Optimisation de classifications hiérarchiques

Henri Ralambondrainy, R. Chifflet

INRIA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. 954 90 20

Rapports de Recherche

N° 70

OPTIMISATION DE CLASSIFICATIONS HIÉRARCHIQUES

Henri RALAMBONDRAINY
Roland CHIFFLET

Avril 1981

OPTIMISATION DE CLASSIFICATIONS HIERARCHIQUES

HENRI RALAMBONDRAINY
ROLAND CHIFFLET*

RESUME

Les méthodes de classification hiérarchiques habituelles fournissent des hiérarchies qui optimisent à chaque étape le critère d'agrégation. Ces méthodes n'optimisent pas de critère global.

La première partie de ce rapport présente un algorithme qui recherche une hiérarchie optimale sur toute la population au sens du critère de l'inertie.

Dans la deuxième partie on recherche des hiérarchies localement optimales. Un des intérêts de cette 2ème partie est de montrer comment la recherche de hiérarchies locales optimales permet de définir sur la population totale une hiérarchie optimale au sens d'un certain critère.

SUMMARY

The methods of hierarchical classification usually applied produce hierarchies which optimize the "aggregation criterion" at each stage. These methods do not optimize a global cirterion.

The first part of this report presents an algorithm which is intended to establish an optimal hierarchy for the whole population with reference to the "inertia criterion".

The second part of the report is devoted to the search for locally optimal hierarchies. One of its points of interest is to show how the search for locally optimal hierarchies makes it possible to define an optimal hierarchy for the whole population with reference to a givere criterion.

* Ministère de l'Economie

SOMMAIRE

1. INTRODUCTION

Les méthodes de classifications hiérarchiques habituelles (BENZECRI [1], BRUNOOGHE [3], CORMAK [7], JAMBU [11], LERMAN [12], WARD [13], etc...) fournissent des hiérarchies en optimisant à chaque étape le critère d'agrégation. Ces méthodes n'optimisent pas de critère global.

Dans cet article, nous avons adopté une approche différente. Nous cherchons à obtenir des structures hiérarchiques qui soient optimales selon un critère global.

D'autres auteurs ont également posé les problèmes sous cette forme. Signalons les travaux de CAROLL et PRUZANSKI [4] et CHANDON [5] : ils recherchent une ultramétrique la plus proche au sens des moindres carrés de l'indice de distance de départ.

La première partie de ce rapport présente un algorithme qui cherche une hiérarchie optimale au sens du critère de l'inertie, c'est à dire dont la somme des inerties des classes est minimale.

C'est l'un des critères proposés par DIDAY dans [9]. L'intérêt de cette méthode est de fournir une hiérarchie ayant globalement des noeuds de plus faibles inerties que les méthodes habituelles.

La deuxième partie de ce rapport présente un algorithme qui recherche des k-structures hiérachiques optimales, c'est à dire des partitions à k classes dont chaque classe est munie d'une hiérarchie. L'intérêt de cet algorithme réside en la possibilité de bonne description des données de grandes tailles,

On pourra se reporter à [6] pour la présentation d'autres algorithmes et problèmes d'optimisation de hiérarchies.

2. HIERARCHIES LOCALEMENT OPTIMALES

2.1. INTRODUCTION

Les algorithmes de classification automatique hiérarchiques fournissent des hiérarchies de parties d'un ensemble donné. Ces hiérarchies ne sont en général pas optimales au sens d'un critère global. On se propose en utilisant les Méthodes des Nuées Dynamiques (MND) de trouver des hiérarchies optimales selon un critère relatif à l'inertie.

2.2. LE PROBLEME D'OPTIMISATION

Soit $E \subset R^p$ l'ensemble des objets à classer de cardinal égal à n. On suppose E muni d'une distance quadratique d.

En suivant les principales étapes de la MND telles qu'elles ont été décrites dans [10] on définit :

- l'espace de recouvrement S :

  S = $\mathbb{H}$ ensemble des hiérarchies dichotomiques définies sur E.

- l'espace de représentation $\mathbb{L}$:

  $\mathbb{L} = [\mathbb{R}^p]^N$ N = 2n - 1 nombre des paliers d'une hiérarchie dichotomique définie sur E.

- le critère à optimiser :

  $$\underset{\substack{H\in\mathbb{H}\\ L\in\mathbb{L}}}{\text{Min}}\ W(H,L) = \sum_{B\in H}\ \sum_{\ell\in L} I(B,\ell)$$

  avec $I(B,\ell) = \Sigma\{p_x\, d(x,\ell),\ x \in B\}$ où p_x est le poids de l'élément x, ℓ sera le centre de gravité du noeud B de la hiérarchie H.
  On recherche donc parmi toutes les hiérarchies dichotomiques définies sur E celle dont la somme des inerties des paliers est minimale. L'intérêt de ce critère est qu'il exprime, si H est meilleure que H', que globalement la hiérarchie H a des noeuds de plus faibles inerties que H'.

2.3. L'ALGORITHME

Il nécessite la définition de :

- la fonction de représentation g :

  On définit une fonction g de $\mathbb{H}$ dans $\mathbb{L}$

  $g : \mathbb{H} \to \mathbb{L}$

  $H \to g(H) = L = (g_1,\ldots,g_N) \in [\mathbb{R}^p]^N$

  g_i est défini comme le centre de gravité du noeud B_i.

