

# REVUE DE STATISTIQUE APPLIQUÉE

R. CHIFFLET

## **Recherche de partitions hiérarchisées optimales ( $k$ -structure de hiérarchies optimales)**

*Revue de statistique appliquée*, tome 30, n° 1 (1982), p. 25-39

[http://www.numdam.org/item?id=RSA\\_1982\\_\\_30\\_1\\_25\\_0](http://www.numdam.org/item?id=RSA_1982__30_1_25_0)

© Société française de statistique, 1982, tous droits réservés.

L'accès aux archives de la revue « *Revue de statistique appliquée* » (<http://www.sfds.asso.fr/publicat/rsa.htm>) implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

# RECHERCHE DE PARTITIONS HIERARCHISEES OPTIMALES (k-structure de hiérarchies optimales)

R. CHIFFLET

Chargé de Mission à la Direction de la Prévision  
(Travail fait à l'INRIA de Mai 1980 à Août 1980)  
dans l'Equipe de M. DIDAY

## SOMMAIRE

1. INTRODUCTION
  2. FORMALISATION DU PROBLEME
    - 2.1. Définitions
    - 2.2. Formalisation du problème
  3. APPLICATION DE LA METHODE DES NUEES DYNAMIQUES A LA RECHERCHE D'UNE SOLUTION DU PROBLEME POSE
    - 3.1. L'espace des recouvrements et l'espace des représentations
    - 3.2. Le critère à optimiser
    - 3.3. L'algorithme
      - 3.3.1. Les fonctions d'affectation
      - 3.3.2. Les fonctions de représentation
      - 3.3.3. L'algorithme
      - 3.3.4. Remarques
  4. LE PROGRAMME ETABLI
    - 4.1. Les résultats obtenus
    - 4.2. Un exemple de sortie du programme établi
    - 4.3. Quelques remarques relatives au programme et à l'algorithme
      - 4.3.1. Remarques relatives au programme
      - 4.3.2. Remarques relatives à l'algorithme
  5. CONCLUSION
- BIBLIOGRAPHIE

## 1. INTRODUCTION

Les méthodes de classification automatique du type Nuées dynamiques ont longtemps semblé ne pouvoir déterminer sur une population donnée que des structures du genre partition localement optimales au sens d'un certain critère. Les tra-

vaux de DIDAY [9] et de CHIFFLET [6] ont montré qu'elles pouvaient, en fait, s'appliquer pour la recherche sur un ensemble de hiérarchie optimale au sens d'un critère (signalons ici un algorithme qui a été défini en [6] visant à doter un ensemble d'une hiérarchie dont la somme des inerties des éléments la composant est minimale). Une généralisation du problème de la recherche d'une hiérarchie optimale a été faite en [6] : En [6] ont été définis des algorithmes visant à munir une population donnée d'une partition dont chaque classe est dotée d'une structure hiérarchique (arborescence, arbre, hiérarchie, ultramétrique) optimale au sens d'un critère dépendant de ces structures.

Les méthodes classiques de classification automatique hiérarchiques peuvent, quant à elles, se classer dans les catégories suivantes :

- celles qui n'optimisent aucun critère global (ou qui optimisent un critère global pas à pas). Citons parmi celles-ci la majeure partie des méthodes ascendantes ou descendantes hiérarchiques (BENZECCI [1] ; BRUNOOGHE [3], JAMBU [11], LERMAN [12], WARD [14] ;
- celles qui visent à optimiser un critère global (CAROLL [5], CHANDON [6] définissent des algorithmes pour munir un ensemble E d'une ultramétrique la plus proche au sens des moindres carrés d'une mesure de ressemblance donnée sur E) ;
- celles qui optimisent un critère global (single linkage).

Nous nous intéressons, dans cet article, au problème de la recherche de partitions hiérarchisées par la donnée sur chacune de leur classe d'une hiérarchie optimale au sens d'un certain critère. L'algorithme présenté ici est une variante de ceux que nous avons définis en [6] et [11] (respectivement dans les chapitres "Hiérarchie pré-indicée optimale" et "Ultramétriques adaptatives"). Nous donnerons une sortie du programme que nous avons établi et, en conclusion, les perspectives de travail que nous nous sommes fixée quant à l'application de cet algorithme.

## 2. FORMALISATION DU PROBLEME

### 2.1. DEFINITIONS

#### Hiérarchie binaire

On appelle hiérarchie binaire définie sur un ensemble fini E tout ensemble H de parties de E tel que :

- (i)  $\forall x \in E ; \{x\} \in H$
- (ii)  $E \in H$
- (iii)  $\forall (B, B') \in H^2 \quad B \cap B' = \begin{cases} B \\ \emptyset \\ B' \end{cases}$
- (iv)  $\forall B \in H \quad \text{Card } B > 1 \implies \exists (B_1, B_2) \in H^2 \quad B_1 \not\subset B ; B_2 \not\subset B$   
et  $B_1 \cup B_2 = B$

### k- Structure de hiérarchie

Soit  $k$  un entier donné à priori. On appelle  $k$ -structure de hiérarchie définie sur un ensemble fini  $E$  la donnée d'un  $k$  uple  $\bar{H} = (H_1, \dots, H_k)$  de hiérarchie vérifiant les propriétés suivantes :

