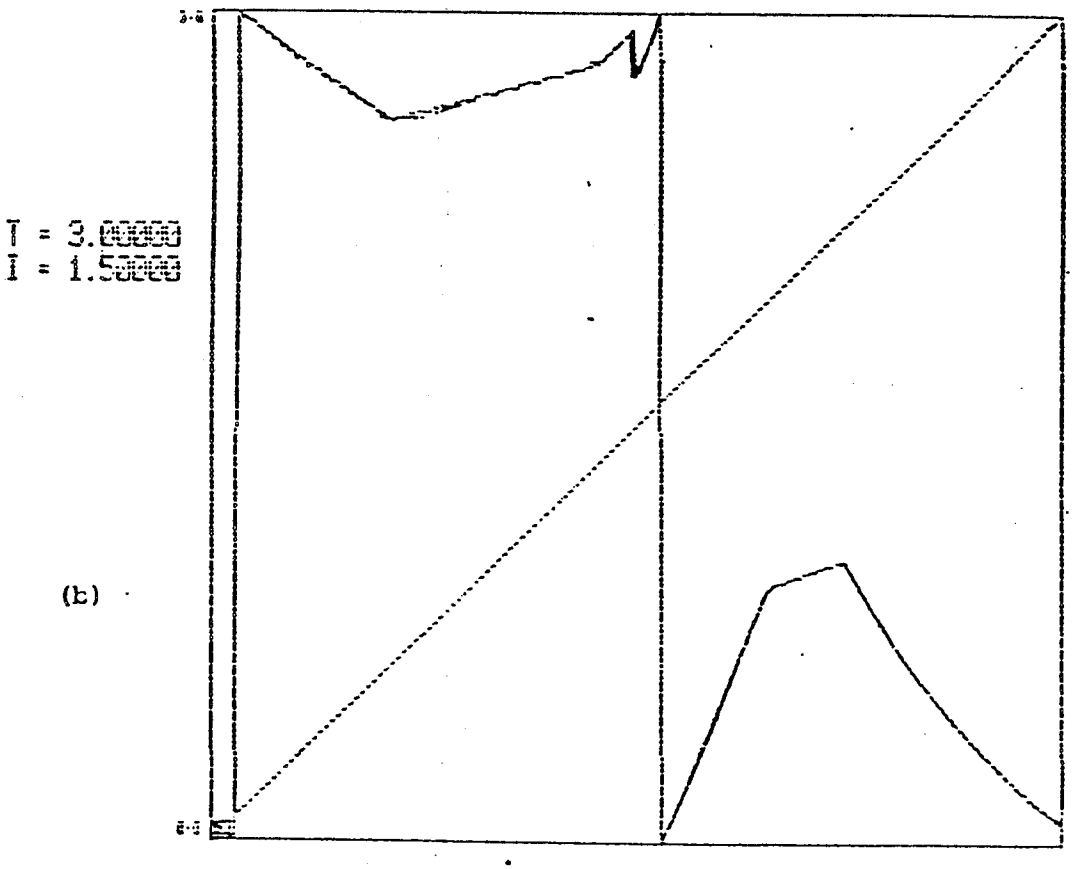
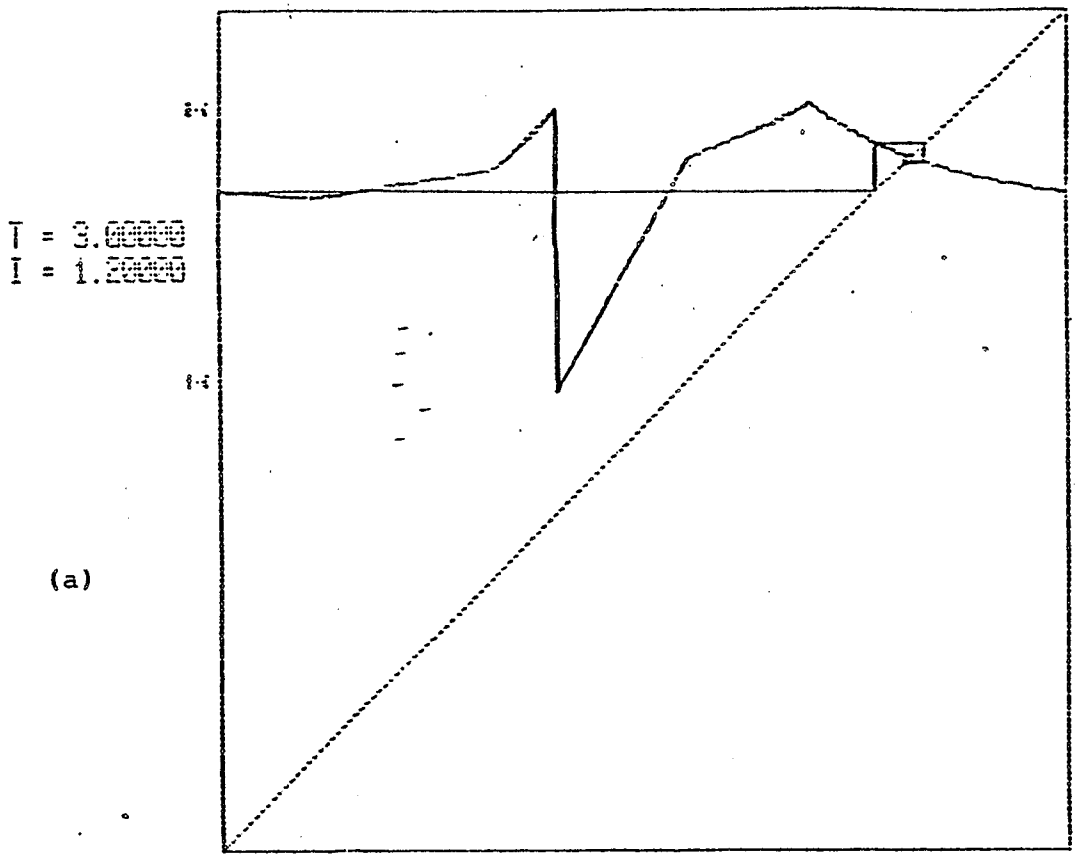
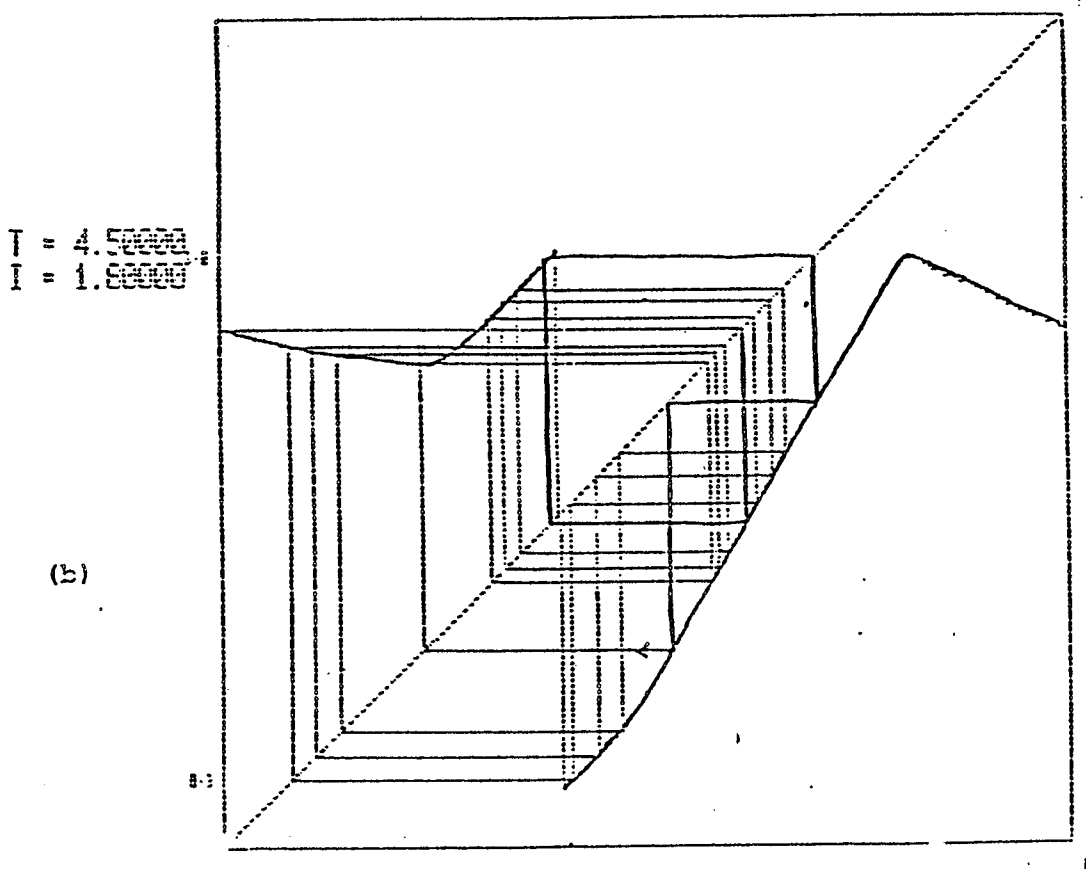


Figure 2.4 Fonctions de réponse du système 1 (a) et 2 (b).



(a)



(b)

Figure 2.5 Fonctions de réponse du système 3 (a) et 4 (b).

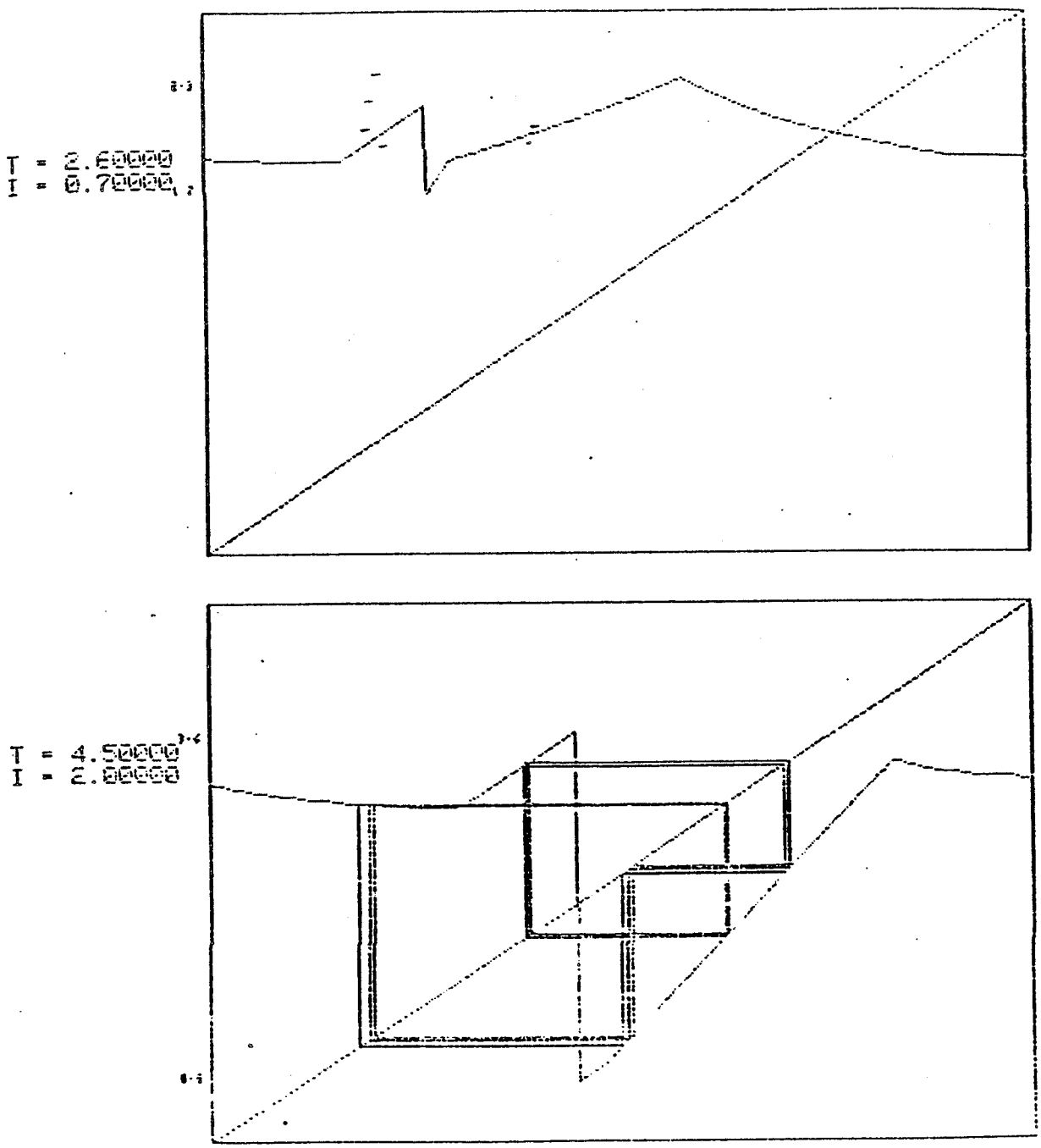


Figure 2.6 Fonction de réponse du système 1.

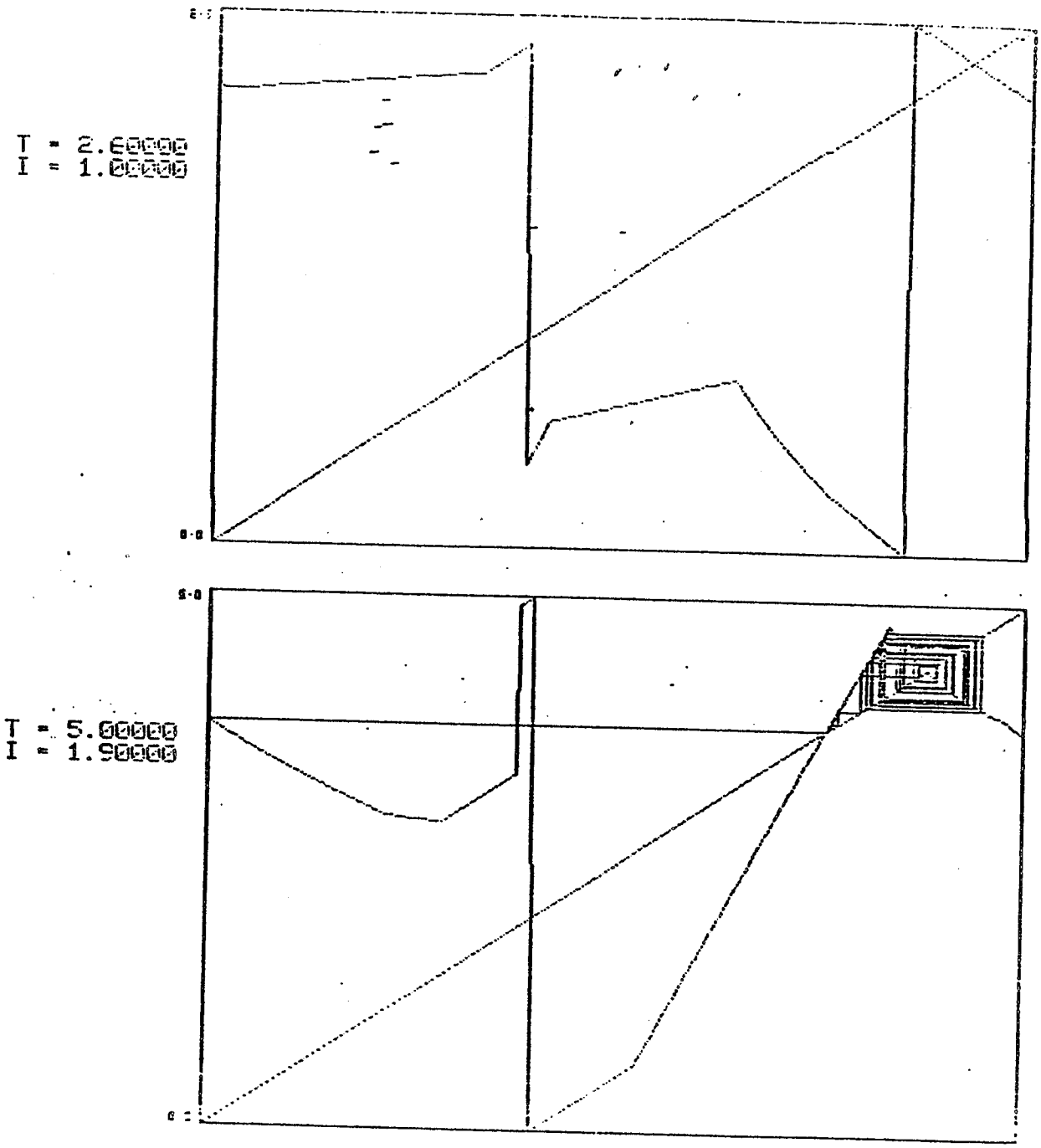


Figure 2.7 Fonction de réponse du système 2.