- la fonction d'affectation f :

  f est une application de $\mathbb{L} \times \mathbb{H}$ dans $\mathbb{H}$

  $f : \mathbb{L} \times \mathbb{H} \to \mathbb{H}$

  $(L,H) \xrightarrow{f} H' = f(L,H)$

  la fonction f permet, à partir d'un couple (L,H), c'est à dire d'une hiérarchie et l'ensemble des centres de gravité de ses paliers, de générer une nouvelle hiérarchie $H' \in \mathbb{H}$ de la manière suivante :

soit B un noeud de H, appelons l'ensemble des successeurs de B, S_B :

$S_B = \{B_i \in H \mid B \subset_{\neq} B_i\}$

Soit B' $\in S_B$, on définit $S_{B-B'} = S_B - S_{B'}$ (ensemble des successeurs de B ne contenant pas B').

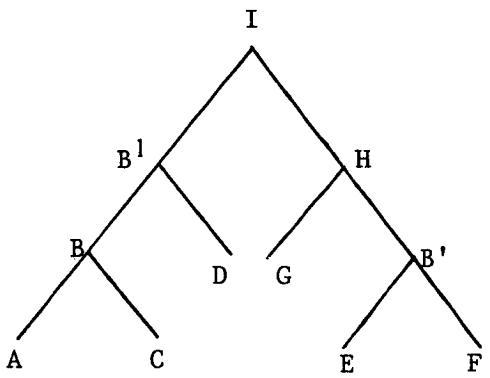
On appelle B^1 le successeur immédiat de B que l'on suppose ne contenant pas B'.

On obtient une nouvelle hiérarchie H' en "affectant" B à B' de la manière suivante :

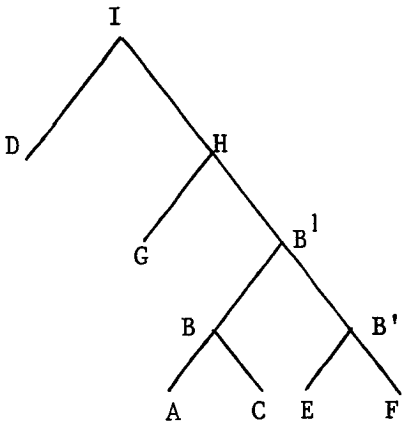
Soient $H = \{B_i \mid i = 1,N\}$

$H' = \{B'_i \mid i = 1,N\}$

* les noeuds n'appartenant pas $S_{B-B'}$ et $S_{B'-B}$ demeurent inchangés
  si $B_i \notin S_{B-B'} \cup S_{B'-B}$ $\quad B'_i = B_i$
* les noeuds de $S_{B-B'}$ et $S_{B'-B}$ sont modifiés de la manière suivante :
  $B^1 \in H \rightarrow B'^1 = B \cup B' \in H'$
  $B_i \in S_{B-B'} - B^1 \rightarrow B'_i = B_i - B \in H'$
  $B_i \in S_{B'-B} \rightarrow B'_i = B_i \cup B \in H'$
* la figure 2.3.1. illustre le passage d'une hiérarchie H à une hiérarchie H' après affectation de B à B'.



Hiérarchie initiale H



Hiérarchie finale H' après affectation de B à B'

FIGURE 2.3.1.

- construction de l'algorithme

L'algorithme consiste à partir d'une hiérarchie initiale H_0 et à optimiser le critère W par itérations successives en L (H fixée) en utilisant la fonction de représentation g puis en H (L fixé) en utilisant la fonction d'affectation f jusqu'à l'obtention d'une solution stable.
Cet algorithme converge vers une solution non nécessairement optimale dépendant de la hiérarchie de départ. Il peut se formaliser à l'aide de deux suites v_n et u_n suivantes :

$v_n = (H^n, L^n) \in \mathbb{H} \times \mathbb{L}$ où $H^n = f(L^{n-1}, H^{n-1})$

$L^n = g(f(L^{n-1}, H^{n-1}))$

$u_n = W(v_n)$

2.4. PROPRIETES DE L'ALGORITHME

Afin de démontrer la convergence de l'algorithme, nous allons établir le théorème suivant :

soient $\delta_1(B, S_{B'-B}) = \sum_{B_i \in S_{B'-B}} \frac{p(B)\, p(B_i)}{p(B_i)+p(B)}\, d^2(g_B, g_{B_i}) + I(B) \text{ Card } S_{B'-B} + I(B \cup B')$

$\delta_2(B, S_{B-B'}) = \sum_{\substack{B_i \in S_{B'-B} \\ B_i \neq B'}} \frac{p(B)\, p(B_i)}{p(B_i)-p(B)}\, d^2(g_B, g_{B_i}) + I(B)(\text{Card } S_{B-B'} - 1) + I(B^1)$

et $\Delta(B,B') = \delta_1(B, S_{B'-B}) - \delta_2(B, S_{B-B'})$

on a alors :

THEOREME 1 : les critères W(H',L') et W(H,L) sont liés par l'expression suivante :

$$\boxed{W(H',L') = W(H,L) + \Delta(B,B')}$$

Démonstration : on a $W(H,L) = \sum_{B_i \in H} I(B_i)$; $W(H',L') = \sum_{B'_i \in H'} I(B'_i)$ et $H' = f(H,L)$.

