



Application de règles de gestion à un système de fabrication : classification des objectifs atteints en vue de leur utilisation

Fabrice Bonneau, Jean-Marie Proth

► To cite this version:

Fabrice Bonneau, Jean-Marie Proth. Application de règles de gestion à un système de fabrication : classification des objectifs atteints en vue de leur utilisation. [Rapport de recherche] RR-0372, INRIA. 1985, pp.45. inria-00076184

HAL Id: inria-00076184

<https://inria.hal.science/inria-00076184>

Submitted on 24 May 2006

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

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



CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105

78153 Le Chesnay Cedex
France

Tel (3) 954 90 20

Rapports de Recherche

N° 372

APPLICATION DE RÈGLES DE GESTION À UN SYSTÈME DE FABRICATION : CLASSIFICATION DES OBJECTIFS ATTEINTS EN VUE DE LEUR UTILISATION

**Fabrice BONNEAU
Jean Marie PROTH**

Mars 1985

APPLICATION DE REGLES DE GESTION
A UN SYSTEME DE FABRICATION :
CLASSIFICATION DES OBJECTIFS ATTEINTS
EN VUE DE LEUR UTILISATION

Fabrice BONNEAU
Jean Marie PROTH

PROJET SAGEP

ABSTRACT

In the following, we propose two methods usable to classify the results obtained using various rules to manage a production system.

The main purpose of this work is to obtain classes of results which can be qualified and easily represented.

RESUME

Dans la suite, nous proposons deux méthodes utilisables pour classer les résultats obtenus en utilisant différentes règles pour gérer un système de production.

L'objectif essentiel de ce travail est d'obtenir des classes de résultats qu'il est possible de qualifier et de représenter simplement.

Notre but est de classer les objectifs atteints en appliquant à un système de fabrication des règles de gestion variées sur une période donnée, connaissant son état initial. Nous nous intéressons à une suite de critères (ou indicateurs) qui caractérisent l'évolution du système tout au long de la période d'étude et son état en fin de période. Nous appelons objectif les valeurs prises par cet ensemble d'indicateurs pour une expérience donnée, c'est à dire pour un état initial et des règles de gestion donnés.

Nous recherchons des classes d'objectifs qui puissent être décrites en termes significatifs pour les utilisateurs. Cette exigence nous a conduit à proposer des classes homogènes pour tout ou partie des critères qui interviennent. Cette démarche nous a permis de trouver des représentations graphiques des classes obtenues.

I. CLASSES HOMOGENES

I.1. Généralités

Un objectif de production est une suite de valeurs numériques prises par des indicateurs. Nous dirons, pour utiliser un vocabulaire habituel en analyse des données, qu'il s'agit d'une observation dans un espace de dimension p , où p est le nombre d'indicateurs.

Partant d'un ensemble d'observations, on cherche habituellement une partition de cet ensemble telle que tous les éléments d'un même sous-ensemble de la partition soient "proches" au sens d'une distance ou d'un indice de similarité donné.

Chaque sous-ensemble de la partition est appelé classe. Notre approche est sensiblement différente. Nous cherchons à regrouper dans une même classe des observations telles que les valeurs des indicateurs correspondants appartiennent aux mêmes intervalles, que nous cherchons à qualifier.

Soient par exemple deux indicateurs qui sont le taux de fonctionnement d'une machine et le stock moyen des produits finis. Supposons que le stock moyen des produits finis relevé dans l'ensemble des observations varie entre 0 et 100. Nous pourrions décider de retenir les échelles suivantes :

pour le taux de fonctionnement :

1. de 0 à 30% du temps total. Nous qualifions ces valeurs de "faibles".
2. de 30% à 70% du temps total. Nous qualifions ces valeurs de "moyennes".
3. de 70% à 100% du temps total. Nous qualifions ces valeurs d'"élevées".

pour le stock moyen :

1. de 1 à 25 : "faible".
2. de 25 à 50 : "moyen faible".
3. de 50 à 75 : "moyen élevé".
4. de 75 à 100 : "important".

Dans cet exemple, $3 \times 4 = 12$ classes sont potentiellement possibles.

Chacune de ces classes peut être caractérisée par le couple de qualificatifs qui s'attache aux indicateurs. On parlera par exemple de la classe des observations à "taux de fonctionnement faible et stock moyen élevé". On pourra également utiliser des qualificatifs qui sont la synthèse des qualificatifs attachés aux classes.

I.2. Pratique de la classification

L'exemple précédent suggère que la classification est simple.

Supposons qu'une observation utilise p indicateurs qui prennent des valeurs quantitatives :

$$\mathcal{E} = \{E_1, E_2, \dots, E_p\} \quad (1)$$

Supposons encore que, pour $i=1,2,\dots,p$:

M_i désigne la valeur maximale prise par E_i au cours des observations.

m_i désigne la valeur minimale prise par E_i au cours des observations.

e_i est le nombre d'intervalles choisis sur $[m_i, M_i]$. On appelle e_i "finesse d'observation de E_i ".

Pour $i=1,2,\dots,p$, on retient pour l'indicateur E_i les intervalles suivants :

$$I_{k_i}^i = [m_i + k(M_i - m_i)/e_i, m_i + (k+1)(M_i - m_i)/e_i], \quad (2)$$

$$k=0,1,\dots,e_i-1$$

Si, pour une observation $\mathcal{E}^*$, l'indicateur E_i prend une valeur située dans $I_{k_i}^i$ ($i=1,2,\dots,p$), alors $\mathcal{E}^*$ sera affecté à la classe homogène

$\prod_{i=1}^p I_{k_i}^i$, que l'on désignera encore, pour simplifier, par $(k_1, k_2, \dots, k_p)$.

Ce p-uple sera appelé représentant de la classe.

Dans l'approche précédente, nous prendrons soin d'écarter les observations pour lesquelles les valeurs prises par un ou plusieurs paramètres sont jugées accidentelles.

Avec les notations que nous venons d'introduire, le nombre de classes homogènes potentiellement possibles est :

$$N = \prod_{i=1}^p e_i \quad (3)$$

N augmente très vite avec e_i et p , et l'usage que nous voulons faire de ces classes nous interdit d'en utiliser un nombre trop important.

Cette contradiction est facile à surmonter. En effet :

1. les indicateurs ne sont pas indépendants. Certaines classes homogènes ne contiennent, de ce fait, aucune observation : cela permet de les éliminer.

2. on cherche à regrouper les classes homogènes contenant trop peu d'observations avec des classes homogènes plus importantes, et ceci de deux manières :

a. ou bien on élimine un indicateur jugé responsable de la présence des classes homogènes de faible cardinal et, de ce fait, considéré comme d'importance secondaire,

b. ou bien on rattache chaque classe homogène de faible cardinal à une classe homogène importante proche, au sens que nous préciserons plus loin.

3. les classes homogènes de faible cardinal et qu'il n'est pas possible de rattacher à une classe plus importante seront éliminées. Nous acceptons donc de perdre ce type d'information.

Nous préciserons ces démarches dans le paragraphe suivant.

II. TRAITEMENT DES CLASSES HOMOGÈNES

Nous proposons deux méthodes qui ont pour but de diminuer le nombre de classes en les regroupant. Nous appelons noyaux des classes les classes homogènes sur lesquelles ont été effectués les regroupements.

Nous utiliserons les grandeurs caractéristiques suivantes :

1. Nombre final de classes.
2. Pourcentage d'éléments éliminés.
3. Coefficient de précision. C'est le rapport de la somme des cardinaux des noyaux sur le total des observations retenues.

II.1. Conventions d'écriture, définitions et notations

Soient :

$$\text{et } \left. \begin{array}{l} K^P = \{k_1, k_2, \dots, k_p\} \\ L^P = \{l_1, l_2, \dots, l_p\} \end{array} \right\} k_i, l_i \in \{1, 2, \dots, e_i\} \text{ pour } i=1, 2, \dots, p$$

deux classes de l'ensemble U^P des classes homogènes à p indicateurs.

On considère une application de $K^P \times L^P$ dans

$$\{0, 1, \dots, \max_{i \in \{1, \dots, p\}} e_i\} \times \{1, 2, \dots, p\}$$

définie de la manière suivante :

$$d(K^P, L^P) = (d_1, d_2)$$

$$\text{avec : } \begin{cases} d_1 = \max_{r \in \{1, \dots, p\}} |l_r - k_r| \\ d_2 = \text{card}\{r / r \in \{1, 2, \dots, p\} \text{ et } |l_r - k_r| \neq 0\} \end{cases} \quad (4)$$

d_1 est le nombre maximum d'intervalles séparant les intervalles occupés par les valeurs prises par les indicateurs, et d_2 est la nombre d'indicateurs dont les valeurs dans K^P et L^P occupent des classes différentes. d est appelé indicateur de dissemblance entre classe homogènes.

Nous décrivons un regroupement élémentaire entre classes de la manière suivante :

$$\{k / (d_1, d_2) / \mathcal{Q}_1\} \quad (5)$$

où k est un entier, $\mathcal{Q}_1$ un ordre sur l'ensemble des classes d'au moins k éléments.