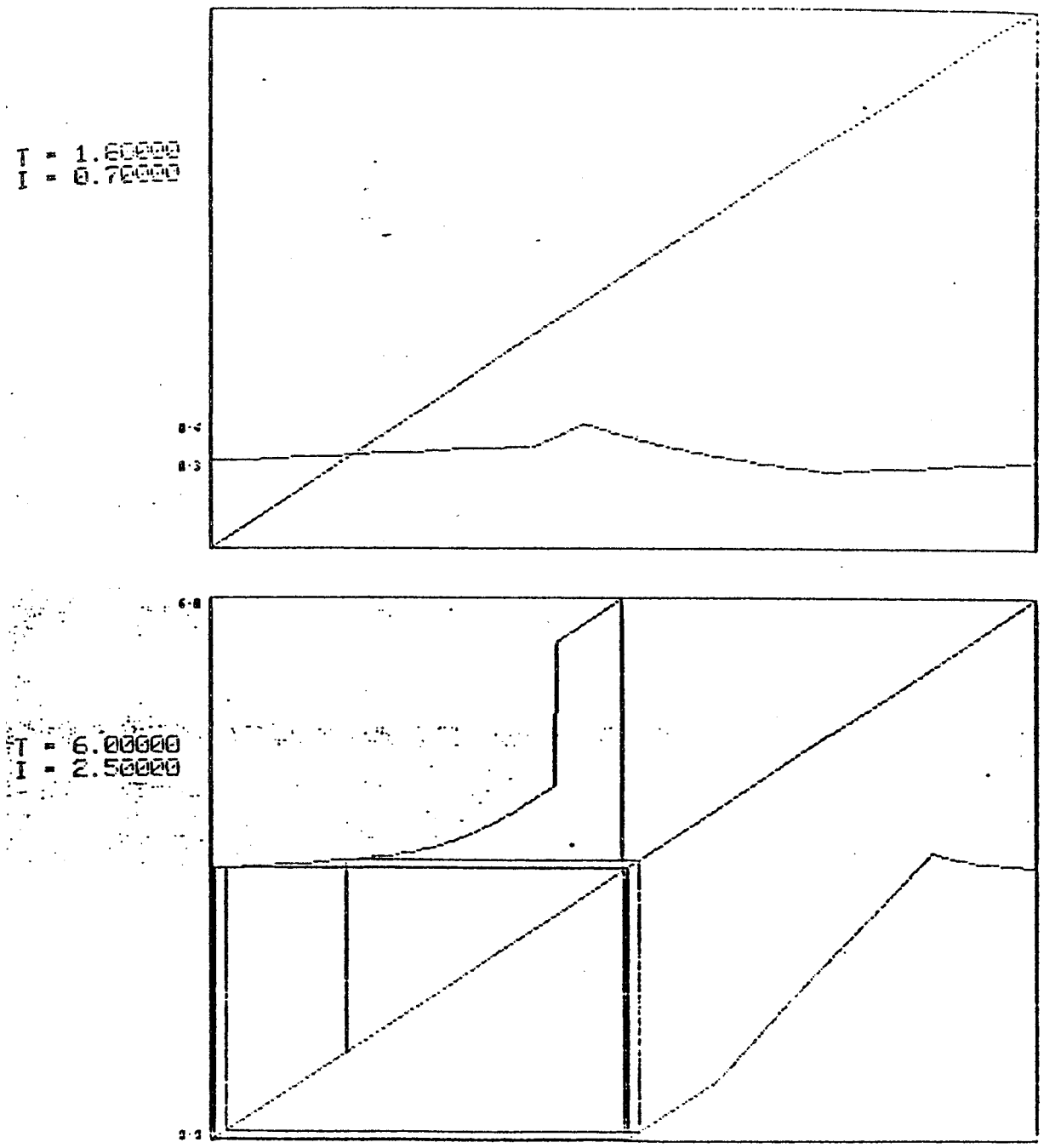


Figure 2.8 Fonction de réponse du système 3.

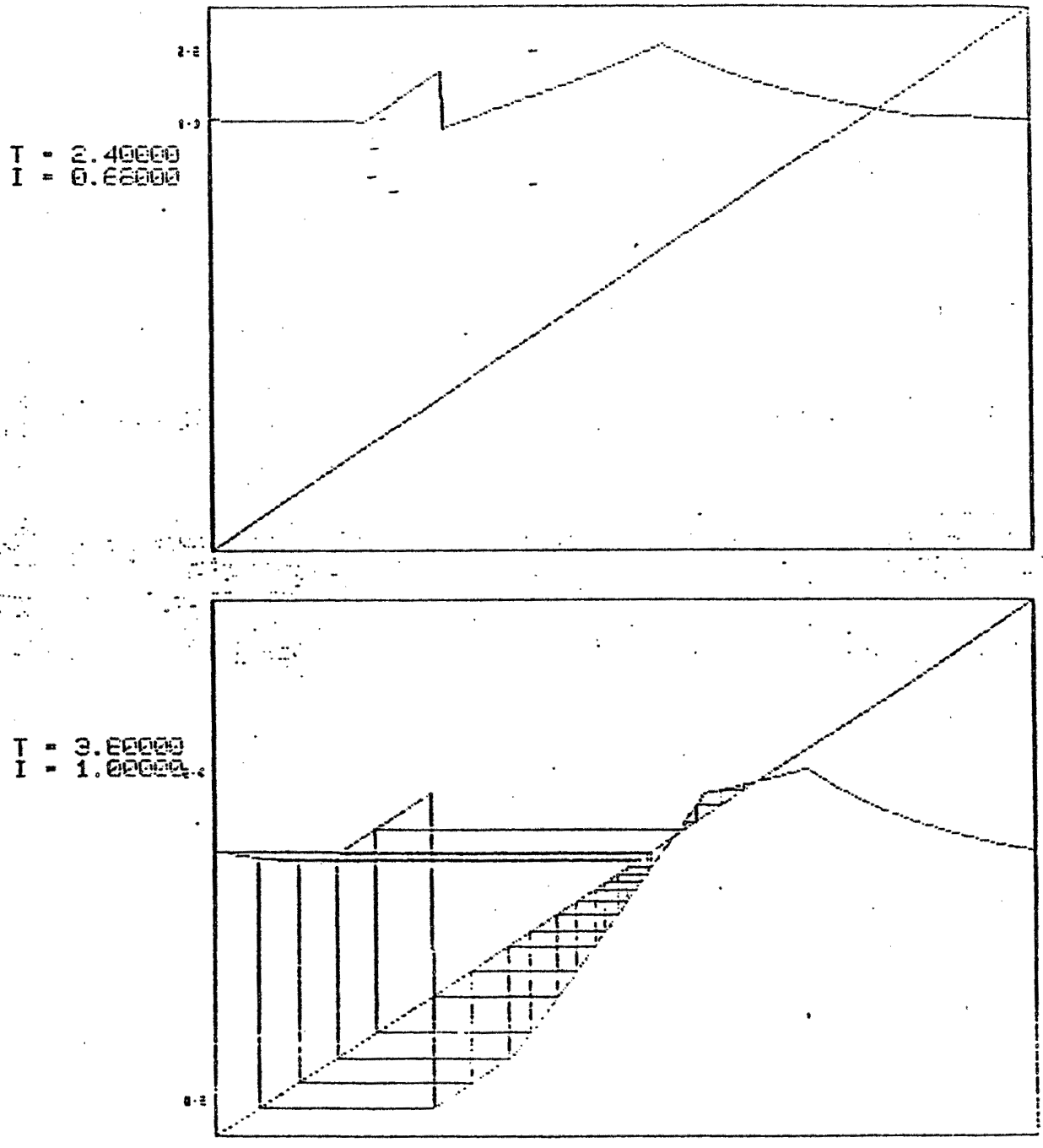


Figure 2.9 Fonction de réponse du système 4.

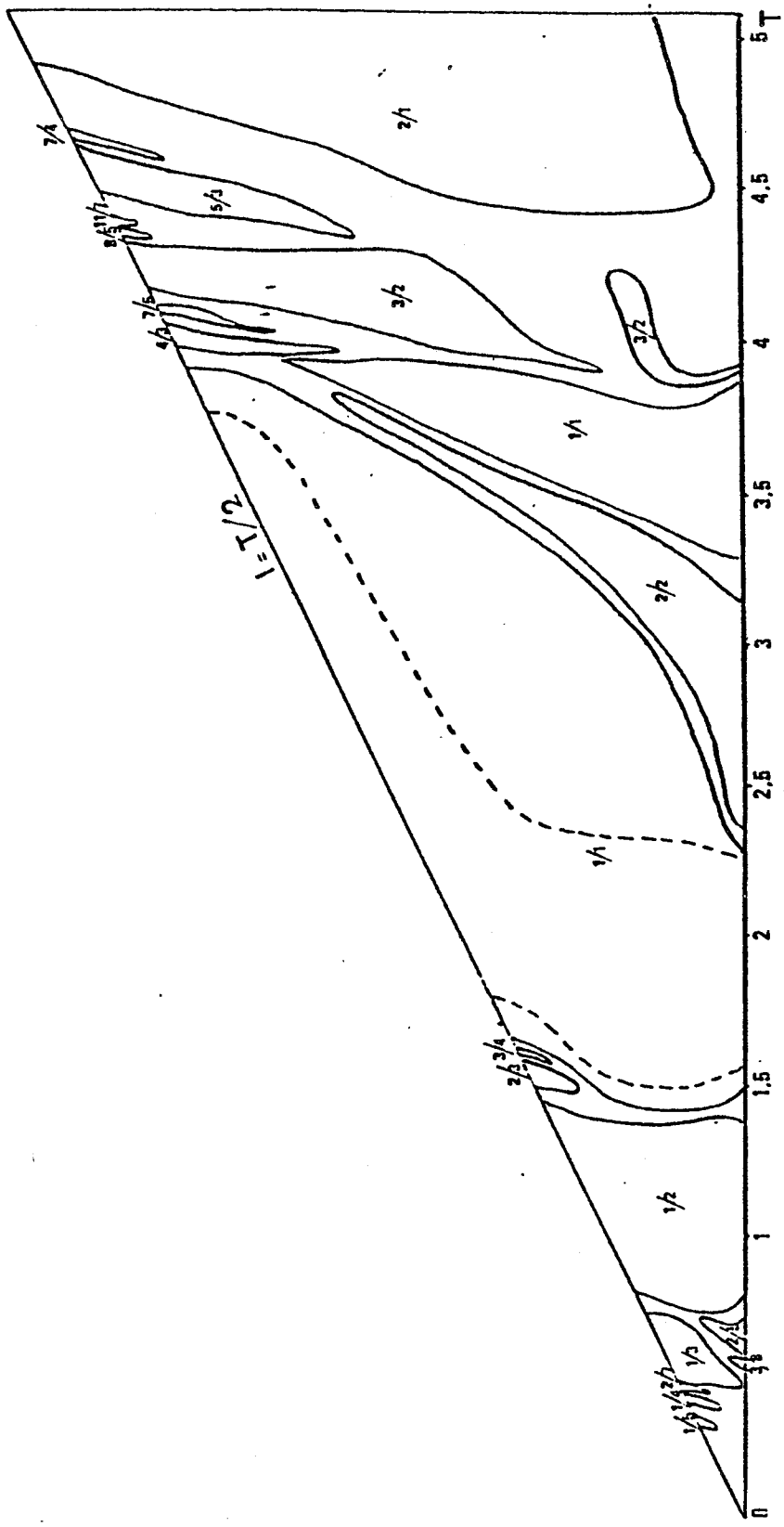


Figure 2.10 Schéma de bifurcation du système 1 (augmentation maximale de volume ΔV prise en compte). La ligne en pointillé montre le changement de la zone 1/1, lorsque l'on passe du système 1 au système 2 (allongement maximal pris en compte).

REFERENCES

[1]. BACONNIER, P., BENCHETRIT, J., et PHAM DINH, T.(1983) Simulation of the entrainment of the respiratory rhythm by two conceptually different models. Lecture Notes in Biomaths, 49, 1-16.

[2]. PHAM DINH, T., DEMONGEOT, J., BACONNIER, P. ET BENCHETRIT, G. (1983) Simulation of a biological oscillator : the respiratory rhythm. J. Theor. Biol., 103 : 113-132

[3]. BACONNIER, P. BENCHETRIT, G., DEMONGEOT, J. et PHAM DINH, T. (1983). ENtraiment of the respiratory rhythm : I. Experiment and Phsiological model II. mathematical models, in Whipp B.J. and Wiberg D.M. eds,Modelling and control of Breathing, 126-143, Elsevier, Amsterdam (1983).

[4]. GRAVES, C., GLASS, L., LAPORTE, D., MELOCHE, R. et GRASSINO, A. Respiratory phase locking during mechanical ventilation inanesthetized human subjects (soumis à J. Appl. Physiol.).

[5]. PETRILLO, G.A., GLASS, L.,et TRIPPENBACH, T. (1983). phase locking of the respiratory rhythm in cats to a mechanical ventilator. Can. J. Physiol. and Pharm., 61 : 559-607.

[6]. BACONNIER, P. BENCHETRIT, G., DEMONGEOT, J., EBERHARD, A. et PHAM DINH, T. (1984) Respiratory rhythm : pattern analysis. In : Oscillations of physiological systems : dymamics and control, 37-43. Institute of Measurement and Control, London.

[7]. GAMBAUDO, J.M., LANFORD, O. and TRESSER, C. (1984). Dynamique symbolique des rotations. C.R. Acad. Sc., 229 : 823-826.

[8]. GUEVARA , M.R. , GLASS , L. , and SHRIER, A. (1981). Phase locking , period doubling bifurcations, and irregular dynamics in periodically stimulated cardiac cells. Science , 214 : 1350-1353.

[9]. GLASS , L. and PEREZ , R. (1982). Fine structure of phase locking. Physical Rev. Letters , 48 : 1772-1775.

[10] P.BACONNIER , T.PHAM DINH , G.BENCHETRIT et J.DEMONGEOT Entraînement d'un oscillateur biologique : De l'expérience au concept . IV^{e} Séminaire de l'Ecole de BIOLOGIE THEORIQUE , Editions du CNRS - Paris 1985.