Décomposons W(H,L) :

$$W(H,L) = \sum_{B_i \notin S_{B-B'} \cup S_{B'-B}} I(B_i) + \sum_{B_i \in S_{B-B'}-B^1} I(B_i) + I(B^1) + \sum_{B_i \in S_{B'-B}} I(B_i)$$

D'après la définition de la fonction d'affectation f :

$$W(H',L') = \sum_{B_i \notin S_{B-B'} \cup S_{B'-B}} I(B_i) + \sum_{B_i \in S_{B-B'}-B^1} I(B_i - B) + I(B \cup B') + \sum_{B_i \in S_{B'-B}} I(B_i \cup B)$$

D'après le théorème de Huygens :

$$I(B_i \cup B) = I(B_i) + I(B) + \frac{p(B)\ p(B_i)}{p(B_i)+p(B)}\ d^2(g_B, g_{B_i})$$

$$I(B_i - B) = I(B_i) - I(B) - \frac{p(B)\ p(B_i)}{p(B_i)-p(B)}\ d^2(g_B, g_{B_i})$$

(si $B \subset B_i$)
on montre alors facilement que :

$$W(H',L') - W(H,L) = \delta_1(B, S_{B'-B}) - \delta_2(B, S_{B-B'})$$

L'expression (2) montre que si l'on choisit d'affecter des noeuds B et B' tels que $\Delta(B,B') < 0$; on a le résultat suivant :

PROPOSITION 1 : la suite u_n converge en décroissant.

Démonstration : on a
$u_n = W(H^n, L^n) < W(H^{n-1}, L^{n-1}) = u_{n-1}$
car $W(H^n, L^n) - W(H^{n-1}, L^{n-1}) = \Delta(B,B') < 0$
La suite u_n est décroissante et minimisée par 0 donc (parce que positive) converge.

PROPOSITION 2 : la suite v_n converge en atteignant sa limite.
On sait que toute suite convergente sur un ensemble fini atteint sa limite or c'est le cas de la suite u_n puisque l'espace est fini.
Supposons que la convergence soit atteinte à l'itération M donc
$u_{M+1} = u_M$ d'où $W(v_{M+1}) = W(v_M)$.
On a alors $v_{M+1} = v_M$ c'est à dire $L^{M+1} = L^M$, $H^{M+1} = H^M$; en effet l'affectation ne se fait que s'il existe B et B' tels que $\Delta(B,B') < 0$ donc à la convergence il n'existe pas de $\Delta(B,B') < 0$ et $H^M = H^{M+1}$.

2.5. GENERALISATION

On généralise facilement l'algorithme

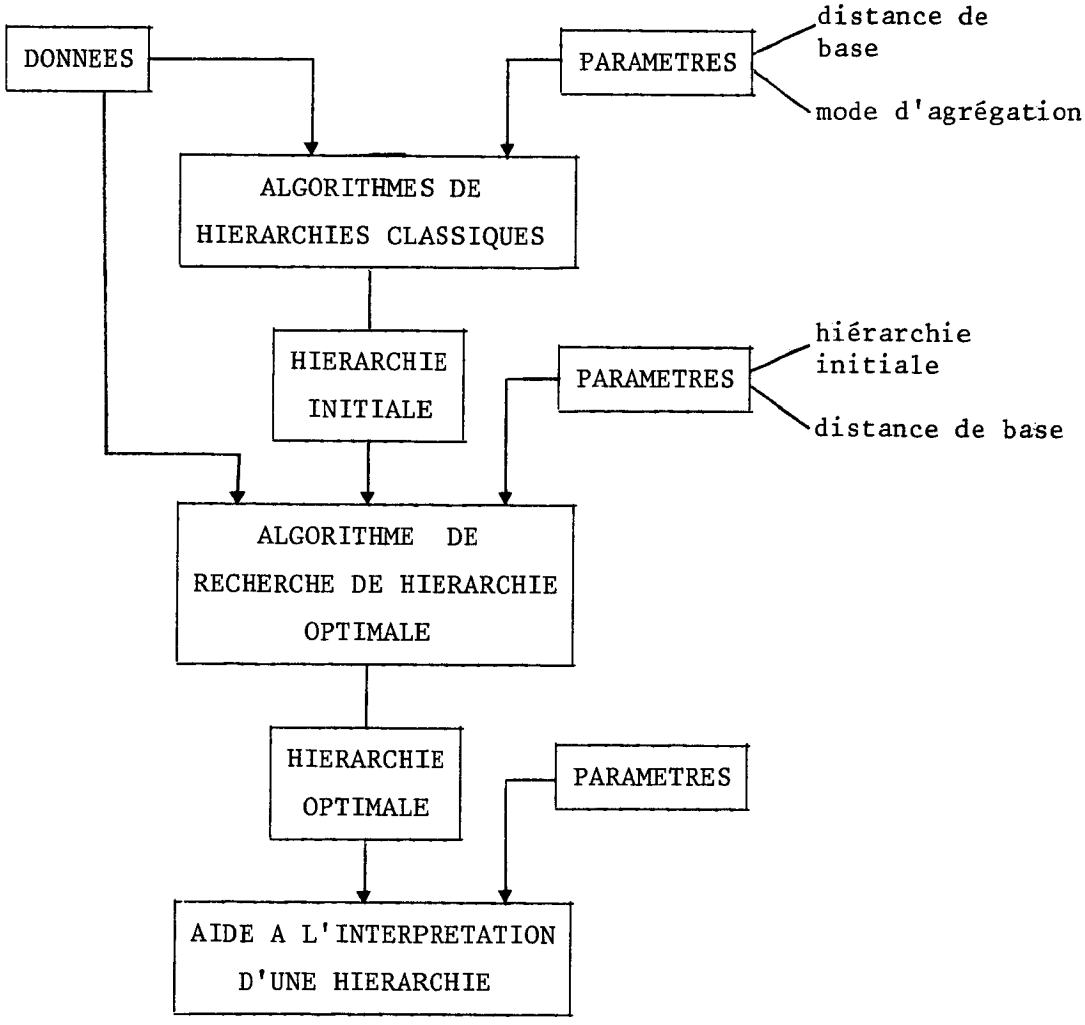
- aux hiérarchies non totales c'est à dire dont les éléments terminaux ne sont pas singletons, cela permet de se limiter à améliorer par exemple, les parties hautes d'une hiérarchie.
- aux k-hiérarchies $\overline{H} = (H_1, H_2, \ldots, H_k)$ où H_k est une hiérarchie totale. Ceci présente un intérêt dans le traitement de grands ensembles de données où l'on désire avoir un ensemble de hiérarchies localement optimales par rapport à un critère global.