Cette formulation indique qu'il y a regroupement des classes dont le cardinal est strictement inférieur à k (classe absorbables) sur les classes dont le cardinal est supérieur ou égal à k (noyaux) si l'indicateur de dissemblance entre le noyau et la classe absorbable est (q_1, q_2) avec $q_1 \leq d_1$ et $q_2 \leq d_2$. Les noyaux sont pris dans l'ordre $\mathcal{Q}_1$.

Dans la suite, nous supposons connus deux entiers :

1. s , qui est le seuil à partir duquel une classe est considérée comme importante.
2. ts , appelé seuil de tolérance ($ts \leq s$). Toute classe non regroupée et de seuil inférieur à ts est éliminée.

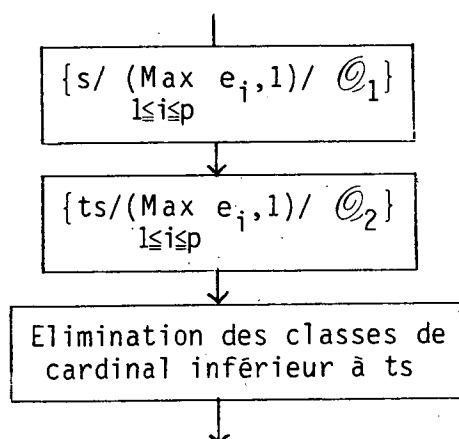
Nous détaillons maintenant les deux méthodes de regroupement esquissées plus haut.

II.2. Méthode de regroupement par élimination d'un indicateur

L'algorithme procède en trois étapes (s et ts viennent d'être définis) :

1. regroupement élémentaires des classes de cardinal inférieur à s sur les classes de cardinal supérieur à s dont les valeurs prises par les indicateurs occupent les mêmes intervalles à l'exception d'un seul, et pour ce dernier l'écart peut-être quelconque.
2. même opération en remplaçant s par ts .
3. élimination des classes non regroupées et de cardinal inférieur à ts .

Avec les notations présentées plus haut, cet algorithme se résume comme suit :



L'ordre $@_1$ est donné par l'utilisateur. Il peut être l'ordre croissant des cardinaux des classes. Les regroupements finalement obtenus dépendent de ces ordres. Il n'existe cependant aucun critère d'optimalité sur les regroupements, donc sur les ordres.

L'intérêt principal de cette méthode est que nous sommes assurés que les classes contiennent des observations dont tous les indicateurs, sauf peut-être l'un d'entre eux, sont contenus dans le même intervalle. Comme nous le verrons plus loin, cette approche permettra une représentation aisée des résultats. Seul le noyau sera qualifié.

Le programme que nous avons mis au point pour appliquer cette méthode donne pour chaque noyau :

1. la liste des observations qui le composent.
2. pour chaque indicateur, la liste des observations qui sont rattachés au noyau lorsque l'indicateur est écarté.

Par exemple, sur 250 observations définies chacune par trois indicateurs, nous avons obtenu, pour l'un des regroupements :

<u>Classe n° 4</u>	<u>Cardinal 15</u>
--------------------	--------------------

Noyau : 3,8,12,90,101,102,181,200,241.

Elimination de l'indicateur 1 : 1,5.

Elimination de l'indicateur 2 : /.

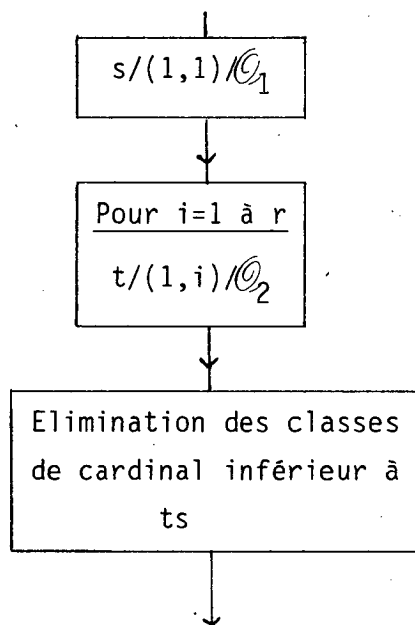
Elimination de l'indicateur 3 : 18,54,74,75.

II.3. Méthode de regroupement au plus proche voisin

Dans cette méthode, on décide de regrouper deux classes si les deux composantes de leur indicateur de dissemblance sont faibles. Cela signifie que nous regroupons entre elles des classes de type "noyau" et des classes absorbables dont peu d'indicateurs diffèrent et telles que les indicateurs qui diffèrent s'écartent les uns des autres d'un nombre également faible d'intervalles.

En d'autres termes, on regroupe des classes dont tous les indicateurs sont proches ou égaux. Il est donc possible de regrouper des classes dont plus d'un indicateurs ont leurs valeurs situées dans des intervalles différents, mais proches.

En utilisant les notations introduites précédemment, l'algorithme que nous retenons est le suivant (s et t_s ont été introduits en II.1.):



r est le nombre d'indicateurs dont on admet qu'ils puissent prendre leurs valeurs dans des intervalles différents.

L'algorithme précédent montre que l'on regroupe d'abord les classes de cardinal inférieur à s et les classes de cardinal supérieur ou égal à s à condition qu'un indicateur au plus prenne ses valeurs dans des classes différentes, et à condition que ces classes soient adjacentes.

Les ordres sont données par l'utilisateur.

II.4. Comparaison des méthodes

Il n'est pas possible, à priori, de dire si le pourcentage d'éliminés ou le coefficient de précision seront plus faibles ou plus importants en utilisant la 1^{ère} ou la seconde méthode. Cependant il est clair que d'une manière générale le pourcentage d'éliminés et le coefficient de précision sont des fonctions respectivement croissantes et décroissantes :

- du nombre p d'indicateurs pour la méthode 1.
- des finesses d'observations e_i pour la méthode 2.

De plus on peut affirmer que le nombre définitif de classes obtenue par la méthode 1 est inférieur ou égal à celui obtenu par la méthode 2.

Preuve :

On peut considérer en tout point du développement de l'algorithme que l'ensemble des classes est divisé en 3 groupes :

- $G_1 = \{b / \text{Card } b \geq s\}$
- $G_2 = \{b / ts \leq \text{Card } b < s\}$
- $G_3 = \{b / \text{Card } b < ts\}$

A la 1^{ère} étape

On a un regroupement du type :

$\{s / (\text{Max } e_i, 1) / \mathcal{C}\}$ pour la méthode 1

$\{s / (1, 1) / \mathcal{C}\}$ pour la méthode 2

et donc un nombre plus élevé de classes est regroupé si la méthode 1 est appliquée.

On a ainsi à la fin de la 1^{ère} étape :

$$\text{Card } [G_2(\text{méth } 1)] \leq \text{Card}[G_2(\text{méth } 2)]$$

A la 2^{ème} étape

G_1 et G_2 restent fixes puisque les regroupements s'effectuent avec un seuil de ts.

A la fin de l'algorithme

On a :

$$\text{Card}(G_1 \cup G_2) \text{ (méth 1)} \leq \text{Card}(G_1 \cup G_2) \text{ (méth 2)}$$

or le cardinal de $G_1 \cup G_2$ n'est autre que le nombre définitif de grandes classes.

Dans les deux cas, on qualifie les regroupements en qualifiant les noyaux autour desquels s'effectuent les regroupements.

Par contre, l'interprétation des qualificatifs est très différente suivant que l'on aura utilisé la première ou la seconde méthode.

Dans le premier cas, il faudra admettre qu'une des composantes de la qualification puisse être totalement inadéquate pour certains éléments du regroupement.

Dans le second cas,, une ou plusieurs de composantes de la qualification peuvent être vraies avec une certaine marge d'erreur pour certains éléments du regroupement.

III. REPRESENTATION GRAPHIQUE DES REGROUPEMENTS

Le travail que nous exposons ici a pour objectif une évaluation en "temps réel" de l'évolution dans le temps d'un système de fabrication. Aussi avons-nous fait un effort particulier pour représenter les résultats ci-dessus sous une forme aussi compacte que possible.

La représentation du noyau est indépendante de la méthode utilisée, contrairement à la représentation des éléments regroupés.

III.1. Représentation du noyau

Considérons le cas où 4 indicateurs sont pris en compte.

Le noyau est représenté par un disque de rayon proportionnel au rapport du nombre d'éléments du noyau sur le nombre total d'éléments de la classe (c'est à dire du noyau et des classes homogènes absorbées par ce noyau). Ce disque est en outre divisé en un nombre de secteurs angulaires égal au nombre d'indicateurs qui interviennent dans les observations. Chaque secteur est lié à un indicateur. La couleur du secteur est fonction de l'intervalle occupé par la valeur prise par l'indicateur. Elle est d'autant plus foncée que l'intervalle correspond à des valeurs élevées.

La figure 1 représente cette situation dans le cas de quatre indicateurs.