[11] IV^{e} Séminaire de l'Ecole de BIOLOGIE THEORIQUE , Edition du CNRS - Paris 1985.

Chapitre Trois

Automate à seuil pour la reconnaissance de forme

§3.1 Introduction:

La notion d'automate à seuil a été introduite en 1943 par W. S. Maculloch et W. Pitts [7] pour modèliser l'activite nerveuse . Pour cela , ils proposent de modèliser la cellule nerveuse ou neurone par un " neurone formel " . C'est à dire un automate à seuil dont les entrées représenteraient les neurones afférents et les sorties , les neurones efférents ; le seuil est alors en fait le seuil d'excitabilité du neurone .

Depuis 1943 , les Biologistes , Physiciens et Mathématiciens ont beaucoup avancé dans ce domaine . Maintenant grâce à de nouvelles techniques et à la puissance des ordinateurs , on peut réaliser des simulations de grands réseaux . De plus le comportement théorique de réseaux d'automates à seuil est mieux connu depuis les travaux de E . GOLES [5] . Actuellement, on tente d'exploiter les automates à seuil en informatique , essentiellement pour la reconnaissance de formes (et pour des machines à apprendre) . Récemment , à la suite de travaux du physien américain J . HOPFIELD [6] , F . FOGELMAN-SOULIE a dévéloppé une construction d'automate à seuil pour la reconnaissance de formes ([1] à [4]) .

Nous étudions dans ce chapitre :

* Une nouvelle règle de construction de l'automate à seuil (la règle L-H *) plus efficace que la règle L-H utilisée jusqu'ici .

* Un nouvel algorithme , plus simple et sous certains aspects plus efficace que les précédents car ne procédant pas par itération .

Les exemples présentés en fin de chapitre illustrent nos résultats. Ils ont été programmés sur MICAL surtout pour la reconnaissance de caractères chinois.

§ 3.2 RAPPELS : La définition de l'automate à seuil et les propriétés importantes :

Cette section a pour but de rappeler un certain nombre de notions et de résultats essentiels qui concernent les automates à seuil. La plupart des démonstrations de ces résultats peuvent être trouvées dans les références [1] à [5].

Définition 3.3.1

Soit $F : \{0,1\}^n \to \{0,1\}^n$ une application dont les composantes sont $f_1, f_2 \ldots f_n$ où les f_i sont des fonctions à seuil, c'est à dire :

$$\forall \quad i \in \{1,2,\ldots n\} \text{ et } \quad \forall \quad x \in \{0,1\}^n$$

$$f_i(x) = \begin{cases} 1 & \text{si } \sum_{j=1}^{n} a_{ij} x_j - b_i \geq 0 \\ 0 & \text{sinon.} \end{cases}$$

où $a_{ij} \in \mathbb{R}$ est le poids synaptique de la connexion de la cellule j à la cellule i.

$A = (a_{ij})$ $i,j = 1, \ldots, n$ est la matrice d'interaction.

$b = (b_i)$ $i = 1, \ldots n$ est le vecteur des seuils.

Proposition 3.2.1

Pour chaque f_i, il existe une fonction à seuil g_i et $\bar{b} = (\bar{b}_i)$ $i = 1, \ldots, n$; tel que :

$$1. \quad \forall \quad x \in \{0,1\}^n ; \qquad f_i(x) = g_i(x).$$

$$2.\quad g_i(x) = \begin{cases} 1 & \text{si } \sum_{j=1}^{n} a_{ij}x_j - b_i > 0 \\ 0 & \text{si } \sum_{j=1}^{n} a_{ij}x_j - b_i < 0 \end{cases}$$

C'est à dire on peut obtenir un seuil strict pour chaque f_i.

Voir [2] .

Définition 3.2.2 :

Une **itération séquentielle** sur F associée à une permutation π de $[1,2,\ldots,n]$ est définie par un état initial $x(0)\in[0,1]^n$ et la règle de reévaluation suivante :

$\forall\ t\geq 0\quad x(t+1) = F_\pi[x(t)]$ avec F_π définie par

$x_{i(1)}(t+1) = f_{i(1)}[y_1, y_2, \ldots, y_n]$ où $y_j = x_j(t)$.

$x_{i(2)}(t+1) = f_{i(2)}[y_1, y_2, \ldots, y_n]$ où $y_j = x_j(t)$, $\forall\ j \neq i(1)$

$y_{i(1)} = x_{i(1)}(t+1)$

$x_{i(m)}(t+1) = f_{i(m)}[y_1, y_2, \ldots, y_n]$ où $y_j = x_j(t)$ $\forall j \notin [i(1),\ldots, i(m-1)]$

$y_j = x_j(t+1)$ $\forall j \in [i(1),\ldots,i(m-1)]$

$x_{i(n)}(t+1) = f_{i(n)}[y_1, y_2, \ldots, y_n]$ où $y_j = x_j(t+1)$ $\forall\ j \neq i(n)$

$y_{i(n)} = x_{i(n)}(t)$.

où $[i(1), i(2), \ldots i(n)] = [\pi^{-1}(1), \pi^{-1}(2), \ldots \pi^{-1}(n)]$ et $x_i(t)$ est l'état de la cellule i à l'instant t .

Définition 3.2.3 :

Une **itération aléatoire** sur F avec une stratégie $(i(t))_{t\geq 0}$ est définie par un état initial x(0) et la règle de reévalution définie de la manière suivante:

$$\forall\, t \geq 0 \quad x(t+1) = F_{i(t)}[x(t)] \quad \text{avec } F_{i(t)} \text{ définie par}$$
$$\forall\, t \geq 0 \quad x_j(t+1) = x_j(t) \qquad j \neq i(t)$$
$$x_{i(t)} = f_{i(t)}[\, x(t)\,].$$

Définition 3.1.4

Soit $A = (a_{ij})$ la matrice d'interaction et $b = (b_i)$ le vecteur des seuils, alors **l'énergie d'un état** x est définie par

$$E(x) = -(1/2) \sum_i x_i \sum_{j \neq i} a_{ij} x_j + \sum_i (b_i - a_{ii}) x_i$$

dans le cas de mémoire associative , on prendra systématiquement $a_{ii} = 0$, (en raison de la diagonale non négative démandée par le théorème d'énergie décroissante) d'où

$$E(x) = -(1/2) \sum_i x_i \sum_{i \neq j} a_{ij} x_j + \sum_i b_i x_i$$

Théorème 3.2.1 (Théorème d'énergie décroissante) [2]

Si $A = (a_{ij})$ est une matrice symétrique à diagonale non-négative, alors pour toutes les itérations séquentielles et toutes les itérations aléatoires on a

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

la démonstration a été faite dans [2] pour les itérations sequentielles et éteudue dans [4] aux itérations aléatoires.

Théorème 3.2.2

Si $A = (a_{ij})$ est une matrice symétrique à diagonale non-négative, alors quelle que soit la configuration x(0) de départ, toutes les itérations séquentielles sur F se stabilisent sur un point fixe de F (en particulier , il n'y a pas de comportement cyclique).

Voir [3].

Dans le problème d'automate à seuil pour la reconnaissance de formes , on s'intéresse beaucoup aux problèmes suivants:

1) Etant donné une famille (finie) de formes $x_S = \{x^s\}_{s \in S}$ avec $x^s \in (0,1)^n$, comment construire un automate à seuil admettant ces x^s comme points fixes ?

2) On considère ici les automates à seuil vérifiant la propriété suivante:

2-a : $\forall\ s \in S$, x^s est un point fixe , c'est à dire :

$$F_\pi(x^s) = x^s$$

2-b : x^s est un point attractif

$$\forall\ s, \exists\ V_s : x^s \in V_s \text{ et}$$

$$\forall\ x(0) \in V_s \text{ on a } F_\pi^{T+t}(x(0)) = x^s$$

où F_π est une itération séquentielle (On peut remplacer F_π par $F_{i(t)}$ pour le cas où il y a une itération aléatoire) sur F , pour ces deux conditions , on donne les deux définitions suivantes :

Définition 3.2.5

Si F satisfait la condition 1) ci-dessus , alors on dit que F est **une x_S-mémoire** .

Définition 3.2.6

Si F satisfait la condition 2) ci-dessus, alors on dit que F est **une x_s-mémoire associative**.

Avec ces deux définitions, on sait qu'une x_s-mémoire associative est une x_s-mémoire dont tous les points $\{x^s\}$ sont attactifs dans un certain voisinage.

Définition 3.2.7 (voisinage premier)

Soit $x \in \{0,1\}^n$, alors le **voisinage premier** de x est défini par

$$V_1(x) = \{x\} \cup C_1(x)$$

où $C_1 = \{ x^*_i = (x_1, x_2, \dots, x_{i-1}, \overline{x_i}, x_{i+1}, \dots, x_n)\ i = 1,\dots,n \}$ et $\overline{1} = 0$ et $\overline{0} = 1$.

Définition 3.2.8 Gradient de E(x)

$$\text{Avec } E(x) = -(1/2) \sum_i \sum_{i \neq j} a_{ij} x_i x_j + \sum_i b_i x_i$$

alors le **gradient de E(x)** est défini par

$$\text{grad } E(x) = \Big(- \sum_{j \neq i} a_{ij} x_i + b_i \Big)_i$$

pour un automate à seuil F, on veut savoir sous quelles conditions, il peut être une X_s-mémoire ou une X_s-mémoire associative ? dans [4], on a les résultats suivants:

1. Condition nécessaire et suffisante de X_S-mémoire.

Supposons que $F : \{0,1\}^n \to \{0,1\}^n$ soit un automate à seuil avec une matrice d'interaction A et un seuil b ; $X_S=\{x^S\}_{s\in S}$ est une famille de formes données, alors F est une X_S-mémoire, si et seulement si la condition suivante est vérifiée :

$$\forall\ j\in(1,2,\ldots,n)$$

$$\text{Max}\ \{\sum_{j\neq i} a_{ij}\, x^s{}_j \ /\ s\in S :\ x^s{}_j=0\} < b_i \leq \text{Min}\ \{\sum_{j\neq i} a_{ij} x^s{}_j \ /\ s\in S : x^s{}_j=1\} \quad (1)$$

2. Condition suffisante de X_S-mémoire associative :

Soit F une X_S-mémoire et $(i_t)_{t\geq 0}$ une strategie aléatoire, si $\forall i,j \in (1,2,\ldots,n)$, on a :

$$\forall\ s\in S\ ,\ [\text{grad}_j E(x^S) + a_{ij}(x^s{}_i - \bar{x}^s{}_i)][x^s{}_j - \bar{x}^s{}_j] \leq 0 \quad (2)$$

Alors F est une X_S-mémoire associative . (les points fixe sont attractifs dans leur voisinage premier).

§3.3 Comparaison de la règle L-H et de la règle L-H* :

D'après le théorème d'energie décroissante, si on construit un automate à seuil ayant une matrice d'interaction symétrique et à diagonale non-négative, alors toutes les itérations séquentielles ou aléatoires se stabilisent sur un point fixe ; Maintenant, le problème est le suivant : Existe - il une règle permettant de construire pour des formes données un automate à seuil qui est une X_S-mémoire associtive ? (ou X_S-mémoire) La réponse est positive, on a ici une règle qui est proposée par LITTLE et J. HOPFIELD pour une petite famille de formes aléatoires :

___La règle L-H :

1) tirage aléatoire uniforme de x^S c'est à dire :
$\forall s\in S \quad x^S$ est équidistribué en 0 et en 1.

2) pseudo-orthogonalité
$\forall\, s\in S \;,\; s\neq\sigma \quad \sum_j x^\sigma{}_j (2x^S{}_j - 1)$ a pour moyenne 0.

3) $\forall i\in \{1,2,\ldots,n\} \quad b_i=0$ (seuil nul)

4) $\forall\, i,j \;\; i\neq j \quad a_{ij} = \sum_{s\in S} (2x^S{}_i - 1)(2x^S{}_j - 1)$; $\quad a_{ii}=0$.

Parallèlement, si on représente les formes dans l'espace $\{-1,1\}^n$ au lieu de $\{0,1\}^n$, alors on définit l'automate à seuil et la règle L-H* de la manière suivante :

Définition 3.3.1 :

Soit $F : \{-1,1\}^n \to \{-1,1\}^n$ une application dont les composantes sont $f_1,\ldots,f_n$ où les f_i sont des fonctions à seuil, c'est à dire

$$\forall\, i\in\{1,\ldots,n\} \quad \text{et} \quad \forall\, x\in\{-1,1\}^n$$

$$f_i(x) = \begin{cases} 1 & \text{si } \sum_{j=1}^{n} a_{ij} x_j - b_i \geq 0 \\ -1 & \text{sinon} \end{cases}$$

où a_{ij} est le poids synaptique de la connexion de la cellule j à la cellule i ;
$A = (a_{ij})$ $i,j = 1,\ldots,n$ est la matrice d'interaction ;
$b = (b_i)$ $i = 1,\ldots,n$ est le vecteur des seuils;

On définit la règle L-H* comme suit :

___ La règle L-H* :

1) tirage aléatoire de x^S, c'est à dire
$\forall s \in S$, x^S est équidistribué en -1 et en 1 ;

2) pseudo-orthogonalité :
$\forall s, \sigma \in S \quad s \neq \sigma$, $\sum_i x^{\sigma}_i x^S_i$ a pour moyenne 0 ;

3) $\forall i \in \{1, \ldots, n\} \quad b_i = 0$; (seuil nul)

4) $\forall i,j \quad i \neq j \quad a_{ij} = \sum x^S_i x^S_j \quad a_{ii} = 0$

Remarque :

1° Avec cette nouvelle définition d'automate à seuil , on peut reprendre de façon parallèle les autres définitions , par exemple : les définitions séquentielle ; aléatoire ; X_S-mémoire ; X_S-mémoire associative etc. De plus tous les théorèmes de la section précédente restent vrais pour les nouvelles définitions. En effet , entre les nouvelles définitions et les anciennes , le passage se fait par transformation de variable :

$$\begin{array}{ccc} y & = & 2x - 1 \\ (-1, 1) & \rightarrow & (0,1) \\ y & \rightarrow & x \end{array}$$

2° La condition 4) de la règle L-H (où L-H*) nous permet facilement d'ajouter une nouvelle forme x^U dans une famille d'anciennes formes (x^S), en effet : si A est la matrice d'interaction associée à (x^S) et B est la matrice d'interaction associée à x^U , alors A+B est la matrice d'interaction associée à $(x^S)U(x^U)$. Ceci nous permet d'analyser la matrice d'interaction forme par forme .

Dans ce qui suit , nous allons nous intéresser seulement aux règles L-H et L-H* , nous allons voir comment l'automate à seuil construit par les règle L-H et L-H* satisfait à la condition de X_S-mémoire , et également nous ferons la

comparaison entre ces deux types d'automates . Selon la règle L-H , on a

$$a_{ij} = \sum_{s\in S} (2x^S{}_i - 1)(2x^S{}_j - 1)$$

$$\sum_{j\neq i} a_{ij} x^S{}_j = \sum_{j\neq i} \Big(\sum_{\sigma\in S} (2x^\sigma{}_i - 1)(2x^\sigma{}_j - 1)\, x^S{}_j$$

$$= \sum_{j\neq i} (2x^S{}_i - 1)(2x^S{}_j - 1)\, x^S{}_j + \sum_{j\neq i} x^S{}_j \sum_{\sigma\neq s} (2x^\sigma{}_i - 1)(2x^\sigma{}_j - 1)$$

Donc , pour que cet automate à seuil soit une X_S-mémoire , il faut que la condition suivante soit satisfaite :

$$\text{Max}\Big(-\sum_{j\neq i} x^S{}_j + \sum_{j\neq i} x^S{}_j \sum_{\sigma\neq s} (2x^\sigma{}_i - 1)(2x^\sigma{}_j - 1) \;/\; s : x^S{}_i = 0 \Big) < b_i \leq$$

$$\text{Min}\Big(\sum_{j\neq i} x^S{}_j + \sum_{j\neq i} x^S{}_j \sum_{\sigma\neq s} (2x^\sigma{}_i - 1)(2x^\sigma{}_j - 1) \;/\; s : x^S{}_i = 1 \Big) \qquad (3)$$

D'après la règle L-H , $b_i = 0$ $i = 1, \ldots, n$ alors cette condition est bien satisfaite pour $S = 1$ (c'est à dire pour mémoriser une forme) et pour un petit nombre de formes (S petit) .