2.6. PRESENTATION DU PROGRAMME

Le programme nécessite en entrée les paramètres concernant les données : nombre d'individus, nombre de variables, format de lecture, distance sur les individus, les paramètres concernant la hiérarchie initiale : critère d'agrégation, format de lecture.

Le programme édite la valeur du critère à chaque itération, les descriptions et les arbres correspondants à la hiérarchie initiale et optimale.

La hiérarchie optimale peut être évidemment interprétée par les aides à l'interprétation habituelles [11].



Etapes de traitements pour la recherche d'une hiérarchie optimale

FIGURE 2.6.1.

2.7. EXEMPLES

Nous avons appliqué l'algorithme sur deux tableaux de données mis sous forme disjonctifs complets

1) le tableau des données concernant la charge d'un ordinateur ; nombre d'individus : 40, nombre de variables : 59 (cf. Figures 2.7.1., 2.7.2., 2.7.3.).

2) le tableau de données concernant une enquête sur le nucléaire ; nombre d'individus : 51, nombre de variables : 100. (cf. Figures 2.7.4, 2.7.5)

L'algorithme nécessitant l'initialisation par une hiérarchie de départ plutôt que de partir d'une hiérarchie au hasard, nous sommes partis d'une hiérarchie obtenue par les méthodes usuelles.

Les tableaux 2.7.1. et 2.7.2. recensent les résultats des différents essais desquels ressortent les remarques suivantes :

1) le meilleur critère pour les hiérarchies habituelles est obtenu évidemment par la méthode "minimisation de l'inertie d'une classe". Cette hiérarchie n'a pu être que très légèrement améliorée étant proche d'un optimum local. Toutes les hiérarchies optimales ont un critère meilleur que cette dernière.

2) le tableau 2.7.3. montre une similitude de comportement du critère, les hiérarchies optimales se classant de la même manière dans les deux cas. Il serait intéressant d'étudier de près les raisons d'un tel comportement.

3) la méthode améliore en "général" les inerties des paliers élevés et donc les variances intra-classe des partitions obtenues en coupant la hiérarchie à ces niveaux.

Les deux tableaux de données ne présentent évidemment aucun intérêt statistique étant donné le nombre peu important d'individus et le nombre élévé de variables. Ils nous ont permis cependant de voir le comportement du critère.

Hiérarchies initiales \ Valeur du critère	initial	final	gain	itération
Saut minimum (single linkage)	34,047	13,051	62%	15
Diamètre (complete linkage)	15,442	13,133	20%	5
Maximisation du moment centré d'ordre 2 d'une partition (Ward)	13,284	12,987	3%	6
Minimisation de l'inertie d'une classe	13,182	13,171	0,1%	1

Meilleur critère initial : 13,182
final : 12,987

Données sur la charge d'un ordinateur

40 individus x 59 variables

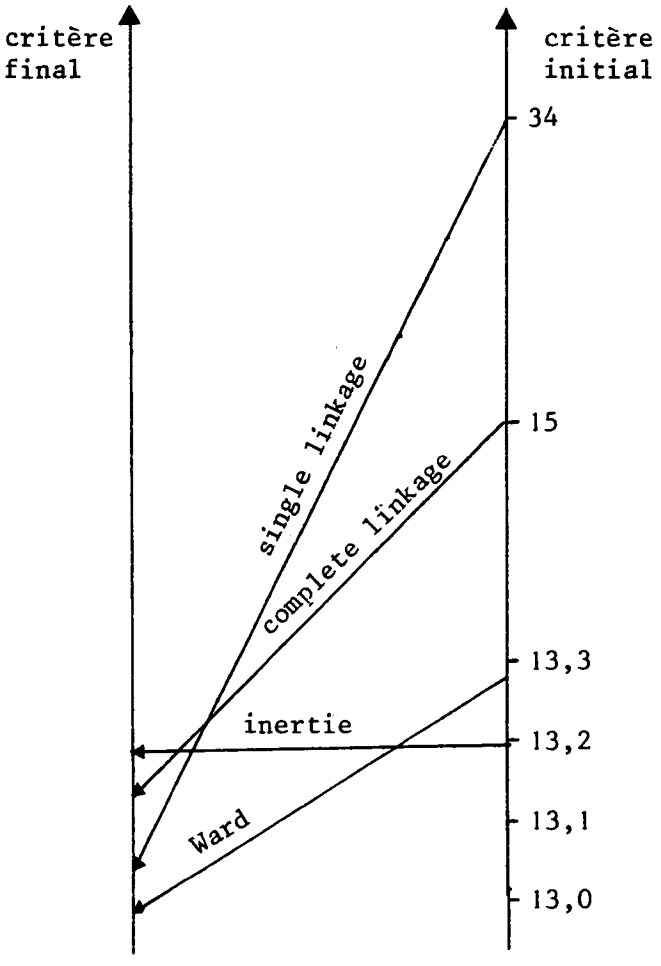
TABLEAU 2.7.1.