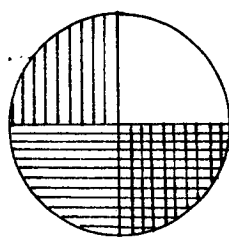


Fig. 1

III.2. Représentation des regroupements

Elle dépend, comme nous l'avons dit plus haut, de la méthode de regroupement utilisée.

III.2.1. Méthode de regroupement par élimination d'un indicateur

Sur chaque secteur circulaire représentant un indicateur E_i ($i=1,2,\dots,p$) on applique un secteur angulaire qui représente les éléments rattachés au noyau. L'angle de l'ouverture de ce secteur est d'autant plus important que le nombre d'éléments rattachés est grand. La figure 2 donne un exemple de cette représentation. Rappelons que

les éléments rattachés ont toutes les caractéristiques des éléments du noyau, sauf pour ce qui concerne l'indicateur de rattachement.

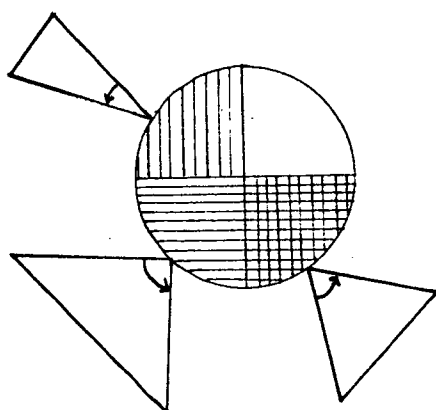


Fig. 2

III.2.2. Méthode de regroupement au plus proche voisin

Cette méthode conduit à un regroupement en couronnes. La couronne la plus proche du noyau regroupe les éléments dont un seul indicateur prend sa valeur dans un intervalle différent de celui qui est occupé par les valeurs prises par ce même indicateur pour les éléments du noyau. De manière générale, la r^{me} couronne (comptée à partir du noyau) regroupe les éléments dont r indicateurs prennent leur valeur dans un intervalle différent de celui qui est occupé par les valeurs prises par ces mêmes indicateurs pour les éléments du noyau. Rappelons que les intervalles utilisés par les observations regroupées sont adjacents aux intervalles caractérisant le noyau. Les observations des couronnes restent donc proches de celles du noyau.

La figure 3 donne une telle représentation. L'importance des couronnes dépend du nombre d'éléments qu'elles contiennent.

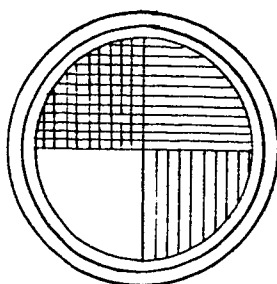


Fig. 3

IV. EXEMPLE D'APPLICATION

IV.1. Présentation du système

On considère 10 machines capables de produire 10 types de produits différents. Il s'agit d'un atelier de type "job-shop", c'est à dire dans lequel les machines ne sont pas utilisées impérativement dans un ordre déterminé. La demande s'exprime en pourcentage de produits de chaque type à fabriquer.

Les gammes de fabrication sont connues : nous connaissons l'ordre de passage d'un produit sur les machines, ainsi que les temps de passage. Les machines sont supposées ne pas tomber en panne. Les pièces sont transportées par chariots (une pièce par chariot).

On applique à ce système des règles de gestion :

règle 1 : Priorité au produit ayant la gamme la plus courte.

règle 2 : Priorité au produit ayant la gamme la plus longue

règle 3 : Priorité au produit auquel il reste le plus petit temps de transformation.

règle 4 : Priorité au produit ayant séjourné le plus longtemps dans le système.

La période d'application de la règle choisie est une constante.

On observe les indicateurs suivants :

1. quantité totale de pièces fabriquées.
2. nombre de chariots (maxi) utilisés.
3. moyenne des écarts (en valeur absolue) entre les quotas obtenus et les quotas demandés. Les écarts sont relevés à chaque fin de fabrication d'une pièce.

IV.2. Présentation des résultats

Nous avons effectué 293 simulations du système de production que nous venons de décrire. Nous avons, pour chaque simulation, généré au hasard les quotas à réaliser et appliqué successivement au système, durant la période connue, les quatre règles décrites plus haut en réparant chaque fois d'un système au repos.

Nous avons ainsi recueilli 4x293 triplets de valeurs pour les indicateurs.

Nous choisissons un seuil de 9 et un seuil de tolérance de 6. Les ordres sont ceux de l'enregistrement dans les fichiers.

Les grandeurs caractéristiques pour les 4 règles sont données par le tableau suivant :

Méthodes	Règles	Pourcentage d'éliminés	Coefficient de précision	Nombre de classes	Représentants
Elimination d'un indicateur	1	3.75	0.76	11	(3,3,3)(1,4,3)(2,4,3) (2,3,3)(1,4,4)(2,3,4) (3,4,3)(1,3,4)(2,2,3) (2,4,4)(1,3,3)
	2	0.34	0.73	13	(2,1,3)(1,4,3)(2,4,2) (3,1,3)(2,2,3)(1,4,4) (2,3,3)(1,3,4)(2,4,3) (1,3,3)(3,2,3)(2,3,2)(4,1,2)
	3	5.12	0.79	14	(4,3,3)(4,4,2)(3,4,2)(4,1,2) (3,2,2)(4,2,2)(3,2,3)(4,4,3) (4,2,3)(4,3,2)(4,1,3)(3,1,3) (3,3,3)(3,3,2)
	4	3.07	0.79	8	(3,3,1)(2,3,2)(1,4,2) (3,2,1)(2,4,1)(2,3,1) (2,4,2)(3,4,1)
Plus proche voisin	1	3.75	0.76	11	(3,3,3)(1,4,3)(2,4,3) (2,3,3)(1,4,4)(2,3,4) (3,4,3)(1,3,4)(2,2,3) (2,4,4)(1,3,3)
	2	0.68	0.73	13	(2,1,3)(1,4,3)(2,4,2) (3,1,3)(2,2,3)(1,4,4) (2,3,3)(1,3,4)(2,4,3) (1,3,3)(3,2,3)(2,3,2)(4,1,2)
	3	4.78	0.81	15	(4,3,3)(4,4,2)(3,4,2)(4,1,2) (3,2,2)(4,2,2)(3,2,3)(4,4,3) (4,2,3)(4,3,2)(4,1,3)(3,1,3) (3,3,3)(3,3,2)(2,1,2)
	4	1.02	0.77	8	(3,3,1)(2,3,2)(1,4,2) (3,2,1)(2,4,1)(2,3,1) (2,4,2)(3,4,1)

Le programme fournit également les résultats détaillés des regroupements. On obtient par exemple, pour la règle 3, les résultats suivants :

a) Lorsque la méthode "Elimination d'un indicateur" est appliquée

Les résultats et les représentations graphiques sont données ci-dessous. Le travail a encore été effectué à un seuil de 9 et un seuil de tolérance de 6.

Nous présentons d'abord l'ensemble des résultats obtenus pour la règle 3. On trouve d'abord les résultats généraux, puis les résultats détaillés pour chacune des classes obtenues.

La figure 1 nous donne dans le détail la représentation de la classe numéro 3. La figure 2 nous donne la représentation des 14 classes obtenues à l'aide de la règle numéro 3. Certaines informations ont été omises pour plus de clarté.

CLASSIFICATION DE LA MEMOIRE ASSOCIEE A LA REGLE 3

Le nombre de classes homogenes de la memoire associee a 3 est:
42

Seuil: 9 Tolerance: 6

METHODE DE REGROUPEMENT PAR RELAXATION D'UN INDICATEUR

Pourcentage d'elimines:
5.12%

Coefficient de precision:
0.7878

Le nombre definitif de classes est:
14

Liste des representants:

4 3 3 , 4 4 2 , 3 4 2 , 4 1 2 , 3 2 2 , 4 2 2 , 3 2 3 , 4 4 3 , 4 2 3 , 4 3 2 ,
4 1 3 , 3 1 3 , 3 3 3 , 3 3 2 ,

COMPOSITION DEFINITIVE DES CLASSES

CLASSE NC 1: REPRESENTANT: 4 3 3 CARDINAL: 17

Noyau

2 9 22 23 24 77 142 152 172 220 250 271

Branches ou l'indicateur 1 est relaxe
4 21 71 94 180

Branches ou l'indicateur 2 est relaxe

Branches ou l'indicateur 3 est relaxe

CLASSE NC 2: REPRESENTANT: 4 4 2 CARDINAL: 18

Noyau

3 13 34 37 38 68 98 103 109 115 151 170 208 216 269 274

Branches ou l'indicateur 1 est relaxe

Branches ou l'indicateur 2 est relaxe

Branches ou l'indicateur 3 est relaxe
31 282

CLASSE NO 3:

REPRESENTANT: 3 4 2

CARDINAL: 23

Noyau

5 59 69 81 87 110 117 141 148 178 204 218

Eranche ou l'indicateur 1 est relaxe

Eranche ou l'indicateur 2 est relaxe

15 47 89 144 174 187 201 232

Eranche ou l'indicateur 3 est relaxe

106 191 293

CLASSE NO 4:

REPRESENTANT: 4 1 2

CARDINAL: 29

Noyau

6 19 29 62 67 88 92 100 104 122 145 153 167 171 200 217 224 240 277 279

Eranche ou l'indicateur 1 est relaxe

1 17 20 50 133 225 238 8 111

Eranche ou l'indicateur 2 est relaxe

Eranche ou l'indicateur 3 est relaxe

CLASSE NO 5:

REPRESENTANT: 3 2 2

CARDINAL: 24

Noyau

7 11 53 54 56 65 116 123 140 147 149 166 179 181 219 226 245 266 280

Eranche ou l'indicateur 1 est relaxe

40 61 156 259 195

Eranche ou l'indicateur 2 est relaxe

Eranche ou l'indicateur 3 est relaxe

CLASSE NO 6:

REPRESENTANT: 4 2 2

CARDINAL: 23

Noyau

12 27 44 49 91 95 97 107 113 118 126 136 168 169 173 223 230 231 242 265
268 286 288

Eranche ou l'indicateur 1 est relaxe

Eranche ou l'indicateur 2 est relaxe

Eranche ou l'indicateur 3 est relaxe

CLASSE NC 7: REPRESENTANT: 3 2 3 CARDINAL: 21

Noyau

16 28 52 58 70 72 93 102 112 252 261 273 275 283

Branché ou l'indicateur 1 est relaxé

30 32 212 237 254 249 284

Branché ou l'indicateur 2 est relaxé

Branché ou l'indicateur 3 est relaxé

CLASSE NC 8: REPRESENTANT: 4 4 3 CARDINAL: 12

Noyau

18 128 154 183 234 244 251 262 287 290

Branché ou l'indicateur 1 est relaxé

211 239

Branché ou l'indicateur 2 est relaxé

Branché ou l'indicateur 3 est relaxé

CLASSE NC 9: REPRESENTANT: 4 2 3 CARDINAL: 22

Noyau

25 36 45 51 83 90 96 105 125 130 137 143 150 163 164 193 197 215 227 235
267 278

Branché ou l'indicateur 1 est relaxé

Branché ou l'indicateur 2 est relaxé

Branché ou l'indicateur 3 est relaxé

CLASSE NC 10: REPRESENTANT: 4 3 2 CARDINAL: 24

Noyau

26 79 120 124 131 138 139 157 159 165 186 228 241 263 272 281

Branché ou l'indicateur 1 est relaxé

14 33 66 135 177 210 246 292

Branché ou l'indicateur 2 est relaxé

Branché ou l'indicateur 3 est relaxé

CLASSE NC 11: REPRESENTANT: 4 1 3 CARDINAL: 20

Noyau

35 48 84 121 158 161 188 213 233 243 247 248 276

Branche ou l'indicateur 1 est relaxe

10 184 192 291 85 134 253

Branche ou l'indicateur 2 est relaxe

Branche ou l'indicateur 3 est relaxe

CLASSE NC 12: REPRESENTANT: 3 1 3 CARDINAL: 16

Noyau

41 55 63 64 78 99 176 196 203 205 209 236 255 256 258

Branche ou l'indicateur 1 est relaxe

Branche ou l'indicateur 2 est relaxe

Branche ou l'indicateur 3 est relaxe

86

CLASSE NC 13: REPRESENTANT: 3 3 3 CARDINAL: 16

Noyau

42 57 73 80 101 162 199 202 207 214 222 257 264 270

Branche ou l'indicateur 1 est relaxe

Branche ou l'indicateur 2 est relaxe

Branche ou l'indicateur 3 est relaxe

108 289

CLASSE NC 14: REPRESENTANT: 3 3 2 CARDINAL: 13

Noyau

46 74 75 76 82 119 132 189 194 198 229 260 285

Branche ou l'indicateur 1 est relaxe

Branche ou l'indicateur 2 est relaxe

Branche ou l'indicateur 3 est relaxe

CLASSE No 3

0,0%

 =faible

 =moyen

 =eleve

 =tres eleve

34,8%

52,2%

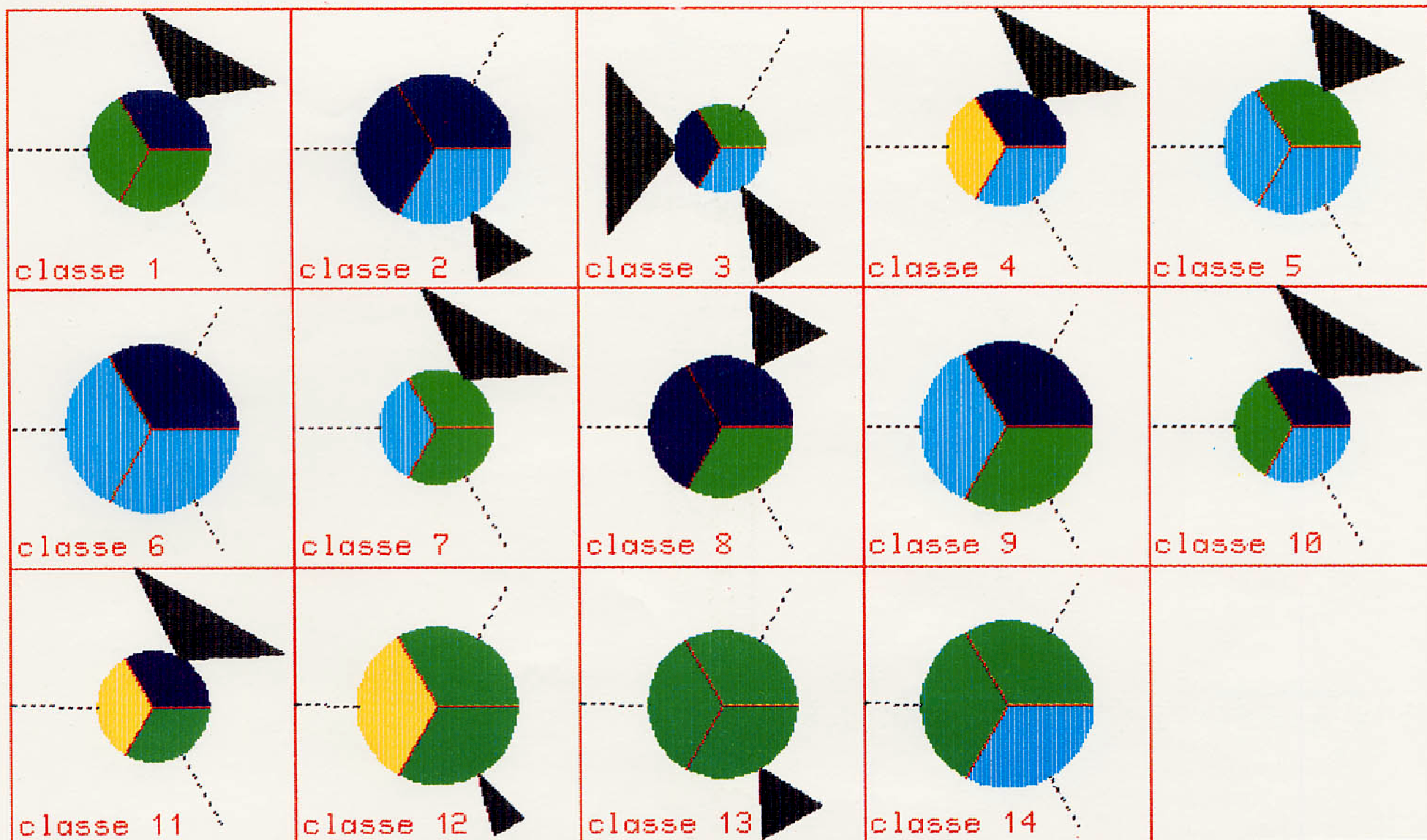
13,0%

1:nombre de pieces
fabriquees

2:nombre de chariots
utilises

3:ecart moyen entre
quotas souhaitees et
quotas obtenus

Fig.2



representation de l'ensemble des
classes pour la regle no 3
methode:1 seuil:9 tolerance:6

b) Lorsque la méthode "Plus proche voisin" est appliquée

Les résultats et les représentations graphiques sont données ci-dessous en suivant la même progression que dans a).