- chaque hiérarchie  $H_i$  est définie sur une partie  $P_i$  de  $E$  ceci pour tout  $i \in [1, k]$
- l'ensemble  $P = \{P_1, \dots, P_k\}$  forme une partition de  $E$  en  $k$  classes appelée support de  $\bar{H}$

### Indice de distance entre partie de E

On appelle indice de distance entre parties de  $E$  toute application  $C$  de  $(\mathcal{P}(E) - \{\emptyset\})^2$  dans  $\mathbb{R}^+$  vérifiant :

- (i)  $\forall (B, B') \in (\mathcal{P}(E) - \{\emptyset\})^2 \quad C(B, B') = C(B', B)$
- (ii)  $\forall x \in E \quad C(\{x\}, \{x\}) = 0$
- (iii)  $\forall (x, y) \in E^2 \quad x \neq y \Rightarrow C(\{x\}, \{y\}) > 0$

### Remarque 1

Nous supposons à partir de maintenant que  $E$  est muni d'un indice de distance noté  $C$ .

### Ultramétrie associée à une hiérarchie binaire

Soit  $H$  une hiérarchie binaire définie sur  $E$  ; on définit alors sur  $E \times E$  une ultramétrie  $[6]$  notée  $d_H$  par :

- (i)  $\forall x \in E \quad d_H(x, x) = 0$
- (ii)  $\forall (x, y) \in E^2$

$$x \neq y \Rightarrow d_H(x, y) = \sup (\delta(B), C(B, B'), \delta(B'))$$

- où  $B$  et  $B'$  sont les 2 éléments de  $H$ , dont l'intersection est vide, la réunion appartient à  $H$  et tels que  $x \in B$  et  $y \in B'$
- et
- où  $\forall B'' \in \{B, B'\} \quad \delta(B'') = \sup_{(\lambda, \mu) \in B''^2} d_H(\lambda, \mu)$

$(\delta(B''))$  est le diamètre de  $B''$

## 2.2. FORMALISATION DU PROBLEME

### Dispersion d'une partie B de E relativement à un couple (x, H)

On appelle dispersion d'une partie  $B$  de  $E$  relativement à un couple  $(x, H)$  où  $x$  est un point de  $E$  et  $H$  une hiérarchie définie sur  $B$  et on note  $I(B, (x, H))$  la quantité

$$I(B, (x, H)) = \sum_{y \in B} d_H(x, y)$$

### Dispersion d'une partie B de E relativement à une hiérarchie H définie sur B

On appelle dispersion d'une partie B de E relativement à une hiérarchie H définie sur B et on note  $\bar{I}(B, H)$  la quantité

$$I(B, H) = \min_{x \in B} I(B, (x, H))$$

Tout élément  $x_H$  de B tel que  $I(B, (x_H, H)) = \bar{I}(B, H)$  sera appelé élément représentatif de B relativement à H.

### Critère associé à une partition structurée par la donnée sur chacune de ses classes à une hiérarchie.

Soit  $P = \{P_1, \dots, P_k\}$  une partition structurée par la donnée sur chacune de ses classes  $P_i$  d'une hiérarchie  $H_i$ . On appelle critère associé à la partition hiérarchisée  $(P, H)$  (où  $H = (H_1, \dots, H_k)$ ), ou critère associé à  $\bar{H}$  et on note  $\bar{W}(\bar{H})$  la quantité :

$$\begin{aligned} W(H) &= \sum_{i=1}^k I(P_i, H_i) \\ &= \sum_{i=1}^k I(P_i, (x_{H_i}, H_i)) \end{aligned}$$

### Le problème posé

Le problème posé à savoir : Trouver une partition hiérarchisée (par la donnée sur chacune de ses classes d'une hiérarchie) optimale se formalise en :

Trouver une k-structure de hiérarchie  $\bar{H}^*$  telle que la quantité  $\bar{W}(\bar{H}^*)$  soit minimale.

#### Remarque 2

Pour toute k-structure de hiérarchie  $\bar{H}$  la quantité  $\bar{W}(\bar{H})$  représente en fait l'adéquation notée  $W(P, (x^H, H))$  entre le support P de  $\bar{H} = (H_1, \dots, H_k)$  et le couple  $(x^H, \bar{H})$  où

$$x^H = (x_{H_1}, \dots, x_{H_k})$$

La quantité  $\bar{W}(\bar{H})$  n'est donc que l'adéquation entre une partition P et un couple  $L = (x^H, \bar{H})$  de k uple de points de E et de k-structure de hiérarchie.

Nous sommes donc tout naturellement amenés à utiliser la méthode des Nuées Dynamiques pour la recherche d'un élément  $\bar{H}^*$  optimal.

## 3. APPLICATION DE LA METHODE DES NUÉES DYNAMIQUES A LA RECHERCHE D'UNE SOLUTION DU PROBLEME POSE

### 3.1. L'ESPACE DES RECOUVREMENTS ET L'ESPACE DES REPRESENTATIONS

On appelle espace des recouvrements et on désigne par  $P_k$  l'ensemble des partitions de E en k classes. On appelle espace des représentations associé à  $P_k$

et on désigne par  $L$  l'ensemble  $E^k \times H$  où  $H$  désigne l'ensemble des  $k$ -structures de hiérarchies définies sur  $E$ .