Hiérarchies initiales \ Valeur du critère	initial	final	gain	itération
Saut minimum (single linkage)	73,631	17,265	77%	25
Diamètre (complete linkage)	29,064	17,367	40%	12
Maximisation du moment centré d'ordre 2 d'une partition (Ward)	18,626	17,138	8%	9
Minimisation de l'inertie d'une classe	17,535	17,401	0,73%	3

Meilleur critère initial : 17,535
final : 17,138

Données d'enquête : 51 individus x 100 variables

TABLEAU 2.7.2.



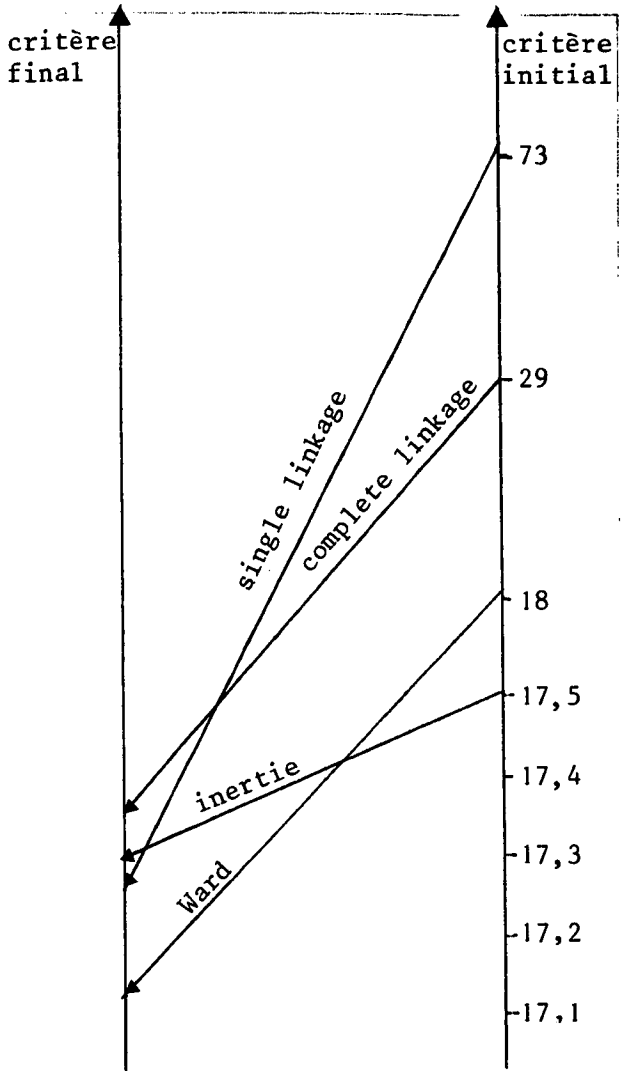


Tableau I : 40 individus
59 variables

Tableau II : 51 individus
100 variables

TABLEAU 2.7.3.