CLASSIFICATION DE LA MEMOIRE ASSOCIEE A LA REGLE 3

Le nombre de classes homogenes de la memoire associee a 3 est:
42

Seuil: 9 Tolerance: 6

METHODE DE REGROUPEMENT AU PLUS PROCHE VOISIN

Nombre maximum d'indicateurs differents: 2

Pourcentage d'elimines:
4.78%

Coefficient de precision:
0.8100

Le nombre definitif de classes est:
15

Liste des representants:

4 3 3 , 4 4 2 , 3 4 2 , 4 1 2 , 3 2 2 , 4 2 2 , 3 2 3 , 4 4 3 , 4 2 3 , 4 3 2 ,
4 1 3 , 3 1 3 , 3 3 3 , 3 3 2 , 2 1 2 ,

COMPOSITION DEFINITIVE DES CLASSES

CLASSE NC 1: REPRESENTANT: 4 3 3 CARDINAL: 12

Noyau

2 9 22 23 24 77 142 152 172 220 250 271

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NC 2: REPRESENTANT: 4 4 2 CARDINAL: 16

Noyau

3 13 34 37 38 68 98 103 109 115 151 170 208 216 269 274

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NC 3: REPRESENTANT: 3 4 2 CARDINAL: 17

Noyau

5 59 69 81 87 110 117 141 148 178 204 218

Branche ou il y a une variation sur 1 indicateur
106 191 293

Branche ou il y a une variation sur 2 indicateurs
211 239

CLASSE NC 4: REPRESENTANT: 4 1 2 CARDINAL: 28

Noyau

6 19 29 62 67 88 92 100 104 122 145 153 167 171 200 217 224 240 277 279

Branche ou il y a une variation sur 1 indicateur
15 47 89 144 174 187 201 232

Branche ou il y a une variation sur 2 indicateurs

CLASSE NO 5: REPRESENTANT: 3 2 2 CARDINAL: 24

Noyau

7 11 53 54 56 65 116 123 140 147 149 166 179 181 219 226 245 266 280

Branche ou il y a une variation sur 1 indicateur
40 61 156 259

Branche ou il y a une variation sur 2 indicateurs
129

CLASSE NC 6: REPRESENTANT: 4 2 2 CARDINAL: 23

Noyau

12 27 44 49 91 95 97 107 113 118 126 136 168 169 173 223 230 231 242 265
268 286 288

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NC 7: REPRESENTANT: 3 2 3 CARDINAL: 20

Noyau

16 28 52 58 70 72 93 102 112 252 261 273 275 283

Branche ou il y a une variation sur 1 indicateur
30 32 212 237 254

Branche ou il y a une variation sur 2 indicateurs
175

CLASSE NO 8:

REPRESENTANT: 4 4 3

CARDINAL: 12

Noyau

18 128 154 183 234 244 251 262 287 290

Branche ou il y a une variation sur 1 indicateur

31 282

Branche ou il y a une variation sur 2 indicateurs

CLASSE NO 9:

REPRESENTANT: 4 2 3

CARDINAL: 22

Noyau

25 36 45 51 83 90 96 105 125 130 137 143 150 163 164 193 197 215 227 235
267 278

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NO 10:

REPRESENTANT: 4 3 2

CARDINAL: 16

Noyau

26 79 120 124 131 138 139 157 159 165 186 228 241 263 272 281

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NO 11:

REPRESENTANT: 4 1 3

CARDINAL: 13

Noyau

35 48 84 121 158 161 188 213 233 243 247 248 276

Branche ou il y a une variation sur 1 indicateur

Branche ou il y a une variation sur 2 indicateurs

CLASSE NO 12:

REPRESENTANT: 3 1 3

CARDINAL: 23

Noyau

41 55 63 64 78 99 176 196 203 205 209 236 255 256 258

Branche ou il y a une variation sur 1 indicateur

10 184 192 291 86

Branche ou il y a une variation sur 2 indicateurs

114 146 190

CLASSE NC 13:

REPRESENTANT: 3 3 3

CARDINAL: 20

Noyau

42 57 73 80 101 162 199 202 207 214 222 257 264 270

Branches ou il y a une variation sur 1 indicateur

21 71 94 180 289

Branches ou il y a une variation sur 2 indicateurs

127

CLASSE NC 14:

REPRESENTANT: 3 3 2

CARDINAL: 19

Noyau

46 74 75 76 82 119 132 189 194 198 229 260 285

Branches ou il y a une variation sur 1 indicateur

108 135 177 210 246 292

Branches ou il y a une variation sur 2 indicateurs

CLASSE NC 15:

REPRESENTANT: 2 1 2

CARDINAL: 14

Noyau

1 17 20 50 133 225 238

Branches ou il y a une variation sur 1 indicateur

8 111

Branches ou il y a une variation sur 2 indicateurs

85 134 253 185 195

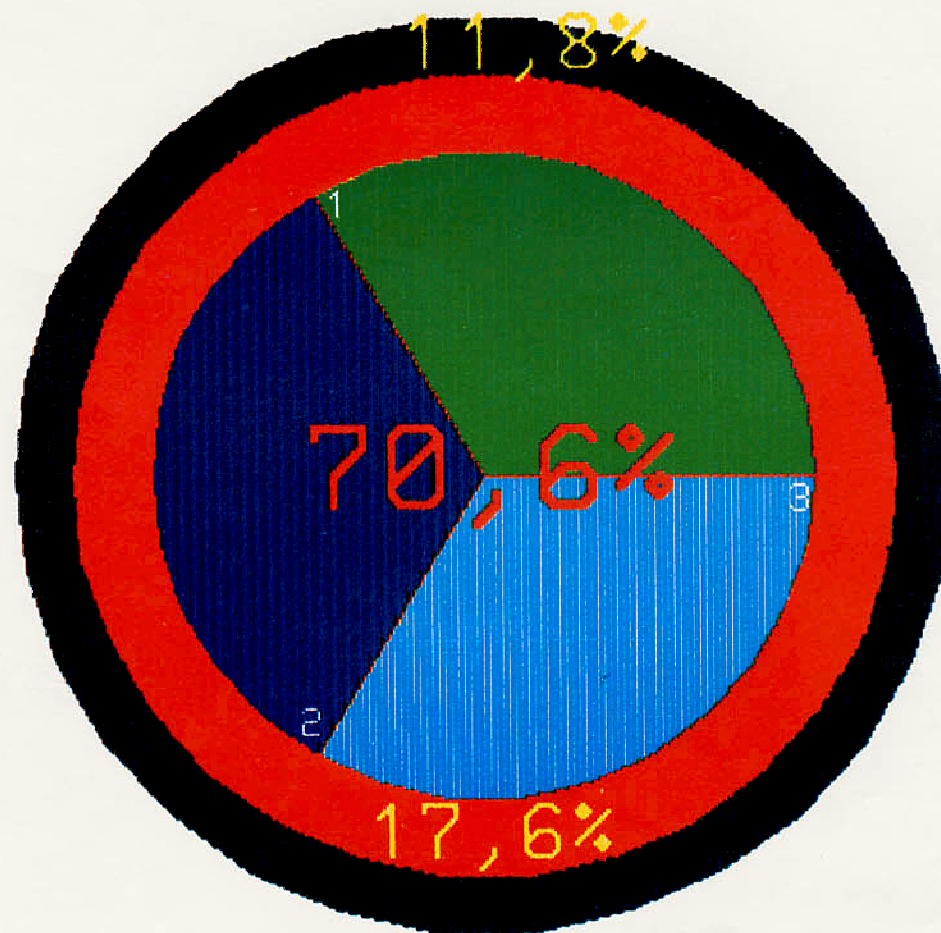
CLASSE No3

 =faible

 =moyen

 =eleve

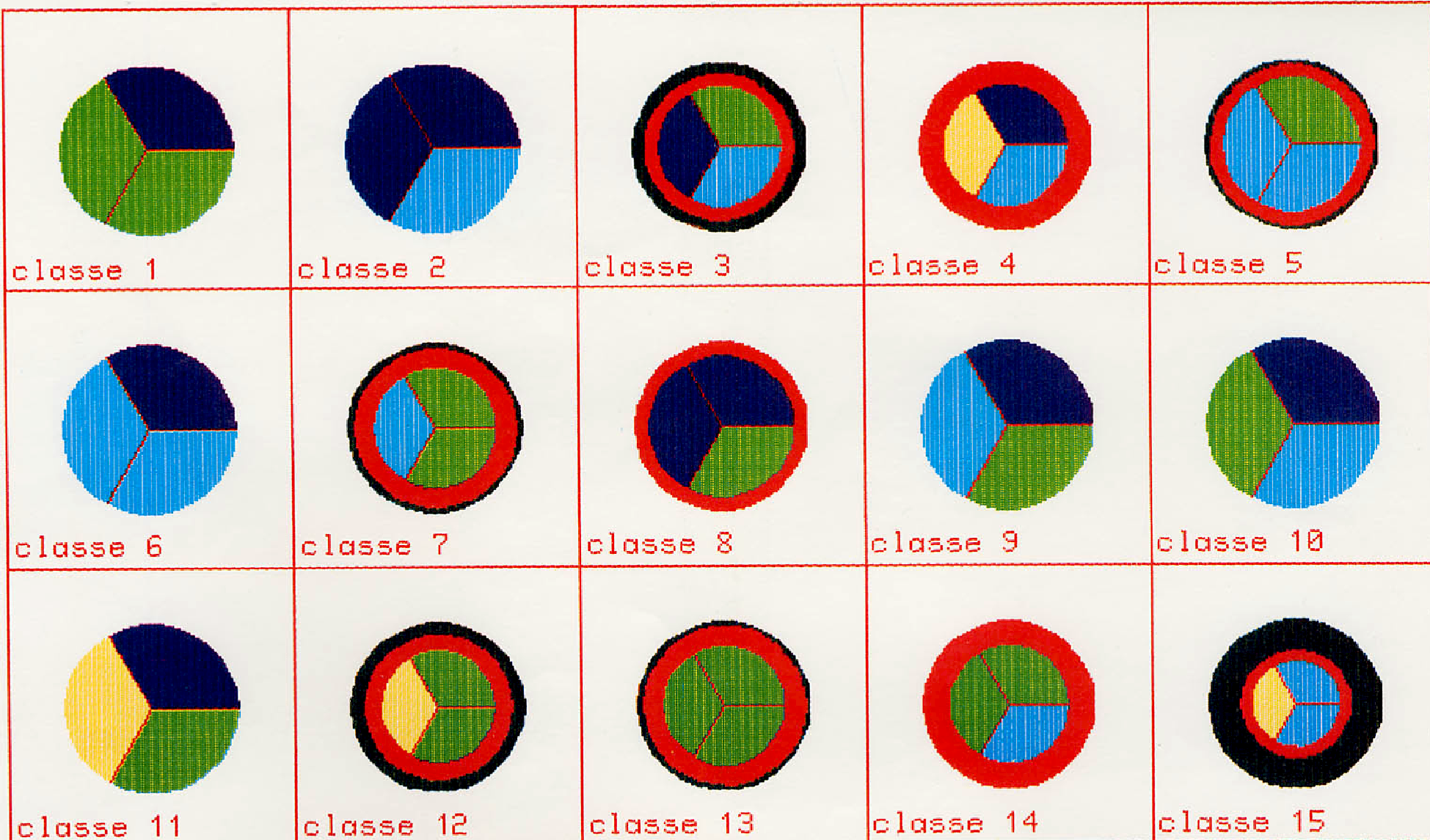
 =tres eleve



1:nombre de pieces fabriquees

2:nombre de chariots utilises

3:ecart moyen entre quotas souhaitees et quotas obtenus



representation de l'ensemble des
classes pour la regle no 3

methode:2 seuil:9 tolerance:6

nombre maximum d'indicateurs differents:2

CONCLUSION

Le travail que nous venons de présenter a un objectif simple : visualiser les résultats atteints par un système de production afin de permettre aux utilisateurs de les évaluer qualitativement en temps réel. Nous venons de proposer deux approches qui semblent satisfaisantes. L'étape suivante de notre démarche sera d'intégrer ces approches dans une boucle d'ordonnancement que nous appliquerons d'abord à un logiciel de simulation par événements.

```

c-----
c-----
c-----
c-----
c-----
c      ** *      ** *** *** **
c      *  *      * * * * *
c      *  *      **** *** **
c      *  *      * * * * *
c      ** *** * * * * *** **
c-----
c-----
c-----
c-----
c-----

```

```

c
c
c      I I
c      I PRESENTATION DU PROGRAMME I
c      I I
c-----
c
c

```

c Ce logiciel classe un ensemble d'observations de p indicateurs apres appli-
c cation de n regles de gestion a tft etats initiaux.Cette operation s'effectue
c en 3 etapes:

c -decoupage de l'ensemble des observations en intervalles,on aboutit ainsi a
c un fichier de tfixn p-uplets d'entiers(un entier representant un intervalle)
c ce fichier sera appele "fichier qualitatif".
c -structuration en classes homogenes des observations issus de l'application
c d'une regle particuliere
c -traitement des classes homogenes par une des deux methodes (relaxation d'un
c indicateur,plus proche voisin)

c Les resultats sortent sous 3 formes differentes:
c -une sortie directe a l'ecran avec 2 options (grandeurs caracteristiques et
c representants des classes ou resultats complets)
c -un fichier destine a la representation graphique
c -un fichier destine a une analyse discriminante

c-----LISTE DES PRINCIPALES VARIABLES

c On indique pour chaque variable dans quel segment elle est utilisee("comm"
c pour variable commune,"main" pour variable utilisee dans le programme princi-
c pal eventuellement en tant que parametre d'appel d'un sous-programme)

```

c
c      Variables scalaires ou de caracteres:
c      -----
c      main      rep:variable binaire (oui/non)
c      main      n:nombre de regles
c      comm      p:nombre d'objectifs
c      comm      tft:nombre d'etats initiaux
c      main      nq:echelle de quantification
c      main      s:seuil de separation entre grandes et petites classes
c      main      ts:seuil de tolerance pour l'elimination des petites classes
c               pris ici toujours egal a s-E(s/3)
c      main      meth:no de la methode(1/2)
c      main      lo:no de l'option pour l'ecriture des resultats(1/2)
c      main      choix:variable binaire (oui/non)
c      main      imax:nombre maximum d'indicateurs differents(methode 2)
c      main      nc:nombre de classes homogenes
c      main      ncp:nombre de classes principales
c      minimaxi  tor:coefficient multiplicatif de l'ecart-type pour ecartier
c               les queues de distribution (ici tor=2)

```



```

c
c      Tableaux:
c      -----
c      main      me:tableau qualitatif pour une regle et pour une experience
c      comm      cl:tableau des classes exactes puis secondaires
c      comm      mc:tableau des cardinaux des classes de cl, puis de cp
c      comm      cp:tableau des classes principales
c      comm      mcp:tableaux des cardinaux des classes de cp
c      main      mt:tableau de travail pour le decoupage en classes exactes
c      quats     c1,c2:tableaux des bornes du fichier des observations
c

```

c Les variables non mentionnees sont des variables muettes de sous-programme
c prenant lors du deroulement du programme les valeurs des variables ci-dessus,
c ou bien des variables secondaires pour lesquelles des commentaires sont don-
c nes dans les sous-programmes.

```

c
c-----
c-----LISTE DES FICHIERS
c-----
c

```

```

c      Fichiers sequentiels:
c      -----
c      fichier no 1:valeurs des parametres fondamentaux(n,p,tfi),
c      des moyennes et ecart-types des observations, le
c      le programme y ecrit egalement les bornes min
c      et max des observations centrees-reduites
c      fichier no 10:contient les regles,les seuils,les methodes
c      dans l'ordre ou ils doivent etre traitees;ce
c      fichier se remplit automatiquement.
c      fichier no 30:fichier-resultat destine a l'A.I
c      fichier no 83:fichier des observations centrees-reduites,
c      rangees pour chaque etat initial dans l'ordre
c      croissant des regles de gestion qu'on leur a
c      applique.La taille de ce fichier est nxfi
c

```

```

c      Fichier acces directe:
c      -----
c      fichier no 25:fichier-resultat destine au traitement gra-
c      phique
c      fichier no 84:fichier qualitatif(voir plus haut).La taille
c      de ce fichier est nxfi
c

```

c Remarque:pour le bon deroulement du programme seuls les fichiers 1 et 83 doi-
c vent etre remplis,les autres parametres sont donnees par l'utilisa-
c teur en mode conversationnel

```

c
c-----
c-----REMARQUES SUR LE DIMENSIONNEMENT DES TABLEAUX ET FICHIERS:
c-----
c

```

c Si le nombre d'indicateurs devient plus grand que 5, augmenter la dimension
c de mt et me et la 2eme dimension de cp et mcp;
c changer egalement mf,mg,mh des sous-programmes et la taille des enregistre-
c ments du fichier 84(recl).

c Les tableaux cl,mt,mcl sont tels que le nombre de classes homogenes possible
c est 120 et le nombre d'elements par classe 250.

c Les tableaux cp et mcp sont tels que le nombre de grandes classes possible
c est 30 et le nombre d'elements par composante 200.

c Pour tout changement de dimension,modifier egalement les bornes des boucles
c d'initialisation a 0.

c

REMARQUES SUR LES FORMATS D'ENTREE ET DE SORTIE

Seul le format 101 du sous-programme ecr doit etre change, si le nombre d'indicateurs change. Neanmoins si ce nombre devient superieur a 5 il faut modifier les formats d'ecriture et de lecture dans le fichier 84 et les formats 100 et 101 de quabs, ainsi que le format 100 de minimaxi.

De plus, si le nombre de regles devient superieur a 9, il faut reconsiderer les formats d'ecriture et de lecture dans 10 et dans 0 (changer i1 en i2).

Enfin, si les dimensions des tableaux cl, mc, cp, mcp sont changees on doit egalement modifier les formats d'ecriture de ecr et rempl.

I	I
I PROGRAMME PRINCIPAL	I
I	I

```
character*1 rep,choix
integer p,ts,s,tfi,s1,s2
integer cl,cp
dimension mt(120,5),me(5)
common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
open(1,form="formatted")
open(10,form="formatted")
open(25,access="direct",form="formatted",recl=25)
open(30,form="formatted")
open(83,form="formatted")
open(84,access="direct",form="formatted",recl=10)
```

```

c-----
c-----INITIALISATION DES PRINCIPALES VARIAELES D'ENTREE
c-----
c
      read(1,99)n,p,tfi
99      format(i2,1x,i1,1x,i3)
      write(0,100)
      read(0,104)choix
100     format("Voulez-vous le remplissage des fichiers resultats?")
      write(0,101)
      read(0,102)lo
101     format("Quelle option choisissiez-vous pour l'ecriture des resultats?"
      & ,/,"-resultats complets(1)",/,"-grandeurs caracteristiques(2)")
102     format(i1)
      write(0,103)
103     format(' e fichier qualitatif est il deja rempli?")
      read(0,104)rep
104     format(v)
      if(rep.eq."n") then
      write(0,105)
105     format("Quelle echelle choisissiez vous pour la quantification?")
      read(0,106)nq
106     format(i1)

```

c
c
c
c
c
c

REPLISSAGE DU FICHIER QUALITATIF (FICHIER 84)

call quats(tfi,n,p,nq)

c

```

c-----
c-----CHOIX DES AUTRES VARIABLEES D'ENTREE (REGLES,SEUILS,METHODES)
c-----
c
      endif
      write(0,107)
107    format("Entrez:","/","- les regles que vous desirez classer (de n1 a n2
      &)"",
      &"/","-les seuils que vous souhaitez pour departager petites et grandes cla
      &sses",/,"(s1,s2,s2=0 si vous ne desirez qu'un seuil)",/,"-la meth
      &ode de regroupement choisie(1,2,ou 3 pour les 2)",/,"
      &"-le nombre maximum d'indicateurs pouvant etre differents(methode 2)",/
      &"/,"toutes ces valeurs devant etre ecrites separees par un blanc,en for"
      &,"mat i1 sauf",/,"pour les seuils en format i2")
      read(0,108)n1,n2,s1,s2,iz,imax
108    format(2(i1,1x),2(i2,1x),2(i1,1x))
c
c      Remplissage du fichier de gestion du travail (fichier 10)
c      -----
c
      call ges(n1,n2,s1,s2,iz,imax)
      rewind(10)
2      read(10,109)i
109    format(i1)
      if(i.ne.0) then
c
c-----
c-----DECOUPAGE EN CLASSES HOMOGENES POUR LA REGLE No I
c-----
c
      write(0,110)i
110    format(79("-"),/,"CLASSIFICATION DE LA MEMOIRE ASSOCIEE A LA REGLE "
      &,i1,/79("-"))
c
c      Initialisation a "0"
c      -----
c
      do4 l=1,120
      mc(l)=0
      do5 k=1,250
      cl(l,k)=0
5      continue
      do6k=1,p
      mt(l,k)=0
6      continue
4      continue
      do7 l=1,30
      do8 j=1,p+1
      mcp(l,j)=0
      do9 k=1,200
      cp(l,j,k)=0
9      continue
8      continue
7      continue
      do10 j=1,p
      me(j)=0
10     continue
      nc=0
c
c      Algorithme de classification
c      -----
c
c Boucle sur les observations associees a la regle no i
      do11i=1,tfi

```

```

        j=tfi*(i-1)+11
c Lecture dans le fichier qualitatif du p-uplet associe a l'observation j
        read(84,j,111)(me(k),k=1,p)
111        format(5(1x,i1))
c Test de debut de classification
        if(nc.ne.0) then
c Foucle sur les classes deja constuites(tableau mt)
        do 13 i2=1,nc
        do 14 j1=1,p
c Test pour savoir si le p-uplet associe a j est deja apparu
        if(me(j1).ne.mt(i2,j1))goto13
14        continue
c Si oui:i1 est ajoute a la classe i2 , on incremente le cardinal de i2 de 1
        mc(i2)=mc(i2)+1
        cl(i2,mc(i2))=i1
        goto11
13        continue
        endif
        nc=nc+1
c Sinon:Une nouvelle classe est cree
        do 15 j1=1,p
        mt(nc,j1)=me(j1)
15        continue
        mc(nc)=1
        cl(nc,1)=i1
11        continue
        write(0,112)i,nc
c Ecriture du nombre de classes homogenes obtenues
112        format("Le nombre de classes homogenes de la memoire associee a ",i1,
        &" est:",/,i3)
c
c-----
c-----TRAITEMENT DES CLASSES HOMOGENES
c-----
c
c Lecture dans le fichier 10 et ecriture a l'ecran
        read(10,113)s
113        format(i2)
        ts=s-int(s/3)
        write(0,114)s,ts
114        format(/,"Seuil:",i2,10x,"Tolerance:",i2,/,/)
c
c        Tri entre grandes et petites classes
c        -----
c
        call selec(s,nc,ncp,0)
        ncl=ncp+1
c Le nombre ncl sera utilise lors de lecriture dans cp des classes non re-
c groupees et de cardinal plus grand que ts
c
c        Choix de la methode
c        -----
c
c Lecture dans le fichier 10 et tranchement sur la methode retenue
        read(10,115)meth,imax
115        format(i1,1x,i1)
        if(meth.eq.1) then
c-----
c-----METHODE DE RATTACHEMENT PAR RELAXATION D'UN INDICATEUR
c-----
        write(0,116)
116        format(/,"METHODE DE REGROUPEMENT PAR RELAXATION D'UN INDICATEUR ",/)
c
c        Rattachement des petites classes(<s) aux grandes classes(>s-1)

```

```

c -----
c
c      call regr(i,ncp,nc,10,1,1,1)
c
c      Rattachement des petites classes(<s) aux classes(>ts-1)
c -----
c
c      call selec(ts,nc,ncp,ncp)
c Test d'existence de classes de cardinal entre ts et s non regroupees
c Si test positif,nouveau regroupement
c      if(nc1.ne.(ncp+1))call regr(i,ncp,nc,10,1,nc1,1)
c      else
c -----
c -----METHODE DES PLUS PROCHES VOISINS
c -----
c      write(0,117)imax
117      format(/,"METHODE DE REGROUPEMENT AU PLUS PROCHE VOISIN",/,/,
&"Nombre maximum d'indicateurs differents:",i2,/)
c
c      Rattachement des petites classes(<s) aux grandes classes(>s-1)
c -----
c
c      call regr(i,ncp,nc,1,1,1,2)
c
c      Rattachement des petites classes(<s) aux classes(>ts-1)
c -----
c
c      call selec(ts,nc,ncp,ncp)
c Test d'existence de classes de cardinal entre ts et s non regroupees
c Si test positif,nouveau regroupement (1,1)
c      if(nc1.ne.(ncp+1))call regr(i,ncp,nc,1,1,nc1,2)
c Foucle sur le nombre d'indicateurs jusqu'a imax,des regroupements (1,ireg)
c sont effectues de ireg egal 2 jusqu'a imax
c      if(imax.gt.1) then
c          do18 ireg=2,imax
c          call regr(i,ncp,nc,1,ireg,1,2)
18      continue
c      endif
c
c -----
c -----ECRITURE DES RESULTATS
c -----
c
c      Ecriture des grandeurs caracteristiques
c -----
c
c      endif
c      call pourc(nc,q)
c      call prec(ncp)
c      write(0,118)ncp
118      format("Le nombre definitif de classes est:",/,i2,/,/)
c      call repres(i,ncp)
c Test sur l'option d'ecriture des resultats
c      if(lo.eq.1)call ecr(i,ncp,meth,imax)
c Remplissage des deux fichiers resultats
c      if(choix.eq."o")call remp(ncp,s,i,meth)
c
c -----
c -----FIN DU PROGRAMME
c -----
c
c Lecture dans le fichier 10 et test de fin de travail
c      read(10,119)rep
119      format(a1)

```



```

      subroutine quabs(tfi,n,p,e)
      integer p,tfi,e
      dimension b(5),c1(5),c2(5),kc(5)
c Positionnement en fin de fichier 1
      do 1 j=1,2
      read(1,101)m
1      continue
c l:nombre total d'observations (nombre d'etats initiaux)x(nombre de regles)
      l=tfi*n
c Boucle sur les indicateurs
      do2 j=1,p
c Calcul des bornes du fichier des observations
      rewind(83)
      call minimaxi(j,c1(j),0,p,1)
      rewind(83)
      call minimaxi(j,c2(j),1,p,1)
2      continue
c Ecriture de c1 et c2 dans le fichier 1
      write(1,100)(c1(j),j=1,p),(c2(j),j=1,p)
100      format(10(f6.3,1x))
      rewind(83)
c Boucle sur les etats initiaux
      do3 i1=1,tfi
c Boucle sur les regles
      do4 i=1,n
c Lecture de l'observation associee a l'etat initial i1 et a la regle i
      read(83,101)(b(k),k=1,p)
101      format(2x,5(f7.3,1x))
c Boucle sur les indicateurs
      do5 j=1,p
c Calcul du p-uplet kc representant les intervalles associes a l'observation
      call codage(t(j),kc(j),c1(j),c2(j),e)
5      continue
c Ecriture dans le keme enregistrement du fichier 84:on range les observations
c regle par regle dans l'ordre croissant des etats initiaux
      k=tfi*(i-1)+i1
      write(84,k,102)(kc(j),j=1,p)
102      format(5(1x,i1))
4      continue
3      continue
      return
      end

c
c      Sous-programme calculant les bornes du fichier des observations
c
c -----
      subroutine minimaxi(j,t1,ii,p,1)
      integer p
      dimension a(5)
      bor=2.
c Lecture de la premiere observation dans le fichier 83
      read(83,100)(a(k),k=1,p)
100      format(2x,5(f7.3,1x))
c On s'interesse au jeme indicateur
      t1=a(j)
c t1:valeur du max ou du min
c a(j):valeur de l'indicateur courant
c Boucle sur les observations a partir de la 2eme
      do1 k1=2,1
c Lecture de l'indicateur courant
      read(83,100)(a(k),k=1,p)
      b=abs(a(j))
c Test pour eliminer les queues de distributions
      if(b.le.bor) then

```

```

c Test pour savoir s'il s'agit du max ou du min
  if(ii.ne.1) then
c Test de comparaison avec la valeur courante de t1
  if(a(j).le.t1)t1=a(j)
  else
  if(a(j).ge.t1)t1=a(j)
  endif
  endif
1  continue
  return
  end

```

```

c
c      Sous-programme de codage
c      -----
c

```

```

  subroutine codage(a,k,m1,m2,nq)
  real m1,m2
c m1,m2: bornes min et max de l'intervalle sur lequel a lieu le decoupage
  b=(m2-m1)/nq
c b: largeur de chaque intervalle
  do1 i2=1,nq
  if(a.le.(m1+i2*b)) then
c a tombe dans l'intervalle no k
  k=i2
  goto2
  endif
1  continue
2  return
  end

```

```

c
c      Sous-programme de tri
c      -----
c

```

```

  subroutine selec(ls,nc,ncp,m)
  integer cl,cp,s,p,tfi
  common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c On ecrit les grandes classes et leurs cardinaux dans cp et mcp a partir de
c ncp+1(de 1 si c'est le premier tri)
  ncp=m
  do1 j=1,nc
c Test sur le cardinal des classes
  if(mc(j).ge.ls) then
c Si test positif incrementation de ncp,mise a jour de cl,mc,cp,mcp
  ncp=ncp+1
  do2k=1,mc(j)
  cp(ncp,1,k)=cl(j,k)
  cl(j,k)=0
2  continue
  mcp(ncp,1)=mc(j)
  mc(j)=0
  endif
1  continue
  return
  end

```

```

c
c      Sous-programme gerant le regroupement des classes
c      -----
c

```

```

  subroutine regr(i,ncp,nc,e1,e2,m,meth)
  integer cp,e1,e2,cl,p,tfi
c (e1,e2): indice de dissemblance
  dimension mf(5),mg(5)
  common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c Boucle sur les petites classes

```

```

do 1 l=1,nc
  if(mc(l).ne.0) then
c Lecture du representant de la classe no l
    call lect(i,mf,cl(l,1))
c Boucle sur les grandes classes
    do 2 j=m,ncp
c Lecture du representant de la classe no j
      call lect(i,mg,cp(j,1,1))
      kp=0
c Test de regroupement suivant les valeurs de e1 et.e2
      do3 kreg=1,p
        ne=abs(mf(kreg)-mg(kreg))
        if(ne.ne.0) then
          if(ne.gt.e1)goto2
          kp=kp+1
          no=kreg
          if(kp.gt.e2)goto2
        endif
      3 continue
c Suivant la methode,on regroupe sur la composante associee a un indicateur (1)
c ou associee a e2 (2)
      if(meth.eq.2)no=e2
c Regroupement de la classe l sur la classe j et disparition de la classe l:mi-
c se a jour des tableaux cl,cp,mc,mcp
      do 4 k1=1,mc(l)
        cp(j,(no+1),(mcp(j,(no+1))+k1))=cl(l,k1)
        cl(l,k1)=0
      4 continue
      mcp(j,(no+1))=mcp(j,(no+1))+mc(l)
      mc(l)=0
      2 continue
    endif
  1 continue
  return
end

```

Sous-programme de lecture dans le fichier qualitatif

```

subroutine lect(i,mh,kr)
integer p,tfi,cp,cl
dimension mh(5)
common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c ks:observation associee a l'etat initial kr et le regle i
  ks=tfi*(i-1)+kr
c Lecture dans le fichier qualitatif
  read(84'ks,100)(mh(kt),kt=1,p)
100 format(5(1x,i1))
  return
end

```

Sous-programme d'ecriture des representants des classes

```

subroutine repres(i,ncp)
integer p,cp,cl,tfi
dimension mrep(120),mf(5)
c mrep:liste des representants des classes
  common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c Boucle sur les grandes classes
  do 1 j=1,ncp
c Lecture du representant de la classe j
    call lect(i,mf,cp(j,1,1))
  1

```



```

c Remplissage de mrep
      do 2 k=1,p
        mrep(p*(j-1)+k)=mf(k)
2      continue
1      continue
c Ecriture a l'ecran
      write(0,100)(mrep(k),k=1,p*ncp)
100     format("Liste des representants:",/,5(10(3(i1,1x)," ",/)))
      return
      end

c
c      Sous-programme de calcul du pourcentage d'elimines
c      -----
c
      subroutine pourc(nc,q)
      integer p,tfi,cl,cp
      common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
      nt=0
c nt:nombre d'elements elimines
      do1 i2=1,nc
        nt=nt+mc(i2)
1      continue
      a=tfi
c q:pourcentage d'elimines
      q=(nt/a)*100.
      write(0,100)q
100     format(/,"Pourcentage d'elimines:",/,f5.2,"%",/)
      return
      end

c
c      Sous-programme de calcul du coefficient de precision
c      -----
c
c Ce sous-programme calcule egalement les cardinaux des grandes classes en les
c les ecrivant dans mc
c
      subroutine prec (ncp)
      integer cl,cp,p,tfi
      common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
      ak=0.
      bk=0.
c Foucle sur les grandes classes pour le remplissage de mc
      do 1 j=1,ncp
        mc(j)=0
        do 2 k=1,p+1
          mc(j)=mc(j)+mcp(j,k)
2        continue
1      continue
c Foucle sur les grandes classes pour le calcul du coefficient de precision
      do 3 j=1,ncp
c ak:nombre d'elements issus d'un noyau
c bk:nombre d'observations retenues
        ak=ak+mcp(j,1)
        bk=bk+mc(j)
3      continue
c bk:coefficient de precision
      bk=ak/bk
      write(0,100)bk
100     format("Coefficient de precision:",/,f6.4,/)
      return
      end

c
c      Sous-programme d'ecriture des resultats complets
c      -----

```

```

c
    subroutine ecr(i,ncp,meth,imax)
    integer cl,cp,p,tfi
    dimension mf(5)
    common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c icp:nombre de composantes par classe(p pour la meth1,imax pour la meth2)
    icp=p
    if(meth.eq.2)icp=imax
    write(0,100)
100    format(79("#"),/,"COMPOSITION DEFINITIVE DES CLASSES",/,79("#"),/)
c Foucle sur les grandes classes
    do1j=1,ncp
c Lecture du representant de la classe j
    call lect(i,mf,cp(j,1,1))
c Ecriture de l'en-tete et du noyau de la classe j
    write(0,101)j,(mf(k),k=1,p),mc(j),(cp(j,1,k),k=1,mcp(j,1))
101    format(/,"CLASSE NO ",i2,":",10x,"REPRESENTANT:",3(1x,i1),10x,
    &"CARDINAL:",i3,/,/,"Noyau",/,10(20(i3,1x),/))
c Ecriture des composantes
    do2 k=1,icp
c Test sur la methode utilisee
    if(meth.eq.1) then
        write(0,102)k,(cp(j,k+1,1),l=1,mcp(j,k+1))
102    format(/,"Eranche ou l'indicateur",i2," est relaxe",/,3(20(i3,1x),/))
    else
        write(0,103)k,(cp(j,k+1,1),l=1,mcp(j,k+1))
103    format(/,"Eranche ou il y a une variation sur ",i1," indicateurs",/,
    &3(20(i3,1x),/))
    endif
2    continue
    write(0,104)
104    format(79("-"),/)
1    continue
    return
end

c
c
c      Sous-programme de remplissage des fichiers resultats
c
c
c
c Ce sous-programme peut etre deconnecte du reste du programme,il n'est utile
c que dans l'interface avec d'autres programmes(cf "presentation du programme")
c
    subroutine remp(ncp,s,i,meth)
    integer cp,cl,s,p,tfi
    common cl(120,250),mc(120),cp(30,6,200),mcp(30,6),p,tfi
c Ecriture dans le fichier 25
    write(25'1,100)i,meth,s,ncp
    do 1 j1=1,ncp
    write(25'(j1+1),101)cp(j1,1,1),mc(j1),(mcp(j1,k),k=1,p+1)
1    continue
100    format(i2,1x,i1,i2,1x,i2)
101    format(10(i3))
c Ecriture dans le fichier 30
    do2 j=1,ncp
    j3=1
    do3 j1=1,p+1
    if(mcp(j,j1).ne.0) then
        do4 j2=1,mcp(j,j1)
        cl(j,j3)=cp(j,j1,j2)
        j3=j3+1
4    continue
    endif
3    continue
    write(30,102)(cl(j,k),k=1,mc(j))

```


[illegible]

3

22

3.

3.

3.