### 3.2. LE CRITERE A OPTIMISER

Nous supposons ici donné [6] un procédé permettant d'associer à toute  $k$ -structure d'ultramétriques  $(d_1, \dots, d_k)$  définie sur une partition  $P = \{P_1, \dots, P_k\}$  de  $E$  une ultramétrique  $d$  définie sur  $E$  vérifiant :

Pour tout  $i \in [1, k]$  la restriction de  $\bar{d}$  à  $P_i$  est égale à  $\bar{d}_i$

*Remarque 3*

La démonstrations de la convergence de l'algorithme que nous allons définir, c'est-à-dire en fait la preuve de l'existence de deux fonctions  $f$  et  $g$  vérifiant certaines propriétés, est étroitement liée au procédé d'affectation défini dans [6]

**Ultramétrique associée à une  $k$ -structure de hiérarchie.**

Soit  $\bar{H} = (H_1, \dots, H_k)$  une  $k$ -structure de hiérarchie. On appellera ultramétrique associée à  $\bar{H}$  et on notera  $\bar{d}_H$  l'ultramétrique associée à  $(d_{H_1}, \dots, d_{H_k})$ .

**Le critère à optimiser**

On définit le critère à optimiser (c'est-à-dire en fait à minimiser) comme étant l'application de  $P_k \times L$  dans  $R^+$  vérifiant :

$$\forall \bar{x} = (x_1, \dots, x_k) \in E^k ; \forall \bar{H} \in H ; \forall P = \{P_1, \dots, P_k\} \in P_k$$

$$W(P, (\bar{x}, \bar{H})) = \sum_{i=1}^k \sum_{x \in P_i} d_H(x, x_i)$$

*Remarque 4*

La quantité  $W(P, (\bar{x}, \bar{H}))$  peut se mettre sous la forme  $\sum_{i=1}^k I(P_i, (x_i, \bar{H}))$ ,

où pour tout  $i \in [1, k]$

$I(P_i, (x_i, \bar{H}))$  représente la dispersion de  $P_i$  par rapport à  $x_i$  relativement à  $\bar{H}$ , c'est-à-dire où :

$$\forall i \in [1, k] I(P_i, (x_i, \bar{H})) = \sum_{x \in P_i} d_H(x, x_i)$$

### 3.3. L'ALGORITHME

Nous allons définir un algorithme basé sur la méthode des Nuées Dynamiques [10] visant à obtenir un couple  $(P^*, L^*)$  optimal, c'est-à-dire minimisant la quantité  $W(P^*, L^*)$ .

Nous allons pour ce faire, construire deux fonctions  $f$  et  $g$  appelées fonction d'affectation et fonction de représentation [10]. Nous donnerons des conditions suffisantes quant à la démonstration de la convergence de l'algorithme que nous définirons alors. Nous verrons enfin l'apport de cet algorithme par rapport à celui défini en [10] au chapitre "Ultramétriques Adaptatives".

### 3.3.1. Les fonctions d'affectation

On appelle fonction d'affectation toute application  $f$  de  $L$  dans  $P_k$

$$\begin{aligned} f : L &\longrightarrow P_k \\ L &\longrightarrow f(L) \end{aligned}$$

vérifiant

$$W(f(L), L) = \min_{P \in P_k} W(P, L)$$

### 3.3.2. Les fonctions de représentation

Les fonctions de représentation que nous allons définir seront, en fait, la composée de deux fonctions  $g_1$  et  $g_2$  dont nous allons maintenant parler.

**Les fonctions  $g_1$**  (appelées par la suite fonction de structuration)

On désigne par  $g_1$  toute application

$$\begin{aligned} g_1 : P_k &\longrightarrow H \\ P &\longrightarrow g_1(P) \end{aligned}$$

vérifiant

$$\forall \bar{x} \in E^k$$

$$W(P, (g_1(P), \bar{x})) = \min_{H \in H_P} W(P, (H, \bar{x}))$$

où  $H_P$  désigne l'ensemble des  $k$ -structures de hiérarchie dont le support est  $P$ .

*Remarque 5*

L'existence de telles fonctions  $g_1$  n'est pas évidente a priori. Elle dépend de l'indice de distance entre parties défini sur  $E$ . Nous indiquerons au paragraphe 4 un indice de distance pour lequel nous avons montré l'existence de telles fonctions.

**Les fonctions  $g_2$**

On désigne par  $g_2$  toute application

$$\begin{aligned} g_2 : H &\longrightarrow E^k \\ H &\longrightarrow g_2(\bar{H}) \end{aligned}$$

vérifiant :

$$\left\{ \begin{array}{l} \forall H \in H \quad W(P, (g_2(\bar{H}), \bar{H})) = \min_{\bar{x} \in E^k} W(P, (\bar{x}, \bar{H})) \\ \text{où } P \text{ désigne le support de } \bar{H} \end{array} \right.$$

**Les fonctions de représentation**

On appelle fonction de représentation toute application

$$\begin{aligned} g : P_k &\longrightarrow L \\ P &\longrightarrow g(P) \end{aligned}$$

vérifiant :

$$\forall P \in \mathbf{P}_k \quad g(P) = (g_2 \circ g_1(P), g_1(P))$$

où  $g_1$  et  $g_2$  sont deux fonctions du type de celles définies précédemment données a priori.

### 3.3.3. L'algorithme

Soit  $P^0 \in \mathbf{P}_k$  ; on définit les trois suites  $P^n$ ,  $L^n$  et  $U_n$  par récurrence de la manière suivante :

$$\begin{aligned} \forall n \in \mathbf{N} \quad L^n &= g(P^n) \\ P^{n+1} &= f(L^n) \\ U_n &= W(P^n, L^n) \end{aligned}$$

#### Notations

Désignons pour tout  $n \in \mathbf{N}$  par

$(\bar{y}^n, \bar{H}^n)$  l'élément  $L^n$

$\{P_1^n, \dots, P_k^n\}$  la partition  $P^n$

#### Théorème

Si pour tout  $n \in \mathbf{N}^*$  les conditions 1) et 2) suivantes sont vérifiées :

1)  $\bar{y}^n \in P_1^n \times \dots \times P_k^n$

2)  $\exists H'^n \in \mathbf{H}$  vérifiant :

$\alpha)$  le support de  $H'^n$  est  $P^{n+1}$

$\beta)$   $W(P^{n+1}, L^n) \geq W(P^{n+1}, (\bar{y}^n, H'^n))$

alors

3)  $\forall n \in \mathbf{N} \quad \bar{W}(\bar{H}^n) = U_n$

4) la suite  $U_n$  converge en décroissant en un nombre fini d'itérations

#### Démonstration

Soit  $n \in \mathbf{N}^*$ . Du fait de 1) et de la définition de  $\bar{W}$ , la condition 3) est trivialement vérifiée. Il nous reste à démontrer la condition 4).

On a :

$n = W(P^n, L^n) \geq W(P^{n+1}, L^n)$  ceci par définition de  $f$

Or, d'après 2) il existe  $H'^n \in \mathbf{H}$  dont le support est  $P^{n+1}$  tel que

$$W(P^{n+1}, L^n) \geq W(P^{n+1}, (\bar{y}^n, H'^n))$$

Or, d'après la définition de  $g_1$ , on a :

$$W(P^{n+1}, (\bar{y}^n, H'^n)) \geq W(P^{n+1}, (\bar{y}^n, g_1(P^{n+1})))$$

Par suite, d'après la définition de  $g_2$ , on a :

$$W(P^{n+1}, (\bar{y}^n, g_1(P^{n+1}))) \geq W(P^{n+1}, L^{n+1}) = U_{n+1}$$



Le fait que la suite  $U_n$  converge en décroissant en un nombre fini d'itération est alors trivial. En effet, l'ensemble des valeurs possibles de  $W$  est fini. La suite  $U_n$  est alors une suite décroissante dans un sous-ensemble fini de  $\mathbb{R}$ .

### 3.3.4. Remarques

#### *Remarques sur les hypothèses du théorème*

La condition 1) du théorème ne pose pas de problème. Il existe, en effet, toujours des fonctions  $g_2$  la rendant vraie. Par contre, la démonstration de la validité de la condition 2) de ce théorème est plus délicate. Elle dépend, en fait, essentiellement de l'indice de distance entre parties défini sur  $E$ . Nous indiquerons au paragraphe suivant un indice de distance grâce auquel nous avons pu construire [6] des fonctions  $f$  et  $g$  rendant valides ces conditions 1) et 2).

#### *Remarques sur l'algorithme défini au chapitre "Ultramétries Adaptatives"* de [10]

Une différence essentielle entre l'algorithme défini en [10] au chapitre Ultramétries Adaptatives et celui précité réside dans la définition des fonctions de structuration.

En effet, nous avons défini ici la fonction de structuration  $g_1$  comme associant à toute partition  $P$  "la meilleure  $k$ -structure de hiérarchie dont le support est  $P$ ". Ceci, d'un point de vue pratique, nous a amenés à utiliser un algorithme de classification automatique hiérarchique visant à munir chaque classe de  $P$  d'une hiérarchie optimale. Le coût en temps calcul d'une telle procédure est bien moindre que celui mis pour définir sur la population toute entière, une hiérarchie au moyen de ce dernier algorithme.

Par contre, dans [10], on désigne la fonction de structuration (notée  $h$ ) comme associant à toute partition  $P$  "la meilleure  $k$ -structure d'ultramétries (hiérarchie) relativement à  $P$ ". D'un point de vue pratique, ceci nous conduit à définir sur la population totale une ultramétrie optimale et, de ce fait, à employer un algorithme de classification automatique hiérarchique sur toute la population. On ne peut donc espérer utiliser l'algorithme défini en [10] pour munir un grand ensemble de données d'une structure hiérarchique optimale (les méthodes classiques de classification automatique hiérarchique ne le permettent pas).