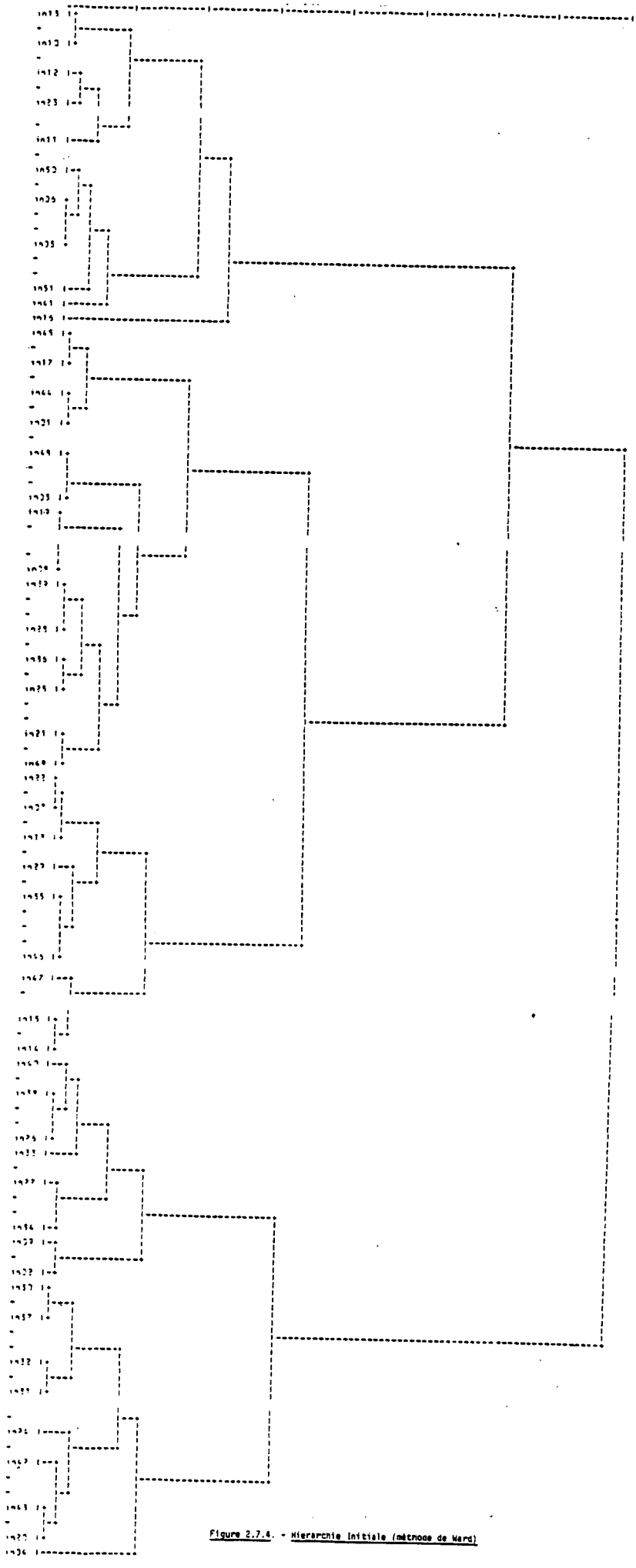


Figure 2.7.4. - Hierarchie initiale (méthode de Ward)

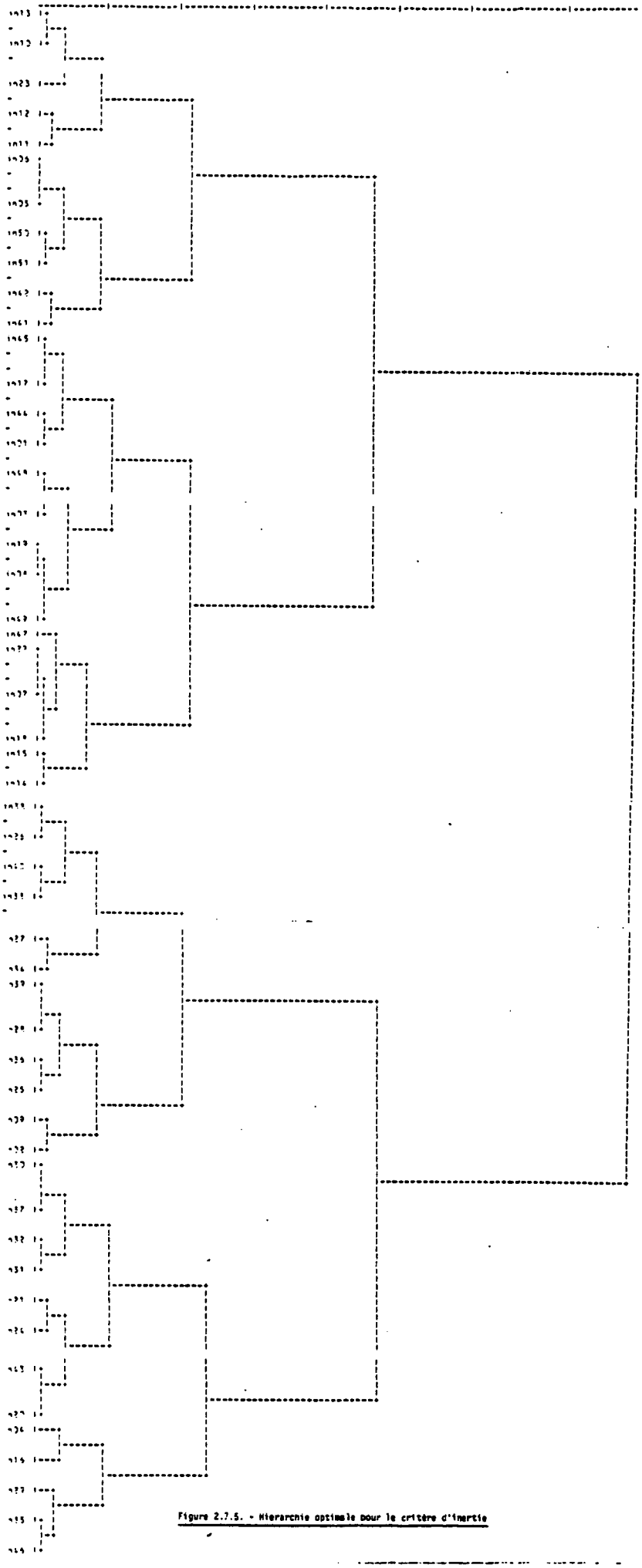


Figure 2.7.5. - Hierarchie optimale pour le critère d'inertie

3. PARTITIONS HIERARCHISEES OPTIMALES

3.1. INTRODUCTION

Nous nous intéressons dans ce paragraphe, au problème de la recherche de partitions hiérarchisées par la donnée sur chacune de leurs classes d'une hiérarchie optimale au sens d'un certain critère de qualité.

Nous définissons un algorithme basé sur la méthode des Nuées Dynamiques visant à obtenir de telles partitions hiérarchisées. L'intérêt de ce paragraphe est de montrer comment la MND peut s'appliquer pour trouver des hiérarchies locales optimales.

3.2. FORMALISATION DU PROBLEME

3.2.1. Définitions

Hiérarchie dichotomique :

On appelle hiérarchie dichotomique 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) $\forall (B,B') \in H^2 \; ; \; B \cap B' = \begin{cases} B \\ \neq \emptyset \\ B' \end{cases}$

iv) $\forall B \in H \; ; \; \text{Card } B > 1 \Longrightarrow \begin{cases} \exists (B_1,B_2) \in H^2 \; ; \; B_1 \subsetneq B \; ; \; B_2 \subsetneq B \\ \text{et } B_1 \cup B_2 = B \end{cases}$

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 $\bar{H} = (H_1,\ldots,H_k)$ de hiérarchies vérifiant les propriétés suivantes :

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

Indice de distance entre parties de E :

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

i) $\forall (B,B') \in (P(E) - \{\emptyset\})^2$; $c(B,B') = c(B',B)$

ii) $\forall x \in E$; $c(\{x\},\{x\}) = 0$

iii) $\forall (x,y) \in E^2$; $x \neq y \Longrightarrow c(\{x\},\{y\}) > 0$

Remarque 1 : Nous supposerons à partir de maintenant que E est muni d'un indice de distance noté c.