Après avoir analysé les deux membres de l'inégalité (3), on remarque que si une forme posséde plus de 1 que de 0 (c'est à dire $\sum_{j\neq i} x^S{}_j > n/2$) alors la condition est plus facile à satisfaire que dans le cas où il y a plus de 0 que de 1 (c'est à dire $\sum_{j\neq i} x^S{}_j < n/2$) ; pour rendre symétrique ce problème , je propose d'utiliser l'espace $\{-1,1\}^n$ plutôt que $\{0,1\}^n$, autrement dit on utilise la règle L-H*, dans ce cas , on a :

$$a_{ij} = \sum_{s\in S} x^S{}_i x^S{}_j$$

$$\sum_{j\neq i} a_{ij}\, x^S{}_j = \sum_{j\neq i} x^S{}_j \Big(\sum_{s\in S} x^S{}_i x^S{}_j \Big) = x^S{}_i \sum_{j\neq i} (x^S{}_j)^2 + \sum_{j\neq i} x^S{}_j \sum_{\sigma\neq s} x^\sigma{}_i x^\sigma{}_j$$

$$= x^S{}_i (n-1) + \sum_{j\neq i} x^S{}_j \sum_{\sigma\neq s} x^\sigma{}_i x^\sigma{}_j$$

La condition pour que cet automate à seuil soit une X_S-mémoire est

$$\text{Max}\ \{ -(n-1) + \sum_{j\neq i} x^S_j \sum_{\sigma\neq s} x^\sigma_i x^\sigma_j \ /\ s : x^S_i=-1 \ \} < b_i \leq$$
$$\text{Min}\ \{ (n-1) + \sum_{j\neq i} x^S_j \sum_{\sigma\neq s} x^\sigma_i x^\sigma_j \ /\ s : x^S_i=1 \ \} \qquad (4)$$

Après comparaison entre (3) et (4) , alors on obtient les propriétés suivantes :

Propriété 3.31 :

$X_S = \{ x^1, x^2, \ldots, x^S \}$ étant une famille de formes , telle que F soit une X_S-mémoire , $A = (a_{ij})$ est la matrice d'interaction associée à X_S , pour la règle L-H*

$$\text{Si } \forall i \ , \quad \sum_j a_{ij} (x^u)_j = p \neq 0 \quad x^u \in X_S \qquad (5)$$

alors $\overline{x^u}$ est aussi un point fixe de F ($x^u = (a1, a2, \ldots, an)$ et $\overline{x^u} = (\overline{a1}, \overline{a2}, \ldots, \overline{an})$ où $ai=1$, $\overline{ai}=-1$, $ai=-1$, $\overline{ai}=1$) si toutes les formes $x^u \in X_S$ satisfont à (5) , alors

$\overline{X_S} = \{ \overline{x^1}, \overline{x^2}, \ldots, \overline{x^S} \}$ est aussi une X_S-mémoire

De plus , soit $x=(x1, x2, \ldots, xn)$ $xi \in \{-1, 1\}$ une forme entrée à reconnaître , si par des itérations séquentielles (ou aléatoires) , x tend vers un point fixe x* et x satisfait à (5) alors

$$\overline{x} \rightarrow \overline{x^*}$$

Démonstration :

En utilisant la condition (5) et le fait que x* soit un point fixe de F , on a :

$$\forall i \in \{1, 2, \ldots, n\} \qquad \sum_{j\neq i} a_{ij} x_j = p$$

alors $\sum_{j\neq i} a_{ij}(\overline{x})_j = \sum_{j\neq i} a_{ij}(-x_j) = -p$

donc par définition de l'automate à seuil

$p>0 \Rightarrow (x(t+1))_i = 1$ et $(\overline{x}(t+1))_i = -1$

$p<0 \Rightarrow (x(t+1))_i = -1$ et $(\overline{x}(t+1))_i = 1$

c'est à dire : $x \to x^* \Rightarrow \overline{x} \to \overline{x}^*$

Remarque :

Pour la règle L-H , on n'a pas cette propriété parce que avec l'espace $(0,1)^n$

$$\sum_{j\neq i} a_{ij}x_j = p \not\Rightarrow \sum_{j\neq i} a_{ij}\overline{x}_j = -p$$

les exemples graphiques illustrant cette propriété sont données dans les figures 3.1 , 3.2 , 3.3 et 3.4 .

Propriété 3.3.2 :

La règle L-H* est plus efficace que la règle L-H .

--- Efficace au sens suivant : Pour obtenir une même capacité de mémorisation de S formes , la règle L-H* necessite en moyenne deux fois moins de cellules (par exemple , si on doit travailler dans $(0,1)^{2n}$ par la règle L-H , on a seulement besoin de travailler dans $(-1,1)^n$ par la règle L-H*) en effet , les deux intervalles suivant sont égaux :

$$\text{L-H: } \operatorname{Max}\left[-\sum_{j\neq i} x^s_j + \sum_{j\neq i} x^s_j \sum_{\sigma\neq s} (2x^\sigma_i - 1)(2x^\sigma_j - 1) \;/\; s : x^s_i = 0 \right] < b_i \leq$$

$$\operatorname{Min}\left[\sum_{j\neq i} x^s_j + \sum_{j\neq i} x^s_j \sum_{\sigma\neq s} (2x^\sigma_i - 1)(2x^\sigma_j - 1) \;/\; s : x^s_i = -1 \right] \qquad (6)$$

$$x^s \in (0,1)^{2n} \qquad s = 1,2,\ldots,S .$$

$$\text{L-H*: } \operatorname{Max}\left[-(n-1) + \sum_{j\neq i} x^s_j \sum_{\sigma\neq s} x^\sigma_i x^\sigma_j \;/\; s : x^s_i = -1 \right] < b_i \leq$$

$$\operatorname{Min}\left[(n-1) + \sum_{j\neq i} x^s_j \sum_{\sigma\neq s} x^\sigma_i x^\sigma_j \;/\; s : x^s_i = 1 \right] \qquad (7)$$

$x^s \in [-1,1]^n \qquad s=1,2,\ldots S$.

Démonstration :

On note $B_{ij} = \sum_{\sigma \neq s} (2x^\sigma{}_j - 1)(2x^\sigma{}_j - 1) \qquad x^u \in (0,1)^{2n}$

$$C_{ij} = \sum_{\sigma \neq s} x^\sigma{}_i x^\sigma{}_j \qquad x^u \in [-1,1]^n$$

$$B_{ij}, C_{ij} \in \{ -S+1, -S+3, \ldots, S-3, S-1 \}.$$

Ici on utilise plusieurs fois le fait que les 0 et les 1 (resp. les -1 et 1) sont équidistribués.

On démontrera seulement l'égalité des deux minimums dans (6) et (7) . (le raisonnement est analogue pour les maximums)

$$T_{2n} = \min \{ \sum_{j=1}^{2n} x^s{}_j + \sum_{j \neq i} x^s{}_j B_{ij} \;/\; s : x^s{}_i = 1 \} \qquad x^s \in (0,1)^{2n}$$

$$T^*_n = \min \{ (n-1) + \sum_{j \neq i} x^s{}_j C_{ij} \;/\; s : x^s{}_i = 1 \} \qquad x^s \in [-1,1]^n$$

Alors $\qquad T_{2n} \sim T^*_n$

a) $\sum_{j \neq i} x^s{}_j$ vaut en moyenne (n-1) quand $x^s{}_i \in (0,1)^{2n}$.

b) Les $x^s{}_i$ étant constitués d'autant de 0 que de 1 (resp. -1 et 1); Les B_{ij} (resp. C_{ij}) sont équidistribés parmi les positife et les négatifs

Alors on peut conclure que $\sum_{j \neq i} x^s{}_j B_{ij}$ (où $x^s \in (0,1)^{2n}$) égale en moyenne $\sum_{j \neq i} x^s{}_j C_{ij}$ (où $x^s \in [-1,1]^n$) (parcequ'il n'y a pas de corrélation entre $x^s{}_j$ et B_{ij} (resp. C_{ij}).

En effectuant la somme , on a démontré l'égalité des deux minimums.

La comparaison de l'efficacité expérimentale est illustrée dans les figures 3.1 , 3.2 , 3.3 et 3.4.

Enfin , on a quelques autres propriétés , elles peuvent nous permettre de mieux comprendre les automates à seuil construits par les règles L-H et L-H*.

Les experimentations d'automates à seuil pour la reconnaissance de formes montrent que certains bits dans une forme sont plus "signifiants" que d'autres ; il y a des traits caractéristiques dans une forme ; autrement dit il y a certains bits qui sont plus sensibles que d'autres .

Le fait de modifier la valeur d'un bit (ou quelques bits) fait passer l'itération d'un bassin d'attraction à un autre : ce sont des bits plus sensibles ; Par contre la modification de certains bits n'a aucune incidence sur l'itération de bassin d'attraction : ce sont des bits moins sensibles . Maintenant on se pose les question suivante :

Comment détecter les points sensibles ?

Est - ce qu'on peut les localiser ?

Existe -il un nombre qui peut mesurer ce genre de sensibilité ?

Voici une propriété qui donne la réponse suivante :

Propriété 3.3.3 :

Soit $X_S = (x^1, x^2, \ldots, x^n)$ $x^i \in (0,1)^n$ une famille de formes à mémoriser . $A=(a_{ij})$ est la matrice d'interaction associée à X_S

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \cdots a_{1n} \\ a_{21} & a_{22} & a_{23} \cdots a_{2n} \\ a_{31} & a_{32} & a_{33} \cdots a_{3n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{n1} & a_{n2} & a_{n3} \cdots a_{nn} \end{pmatrix}$$

on note

$$M_j = \sum_{i=1}^{n} |a_{ij}|$$

et

$$M = \max_{j\in(1,2\ldots n)} (M_j)$$

$$m = \min_{j\in(1,2\ldots n)} (M_j)$$

$$S = M-m$$

Alors les bits k où $M_k=M$ $k\in(1,2,\ldots,n)$ sont les points les plus sensibles. les bits l où $M_l=m$ $l\in(1,2,\ldots,n)$ sont les points les moins sensibles. Le nombre S peut mesurer le différence de la sensibilité entre bits k et bits l, si S est plus grand, alors les bits k sont beaucoup plus sensibles que les bits l.

Démonstration :

D'après la définition de l'automate à seuil, on a :

$$\forall\, i,\ P_i = \sum_{j=1}^{n} a_{ij}x_j \qquad f_i = \begin{cases} 1 & P_i\geq 0 \\ 0 & P_i<0 \end{cases}$$

Pour les indices k réalisant le maximum M la modification de x_k a de fortes chances d'entrainer un changement de signe pour P_i.

Symétriquement la modification de x_l réalisant le minimum m a une très faible probabilité de modifier le signe de P_i.

Si le nombre S est très grand, il indique qu'il y a plus de probabilité pour que $|a_{ik}|$ soit très grand et $|a_{il}|$ soit très petit, L'augmentation de S augmente la sensibilité des bits k et diminue les bits l. La figure 3.8 illustre bien cette

propriété .

Propriété 3.3.4 :

Soit $X_S = (x^V, x^U)$ une famille de deux formes et

$$x^U = (a1, a2, \ldots, an)$$
$$x^V = (b1, b2, \ldots, bn)$$

ai ,bi $\in$ {-1,1} et $A = (a_{ij})$ est la matrice d'interaction associée à X_S , alors quels que soient x^U , x^V , on a toujours :

$$A \neq 0_{nxn}$$

Démonstration :

Ici on peut analyser forme par forme .

Soit A_V la matrice d'interaction associée à x^V ;

A_U la matrice d'interaction associée à x^U ;

alors on sait que :

$$A_V + A_U = A$$

si . A=0 , on a $A_U = -A_V$ D'après la règle L-H* , on peut écrire :

$$A_U = \begin{pmatrix} a1(\,0\ a2\ a3 \ldots an\,) \\ a2(\,a1\ 0\ a3 \ldots an\,) \\ \vdots \\ an(a1\ a2\ a3 \ldots an\,) \end{pmatrix} \qquad A_V = \begin{pmatrix} b1\,(\,0\ b2 \ldots bn\,) \\ b2\,(\,b1\ 0 \ldots bn\,) \\ \vdots \\ bn\,(\,b1\ b2 \ldots\ 0\,) \end{pmatrix}$$

par $A_U = -A_V$ on a :

$$(\,0, a2, \ldots, an\,) = -(b1/a1)(\,0\,, b2\,, \ldots, bn\,) \qquad (10)$$

$$(\,a1, 0, \ldots, an\,) = -(b2/a2)(\,b1, 0\ , \ldots, bn\,) \qquad (11)$$

dans (10)

$$b1 = a1 \Rightarrow x_V = (\,a1, -a2, \ldots, -an)$$

(12)

$b1=-a1 \Rightarrow x_V=(-a1,a2,\ldots,an)$

dans(11)

$b2=a2 \Rightarrow x_V=(-a1,a2,-a3,\ldots,-an)$

(13)

$b2=-a2 \Rightarrow x_V=(a1,-a2,a3,\ldots,an)$

On trouve que (12) et (13) sont en contradiction, donc

$$A_u \neq A_v \ , \quad A \neq 0_{nxn} .$$

propriété 3.3.5 :

Supposons que $X_S=(x^u, x^v)$ est une famille de deux formes et qu'entre x^u et x^v, il y a k points différents

$x^u=(a1,\ldots,ai_1,\ldots,ai_k,\ldots,an)$ $\quad i_1,i_2,\ldots,i_k \in \{1,\ldots n\}$

$x^v=(a1,\ldots,\overline{ai}_1,\ldots,\overline{ai}_k,\ldots,an)$

On note alors

U : la matrice d'interaction associée à x^u ;

V : la matrice d'interaction associée à x^v.

$U+V=W=\begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}$ est la matrice d'interaction associée à X_S.

Alors :

$wi_j=(0\ldots0\ \delta i_1\ 0\ldots0\ \delta i_2\ 0\ldots0\ \delta i_k\ 0\ldots0)$ $\quad j=1\ldots k$;

$\delta i_p = 2\, ai_j ai_p \quad p \in \{1,\ldots,j-1,j+1,\ldots,k\}$

$\delta i_j = 0$ (par la définition de la règle L-H)

Démonstration

$wi_j = ui_j + vi_j$

$ui_j = (a1ai_j, a2ai_j, \ldots, ai_1ai_j, \ldots, ai_kai_j, \ldots, an\, ai_j)$

$vi_j = (a1\overline{ai}_j, a2\overline{ai}_j, \ldots, \overline{ai}_1\overline{ai}_j, \ldots, \overline{ai}_k\overline{ai}_j, \ldots, an\, \overline{ai}_j)$

donc $wi_j=(0\ ,\ 0\ ,\ldots,2ai_1ai_j,\ldots,2ai_kai_j,\ldots,\ 0\ )$

Remarque :

Citons quelques cas intéressants dans la propriété 3.3.5 :

1) Si k=1 :

Dans ce cas $wi_j=(0,0,...0)$ est un vecteur nul . Alors on ne peut pas reconnaître les x^u et x^v ; autrement dit cet automate ne peut pas distinguer deux formes qui n'ont qu'un seul point différent . (c'est normal , car ces deux formes sont premières voisines) .

2) Si k est petit :

Dans ce cas , on sait que dans la $i_j^{ème}$ ligne ($j = 1, \ldots, k$), il n'y a que k-1 éléments non-nuls . Autrement dit il reste peu d'information de x^u et x^v dans les lignes i_j ($j=1,...,k$) ,donc dans cet automate , ces deux formes sont difficiles à reconnaître. Ceci signifie que si deux formes sont plus proches , alors ces deux formes sont plus difficiles à reconnaître d'après la règle L-H (L-H*). La figure 3.9 (b) illustre cette propriété .

3) Si une forme x^u est une partie d'autre forme x^v (ceci arrive très souvent avec les caractères chinois , voir Figure 3.9(a)) c'est à dire : si dans la propriété 3.3.5 , on a $ai_1=ai_2=...=ai_k=0$,alors si on utilise la règle L-H , on ne peut pas mettre les deux formes x^u et x^v ensemble comme points fixes de F . Si on utilise la règle L-H* , il n'y a pas de problème , en effet , par la règle L-H

$$wi_j = (0,\ldots,\delta i_1,\ldots,\delta i_2,\ldots,\delta i_k,\ldots,0)$$
$$x^u = (a1,\ldots, 0 , \ldots,0 , \ldots , 0 , \ldots,an)$$

D'après la définition de l'automate à seuil :

$$f_{i_j}=1 \quad \text{car} \quad \sum_l (w_{i_j})_l (x^u)_l = 0 \quad j=1,2,...k$$

mais , $ai_1=ai_2=\ldots=ai_k=0$ donc x^u ne peut pas être un

point fixe . l'exemple graphique d'illustration est donné dans la Figure 3.9(a).

Propriété 3.3.6:

La règle L-H* et la règle L-H nécessitent le même temps de calcul par pas d'itération.

(La démonstration est évidente)

CONCLUSION:

D'après toutes les propriétés qu'on a montrées ci-dessus ,on conclut que la règle L-H* est plus performante que la règle L-H pour la reconnaissance de formes.

§3.4. Un nouvel algorithme pour la reconnaissance de formes.

Dans les sections précédentes, on a discuté les propriétés du réseau d'automate à seuil qui est construit par les règles L-H et L-H* et on les utilise pour la reconnaissance de formes ,mais d'après les analyses précédentes, on constate que :

1) la capacité de ce réseau pour la reconnaissance de formes est trés limitée. Dans [1] une formule a été donnée pour que K formes au plus soient bien reconnues dans leur voisinage premier:

$$k \sim \frac{n}{8 \ln n}$$

où k: le nombre de formes à reconnaître ;
n: le dimension des vecteurs de formes .

Si on veut reconnaître une grande famille des formes , par exemple $k = 10^4$ caractères chinois , il faut prendre n entre 10^7 et 10^8 . Ce qui signifie qu'on doit construire un réseau de

dimension $10^7 X 10^7$ pour reconnaître les 10^4 formes dans leur voisinage premier. Un tel grand réseau est évidement impossible à réaliser sur ordinateur ;

2) Les conditions 1) et 2) des règles L-H et L-H* sont trés restrictives. Souvent dans la pratique , les formes qu'on veut reconnaitre ne peuvent pas en général vérifier ces deux conditions.

3) Avec les régles L-H et L-H* , pour reconnaître une forme , on a parfois besoin de plusieurs itérations afin d'aboutir au point fixe. Dans la Figure 3.6 , on voit qu'il faut quatre itérations pour obtenir un tel point fixe. Donc , les règles L-H et L-H* ne sont pas très efficaces pour la reconnaissance au point de vue du temps de calcul.

Dans le but de résoudre les problèmes ci-dessus et de trouver une méthode qui puisse avoir une grande capacité et une meilleure efficacité , je propose la construction suivante : Au lieu de mettre les informations de chaque forme dans toutes les cellules du réseau (c'est à dire chaque forme intervient dans tous les a_{ij} $i,j= 1,2,...,n$) , mettre les informations de chaque forme seulement dans une ligne du réseau.

Nous allons d'abord introduire entre deux configurations x et y de $\{-1,1\}^n$ la distance de Hamming usuelle :

d(x,y) =nombre de cellules différentes entre x et y.

Proposition 3.4.1 :

Soit $x,y \in \{-1,1\}^n$,alors la distance de Hamming s'écrit :

$$d(x,y)=\frac{n-\langle x,y\rangle}{2}$$

où $\langle , \rangle$ est le symbole du produit scolaire usuel .

(La démonstration est évidente).

Un exemple :

Soit $x=\begin{pmatrix}-1\\1\\1\\1\end{pmatrix}$ $y=\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}$ alors il y a trois points différents entre x et y, en effet :

$$d(x,y)=\frac{4-(-1-1+1-1)}{2}=\frac{4+2}{2}=3$$

3.4.1 Un algorithme à seuil:

On construit le réseau suivant :

1° Utiliser toujours les vecteurs de $(-1,1)^n$ comme représentation des formes.

2° Définir la matrice mémoire A suivante : chaque ligne de A mémorise une forme, c'est à dire :

Soit $X_S=(x^1,\ldots,x^S)$

et $x^i=(x^i_1,\ldots,x^i_n)$ $\quad i=1,\ldots,S \quad x_j^i\in(-1,1)$

alors $a_{ij}=x_j^i$

$$A=\begin{pmatrix}x^1_1\ldots x^1_n\\x^2_1\ldots x^2_n\\\vdots\\x^S_1\ldots x^S_n\end{pmatrix}=\begin{pmatrix}a_{11}\ldots a_{1n}\\a_{21}\ldots a_{sn}\\\vdots\\a_{s1}\quad a_{sn}\end{pmatrix}$$

A peut être non-carré (lorsque $s\neq n$).

3° Prendre un seuil $b = (b_i) = \begin{pmatrix} k \\ \vdots \\ k \end{pmatrix}$ si on veut une reconnaissance de formes (x^s) dans le voisinage $k^{ème}$ $(1 \leq k \leq n)$ de chaque forme.

Pour montrer comment cet algorithme fonctionne, on va seulement considérer l'exemple suivant :

On veut reconnaître 26 caractères et 10 chiffres. Chaque lettre ou chiffre est représenté par un vecteur de dimension $n(\in[-1,1]^n)$. Soit x la forme entrée.

1) D'abord, on construit la matrice de memoire de la façon suivante :

$$\begin{array}{l} \left.\begin{array}{l} A\rightarrow \\ B\rightarrow \\ \vdots \\ Z\rightarrow \end{array}\right\} 26 \text{ lettres} \\ \left.\begin{array}{l} 0\rightarrow \\ 1\rightarrow \\ \vdots \\ 9\rightarrow \end{array}\right\} 10 \text{ chiffres} \end{array} \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ & & \vdots & \\ a_{26,1} & a_{26,2} & \cdots & a_{26,n} \\ a_{27,1} & a_{27,2} & \cdots & a_{27,n} \\ a_{28,1} & a_{28,2} & \cdots & a_{28,n} \\ & & \vdots & \\ a_{36,1} & a_{36,2} & \cdots & a_{36,n} \end{pmatrix}$$

2) On calcule ensuite :

$$\rho=(1/2)\left(\begin{pmatrix} n \\ \vdots \\ n \end{pmatrix} - Ax \right) = \begin{pmatrix} \frac{n-\langle a^1, x\rangle}{2} \\ \vdots \\ \frac{n-\langle a^{36}, x\rangle}{2} \end{pmatrix} = \begin{pmatrix} d(x,a^1) \\ \vdots \\ d(x,a^{36}) \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_{36} \end{pmatrix}$$

où $d(x,a^i) \in \{0,1,2,\ldots,n\}$ $\quad i = 1, 2, \ldots, 36$

ρ nous indique par ses comportements la distance de x à chacune des 36 formes.

3) on compare le vecteur $\begin{bmatrix} d_1 \\ \vdots \\ d_{36} \end{bmatrix}$ avec le seuil choisi $\begin{bmatrix} k \\ \vdots \\ K \end{bmatrix}$

si $d_i \leq k$ Ceci indique que la forme a^i est dans le voisinage $k^{ème}$ de x.

si $d_i > k$ Ceci indique que la forme a^i est en dehors du voisinage $k^{ème}$ de x.

En changeant le seuil k, on peut trouver un nombre m tel que

$$d_m \leq d_j \qquad j \in \{1,2,\ldots m-1, m+1, \ldots, 36\}$$

et $$d_m = k.$$

Alors on considère a^m comme une des formes la plus proche de x (au sens de la distance de Hamming). Les expérimentations sont illustrées dans la Figure 3.10.

3.4.2 La Méthode de Meilleure Approximation Dans Un Espace Discret:

Notons $X_S = \{x^S\}$, $x^S \in \{-1, 1\}^n$ l'ensemble des formes données, on peut considérer le problème de reconnaissance de formes comme un problème de meilleure approximation dans un espace discret avec la distance de Hamming définie précédemment.

La forme $x \in \{-1, 1\}^n$ étant donnée, on cherche une forme $x^* \in \{x^S\}$, telle que

$$d(x^*, x) = \min_{x^i \in X_S} d(x^i, x) \qquad i = 1,2,\ldots S;$$

On sait que dans le problème ci dessus, la meilleure

approximation x* existe toujours mais peut ne pas être unique. on va construire un réseau pour cette méthode (qui est un peu différente de la précédente):

1) Utiliser toujours les vecteurs $(-1,1)^n$ pour la représentation des formes;

2) Construire la matrice de mémoire comme suit:

bloc	
0	[-1 -1 -1 ..... -1]
1	[les formes qui ont seulement un 1]
2	[les formes qui ont deux 1]
⋮	⋮ ⋮
k-1	[les formes qui ont k-1 1]
k	[les formes qui ont k 1]
k+1	[les formes qui ont k+1 1]
⋮	⋮ ⋮
n	[1 1 1 1]

Maintenant, si on a une forme entrée x à reconnaitre, alors:

1) D'abord, on calcule combien il y a de 1 dans la forme x, par exemple: x possède k 1;

2) Ensuite, on calcule les distances $d_1^{(k)}, \ldots, d_l^{(k)}$ (l est le nombre de formes qui possèdent k 1):

$$d_j^{(k)} = d(x,x_j) \qquad x_j \in k \text{ bloc}$$

on note : $d^1 = \min (d_1^{(k)}, \ldots, d_1^{(k)})$;

Si $d^1 \leq 1$ et $d(x,x_j) = d^1$, alors : x_j est une des formes les plus proches de x (parce que $x_j \notin$ k bloc, $d(x_j,x) \geq 1$).

Sinon on calcule les distances entre x et les formes qui sont dans les bloc k-1 et k+1 $D_1^{(k-1)}, \ldots, D_m^{(k-1)}, d_1^{(k+1)}, \ldots, d_m^{(k+1)}$ (m et n sont respectivement les nombres de formes possèdant k-1 l et K+1 l).

$$d_j^{(k-1)} = d(x,x_j) \qquad x_j \in k-1 \text{ bloc};$$

$$d_j^{(k+1)} = d(x,x_j) \qquad x_j \in k+1 \text{ bloc}.$$

et $$d^{k1} = \min (d_1^{(k-1)}, \ldots, d_m^{(k-1)}, d_1^{(k+1)}, \ldots, d_n^{(k+1)})$$

$$d^2 = \min (d^1, d^{k1})$$

Si $d^2 \leq 2$ et $d(x,x_j) = d^2$ $(x_j \in [$ k bloc, k-1 bloc, k+1 bloc$])$

alors x_j est une des formes les plus proches de x (parce que $x \notin [$ k bloc, k-1 bloc, k+1 bloc], on a $d(x,x_j) \geq 2$)

Sinon on calcule les distances entre x et $x_j \in [$ k-2 bloc, k+2 bloc]

. . .

jusqu'à une réponse positive.

Par cette façon de calculer, on fait moins de calculs et on gagne plus de temps, surtout quand on a une grande famille de formes.

CONCLUSION :

Les méthodes 3.4.1 et 3.4.2 ont sous certains aspects résolu les trois problèmes posés au début de la section ; mais elles ont aussi des inconvenients ; par exemple, par la distance de Hamming, on ne peut pas distinguer deux formes qui ont la même distance avec une forme entrée à reconnaître.

Commentaire des Figures

Figure 3.1 :

Pour la règle L-H* , si

$$X_S = \{ x^1 , x^2 , x^3 , \ldots x^n \}$$

étant une famille de formes , telle que F soit une X_S-mémoire , alors :

$$\overline{X_S} = (\overline{x^1} , \overline{x^2} , \ldots , \overline{x^n})$$

est aussi une X_S-mémoire .

Figure 3.2

Elle montre que pour la règle L-H , si $x^U \in X_S$ est un point fixe , alors $\overline{x^U}$ peut ne pas être un point fixe . Aussi les figures 3.1 et 3.2 indiquent que la règle L-H* est plus symétrique (elle est autoduale) .

Figure 3.3

Elle montre que pour la règle L-H*:
si x^U va vers le point fixe x^*
alors $\overline{x^U}$ va vers le point fixe $\overline{x^*}$ (autodualité)
C'est à dire que si x^U peut être reconnu alors $\overline{x^U}$ aussi .

Figure 3.4

Elle montre que pour la règle L-H:
si x^U va vers le point fixe x^*
alors $\overline{x^U}$ peut ne pas aller le point fixe $\overline{x^*}$
C'est à dire que x^U peut être reconnu sans que $\overline{x^U}$ le soit .

Figure 3.5 :

Comparaison de l'efficacité des règles L-H et L-H* après un pas de l'itération séquentielle, chacune des quatre formes est reconnue. (même dans un voisinage très grand). Ce n'est pas le cas pour la règle L-H.

Figure 3.6

Comparaison de l'efficacité de la règle L-H et L-H* pour la reconnaissance de formes :

Avec les mêmes points de départ : Par la règle L-H*, après un pas l'itération séquentielle, 4 formes sur 5 sont reconnues; mais par la règle L-H, après plusieurs pas de l'itération, aucune forme n'est reconnue.

Figure 3.7

Comparaison de l'efficacité des règles L-H et L-H*.

Avec les mêmes points de départ : Par la règle L-H*, toutes les 5 formes sont reconnues ; mais la règle L-H n'en trouve aucune.

Figure 3.8

Elle indique que dans une forme il y a certains bits qui sont plus sensibles que d'autre (soit par la règle L-H ou bien par la règle L-H*).

Dans la Figure, il y a 10 bits les plus sensibles et 6 bits les moins sensibles. On a

$$M = 140$$

$$m = 88$$

$$S = M - m = 52$$

Quand on change l'un des 10 bits les plus sensibles dans le caractère "天"*, l'automate passe à l'autre point fixe et il ne retrouve pas le caractère " 天 " ; mais quand on change 5 bits entre 6 bits les moins sensibles, il peut encore retrouver le caractère "天".

Figure 3.9 (a)

Elle indique que si une forme x^1 est une partie d'une autre forme x^2, alors la règle L-H ne peut pas reconnaître ensemble des ces deux formes ; mais la règle L-H* le peut .

Figure 3.9 (b)

Elle indique pour une famille de trois formes qui sont très proches, il a du mal à reconnaître . (soit par la règle L-H, soit par L-H*).

Figure 3.10

Elle indique que la méthode de meilleure approximation avec la distance de Hamming pour la reconnaissance de formes est très efficace .

Figure 3.11 et 3.12

Sur les même exemples de la Figure 3.10 , elles montrent les limitations des règles L-H et L-H* lorsqu'on veut mémoriser un trop grand nombres de caractères .

* "le ciel"

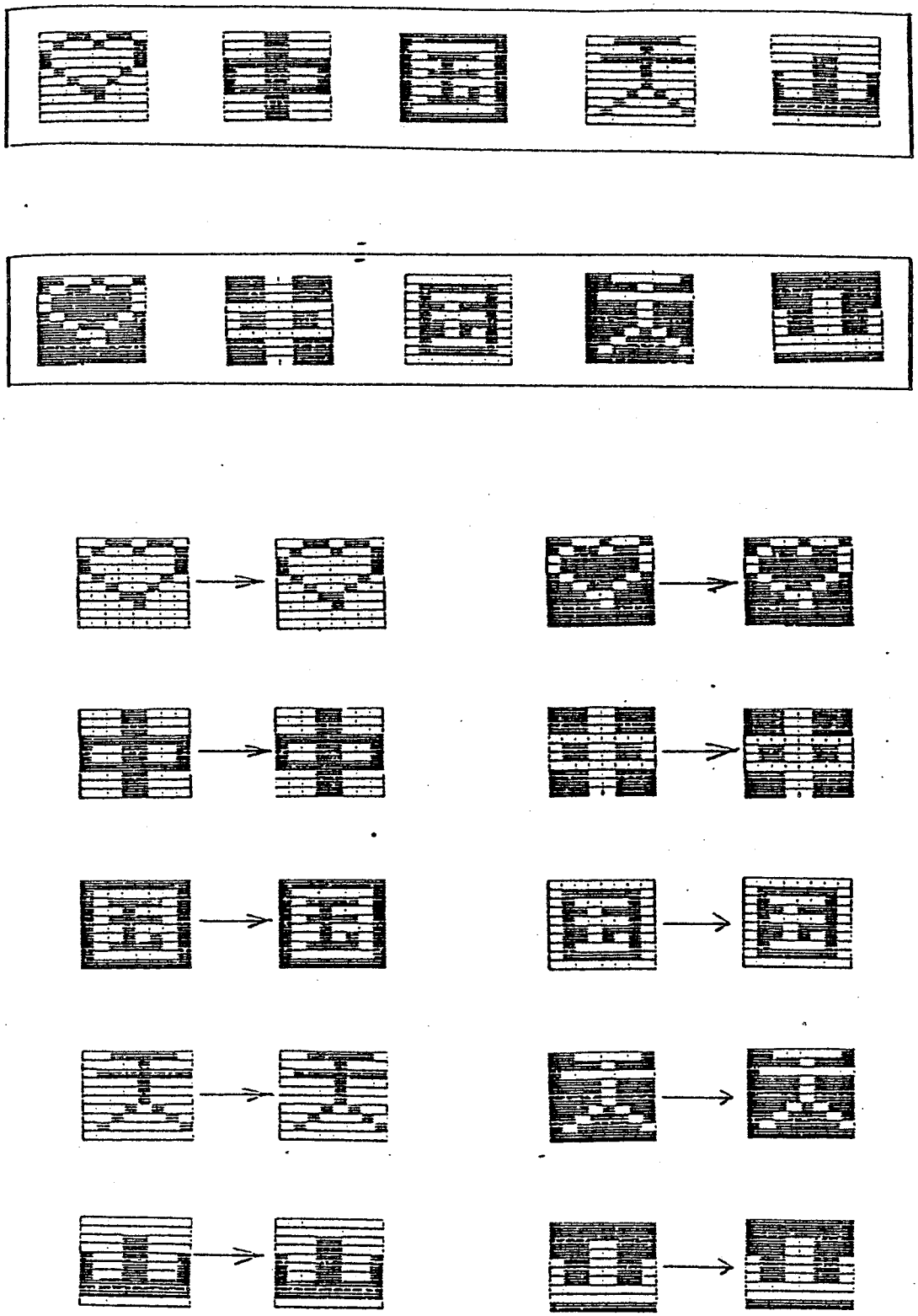


Figure 3.1 Règle L-H*

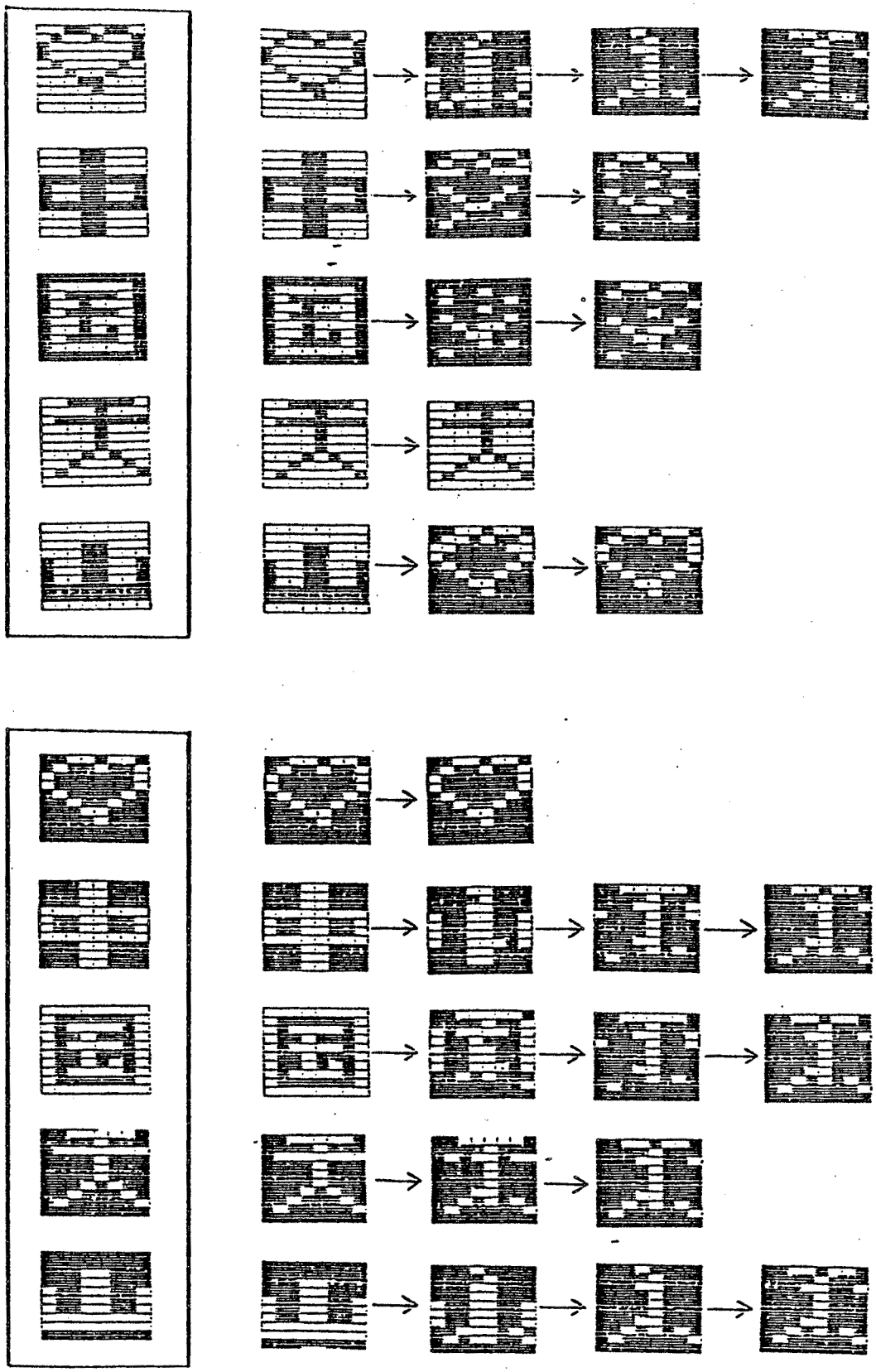
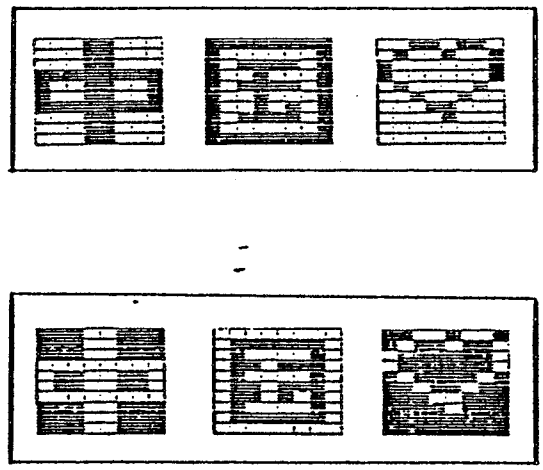


Figure 3.2 Règle L-H



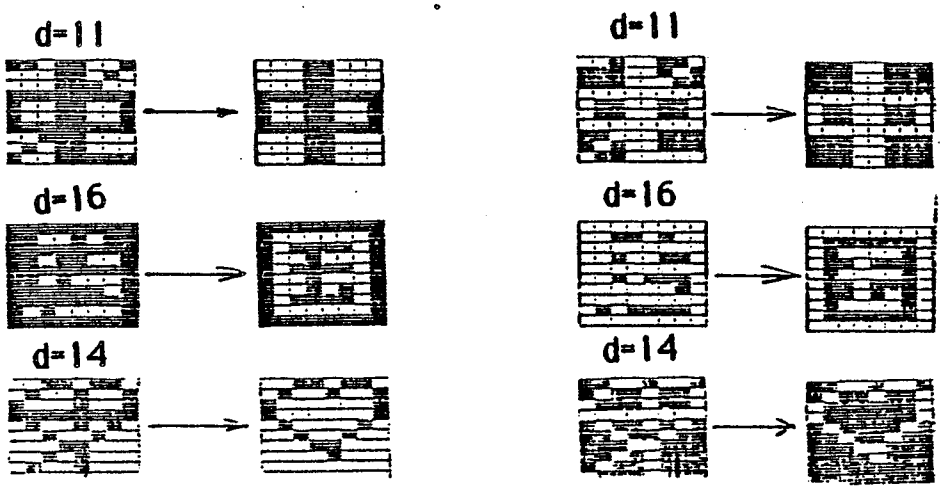


Figure 3.3 Règle L-H*

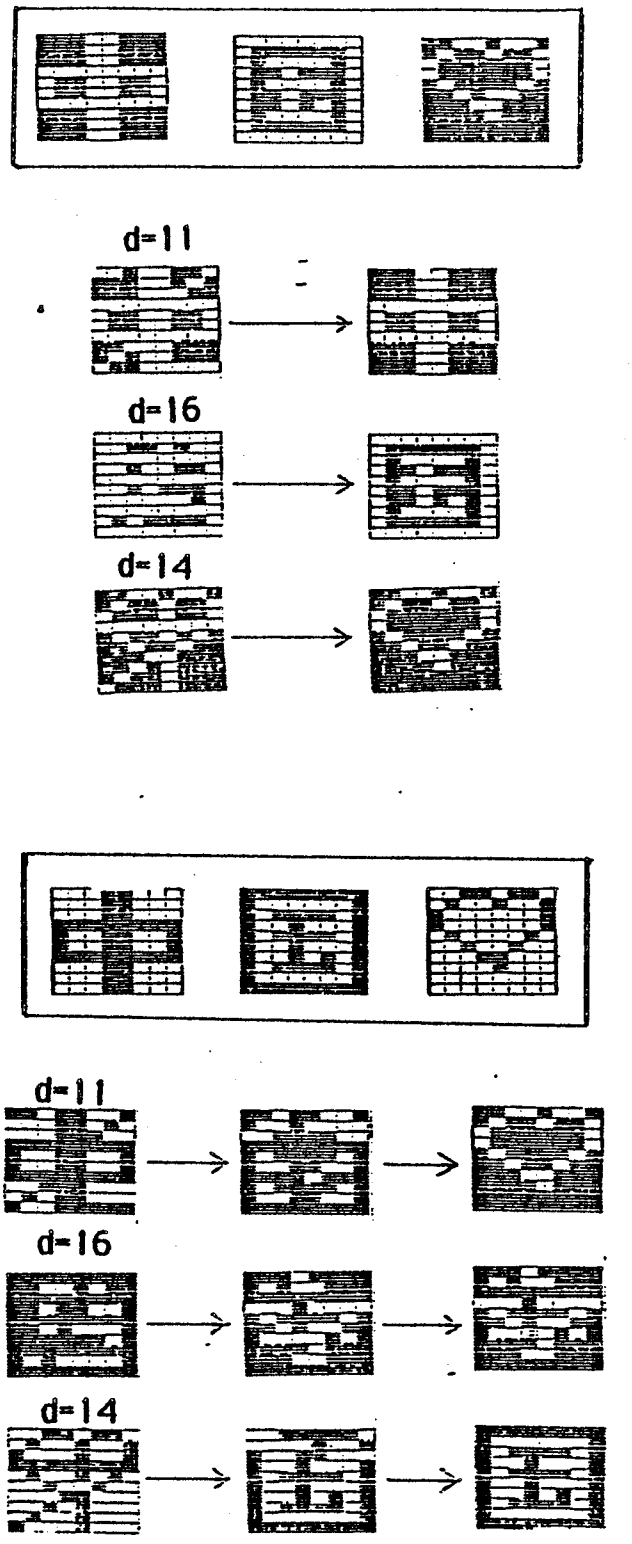
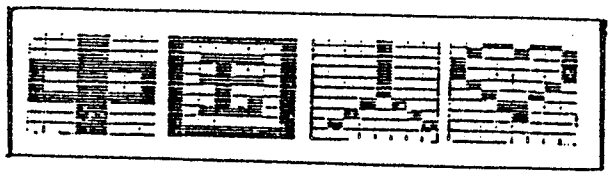


Figure 3.4 Règle L-H



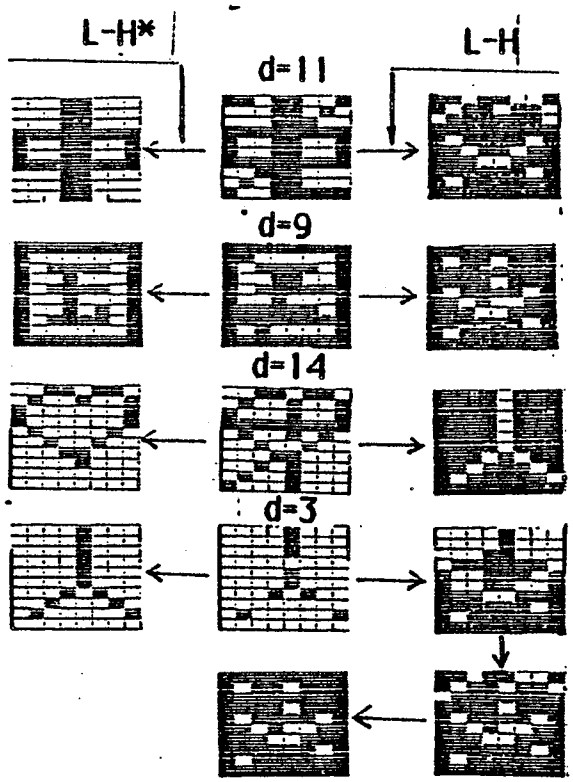
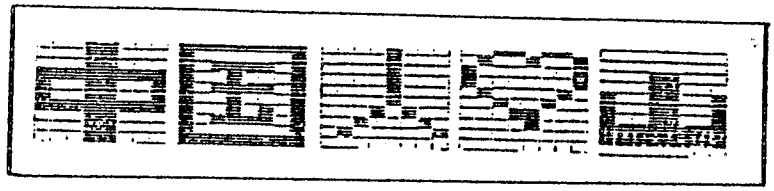


Figure 3.5



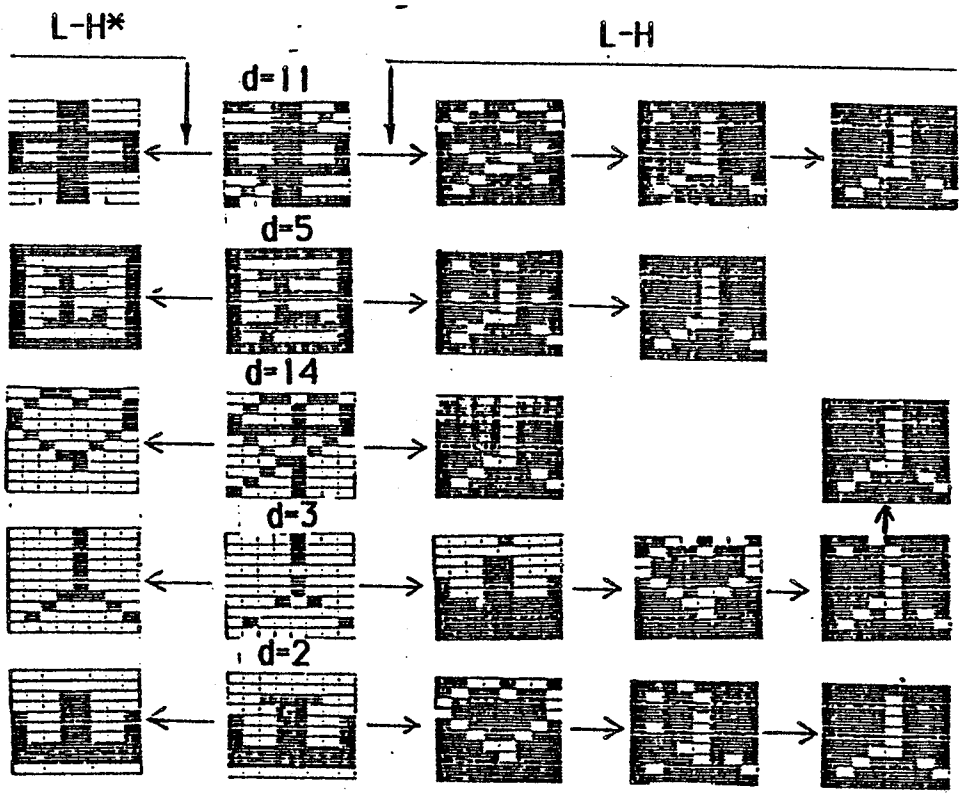
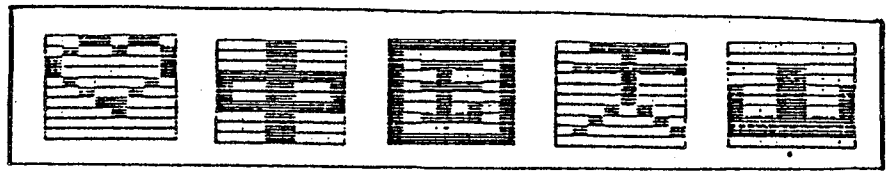


Figure 3.6



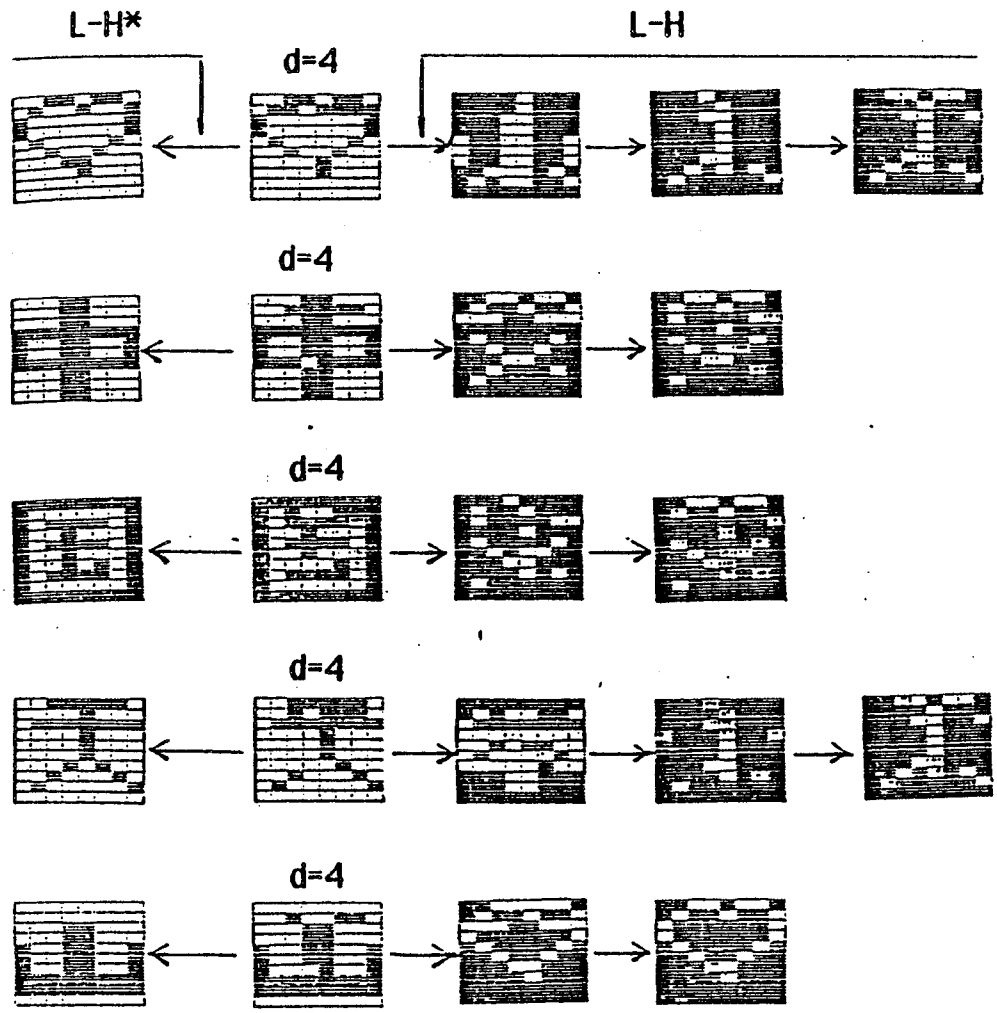


Figure 3.7

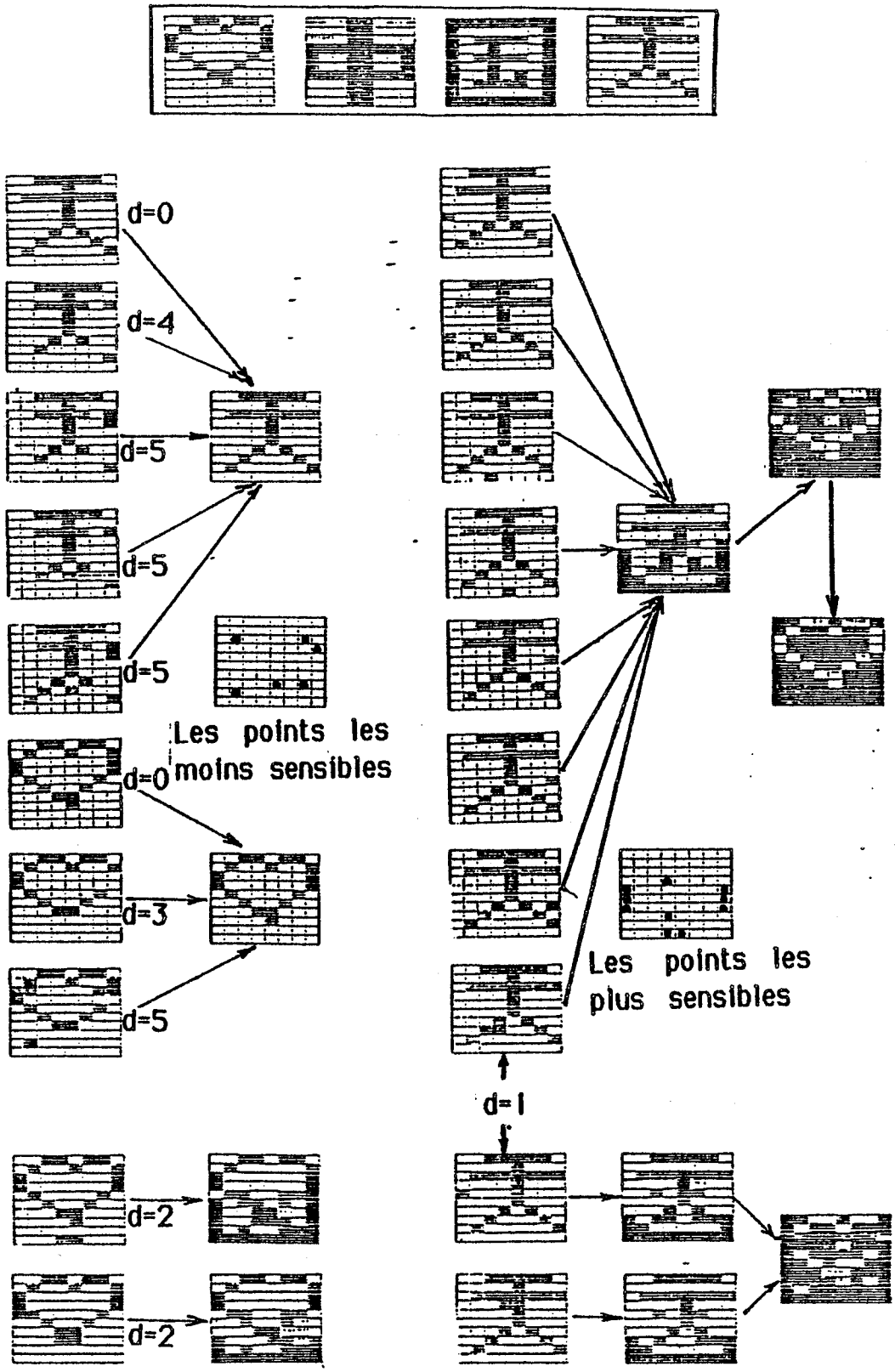
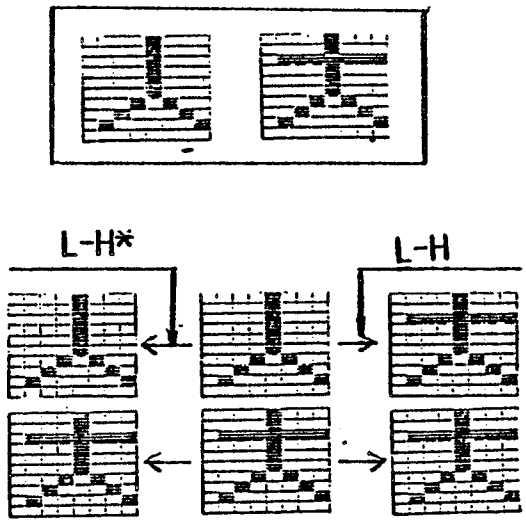
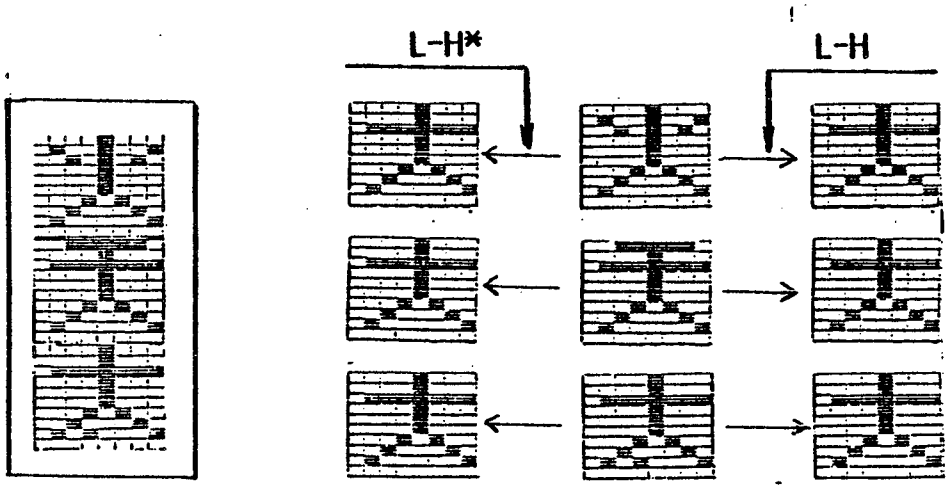


Figure 3.8 Règle L-H

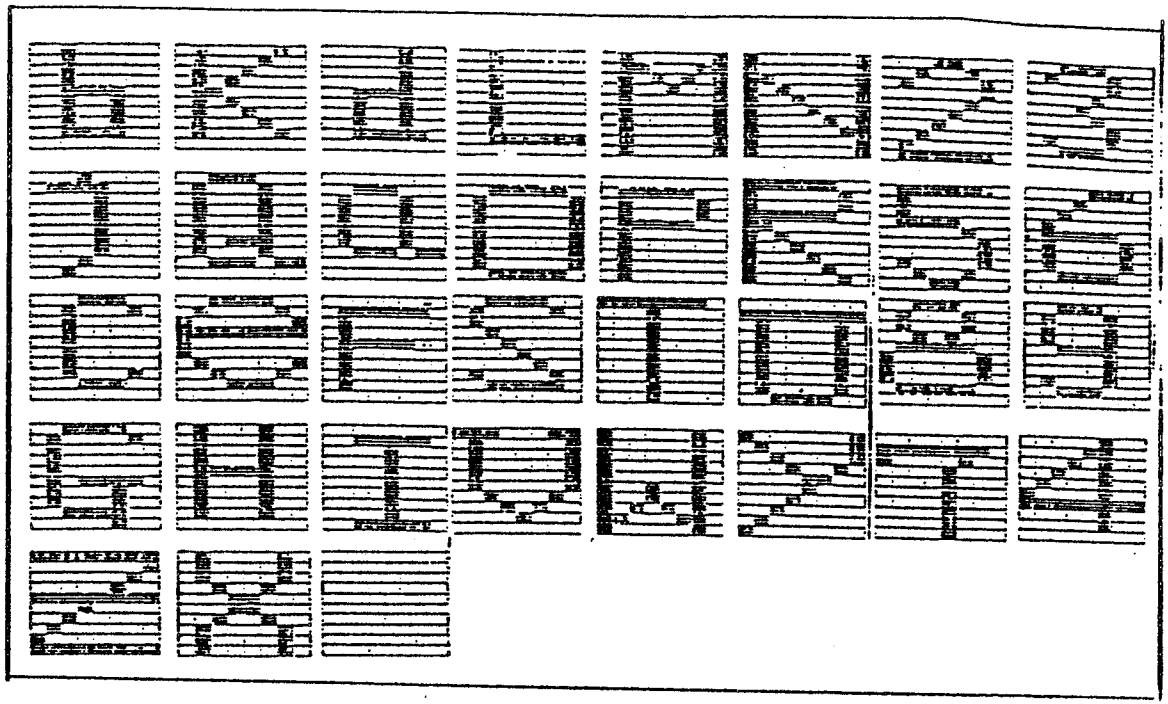


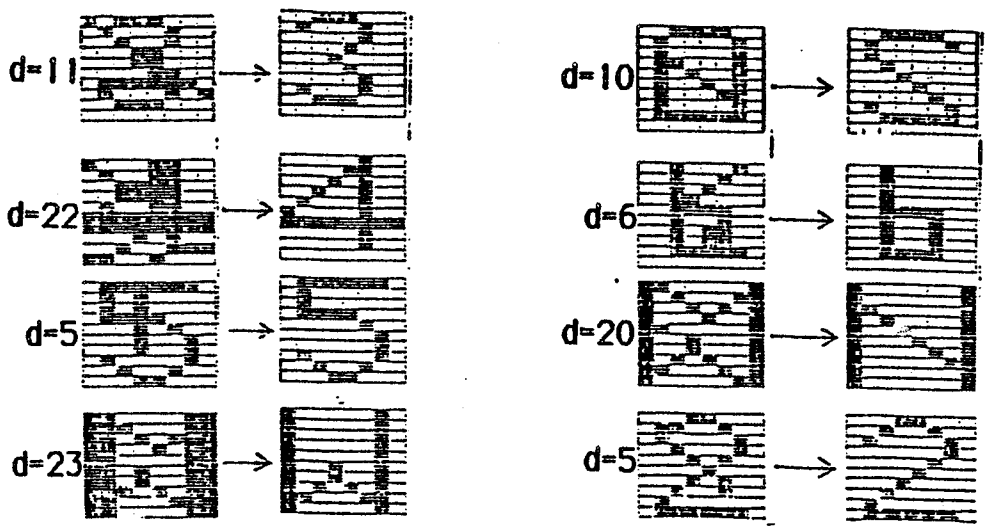
(a)



(b)

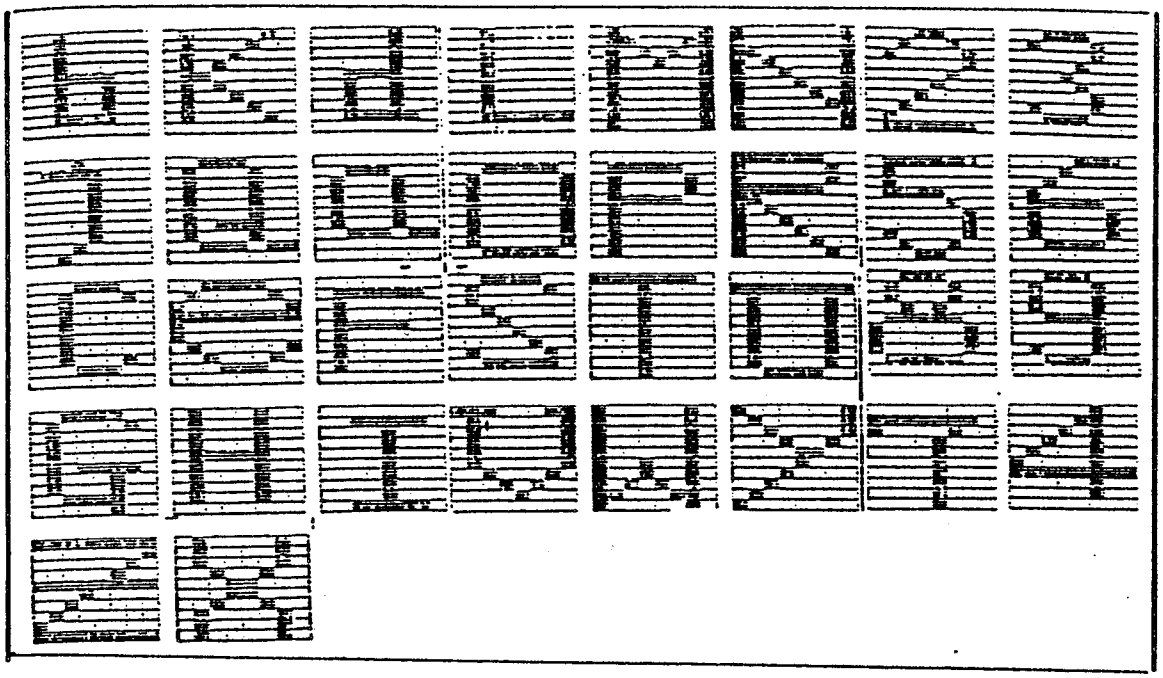
Figure 3.9





La méthode de meilleure approximation

Figure 3.10



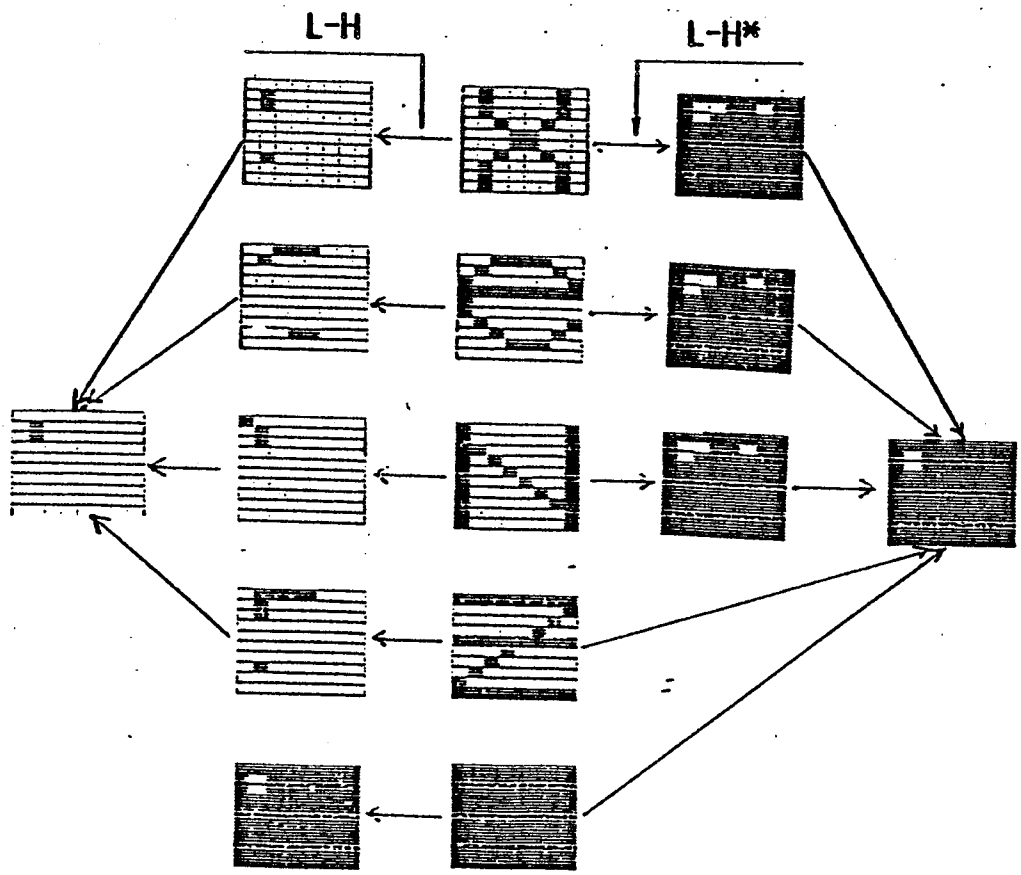
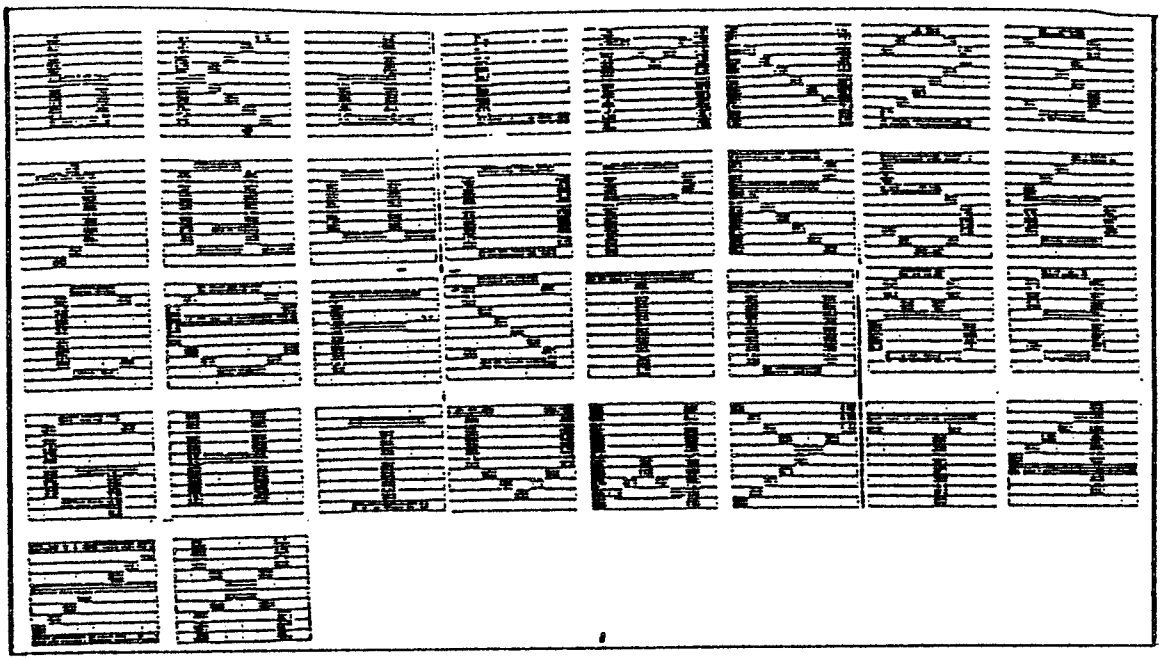


Figure 3.11



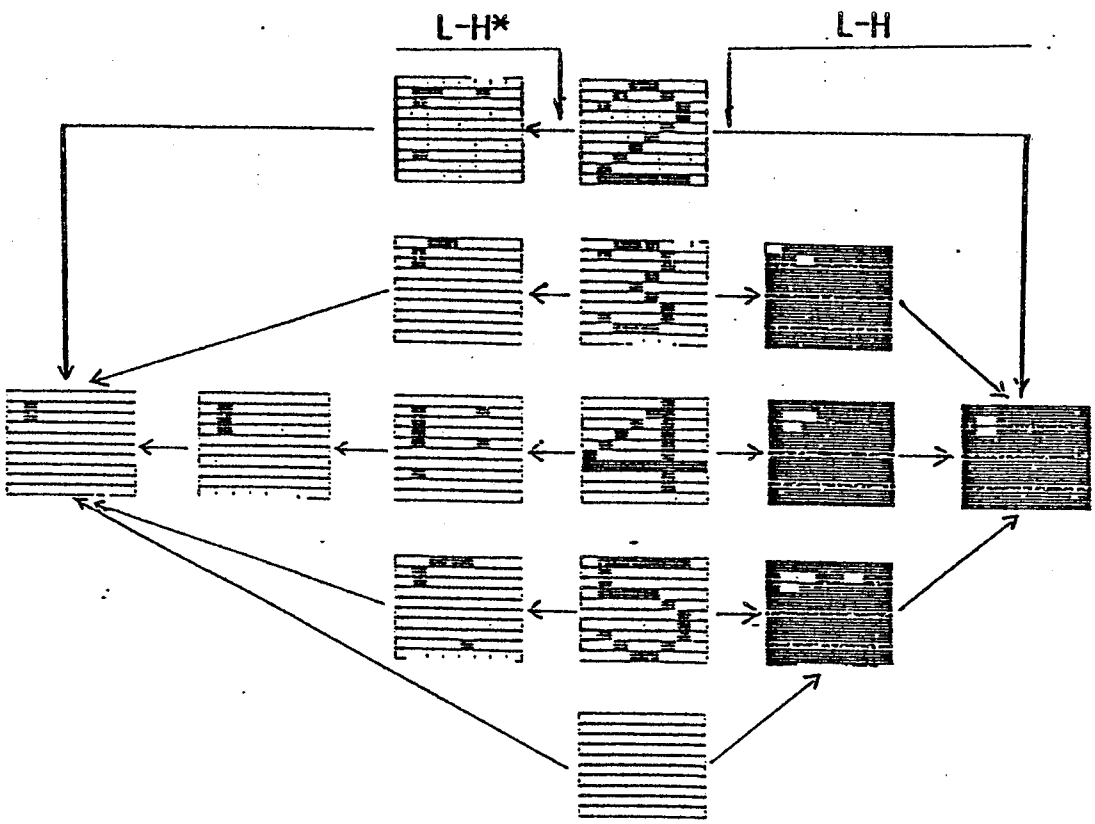


Figure 3.12

Références

[1] Françoise FOGELMAN-SOULIE . " Contribution à une théorie du calcul sur réseaux " Thèse d'Etat , Grenoble (1985) .

[2] Françoise FOGELMAN-SOULIE , E.GOLES-CHACC et G.WEISBUCH . " Transient length in sequential iteration of threshold functions " dans Discrete Applied Mathematics -6, pp 95-98 (1983).

[3] Françoise FOGELMAN-SOULIE , E.GOLES-CHACC et D.PELLEGRIN . " Decressing energy function as a tool for studing theshold network " ; soumis à Discrete Applied Mathematics .

[4] Françoise FOGELMAN-SOULIE et G.WEISBUCH . " Random iterations of threshold networks and associative memory " ; soumis à SIAM J. on computing .

[5] E.GOLES-CHACC . " Comportement dynamique de réseaux d'automates " . Thèse d'Etat , Grenoble (1985) .

[6] J. J. HOPFIELD . "Neural networks and physical systems with emergent collective computational abilities " . Proc. Nat. Acad. Sci. USA , Vol. 79. pp. 2554-2558 . (1982).

[7] W. S. MACCULLOCH, W. PITTS . " A logical calculus of the ideas immanent in nervous activity " . Bull. Mathem. Biophys. 5. pp 115-133 . (1943)

DERNIERE PAGE D'UNE THESE

~~3E CYCLE DOCTEUR INGÉNIEUR OU UNIVERSITÉ~~

DOCTORAT D'UNIVERSITE MATHEMATIQUES

Vu les dispositions de l'arrêté du 16 avril 1974,

Vu les rapports de M. *F. ROBERT* .....

M. .....

M. *onsieur PAN Xin An* ..... est autorisé

à présenter une thèse en vue de l'obtention du grade de DOCTEUR *d'Université* .....

.....

Grenoble, le *20 mai 1985*

Le Président de l'Université Scientifique
et Médicale

M. TANCHE

UNIVERSITÉ SCIENTIFIQUE ET MÉDICALE DE GRENOBLE

M. Tanche

Résumé :

Ce travail concerne l'expérimentation mumérique de trois modéles de MathématiqueS Appliquées :

Le premier chapitre est consacré à la localisation des minima de Pareto relatifs à plusieurs formes quadratiques.

Le second chapitre concerne l'expérimentation numérique d'un modele itératif de la respiration.

Le troisième chapitre , le plus dévéloppé , est consacré à la reconnaissance de caractères par un réseau d'automates à seuil . De nouveaux algorithmes sont proposés , dont on montre l'efficacité accrue . L'expérimentation est conduite sur la reconnaissance de caractères chinois.

Mots clés : minima de Pareto , algorithmes itératifs , réseaux d'automates , automates à seuil , reconnaissance de caractéres , distance de Hamming .