## 4. LE PROGRAMME ETABLI

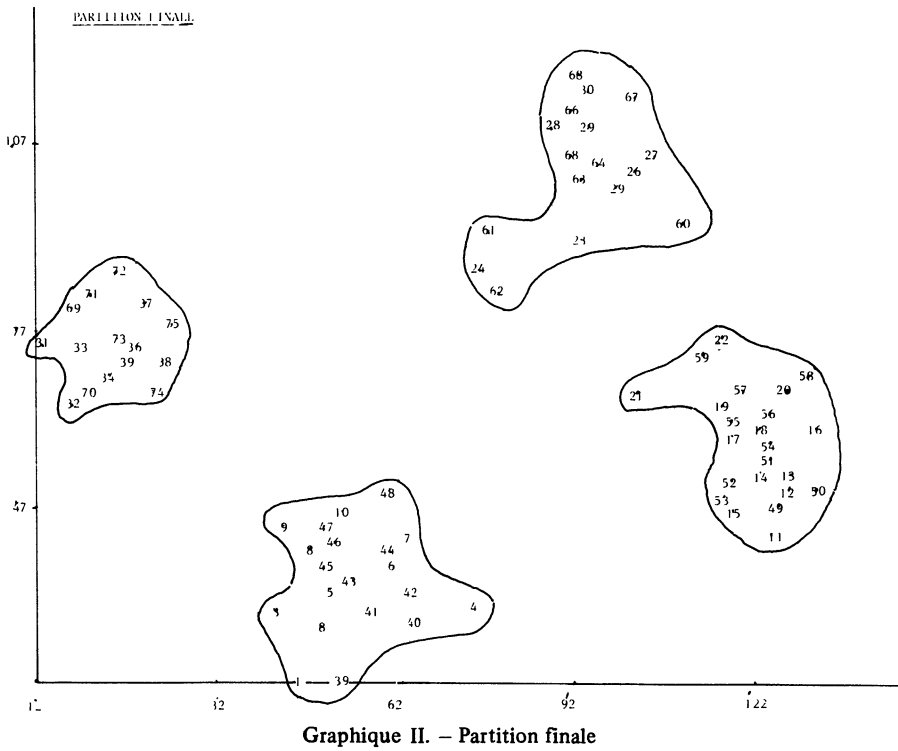
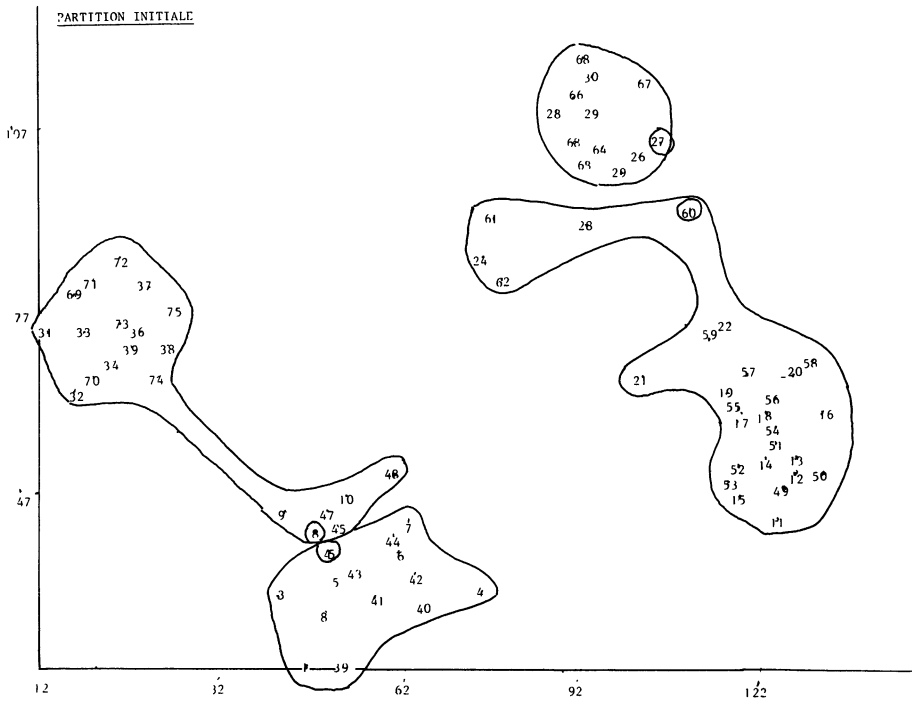
### 4.1. Les résultats obtenus

Nous supposons  $E$  muni de l'indice de distance  $d_{\min}$  défini par :

$$\forall (A, B) \in \mathcal{P}^2(E) ; d_{\min}(A, B) = \underset{(x,y) \in A \times B}{\text{Min}} d(x, y)$$

A partir des résultats des chapitres 2 et 4 de [6], on montre :

- l'existence de deux fonctions  $f$  et  $g$  rendant valides les conditions 1) et 2) du théorème précédent ;



- qu'une hiérarchie optimale sur E est une hiérarchie associée (de manière habituelle) à un arbre de longueur minimum [2] défini sur E ;
- que si  $\bar{H}$  est une k-structure de hiérarchie sur E optimale, alors l'ultramétrie  $d_{\bar{H}}$  associée à  $\bar{H}$  n'est autre que l'ultramétrie sous dominante maximale définie sur E ;
- que si H est une hiérarchie définie sur E optimale alors la k-structure de hiérarchie obtenue en ôtant à H ses k classes de plus grand cardinal n'est, en général, pas optimale.

### Conséquence

La donnée d'une k-structure de hiérarchie optimale sur E permet de munir E d'une hiérarchie optimale.

## 4.2. UN EXEMPLE DE SORTIE DU PROGRAMME ETABLI

Nous donnons ici les résultats du passage de notre programme sur les données de Ruspini [13] en demandant une 4-structure de hiérarchie. Les données de Ruspini sont des données artificielles dans  $R^2$  comportant quatre groupes homogènes bien distincts. De façon intuitive, une 4-structure de hiérarchie optimale consisterait en la donnée :

- de ces quatre groupes,
- sur chacun de ces groupes, d'une hiérarchie optimale.

La 4-structure de hiérarchie obtenue à partir de la partition initiale définie au graphique I, à l'aide de notre programme, vérifie bien les deux conditions précitées. Cette partition initiale a été formée en affectant chaque donnée à l'élément de l'ensemble  $\{8, 45, 60, 27\}$  le plus proche de cette dernière.

Les données :

Nom	Abscisse	Ordonnée
1	45	17
2	50	23
3	41	26
4	74	26
5	51	30
6	62	34
7	63	39
8	46	37
9	43	40
10	51	42
11	125	42
12	127	48
13	127	51
14	122	51
15	120	45
16	131	56
17	119	55
18	122	57

Nom	Abscisse	Ordonnée
19	118	61
20	127	62
21	103	63
22	116	72
23	94	88
24	77	82
25	100	98
26	102	100
27	105	101
28	90	107
29	95	108
30	95	113
31	2	71
32	9	62
33	11	71
34	13	69
35	15	68
36	17	71

Nom	Abscisse	Ordonnée
37	20	78
38	23	68
39	53	18
40	65	23
41	57	26
42	63	28
43	54	31
44	61	38
45	48	35
46	49	38
47	49	41
48	59	46
49	126	45
50	129	47
51	124	53
52	120	50
53	118	47
54	124	56
55	118	57
56	124	58

Nom	Abscisse	Ordonnée
57	120	63
58	130	64
59	113	68
60	109	90
61	78	88
62	78	79
63	94	99
64	97	100
65	93	102
66	92	109
67	103	111
68	93	115
69	8	78
70	11	65
71	11	79
72	14	83
73	15	72
74	19	66
75	23	75

### 4.3. QUELQUES REMARQUES RELATIVES AU PROGRAMME ET A L'ALGORITHME

#### 4.3.1. Remarques relatives au programme

Le programme que nous avons établi a fait apparaître clairement un des avantages de la recherche de k-structure de hiérarchie optimale, à savoir la facilité de lecture au premier abord des résultats.

Ce programme nécessite (sous sa forme actuelle) à chaque itération  $n$  le calcul et le stockage des distances entre les classes de la partition  $P^n$ .

#### 4.3.2. Remarques relatives à l'algorithme

Nous supposons ici disposer d'un algorithme de classification automatique hiérarchique (pour munir chacune des classes de la partition  $P^n$  d'une hiérarchie optimale et pour munir l'ensemble des classes de  $P^n$  d'une hiérarchie optimale) dont le temps de calcul est proportionnel au cardinal de l'ensemble sur lequel il s'applique à la puissance  $\alpha$  ( $\alpha \in \mathbb{R}_+^*$ ).

Nous supposons que toutes les classes de la partition  $P^n$  et de la partition  $P^{n+1}$  ont sensiblement le même nombre d'éléments ; ce nombre d'éléments étant voisin de  $\sqrt{m}$  où  $m = \text{Card } E$ .

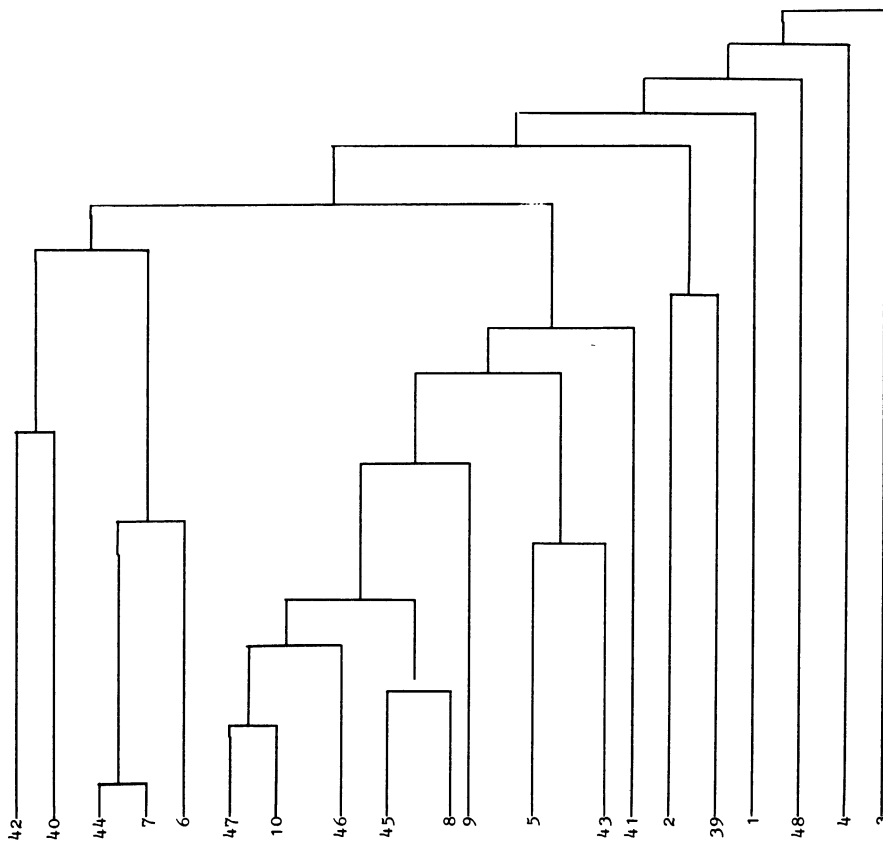
Dans ce cas, si l'on ne tient pas compte du coût du calcul de la matrice des distances entre les classes de la partition  $P^n$ , le coût en temps calcul du passage de l'itération  $n$  à l'itération  $n + 1$  est proportionnel à

$$m^{\alpha/2} (1 + m^{1/2})$$

Ceci est très important, car sous certaines hypothèses notre algorithme permet de munir un ensemble de données en un temps proportionnel à

Sorties

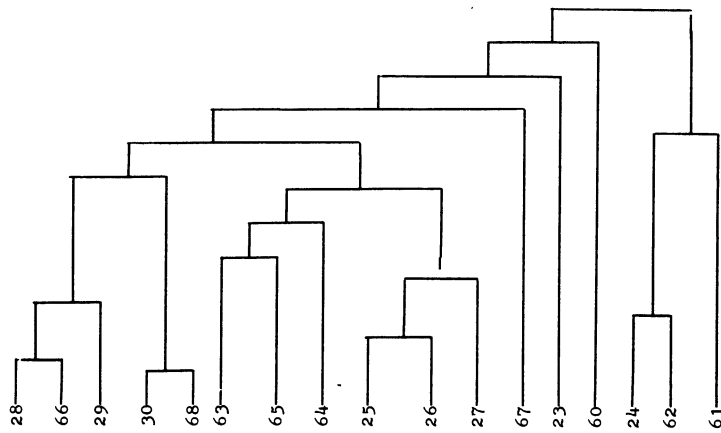
Hiéarchie sur la classe 1



Dispersion de la classe 1  
 Élément représentatif 10  
 Nombre d'éléments 20

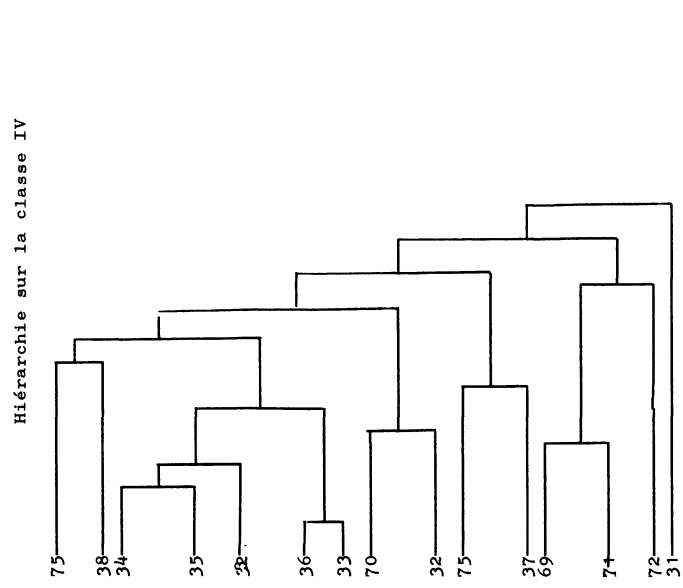
760.000

Hiéarchie sur la classe 2



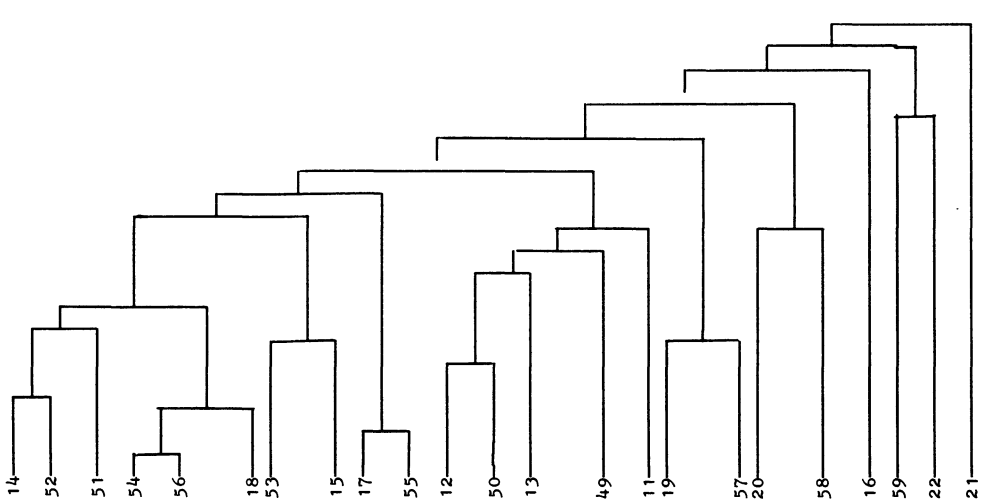
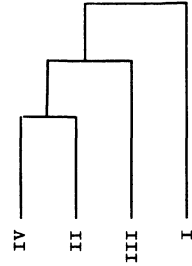
Valeur du critère 1.321.000  
 Élément représentatif 25  
 Nombre d'éléments 17

Hiérarchie sur la classe IV



Valeur du critère 472.000  
 Élément représentatif 34  
 Nombre d'éléments 15

hiérarchie sur l'ensemble des classes



Valeur du critère 549.00  
 Élément représentatif 54  
 Nombre d'éléments 23

$m^{\alpha/2} (1 + m^{1/2})$  en se servant d'un algorithme de classification automatique hiérarchique qui, utilisé seul, nécessiterait un temps machine proportionnel à  $m^\alpha$ .

D'où l'idée de chercher à appliquer cet algorithme pour munir de grands ensembles de données (de 5 000 à 15 000 individus) de hiérarchies localement optimales.

## 5. CONCLUSION

Nous avons cherché à associer à tout ensemble de hiérarchies locales définies sur les classes d'une partition un critère de qualité. Nous avons alors établi un algorithme basé sur la méthode des Nuées Dynamiques visant à obtenir un ensemble de hiérarchies locales optimal au sens de ce critère.

### LE CRITERE

Nous avons donné des procédés pour associer à toute hiérarchie de parties une ultramétrie. Cela nous a permis de définir la dispersion d'une partie relativement à une hiérarchie qui y était établie. Nous avons alors considéré que la qualité d'un ensemble de hiérarchies locales peut s'exprimer comme la somme des dispersions des classes sur lesquelles elles sont définies.

(Nous aurions pu prendre d'autres critères tels que la somme des inerties des classes des hiérarchies locales [6] ou tels que la somme des écarts des hiérarchies à une mesure de ressemblance définie sur la population [6]).

### L'ALGORITHME

Nous avons alors défini un algorithme basé sur la méthode des Nuées Dynamiques visant à obtenir une partition hiérarchisée optimale. Une bonne connaissance a priori des données aide à une convergence rapide de ce dernier.

### LE PROGRAMME ET SES LIMITES ACTUELLES

Le programme que nous avons établi utilise dans sa version actuelle le calcul à chaque itération de la matrice des distances entre les classes de la partition  $P^n$  alors obtenue. Il se sert d'un algorithme de classification automatique hiérarchique (voir le paragraphe 4.3.2.) dont le temps de calcul est proportionnel au cube du cardinal de l'ensemble sur lequel on l'applique.

Dans sa forme actuelle, il ne peut s'appliquer qu'à des ensembles relativement réduits d'individus.

### LES PERSPECTIVES DE TRAVAIL

Nous comptons rendre notre programme opérationnel pour le traitement de grands tableaux de données.

Pour ce faire, nous allons d'une part remplacer l'algorithme de classification automatique hiérarchique que nous utilisons par un algorithme tel que [3]. Nous allons, d'autre part, approximer le calcul de la matrice des distances entre les classes d'une partition donnée afin d'en rendre le coût en temps machine proportionnel au nombre de points de la population.

Nous dirons, en conclusion, que l'intérêt de cet article est double :

- il donne un nouvel algorithme basé sur la méthode des Nuées Dynamiques pour la recherche de partitions hiérarchisées optimales ;
- il montre que cet algorithme peut, sous certaines hypothèses, s'appliquer pour munir de grands ensembles de données (entre 6 000 et 15 000) d'une hiérarchie localement optimale.

### BIBLIOGRAPHIE

- [1] BENZECRI. – *L'Analyse des Données*. Dunod 1973.
- [2] BERGE. – *Graphe et Hypergraphe*, Dunod.
- [3] BRUYNNOOGHE. – *Classification automatique d'un grand tableau de données par la méthode des graphes réductibles*. Colloque IRIA, Septembre 1977.
- [4] CHANDON. – *Construction de l'ultramétrie la plus proche au sens des moindres carrés*. Note interne IAE, Mai 1978.
- [5] CARROL and PRUZANSKY. – *Fitting of hierarchical tree structure (HTS) models mixtures of HTS models and hybrid models via mathematical programming and alternating least squares*. Bell Laboratories Murray Hill, New-Jersey 1974.
- [6] CHIFFLET. – *Structures hiérarchiques locales optimales*. Thèse de 3<sup>e</sup> cycle 1979, Paris.
- [7] CORMACK. – A review of classification. *J.R. Statistic Soc.*, Vol. 134, part 3 1971.
- [8] DIDAY. – *Classification automatique séquentielle pour grands tableaux*, RAIRO, Mars 1975.
- [9] DIDAY. – *Problems of clustering and recent advances*. Invited paper at the 11 European Congress of Statistics Oslo 1978.
- [10] DIDAY et Collaborateur. – *Optimisation en classification automatique*, INRIA, 1980.
- [11] JAMBU. – *Classification automatique pour l'analyse des données*. Dunod 1979.
- [12] LERMAN. – *Méthodes combinatoires et statistiques dans le traitement, des données de comportement*. IRISA, Université de Rennes, 1978.
- [13] RUSPINI. – Numerical methods for fussy clustering. *Information Science*, 2, 1970, 319-350.
- [14] WARD. – *Hiérarchical grouping to optimize an objective function*. JASA, March 1963.