Ultramétrique associée à une hiérarchie dichotomique :

Soit H une hiérarchie dichotomique définie sur E ; on définit alors sur E × E une ultramétrique [6] notée ∂_H de manière récurente par :

i) $\forall x \in E$; $\partial_H(x,x) = 0$

ii) $\forall (x,y) \in E^2$

$x \neq y \Longrightarrow \partial_H(x,y) = \sup(\delta(B), c(B,B'), \delta(B'))$ où B et B' sont les 2 éléments de H dont l'intersection est vide, la réunion appartenant à H et tels que $x \in B$ et $y \in B'$

et où $\forall B'' \in \{B,B'\}$; $\delta(B'') = \sup_{(\lambda,\mu)\in B''^2} \partial_H(\lambda,\mu)$

3.2.2. *Formalisation du problème*

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) de points de B et de hiérarchie définie sur B et on note I(B,(x,H)) la quantité :

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

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 :

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

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.

Remarque : B étant fini, il existe toujours au moins 2 éléments représentatifs de B relativement à une hiérarchie H définie sur B.

Critère associé à une partition hiérarchisée :

Soit $P = \{P_1,\ldots,P_k\}$ une partition de E.Soit $H_1,\ldots,H_k$, k hiérarchies définies respectivement sur $P_1,\ldots,P_k$. On appelle critère associé à la partition hiérarchisée $(P,\bar{H})$ (où $\bar{H} = (H_1,\ldots,H_k)$) ou critère associé à $\bar{H}$ et on note $\bar{W}(\bar{H})$ la quantité :

$$\bar{W}(\bar{H}) = \sum_{i=1}^{k} \bar{I}(P_i,H_i) = \sum_{i=1}^{k} I(P_i,(x_{H_i},H_i))$$

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 $W(P,(x^{\bar{H}},\bar{H}))$ entre le support P de $\bar{H} = (H_1,\ldots,H_k)$ et le couple $(x^{\bar{H}},\bar{H})$ où $x^{\bar{H}} = (x_{H_1},\ldots,x_{H_k})$.

La quantité $\bar{W}(\bar{H})$ n'est donc en fait que l'adéquation entre une partition P et un couple $L = (x^{\bar{H}},\bar{H})$ 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.

3.3. APPLICATION DE LA METHODE DES NUEES DYNAMIQUES A LA RECHERCHE D'UNE SOLUTION AU PROBLEME POSE

3.3.1. *L'espace des recouvrements et l'espace des représentations*

On appelle espace des recouvrements et on désigne par $\mathbb{P}_k$ l'ensemble des partitions de E en k classes. On appelle espace des représentations associé à $\mathbb{P}_k$ et on désigne par $\mathbb{L}$ l'ensemble $E^k \times \mathbb{H}$ où $\mathbb{H}$ désigne l'ensemble des k-structures de hiérarchies définies sur E.

3.3.2. *Le critère à optimiser*

Nous avons montré dans [6] comment associer à une k-structure ultramétrique $\bar{\delta} = (\delta_1,\ldots,\delta_k)$ définie sur une partition $P = \{P_1,\ldots,P_k\}$ une ultramétrique $\hat{\delta}$ définie sur E vérifiant :

Pour tout $i \in [1,k]$ la restriction de $\hat{\delta}$ à P_i est égale à δ_i.

Nous supposerons ici connu ce procédé d'association.

Remarque 3 : La démonstration de la convergence de l'algorithme que nous allons définir, c'est à dire en fait la preuve de l'existence de 2 fonctions f et g vérifiant certaines propriétés, est étroitement liée à ce procédé d'affectation.

Ultramétrique associée à une k-structure de hiérarchie :

Soit $\bar{H} = (H_1,\ldots,H_k)$ une k-structure de hiérarchie ; on appellera ultramétrique associée à $\bar{H}$ et on notera $\partial_{\bar{H}}$ l'ultramétrique associée à $(\partial_{H_1},\ldots,\partial_{H_k})$.

Le critère à optimiser :

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

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

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

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 :

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

3.3.3. *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 2 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 celui défini en [10] au chapitre "Ultramétriques Adaptatives".

3.3.3.1. Les fonctions d'affectation

On appelle fonction d'affectation toute application f de $\mathbb{L}$ dans $\mathbb{P}_k$

$$f : \mathbb{L} \to \mathbb{P}_k$$
$$L \to f(L)$$

vérifiant $W(f(L),L) = \underset{P \in \mathbb{P}_k}{\text{Min}} \quad W(P,L)$

L'existence est assurée par le fait que E est un ensemble fini. Il n'y a pas en général unicité.

3.3.3.2. Les fonctions de représentation

Les fonctions de représentation que nous allons définir seront en fait la composée de 2 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

$$g_1 : \mathbb{P}_k \to \mathbb{H}$$
$$P \to g_1(P)$$

vérifiant $\forall\, \bar{x} \in E^k$; $W(P(\bar{x},g_1(P)) = \underset{\bar{H} \in \mathbb{H}_P}{\text{Min}} \quad W(P,(\bar{x},\bar{H}))$

où $\mathbb{H}_P \subset \mathbb{H}$ désigne l'ensemble des k-structures de hiérarchies dont le support est P.

Remarque 5 : L'existence de telles fonctions g_1 n'est pas évidente à priori. Elle dépend de l'indice de distance entre parties défini sur E. Nous indiquerons au paragraphe 3.4.1 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

$$g_2 : \mathbb{H} \to E^k$$
$$\bar{H} \to g_2(\bar{H})$$

vérifiant $\forall\, \bar{H} \in \mathbb{H}$; $W(P,(g_2(\bar{H}),\bar{H})) = \underset{x \in E^k}{\text{Min}} \quad W(P,(\bar{x},\bar{H}))$

où P désigne le support de $\bar{H}$.

L'existence des fonctions g_2 est due au fait que E est un ensemble fini. IL n'y a en général pas unicité.

Les fonctions de représentation :

On appelle fonction de représentation, toute application

$g : \mathbb{P}_k \to \mathbb{L}$

$P \to g(P)$

vérifiant $\forall P \in \mathbb{P}_k$; $g(P) = (g_2 \circ g_1(P), g_1(P))$ où g_1 et g_2 sont deux fonctions du type de celles définies précédemment données à priori.

3.3.3.3. *L'algorithme*

Soit $P^0 \in \mathbb{P}_k$; on définit les 3 suites P^n, L^n et U_n par récurrence de la manière suivante :

$\forall n \in \mathbb{N}$ $L^n = g(P^n)$

$P^{n+1} = f(L^n)$

$U_n = W(P^n, L^n)$

Notations : Désignons pour tout $n \in \mathbb{N}$ par $(\bar{y}^n, \bar{H}^n)$ l'élément L^n et $\{P_1^n, \ldots, P_k^n\}$ la partition P^n.

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

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

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

α) le support de H'^n est P^{n+1}

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

alors

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

4) La suite U_n converge en décroissant en un nombre fini d'itérations.

Démonstration : Soit $n \in \mathbb{N}^*$. 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 $u_n = W(P^n, L^n) \geq W(P^{n+1}, L^n)$ ceci par définition de f

Or d'après 2), il existe $H'^n \in \mathbb{H}$ 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 de 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'itérations est alors triviale. 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 $\mathbb{R}$.

3.3.3.4. *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 valide ces conditions 1) et 2).

Remarques sur l'algorithme défini au chapitre "Ultramétriques Adaptatives" de [10]

Une différence essentielle entre l'algorithme défini en [10] au chapitre "Ultramétriques Adaptatives" et celui pré-cité 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 amené 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'ultramétriques (hiérarchie) relativement à P". D'un point de vue pratique, ceci nous conduit à définir sur la population totale, une ultramétrique 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).

3.4. LE PROGRAMME ETABLI

3.4.1. Les résultats obtenus

Nous supposons E muni de l'indice de distance ∂_{min} défini par :

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

A partir des résultats démontrés dans [6], il découle :

* 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 2.
* que si $\bar{H}$ est une k-structure de hiérarchie sur E optimale alors l'ultramétrique $\partial_{\bar{H}}$ associée à $\bar{H}$ n'est autre que l'ultramétrique sous dominante maximale définie sur E.
* que si H est une hiérarchie définie sur E optimale alors la k-structure de hiérarchie obtenue en otant 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.

3.4.2. Quelques remarques relatives au programme et à l'algorithme

3.4.2.1. Remarques relatives au programme

Le 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.

3.4.2.2. 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 α ($\alpha \in \mathbb{R}_+^*$) (par exemple, algorithme de PRIM)
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 = 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 les hypothèses, notre algorithme permet de munir un ensemble de données en un temps proportionnel à $m^{\alpha/2}(1 + m^{1/2})$ en se servant d'un algorithme de classification automatique hiérarchique qui, utilisé seul,nécessiterait un temps machine proportionnel à m^{α}.

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.

4. CONCLUSION

L'approche qui a été présentée est intéressante du point de vue théorique car elle pose les problèmes de recherche de structures hiérarchiques en termes de problèmes d'optimisation. Le prolongement naturel serait la recherche d'autres critères : par exemple, critère exprimant l'adéquation entre indice de distance de base et l'ultramétrique induite par la hiérarchie et les algorithmes correspondants. Dans [6] sont présentés l'étude d'autres critères et algorithmes.

Du point de vue pratique, les algorithmes proposés améliorent les hiérarchies et ont permis de mettre en évidence des structures locales optimales facilement interprétables. Les programmes écrits sont cependant encore coûteux en temps machine et l'application à de grands jeux de données sont en cours.

5. BIBLIOGRAPHIE

[1] BENZECRI : L'Analyse des Données.
Dunod. 1973.

[2] BERGE : Graphe et Hypergraphe.
Dunod.

[3] BRUNOOGHE : Classification Automatique d'un grand tableau de données par la méthode des graphes réductibles.
Colloque IRIA. Septembre 1977.

[4] CAROLL et 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

[5] CHANDON : Construction de l'ultramétrique la plus proche au sens des moindres carrés.
Note interne IAE. Mai 1978.

[6] CHIFFLET : Structures hiérarchique locales optimales.
Thèse de 3ème cycle. Paris. 1979.

[7] COMARCK : A review of classification.
J.R. Statistic SOL. 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 11th European Congress of Statistics. Oslo. 1978.

[10] DIDAY et collaboration : 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] WARD : Hierarchical grouping to optimize an objective function.
JASA. March 1963.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique