



HAL
open science

Cross Clustering Heuristics for addressing the Manufacturing Cell Formation problem

Roland de Guio

► **To cite this version:**

Roland de Guio. Cross Clustering Heuristics for addressing the Manufacturing Cell Formation problem. 1992. hal-00651989

HAL Id: hal-00651989

<https://hal.science/hal-00651989>

Preprint submitted on 14 Dec 2011

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.

Cross Clustering Heuristics for addressing the Manufacturing Cell Formation problem

Roland DE GUIO

INSA de Strasbourg

24, boulevard de la Victoire, 67084 Strasbourg Cedex FRANCE

Roland.deguio@insa-strasbourg.fr

This paper, written in 1992, summarizes and regroups in one paper several cross clustering algorithms that are presented in my PhD memoire in French and that are referred to or presented partially in several of my papers. They are also implemented in several recent applications (2010) for manufacturing flows analysis and used in other context like inventive problem solving (2009). It is also used to compare its performance towards metaheuristics including the functions of these algorithms.

SUMMARY

1. INTRODUCTION

2. GENERAL MAXIMIZATION FORMULATION

- 2.1 The quasi-seriation problem
- 2.2 The block-seriation problem
- 2.3 The concept of diagonal blocks

3. THE LINEAR INTEGER MODELS.

- 3.1 Case of binary data
- 3.2 Case of integer data

4. THE QUASI-SERIATION ALGORITHMS

- 4.1 Basic notations
- 4.2 Basic ingredients of the quasi-seriation algorithms
 - 4.2.1 Quasi seriation rows assignment procedure
 - 4.2.2 Quasi seriation columns assignment procedure
- 4.3 A basic quasi-seriation algorithm.
- 4.4 How to get equivalent solutions
- 4.5 A raffined quasi-seriation algorithm

5. BLOCK SERIATION ALGORITHMS

- 5.1 The basic seriation procedure.
- 5.2 A seriation algorithm based on quasi-seriation

6. COMPUTING PERFORMANCES

- 6.1 Computing time complexity
- 6.2 Comparison of block seriation algorithms.

7. EXAMPLE OF USES IN PRODUCTION MANAGING

- 7.1 Analysis of the binary machine-parts incidence matrix
- 7.2 Analysis of the part-loads matrix
- 7.3 Analysis of the traffic beetween machines

8 CONCLUSION

SUMMARY

The first aim of this paper is to present heuristics used to solve the generalized blocks seriation models as developed by Marcotorchino in [1]. The second aim is to show how they can be used to solve several problems in the area of Group Technology in manufacturing .

KEY WORDS:

Optimal clustering	Clustering algorithms
Manufacturing cells	Group technology
Block seriation	

1. INTRODUCTION

A rough statement of the blocks seriation clustering problem would be the following: "We are a direct simultaneous partitioning of an non negative value matrix with n rows and m columns in order to obtain a high density of the largest values in the diagonal blocks".

To clarify the assertion "largest values" let us give some examples.

Example 1:

Matrix A is a binary matrix. The goal is to obtain, by permuting the rows and columns of this matrix, a matrix B in the ones are concentrated in the diagonal blocks. One observes that the set of rows on one side and the set of columns on the other side are clusters solving the block seriation problem as stated previously. For this reason, the clustering problem must be stated in more detail. Some authors predefine the number of clusters ($\{6\};\{16\}$) and add size constraints to the clusters ($\{19\};\{12\};\{1\};\{17\}$). In other block seriation problems some rows or columns of the matrix are left unclassified ($\{18\};\{2\};\{8\}$). J.M Proth [6] and F.Marcotorchino [14] are looking for the best relational match to an ideal block seriation matrix. For others the number of clusters does not matter and the density of ones inside diagonal blocks defines the number of clusters [15].

Example 2:

Matrix A is now composed of non negative values. The blocks seriation problem can be summarized as clustering the rows and columns of A in order to obtain a matrix B whose non zero values are concentrated in diagonal blocks. J.R Kumar, A.Kusiak and A. Vanelli [11] limit cluster size to a given range, as do G. Harhalakis, R. Nagi, and J.M Proth [9]. In this paper we define a model based on a relational analysis approach which generalizes F. Marcotorchino's block seriation model for binary matrices.

Example 3:[7][12]

The matrix is still composed of non negative values, but the number of zeroes does not matter. Let N_{max} be the maximum value of the matrix elements and N_{min} the minimum one. The data we should like to group in the diagonal blocks can, for example, be those belonging to the interval $[\text{limit}...N_{max}]$ where limit belongs to $]N_{min}, N_{max}]$.

In the further developed models, the number of clusters is an unknown quantity and the constraints on the partition are defined by the minimum density of values inside the diagonal blocks. However, presented heuristics can be used to solve blocks seriation problems which include constraints on cluster number or cluster size [15].

The examples previously mentioned are mainly drawn from the field of production management. The list is not exhaustive and one could give more than a hundred references in this one field. In [14] one can find numerous references from a large range of scientific fields, which deal with the block seriation problem without constraints on the number of clusters.

We present two formulations of the block seriation clustering problem: a general maximization formulation, and an equivalent binary linear programming formulation. Each of these formulations is helpful to analysis of the block seriation problem. We also study the quasi seriation problem and present its application to solve the seriation problem. Mathematically there may be more than one maximum to a maximization problem. We present, an heuristic which facilitates discovery of the equivalent solutions if they exist. Finally, we show how to apply previous results in production management.

2. GENERAL MAXIMIZATION FORMULATION.

2.1 The quasi seriation problem.

Given $A[a_{ij}]$ a matrix of non negative integer values with n rows and m columns.

Let I be the set of rows of matrix A .

Let J be the set of columns of matrix A .

N_{max} is the integer which verifies:

$$N_{max} = \max_{\langle i, j \rangle \in I \times J} (a_{ij})$$

N_{min} is the integer which verifies:

$$N_{min} = \min_{\langle i, j \rangle \in I \times J} (a_{ij})$$

The problem consists in maximizing the function defined by:

$$fr(X, Y, p) = \sum_{l=1}^p \sum_{\langle i, j \rangle \in (X_l \times Y_k)} a_{ij} \quad (1)$$

where:

- p is the number of clusters of both partitions we sought.
- $X = \{X_1, X_2, \dots, X_p\}$ is a partition of the set $I - X_0$ in p clusters.
- X_0 is the set of non classified rows.

- $Y = \{Y_1, Y_2, \dots, Y_p\}$ is a partition of the set $J - Y_0$ in p clusters.
- Y_0 is the set of non classified columns.

Let D_q be the set of feasible solutions of the quasi seriation problem.

A triplet (X, Y, p) is said a feasible solution of the quasi seriation problem if the following constraints are verified:

$$1. X \text{ is a partition of } I - X_0 \text{ in } p \text{ clusters.} \quad (2)$$

$$2. Y \text{ is a partition of } J - Y_0 \text{ in } p \text{ clusters.} \quad (3)$$

Remarks

- Constraints (2) and (3) force the integer p to verify:
 $1 \leq p \leq \text{Inf}(m, n)$ where $\text{Inf}(m, n)$ is the lowest of the two values m and n ;
- p is given by the pair (X, Y) .

A triplet (X^*, Y^*, p^*) is said an optimal solution or an optimal triplet of the quasi seriation problem if it satisfies:

$$\text{fr}(X^*, Y^*, p^*) \geq \text{fr}(X, Y, p) \quad \forall (X, Y, p) \in D_q$$

2.2 The block seriation problem.

The block seriation problem can be stated as a quasi seriation problem whose set of feasible solutions D_b is the subset of D_q which verifies:

$$D_b = \{ (X, Y, p) \in D_q / X_0 = \emptyset \text{ and } Y_0 = \emptyset \}. (\emptyset \text{ is the empty set})$$

2.3 The concept of diagonal blocks.

For a given triplet (X, Y, p) we define p sets \mathcal{D}_ℓ for $\ell = 1, \dots, p$ as follows :

$$\mathcal{D}_\ell = \{ (i, j) \in I \times J / (i, j) \in (X_\ell \times Y_\ell) \}.$$

The triplet (X, Y, p) defines the generic terms of the p diagonal blocks of the permuted matrix as the sets \mathcal{D}_ℓ for $\ell = 1, \dots, p$.

One can note that the concept of diagonal blocks depends only on the initial matrix and on the pair (X, Y) . (The number of clusters p is entirely defined by (X, Y)).

3. THE LINEAR INTEGER MODELS

3.1 Case of binary data [13].

F. Marcotorchino puts forth the generalized relational model of blocks seriation for a given binary matrix $A[a_{ij}]$ defined as in section 2 ($N_{\max}=1, N_{\min}=0$).

Let $Z[z_{ij}]$ be a binary matrix with n rows and m columns defined by:

$$\begin{cases} z_{ij} = 1 & \text{if } a_{ij} \text{ belongs to a diagonal block.} \\ z_{ij} = 0 & \text{otherwise.} \end{cases}$$

The problem is to find Z matrix that maximizes the criterion $F(Z)$ where:

$$F(Z) = \sum_{i=1}^n \sum_{j=1}^m \alpha a_{ij} z_{ij} + \beta (1-a_{ij})(1-z_{ij}) \quad (4)$$

subject to the constraints:

$$\left[\begin{array}{l} z_{ij} + z_{ij'} + z_{i'j'} - z_{i'j} \leq 2 \quad \forall (i, i') \in I \times I; \forall (j, j') \in J \times J \quad (C1) \\ z_{i'j'} + z_{i'j} + z_{ij} - z_{ij'} \leq 2 \quad \forall (i, i') \in I \times I; \forall (j, j') \in J \times J \quad (C2) \\ z_{i'j} + z_{ij} + z_{ij'} - z_{i'j'} \leq 2 \quad \forall (i, i') \in I \times I; \forall (j, j') \in J \times J \quad (C3) \\ z_{ij'} + z_{i'j'} + z_{i'j} - z_{ij} \leq 2 \quad \forall (i, i') \in I \times I; \forall (j, j') \in J \times J \quad (C4) \\ \sum_{j=1}^m z_{ij} \geq 1 \quad \forall i \in I \quad (C5) \\ \sum_{i=1}^n z_{ij} \geq 1 \quad \forall j \in J \quad (C6) \\ z_{ij} \in \{0, 1\} \quad \forall (i, j) \in I \times J \quad (C7) \end{array} \right.$$

The constraints (C1), (C2), (C3), (C4) oblige generic terms of matrix Z diagonal blocks to contain only "1" values and non diagonal blocks to contain only "0" values.

F(Z) maximization under (C1), (C2), (C3), (C4), (C7), constraints is an equivalent formulation of the quasi seriation problem for Nmax=1 and Nmin=0 as stated in section 2.1. The proof is given in [14]. Figures 1,2,3,4 illustrate the correspondences between diagonal blocks of matrix A and Z and the triplet (X,Y,p).

Adding (C5) and (C6) called "assignment constraints", forces all rows and columns to be classified ($X_{\circ} = Y_{\circ} = \emptyset$). The number of row families is equal to the number of column families. F(Z) maximization under (C1), (C2), (C3), (C4), (C5), (C6), (C7), is an equivalent formulation of the block seriation problem as stated in section 2.2. The proof is given in [14].

Note that with this formulation the number of clusters "p" is of no use.

It was shown in [14] that F(Z) can be developed in the following way:

$$F(Z) = \sum_{i=1}^n \sum_{j=1}^m (\alpha a_{ij} - \beta(1-a_{ij})) z_{ij} + \sum_{i=1}^n \sum_{j=1}^m \beta(1-a_{ij}) \quad (5)$$

With the following notations

$$f(Z) = \sum_{i=1}^n \sum_{j=1}^m (\alpha a_{ij} - \beta(1-a_{ij})) z_{ij} \quad (6)$$

$$C = \sum_{i=1}^n \sum_{j=1}^m \beta(1-a_{ij}) \quad (7)$$

F(Z) can be rewritten:

$$F(Z) = f(Z) + C. \quad (8)$$

Because C does not depend on z_{ij} , the matrices Z that maximize f(Z) also maximize F(Z).

Simplifications of f(Z) criterions.

α and β are constants of the models. They balance the influence of positive and negative agreements between A and Z. The case $\alpha=\beta=1/2$ was first proposed in [6]. Let us now consider the function

$$g(Z, \alpha, \beta) = \sum_{i=1}^n \sum_{j=1}^m (\alpha a_{ij} - \beta(1-a_{ij})) z_{ij} \quad (9)$$

When $\alpha+\beta \geq 0$ then to find Z that maximizes $g(Z, \alpha, \beta)$ at α and β fixed is equivalent to find Z that maximizes $g(Z, \alpha' = \alpha/\alpha+\beta, \beta' = \beta/\alpha+\beta)$. Note that $\alpha'+\beta'=1$, $0 \leq \alpha' \leq 1$ and $0 \leq \beta' \leq 1$.

When $\alpha+\beta < 0$ to find Z that maximizes $g(Z, \alpha, \beta)$ at α and β fixed is equivalent to find Z that maximizes $g(Z, \alpha' = -\alpha/\alpha+\beta, \beta' = -\beta/\alpha+\beta)$. Note that $\alpha'+\beta' = -1$; $0 \leq -\alpha' \leq 1$ and $0 \leq -\beta' \leq 1$.

So, by only choosing $f(Z)$ criteria where

$$\begin{cases} \alpha+\beta = \varepsilon ; \\ 0 \leq \alpha \leq 1 \\ 0 \leq \beta \leq 1 \\ \varepsilon \in \{-1, +1\} \end{cases}$$

we do not limit the general formulation of the seriation or quasi seriation problem.

$f(Z)$ criterion can then be rewritten:

$$f(Z) = \varepsilon \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta) z_{ij} \text{ with } 0 \leq \beta \leq 1 \text{ and } \varepsilon \in \{-1, +1\} \quad (10)$$

When $\varepsilon = -1$ the matrix Z which maximizes (10) will concentrate zeroes values of A inside diagonal blocks and ones outside diagonal blocks. Because we are attempting to concentrate ones in diagonal blocks we are to choose only $f(Z)$ criterions with $\varepsilon = 1$ and $\beta \in [0, 1]$. The developed below heuristics are easily tractable for the case $\varepsilon = -1$.

$f(Z)$ criterion can then be rewritten:

$$f(Z) = \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta) z_{ij} \text{ with } 0 \leq \beta \leq 1 \quad (11)$$

When $\beta = 0$ the matrix Z defined by $z_{ij} = 1$ for each row and each column is a trivial solution of the previous blocks seriation problems.

Interpretation of B .

a_{ij} is said in a diagonal block if $z_{ij} = 1$. The number of one values of Z is the number of generic terms of A that are inside the diagonal blocks. Let S be this number. S can be interpreted as the size of the diagonal blocks. Let $Nd1$ be the number of ones values of A that are inside the diagonal blocks. Let $Nd0$ be the number of zeroes values of A that are inside the diagonal blocks. $S = Nd1 + Nd0$ because A is a binary matrix.

Theorem 1:

$$\text{When } f(Z) \geq 0, \text{ then } \frac{Nd1}{S} \geq \beta. \quad (12)$$

Proof of theorem 1:

$$f(Z) = \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta) z_{ij} = \left(\sum_{i=1}^n \sum_{j=1}^m a_{ij} z_{ij} \right) - \beta \left(\sum_{i=1}^n \sum_{j=1}^m z_{ij} \right)$$

$$\sum_{i=1}^n \sum_{j=1}^m a_{ij} z_{ij} = Nd1;$$

$$\sum_{i=1}^n \sum_{j=1}^m z_{ij} = S.$$

So ,

$$f(Z) = Nd1 - \beta S ,$$

As $f(Z)$ is positive ; $Nd1/S \geq \beta$. \square

β can then be interpreted as the minimum concentration of one values inside diagonal blocks.

Theorem 2:

An optimal solution Z^* of the quasi seriation problem verifies:

$$f(Z^*) \geq 0. \quad (13)$$

Proof of theorem 2

Let a_{xy} be a generic term of matrix A which satisfies $a_{xy} = 1$.

Let Zq be the matrix of n rows and m columns defined by:

$$\begin{cases} z_{ij} = 1 & \text{if } i=x \text{ and } j=y; \\ z_{ij} = 0 & \text{otherwise.} \end{cases}$$

$$f(Zq) = a_{xy} - \beta = 1 - \beta \geq 0. \quad (14)$$

Zq is a feasible solution of the quasi seriation problem and $f(Zq)$ is positive.

An optimal solution Z^* of the quasi seriation problem verifies $f(Z^*) \geq f(Z)$ for each feasible solution Z .

So,

$$f(Z^*) \geq f(Zq). \quad (15)$$

(14) and (15) yield to $f(Z^*) \geq 0$. \square

Theorem 1 and 2 yield the result that the concentration of one values inside diagonal blocks of A defined by the optimal solution of a quasi seriation problem is greater than β .

Theorem 3 shows that it is sometimes possible to draw a conclusion about the minimum concentration inside diagonal blocks without computing $f(Z^*)$.

Theorem 3:

Let us suppose matrix A to have at least one "1" value per row and per

column. Let Z^* be an optimal solution of the seriation problem. We define $\text{Sup}(m,n)$ as the greatest of the two values m, n and $\text{Inf}(m,n)$ as the lowest of the two values m and n .

$$\text{Then, } f(Z^*) \geq (1-\beta)\text{Sup}(m,n) - \text{Inf}(m,n). \quad (16)$$

When in addition β satisfies

$$\beta \leq 1 - \frac{\text{Inf}(m,n)}{\text{Sup}(m,n)} \quad (17)$$

$$\text{then, } f(Z^*) \geq 0 \quad (18)$$

and the concentration of one inside diagonal blocks of A will be greater than β .

Example of use of theorem 3.

A is a binary matrix with 1000 rows and 100 columns. We want to solve the block seriation problem under the constraint that the concentration of ones inside diagonal blocks is greater than $1/2$. Give a range of β which satisfies the concentration constraint.

Using (17) we find that if β is lower than 0.9 the concentration inside the diagonal blocks of the maximum will be greater than β . So if we choose β in $[0.5, 0.9]$ we are sure that concentration is greater than $1/2$.

Proof of theorem 3

1. We will first prove theorem 3 in the case $n \geq m$.

We are to find a feasible solution Z_1 to the seriation problem and show that (16) and (17) are verified for Z_1 .

As matrix A is supposed to have at least one "1" value per row and per column, we can, for each row $i=1, \dots, n$, define a column $j(i)$ which verifies $a_{ij(i)} = 1$.

The matrix Z defined as follows,

$$\left. \begin{array}{l} z_{ii} = 1 \quad \forall i \in I \quad / \quad i \leq m \\ z_{ij} = 0 \quad \forall (i,j) \in (I \times J) \quad / \quad i \neq j \quad \text{and } i \leq m \end{array} \right\} (19)$$

$$\left. \begin{array}{l} z_{ij(i)} = 1 \quad \forall i \in I \quad / \quad i > m \\ z_{ij} = 0 \quad \text{otherwise} \end{array} \right\} (20)$$

is a feasible solution of the seriation problem.

Let us now compute $f(Z_1)$.

$$\begin{aligned} f(Z_1) &= \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta) z_{ij} \\ &= \sum_{i=1}^m \left(\sum_{j=1}^m (a_{ij} - \beta) z_{ij} \right) + \sum_{i=m+1}^n \left(\sum_{j=1}^m (a_{ij} - \beta) z_{ij} \right) \end{aligned}$$

$$\sum_{j=1}^m (a_{ij} - \beta) z_{ij} = a_{ii} - \beta \quad \forall i \in I / i \leq m \text{ (see (19))}$$

$$\sum_{j=1}^m (a_{ij} - \beta) z_{ij} - \sum_{j=1}^m (a_{ij} - \beta) = 0 \quad \forall i \in I / i > m \text{ (see (20))}$$

because $a_{ij(i)} = 1 \quad \forall i = 1, \dots, n.$

$f(Z_1)$ becomes,

$$\begin{aligned} f(Z_1) &= \sum_{i=1}^m (a_{ii} - \beta) + \sum_{i=m+1}^n (a_{ij(i)} - \beta) \\ &= \sum_{i=1}^m a_{ii} - \sum_{i=1}^m \beta + \sum_{i=m+1}^n (1 - \beta) \\ &= \left(\sum_{i=1}^m a_{ii} \right) - \beta m + (n-m)(1-\beta) \\ &= \left(\sum_{i=1}^m a_{ii} \right) + (1-\beta)n - m \\ &= \left(\sum_{i=1}^m a_{ii} \right) + (1-\beta)\text{Sup}(m,n) - \text{Inf}(m,n). \end{aligned}$$

As A is a binary matrix $\sum_{i=1}^m a_{ii} \geq 0$ and,

$$f(Z_1) \geq (1-\beta)\text{Sup}(m,n) - \text{Inf}(m,n)$$

That proves (16) in the case $n \geq m$.

The quantity $(1-\beta)\text{Sup}(m,n) - \text{Inf}(m,n)$ is positive when

$$\beta \leq 1 - \frac{\text{Inf}(m,n)}{\text{Sup}(m,n)}.$$

So, if β verifies (16), $f(Z_1)$ is positive. As $f(Z^*)$ is greater than $f(Z_1)$ we can conclude that $f(Z^*)$ is positive. We have proved theorem 3 for $n \geq m$.

2. We will now prove theorem in the case $n \leq m$.

As matrix A is supposed to have at least one "1" value per row and per column, we can, for each column $j=1, \dots, m$, define a row $i(j)$ which verifies $a_{i(j)j} = 1$.

The matrix Z_2 defined as follows,

$$\left[\begin{array}{l} z_{jj} = 1 \quad \forall j \in J / \quad j \leq n \\ z_{ij} = 0 \quad \forall (i,j) \in (I \times J) / \quad i \neq j \quad \text{and } j \leq n \end{array} \right\} (21)$$

$$\left[\begin{array}{l} z_{i(j)j} = 1 \quad \forall i \in I / \quad j > n \\ z_{ij} = 0 \quad \text{otherwise.} \end{array} \right\} (22)$$

is a feasible solution to the seriation problem.

Let us now compute $f(Z_2)$.

$$f(Z_2) = \sum_{i=1}^n \sum_{j=1}^m (a_{ij}^{-\beta}) z_{ij} - \sum_{j=1}^n \sum_{i=1}^m (a_{ij}^{-\beta}) z_{ij} + \sum_{j=1}^n \sum_{i=1}^m (a_{ij}^{-\beta}) z_{ij} + \sum_{j=n+1}^m \sum_{i=1}^n (a_{ij}^{-\beta}) z_{ij}$$

As,

$$\sum_{i=1}^n (a_{ij}^{-\beta}) z_{ij} = a_{jj}^{-\beta} \text{ for } j \leq n \text{ (see(21))}$$

and

$$\sum_{i=1}^n (a_{ij}^{-\beta}) z_{ij} = a_{i(j)}^{-\beta} \text{ for } j > n \text{ (see(22))}$$

$$\begin{aligned} f(Z_2) &= \sum_{j=1}^n (a_{jj}^{-\beta}) + \sum_{j=n+1}^m (1-\beta) \\ &= \left(\sum_{j=1}^n a_{jj} \right) - \sum_{i=1}^n \beta + (m-n) \cdot (1-\beta) \\ &= \left(\sum_{j=1}^n a_{jj} \right) - \beta \cdot n + (m-n) \cdot (1-\beta) \\ &= \left(\sum_{j=1}^n a_{jj} \right) + (1-\beta) \cdot m - n \\ &= \left(\sum_{j=1}^n a_{jj} \right) + (1-\beta) \cdot \text{Sup}(m,n) - \text{Inf}(m,n). \end{aligned}$$

$\left(\sum_{j=1}^n a_{jj} \right)$ is greater than or equal to zero. So,

$$f(Z_2) \geq (1-\beta) \cdot \text{Sup}(m,n) - \text{Inf}(m,n). \quad (23)$$

That proves (16) in the case $n \leq m$.

Now, if β verifies (17) using (23) we get

$$f(Z_2) \geq 0. \quad (24)$$

As $f(Z^*) \geq f(Z_2)$ one get (18) using (24) for the case $n \leq m$. \square

Quality measures of the result.

Let Na_1 and Na_0 be the number of "1" and "0" values of matrix A. Let F_{\max} be the maximum value of $F(Z)$ (see(5)) when the seriation is "perfect" (i.e diagonal blocks of A contain only ones and non diagonal blocks contain only zeroes).

$$F_{\max} = \alpha \cdot Na_1 + \beta \cdot Na_0. \quad (25)$$

Note that F_{\max} is independent of matrix Z generic terms. For α and β given, F_{\max} depends on the two values Na_1 and Na_0 of A. As Na_1 plus Na_0 is equal to the product $m \cdot n$, F_{\max} can be considered as dependent on two of the three values $Na_1, Na_0, m \cdot n$.

One has defined [14] the ratio B:

$$B(Z) = F(Z)/F_{\max} \quad (26)$$

as a quality measure of the result, and B(Z) verifies

$$0 \leq B(Z) \leq 1 \quad (27)$$

We are to show that when Z^* maximizes $F(Z^*)$ then $B(Z^*)$ is always greater than a given value depending only on the number of rows and columns of matrix A.

Using (18) B(Z) can be rewritten thus :

$$B(Z) = \frac{f(Z)}{F_{\max}} + \frac{C}{F_{\max}} \quad (28)$$

Computing C (see (7)) gives

$$C = \beta Na_0 \geq 0 \quad (29)$$

(25) to (28) and (16) give:

$$1 \geq B(Z^*) \geq \frac{(1-\beta) \cdot \text{Su}(m,n) - \text{Inf}(m,n)}{F_{\max}} + \beta \frac{Na_0}{F_{\max}} \quad (30)$$

When in addition, β verifies (17) then (18) and (25) to (29) give :

$$B(Z^*) \geq \beta \frac{Na_0}{F_{\max}} \quad (31)$$

(31) is also verified for an optimal solution of the quasi seriation problem .

Example

Given $\beta = \alpha = 0.5$, $n = m = 100$ $Na_1 = 1000$ let us compute the scale of $B(Z^*)$.

$F_{\max} = 0.5 \times 10\ 000$, (see (25));

$C = 0.5 \times 9000$, (see (29));

$$\frac{(1-\beta) \cdot \text{Su}(m,n) - \text{Inf}(m,n)}{F_{\max}} = \frac{0.5 \times 100 - 100}{0.5 \times 10000} = -0.01,$$

(30) gives

$$1 \geq B(Z^*) \geq -0.01 + 0.9 = 0.89.$$

As we are using $f(Z)$ instead of $F(Z)$ as fonction to maximize, we define a new measure of quality of the result:

$$b(Z) = \frac{f(Z)}{f_{\max}} \quad (32)$$

where f_{\max} is the maximum value of $f(Z)$ when the seriation is perfect.

$$f_{\max} = (1-\beta) Na_1 \quad (33)$$

When Z^* is an optimum of the quasi seriation problem, we deduce an interval of $b(Z^*)$ using (16):

$$\frac{(1-\beta) \cdot \text{Su}(m,n) - \text{Inf}(m,n)}{(1-\beta) Na_1} \leq b(Z) \leq 1. \quad (34)$$

When β verifies (17) then

$$0 \leq b(Z^*) \leq 1 \quad (35)$$

3.2 Case of real data.

Matrix A is now composed of real data varying between Nmin and Nmax. The aim is to solve problem examples 1, 2, 3.

The problem is to find Z maximizing the criterion Fr(Z) under the constraints (C1) to (C7) defined in 3.1 where:

$$\text{Fr}(Z) = \sum_{i=1}^n \sum_{j=1}^m (\alpha a_{ij} z_{ij} + \beta (N_{\min} + N_{\max} - a_{ij})(1 - z_{ij})) \quad (36)$$

with $\alpha + \beta = 1$ and $0 \leq \beta \leq 1$

Fr(Z) can be split into two terms in the following way:

$$\text{Fr}(Z) = \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta(N_{\min} + N_{\max})) z_{ij} + \sum_{i=1}^n \sum_{j=1}^m \beta(N_{\max} + N_{\min} - a_{ij}) z_{ij}$$

$$\text{fr}(Z) = \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - \beta(N_{\min} + N_{\max})) z_{ij} \quad (37)$$

$$\text{Cr} = \sum_{i=1}^n \sum_{j=1}^m \beta(N_{\max} + N_{\min} - a_{ij}) z_{ij} \quad (38)$$

$$\text{Fr}(Z) = \text{fr}(Z) + \text{Cr} \quad (39)$$

Where Z is defined as in 3.1. Only fr(Z) depends on Z. Let us compute Fmax the greatest value of Fr(Z) without the constraints (C1) to (C6). Let frmax be the greatest value of fr(Z) without the constraints (C1) to (C6).

The matrix Zmax defined as follows:

$$\begin{cases} z_{\max_{ij}} = 1 \quad \forall (i,j) \in (I \times J) / a_{ij} > \beta(N_{\min} + N_{\max}); \\ z_{\max_{ij}} = 0 \quad \text{otherwise.} \end{cases}$$

verifies $\text{fr}(Z_{\max}) = \text{frmax}$.

Using (39) we deduce that :

$$\text{Fr}(Z_{\max}) = \text{fr}(Z_{\max}) + \text{Cr} = \text{Fmax}.$$

The corresponding value of previous defined B(Z) ratio is :

$$\text{Br}(Z) = \frac{\text{Fr}(Z)}{\text{Fmax}} \quad (40)$$

It is possible to define a br(Z) which correspond to b(Z):

$$\text{br}(Z) \quad \begin{matrix} \text{fr}(Z) \\ \dots \\ \text{frmax} \end{matrix} \quad (41)$$

The previous linear model defines an ideal A matrix as a matrix for which it is possible, by permutations of rows and columns, to point out a clustering of rows and columns verifying that:

- generic terms of A greater than $\beta(N_{\max}+N_{\min})$ are concentrated in diagonal blocks
- generic terms of A lower than $\beta(N_{\max}+N_{\min})$ are outside diagonal blocks.

To solve example problem 3 with the presented formulation one must choose β in such a way that :

$$\text{limit} = \beta(N_{\max}+N_{\min}).$$

4. THE QUASI-SERiation ALGORITHMS.

4.1 Notations

Given a feasible triplet (X, Y, p) of Dq :

1. For $j = 1, 2, \dots, m$ we consider

$$CR_r(j) = \sum_{i \in X_r} (a_{ij} - \beta(N_{\max} + N_{\min})) \text{ for } r=1, \dots, p ;$$

$$X_0^+(j) = \{ i \in X_0 / a_{ij} - \beta(N_{\max} + N_{\min}) > 0 \} ;$$

$$CR_0^+(j) = \sum_{i \in X_0^+(j)} (a_{ij} - \beta(N_{\max} + N_{\min})) ;$$

$r(j)$, the lowest integer which satisfies

$$CR_{r(j)}(j) = \text{Max}_{r=0, \dots, p} CR_r(j) .$$

(42)

2. For $i = 1, 2, \dots, n$ we consider

$$RC_s(i) = \sum_{j \in Y_s} (a_{ij} - \beta(N_{\max} + N_{\min})) \text{ for } s=1, \dots, p ;$$

$$Y_0^+(i) = \{ j \in Y_0 / a_{ij} - \beta(N_{\max} + N_{\min}) > 0 \} ;$$

$$RC_0^+(i) = \sum_{j \in Y_0^+(i)} (a_{ij} - \beta(N_{\max} + N_{\min})) ;$$

$s(i)$, the lowest integer which satisfies

$$RC_{s(i)}(i) = \text{Max}_{s=0, \dots, p} RC_s(i) .$$

(43)

4.2 Two basic ingredients of the quasi-seriation algorithms.

4.2.1 Quasi-seriation rows assignment procedure.

A feasible solution (X^0, Y^0, p^0) is assumed to be known.
 $X^0 = \{ X_1^0, \dots, X_p^0 \}$ and $Y^0 = \{ Y_1^0, \dots, Y_p^0 \}$, This procedure build then a feasible triplet of Dq , (X, Y, p) , which verifies $fr(X, Y, p) \geq fr(X^0, Y^0, p^0)$.

1. Assign Y, p, Y_0 the following values:
 - 1.1 $p = p^0$;
 - 1.2 $Y = Y^0$; $Y_0 = Y_0^0$.
 End of process 1.
2. For each row $i=1, \dots, n$ do
 - 2.1 Compute $s(i)$ and $RC_{s(i)}(i)$ using (43)
 - 2.2 If $RC_{s(i)}(i) > 0$ and $s(i) > 0$, assign row i to $X_{s(i)}$.
 - 2.3 If $RC_{s(i)}(i) \leq 0$, assign row i to X_0 .
 - 2.4 If $RC_{s(i)}(i) > 0$ and $s(i) = 0$,
 - 2.4.1 Increase p from unity.
 - 2.4.2 Assign row i to X_p .
 - 2.4.3 Cancel the columns of the set $Y_0+(i)$ that belong to the set Y_0 and assign them to Y_p .
 End of 2.4
 End of process 2.

At this point we have clustered I and J in p subsets. Y is a partition of $J - Y_0$, but some of the subsets of X may be empty. In the last case the triplet (X, Y, p) is not an admissible solution to the quasi-serialization problem.

3. If (X, Y, p) is not an admissible solution,
 - 3.1 For $s=1, \dots, p$, if $X_s = \emptyset$, assign columns of Y_s to Y_0 and do $Y_s = \emptyset$.
 - 3.2 Compute the number of non empty sets of X and assign p to this number.
 - 3.3 Number the non empty sets of X and Y from 1 to p .
 End of 3.
- At this point the triplet (X, Y, p) is an admissible solution of the quasi-serialization problem.
4. End of the procedure.

At the end of the procedure $fr(X, Y, p)$ is greater or equal to $fr(X^0, Y^0, p^0)$.

proof:

1. Let us first consider the triplet (X, Y, p) at the end of process 2. We see that the triplet (X, Y, p) is one of the triplets which lead to the greatest value of the criterion knowing (X^0, Y^0, p^0) . $fr(X, Y, p)$ is then greater than or equal to $fr(X^0, Y^0, p^0)$.

2. We shall now demonstrate that process 3. does not decrease the criterion. Using (1) we can write:

$$fr(X, Y, p) = \sum_{l=1}^p \sum_{(i,j) \in (X_k \times Y_k)} (a_{ij} - \beta(N_{max} + N_{min}))$$

When $X_s = \emptyset$ the set of pairs $\{(i,j) \in (X_s \times Y_s)\}$ is empty. As process 3. only modifies the pairs $(X_s = \emptyset, Y_s)$ the criterion will not be modified by process 3. \square

4.2.2 Quasi-seriation columns assignment procedure.

X and p are assumed to be known. X is assumed to be a partition of $I - X_0$.

This procedure constructs then, a feasible triplet (X, Y, p) of Dq . The process is the same as for the row assignment procedure, just replacing X by Y , RC by CR , s by r , row by column and column by row.

1. Initialise X, p, X_0 , with the following values:

1.1 $p = p^0$;

1.2 $X = X^0$; $X_0 = X_0^0$.

End of process 1.

2. For each column $j=1, \dots, m$ do

2.1 Compute $r(j)$ and $CR_{r(j)}(j)$ using (42)

2.2 If $CR_{r(j)}(j) > 0$ and $r(j) > 0$, assign column j to $Y_{r(j)}$.

2.3 If $CR_{r(j)}(j) \leq 0$, assign column j to Y_0 .

2.4 If $CR_{r(j)}(j) > 0$ and $r(j) = 0$,

2.4.1 Increase p from unity.

2.4.2 Assign column j to Y_p .

2.4.3 Cancel the rows that belong to $X_0 + (j)$ from the set X_0 and assign them to X_p .

End of 2.4

End of process 2.

At this point we have clustered I and J in p subsets. X is a partition of $J - Y_0$, but some of the subsets of X may be empty. In this last case the triplet (X, Y, p) is not an admissible solution of the quasi-seriation problem.

3. If (X, Y, p) is not an admissible solution,

3.1 For $r=1, \dots, p$, if $Y_r = \emptyset$, assign columns of X_r to X_0 and do $X_r = \emptyset$.

3.2 Compute the number of non empty sets of X and set p to this number.

3.3 Number the non-empty sets of X and Y from 1 to p .

End of 3.

At this point the triplet (X, Y, p) is an admissible solution of the

quasi-serialization problem.
 4. End of the procedure.

At the end of the procedure $fr(X, Y, p)$ is greater or equal to $fr(X^0, Y^0, p^0)$.

Remark.

The row and column assignment procedure build a feasible triplet (X, Y, p) when initialled with $(X^0 = \emptyset, Y^0 = \emptyset, p^0 = 0)$ which is not a feasible solution.

4.2.3 A basic quasi-serialization algorithm.

We define a qualitative variable "assign" which can take the values "row" or "column". We also define the integer "stop" which takes one of the values 0, 1, 2, 3.

It is assumed that (X^0, Y^0, p^0) , and the value of assign are known.

```

1. Do stop=0.
2. If assign = row,
   | 2.1 Compute  $(X, Y, p)$  using the rows assignment procedure,
   | 2.2 Do assign = column and increase stop from unity.
3. If assign = column,
   | 3.1 Compute  $(X, Y, p)$  with the columns assignment
   |   procedure,
   | 3.2 Do assign = row and increase stop from unity.
4. Test
   | 4.1 If  $fr(X, Y, p) > fr(X^0, Y^0, p^0)$ , do  $(X^0, Y^0, p^0) = (X, Y, p)$ 
   |   and go to 1.
   | 4.2 If  $fr(X, Y, p) = fr(X^0, Y^0, p^0)$ ,
   |   4.2.1 If stop=3 go to 5
   |   4.2.2 If stop<3 go to 2
5. End of process.
```

4.3 How to get equivalent solutions.

There may be several optimal solutions to the quasi serialization problem. In this subsection we present ingredients to find some equivalent solutions knowing an optimal triplet (X^0, Y^0, p^0) . Two triplets (X, Y, p) and (X', Y', p') are said equivalent if $fr(X, Y, p) = fr(X', Y', p')$.

Let (X^0, Y^0, p^0) be an optimal triplet.

Ingredient 1.

For each row $i \in X^0$, we consider $\mathcal{S}(i)$ the set of integers $s_{\ell}(i)$ which verify (44)

$$\left\{ \begin{array}{l} RC_{s_\ell(i)}(i) = \text{Max}_{s=0, \dots, p^0} RC_s(i) \\ RC_{s_\ell(i)}(i) > 0 \\ s_\ell(i) > 0. \end{array} \right. \quad (44)$$

We force the elements of $\mathcal{J}(i)$ to satisfy :

$$s_\ell(i) < s_{\ell+1}(i) \text{ for } \ell=1, \dots, \text{card}(\mathcal{J}(i)) \text{ when } \text{card}(\mathcal{J}(i)) \geq 2 \quad (45)$$

As (X^0, Y^0, p^0) is an optimal triplet the integer $s(i)$ (see (43)) is equal to $s_1(i)$ and $\text{card}(\mathcal{J}(i)) \geq 1$.

We assign row i to any subset X_ℓ $\ell=2, \dots, \text{card}(\mathcal{J}(i))$. We obtain then a new triplet (X, Y^0, p^0) which verifies:

$$\text{fr}(X, Y^0, p^0) = \text{fr}(X^0, Y^0, p^0).$$

If (X, Y^0, p^0) is not a feasible solution, we use process 3. of the rows assignment procedure to build a feasible triplet. Remember that process 3. of the row assignment procedure does not modify the value of the criterion.

We have shown how to build an equivalent solution by considering several equivalent assignments of a row.

We define $n_r(X^0, Y^0, p^0)$ as the number of equivalent triplets different from (X^0, Y^0, p^0) which can be obtained by the use of ingredient 1. Then:

$$n_r(X^0, Y^0, p^0) = \left(\prod_{i \in X^0} \text{card}(\mathcal{J}(i)) \right) - 1. \quad (46)$$

Ingredient 2.

For each column $j \in Y^0$, we consider the set $\mathcal{R}(j)$ of integers $r_\ell(j)$ which verify (47)

$$\left\{ \begin{array}{l} RC_{r_\ell(i)}(j) = \text{Max}_{r=0, \dots, p^0} CR_r(j) \\ CR_{r_\ell(i)}(j) > 0 \\ r_\ell(j) > 0. \end{array} \right. \quad (47)$$

We force the elements of $\mathcal{R}(j)$ to satisfy :

$$r_\ell(j) < r_{\ell+1}(j) \text{ for } \ell=1, \dots, \text{card}(\mathcal{R}(j)) \text{ if } \text{card}(\mathcal{R}(j)) \geq 2$$

As (X^0, Y^0, p^0) is an optimal triplet the integer $r(j)$ (see (42)) is equal to $r_1(j)$ and $r_1(j) \geq 1$.

We assign column j to any subset Y_ℓ $\ell=2, \dots, \text{card}(\mathcal{R}(j))$. We obtain then

a new triplet $(X, Y \neq Y^0, p^0)$ which verifies:

$$fr(X^0, Y, p^0) = fr(X^0, Y^0, p^0).$$

If (X^0, Y, p^0) is not a feasible solution, we use process 3. of the column assignment procedure to build a feasible triplet. Remember that process 3. of the column assignment procedure do not modify the value of the criterion.

We have shown how to build an equivalent solution by considering several equivalent assignments of a column.

We define $n_c(X^0, Y^0, p^0)$ as the number of equivalent different from (X^0, Y^0, p^0) that can be obtained by the use of the previous process. Then,

$$n_c(X^0, Y^0, p^0) = \left(\prod_{j \in Y^0} \text{card}(\mathcal{R}(j)) \right) - 1. \quad (48)$$

Let $n_w(X^0, Y^0, p^0)$ be the number of triplet different from (X^0, Y^0, p^0) obtained by the whole previous process (1. and 2.)

$$n_w(X^0, Y^0, p^0) = n_r(X^0, Y^0, p^0) + n_c(X^0, Y^0, p^0)$$

We could apply the whole previous process to each of the $n_w(X^0, Y^0, p^0)$ solutions equivalent to (X^0, Y^0, p^0) and so on until no new equivalent solution is found.

In practice we observe that this ingredient is very efficient for binary matrices and criteria with $\beta=1/2$.

4.4 A refined quasi-seriation algorithm for binary matrices

The following algorithm is efficient for finding an absolute optimum of the quasi seriation problem when $\beta=1/2$ and when A is a binary matrix. Its efficiency is due to the use of equivalent solution ingredients presented in section 4.3. In section 4.3 we assumed that (X^0, Y^0, p^0) was an optimal triplet. Ingredients 1 and 2 can be applied to each triplet issued from the basic quasi seriation algorithm even if it is a limited optimum.

The algorithm.

We suppose (X^0, Y^0, p^0) to be a feasible solution and the value of the qualitative variable "assign" (see section 4.3)

1. Compute a local or absolute maximum (X^*, Y^*, p^*) with the basic quasi seriation procedure.
2. If a new equivalent triplet (X^0, Y^0, p^0) of (X^*, Y^*, p^*) can be found using ingredient 1, (see 4.3) go to 4 else go to 3.
3. If a new equivalent triplet (X^0, Y^0, p^0) of (X^*, Y^*, p^*) can be found using ingredient 2, (see 4.3) go to 4 else go to 8.
4. If assign = row do assign = column else do assign = row;
5. Compute the basic quasi seriation procedure initialled

with (X^0, Y^0, p^0)
| The quasi-serialization procedure produces a triplet (X, Y, p)
6. If $fr(X, Y, p) > fr(X^0, Y^0, p^0)$ do $(X^*, Y^*, p^*) = (X, Y, p)$ and go
| to 2.
7. If $fr(X, Y, p) - fr(X^0, Y^0, p^0)$ go to 2.
8. End.

5. BLOCK SERIATION ALGORITHMS.

5.1 The basic seriation procedure

It is assumed to know $(X^{\circ}, Y^{\circ}, p^{\circ}) \in D_b$. Remember that D_b is the set of feasible triplets of the block seriation problem.

As in section 4.2.3 we define a qualitative variable "assign" which can take the values "row" or "column".

We also define the integer "IT" which takes one of the two values 0, 1.

The algorithm:

1. Do $IT=0$.
2. If assign = row,
 - 2.1 Do $Y=Y^{\circ}$; $p=p^{\circ}$.
 - 2.2 Assign each row $i=1, \dots, n$ to $X_{s(i)}$. (see 4.1)
At this point we have defined (X, Y, p) . Y is a partition of J in p clusters; but some subsets of X may be empty.
 - 2.3 Do assign = column and go to 4.
3. If assign = column,
 - 3.1 Do $X=Y^{\circ}$; $p=p^{\circ}$
 - 3.2 Assign each column $j=1, \dots, n$ to $Y_{r(j)}$ (see 4.1)
At this point we have defined (X, Y, p) . X is a partition of I in p clusters, but some subsets of Y may be empty.
 - 3.3 Do Assign = row and go to 4.
4. If $(X, Y, p) \in D_b$ and $fr(X, Y, p) > fr(X^{\circ}, Y^{\circ}, p^{\circ})$, do
 $(X^{\circ}, Y^{\circ}, p^{\circ}) = (X, Y, p)$ and go to 1.
5. If $(X, Y, p) \in D_b$ and $fr(X, Y, p) = fr(X^{\circ}, Y^{\circ}, p^{\circ})$,
 - 5.1 If $IT = 0$, do $IT = 1$; $(X^{\circ}, Y^{\circ}, p^{\circ}) = (X, Y, p)$ and go to 2.
 - 5.2 If $IT = 1$, go to End.
6. If $(X, Y, p) \in D_b$ and $fr(X, Y, p) < fr(X^{\circ}, Y^{\circ}, p^{\circ})$, do
 $(X, Y, p) = (X^{\circ}, Y^{\circ}, p^{\circ})$ and go to End.
7. If $(X, Y, p) \notin D_b$,
 - 7.1 If assign = row,
 - 7.1.1 Set p° to the number of non empty subsets of Y
 - 7.1.2 Number the non empty sets of Y from 1 to p° .
At this point X is a partition of I in p° subsets and Y is a partition of J in p° subsets, and $p^{\circ} \neq p$.
 - 7.1.3 Do $Y^{\circ} = Y$ and go to 2.
 - 7.2 If assign = column,
 - 7.2.1 Set p° to the number of non empty subsets of X
 - 7.2.2 Number the non empty sets of X from 1 to p° .
At this point Y is a partition of J in p° subsets and X is a partition of I in p° subsets, and $p^{\circ} \neq p$.
 - 7.2.3 Do $X^{\circ} = X$ and go to 3.
8. End.

Remarks:

With this algorithm $p \leq p^{\circ}$. So, if there is no optimal triplet

(X^*, Y^*, p^*) verifying $p^* > p^0$ then it is sure the algorithm will not find an optimal solution. In the following section we present an algorithm for which p may be greater or lower than p^0 .

5.2 A seriation algorithm based on quasi-seriation.

Quasi-seriation heuristics presented in section 4. can be used as seriation algorithm if the optimum of the quasi seriation problem verifies seriation constraints.

Let (X_q, Y_q, p_q) be the optimum of the quasi seriation problem and (X_s, Y_s, p_s) the optimum of the seriation problem for the same criterion. Then, $fr(X_q, Y_q, p_q)$ is greater than or equal to $fr(X_s, Y_s, p_s)$.

These two remarks lead us to initialise the basic seriation algorithm with the quasi seriation algorithm.

The algorithm.

1. Compute an initial triplet (X^0, Y^0, p^0) with the quasi seriation procedure.
2. If $(X^0, Y^0, p^0) \in D_b$, then go to End.
- 2'. If A is a binary matrix and $\beta=1/2$ and $(X^0, Y^0, p^0) \notin D_b$,
 - 2'.1 Try to build an admissible equivalent triplet of (X^0, Y^0, p^0) using ingredients of section 4.3.
 - 2'.2 If an admissible equivalent triplet is found go to End.
3. If $(X^0, Y^0, p^0) \notin D_b$
 - 3.1 If $X_0^0 \neq \emptyset$ and $Y_0^0 = \emptyset$, do assign=row and go to 4.
 - 3.2 If $X_0^0 = \emptyset$ and $Y_0^0 \neq \emptyset$, do assign=column and go to 4.
 - 3.3 If $X_0^0 \neq \emptyset$ and $Y_0^0 \neq \emptyset$,
 - 3.3.1 Do $p=p^0$;
 - 3.3.2 For each row $i=1, \dots, n$, if $i \in X_0^0$, increase p from unity and assign row i to X_p .
 - 3.3.3 Do $(X^0, Y^0, p^0) = (X, Y^0, p)$ and do assign=column
4. Compute the basic seriation procedure.
5. End.

6.COMPUTING PERFORMANCES.

6.1 Computing time complexity.

The seriation algorithms involve about the same computing time complexity as that of F.Marcotorchino and are lower than H. Garcia and als, because of the use of Marcotorchino's formulation.

Tests were conducted on an Olivetti M280 micro-computer. Programs were written in Turbo Pascal 4.0. Matrices in several sizes were created randomly. Each row and column has a minimum of one "1" value. Global "1" values are at 10% . Tests were done on about one thousand matrices.

We verified that the computing time of one loop increased linearly with the product $m.n$. The average number of loops was 10. Recently we treated an industrial case with more than 20 000 rows and 34 columns in less than 4 minutes.

6.2 Comparison of block seriation algorithms

Let us now compare the presented algorithms with two similar ones (A) presented in [6] and (B) presented in [14].

With algorithm (A) the user gives the number of families and the algorithm gives a solution with the same number of families. Mathematically, if one want to use algorithm (A) to solve the model we presented in this paper one must run the algorithm at least $\text{Inf}(m,n)$ times. In addition, algorithm (A) do not always compute a feasible solution (some pairs $(X_k = \emptyset, Y_k \neq \emptyset)$ or $(X_k \neq \emptyset, Y_k = \emptyset)$ may exist in the given solution).

With algorithm (B) it is useless to give the number of clusters. The algorithm do not always lead to a feasible seriation solution because it cannot decrease the number of families. The number of initial families must be lower than the optimal one.

Algorithms we presented in this paper always lead to a feasible solution. The number of families may increase and decrease during the algorithm.

We note (C) the seriation algorithm using the basic quasi seriation algorithm and (D) the one using the refined quasi seriation algorithm. The matrix example is taken from reference [3]. Chosen criterion is $\beta=1/2$. The result is given for 18 trials, the number of families of the initial partition was one for the first trial, two for the second trial, three for the third one ,etc...

Initial matrix is:

		111111111
		123456789012345678
ville	1	111110000000000000
métropole	2	011111000000000000
cit�	3	101010000000000000
capitale	4	011111000000000000
agglom�rat	5	101010111100000000
centre	6	111111000000000000
village	7	000000111111111000
localit�	8	000010111111000000
bourgade	9	000000101100000000
bourg	10	000000101111010000
trou	11	000000100111100100
patelin	12	000000100111100110
bled	13	000000100011100000
hameau	14	000000110000010000
commune	15	000010101100001001
coin	16	000000010011100100
pays	17	001000001111101010
municipalit�	18	100000000000001001

Algorithms	(A)	(B)	(C)	(D)
number of feasible solution	3.0	2.0	18.0	18.0
best seriation criterion value	19.5	23.5	28.0	28.0
worst seriation criterion value	-64.0	23.5	22.0	28.0

Table 1: Comparison of four block-seriation algorithms

Discussion:

- the number of obtained seriation of (C) and (D) are very high compared to (A) and (B). Initial solutions were chosen randomly and were the same for each algorithm.
- (C) found 22 as criterion value once. The others values were distributed the following way:
 - 1 time 26.00
 - 3 times 27.50
 - 4 times 28.00 which is the best solution for this matrix.
- (D) for this matrix the ingredient of equivalent solution was very efficient.
- The quasi seriation loop never gave a seriation as result. Starting with all the rows and columns unclassified ($X_0=Y_0=\emptyset$), the algorithm (D) finds the optimum quasi seriation solution each time, despite the line and column numbering and the first assignment. The ingredient of equivalent solutions was helpful in obtaining this

result.

The quasi seriation loop is another method for solving certain particular cases of quadri-decomposition models. [4]

We made trials on numerous matrices and found out that results obtained in previous example seem to be general except that algorithm (B) resulted in a seriation more often than (A) algorithm.

For binary matrices chosen randomly we often found more than 10^6 equivalent solutions only by using the basic quasi seriation algorithm. For industrial cases with several thousand rows, with no more than 10% of ones values we often found several hundred equivalent solutions with the basic quasi seriation algorithm even when $br(Z)$ ratio was greater than 0.80. Using the equivalent solution ingredient the number of equivalent solutions generally collapses.

7. EXAMPLE OF USES IN PRODUCTION MANAGEMENT.

Section 3 and section 4 algorithms can be used as a method of classification for factory objects such as products, process-plannings, product price estimates etc.. The following pages describe some examples of its application in solving certain manufacturing unit formation problems.

7.1 Analysis of the binary machine-parts incidence matrix. [16]

Matrix $A[a_{ij}]$ is a binary element. Rows are parts, and columns pools of machines. A pool is a group of machines performing the same task(s). Element a_{ij} is equal to 1 when a pool of machines j belongs to part i 's process planning.

The number of "1" in such a matrix in a range of $0.15mn$ to $0.35mn$. (mn is the number of rows multiplied by the various number of columns of $A[a_{ij}]$).

The real problem is more complex than the linear model developed in [6]. That's why several trials are necessary.

Some rules obtained through experiments.

A good seriable matrix is a matrix in which the optimal solution is composed of groups with a high density of "1" inside diagonal blocks and less than 10% of "1" outside, whatever the β value.

Usually the number of families decreases with β .

minimizing family size constraints by hierarchical optimization

In a production management problem matrix A size is greater than 10^5 and the number of "1" is about 20%. For $\beta=1/2$ the number of families optimizing $f(Z)$ may be more than 50. The natural number of families obtained by solving section 2 and 3 block seriation models gives a lot of information on data typology to the cell designer; but

the result is not always satisfactory in reality. For instance, the size of some cells (families of pools) may not be great enough for economic reasons.

One way to obtain a solution with fewer families is to decrease β value as suggested by experience rules. Figure 5 shows the obtained hierarchy for matrix of figure 6. Figure 7 represents the number of family variations with β . Decreasing β is of no help in obtaining solutions between 2 and 5 families.

The natural number of families that optimizes the criterion is too large for $\beta > 0,6$ and not large enough for $\beta < 0,6$.

Another way to obtain solutions with several numbers of families is to compute a hierarchy of partitions. This is possible using the section 5 algorithm. Figures 8 to 11 describe the process in this academic example where the initial matrix is represented by figure 6. After the block seriation algorithm, the natural solution is obtained in 11 families (figure 8). This binary matrix is summed up on figure 9; each number represents the number of "1" values of a block in the reorganized A matrix. Table (figure 9) can be interpreted as an A matrix described in example 3 (introduction). By applying the seriation algorithm to this matrix, the 11 families are grouped into larger ones in order to minimize non zero values outside the new blocks. Figure 11 is the binary representation of the figure 10 result. So now figure 10 is the initial matrix for the next aggregation level and so on.. The whole hierarchy is given in figure 12.

A third way to obtain solutions with several numbers of families is to compute the equivalent solutions if they exist.

If the constraint of keeping previously level families included in the new larger one (hierarchy of partitions) does not exist, the different level results can eventually be optimized by the basic seriation procedure (in which the number of families cannot increase starting with the reorganized binary matrix. This last method is useful but a good initial partition is necessary. This is obtained through the hierarchy of partitions. If one starts with randomly processed partitions, the obtained solution is often given in one family or the criterion value is poor.

Figure 13 sums up the results for the three aggregation methods.

7.2 Analysis of the part-loads matrix.

Some authors prefer to study the cell formation problem by analyzing the part load matrix. Load is the time a machine is working. Data is given in positive values with about 20% of non zeroes. The objective function is $fr(Z)$. Minimum value is given by solving $limit > \beta(N_{max}+N_{min})$ (see section 3.2) limit is the lower value expected in diagonal blocks through the objective function.

We compared the solution obtained by the proposed heuristic to solutions obtained by J.A Ventura and alls in [19]. Matrix treated in [19] are most of proposed matrices in the production research literature in the last twenty years. J.A Ventura and als algorithm is compared to the two algorithms of Kumar and als [11] and proves that their algorithm is far superior to Kumar's algorithm. Our algorithm

reveals each time a better or the same solution as those presented in [19].

For S1 and S2 (table2) there are five values greater than limit outside the diagonal blocks. For S3 and S4 there is only one value greater than limit outside diagonal blocks.

7.3 Analysis of the traffic matrix.

Some papers deal with a symmetrical matrix which in generic terms is a measure of the traffic between machines. For instance a_{ij} could be the number of parts moving between machine i and machine j . The set of rows is identical to the set of columns. The problem consists in a block seriation problem with the additional constraint that the partitions of rows and columns are the same. This additional constraint is costless to implement using previous presented algorithms. One need only define the set of rows in the same variable as the set of columns. The presented algorithm then becomes equivalent to the heuristic presented in [13] as proved in [8]. In [5] one can find a comparison between this algorithm and algorithms written in order to solve this specific case.

8. CONCLUSION

The presented block seriation method is helpful in analysing industrial data. The presented heuristics solve the block seriation problem even in the case of symmetrical matrices, they always produce a feasible solution to the seriation problem and sometimes, are able to propose several maximum if they exist. We pointed out the linkage between the proposed function to maximize and the concentration of given values of generic terms inside the diagonal blocks allowing new approaches in the practical identification of the manufacturing cells [19]. They are currently implemented on manufacturing cell design system S.A.F.I.R [7] and on group technology data analysis systems G.T.S .

References.

- [1] E. Barnes
" An algorithm for partitioning the nodes of a graph ",SIAM
Journal of Algorithms and Discrete Methods, Vol3, No4,pp
541-550 1982.
- [2] Ch. Bedecarrax
" Quadri-decomposition en analyse relationnelle et applications à la
sériation"
IBM Paris Scientific Center Technical Report F117 1987.
- [3] Ch. Bedecarrax
" Quadri-décomposition: modèle général."
IBM Paris Scientific Center Technical Report F132 1988.
- [4] Ch. Bedecarrax , I. Warnesson
" Relational analysis and dictionaries "
Applied Stochastic Models and Data Analysis vol5 p131-151 1989.
- [5] P. Beziat ; P. Massote
"Plant layout optimization tool using group technology."
Colloque International sur les méthodes de bloc sériation et
applications " Strasbourg 3-4-5 avril 1990 Recueil des conférences.
ENSAIS 24, bd de la victoire 67084 Strasbourg Cedex France
- [6] H. Garcia ; J.M. Proth
" Group Technology in Production Management the Short Horizon
Planning Level"
Journal of Applied Stochastic Models and Data
Analysis, Vol11, N°1, pp.25-34, J.Wiley(1985)
- [7] R. De Guio
" Contribution à l'organisation d'ateliers en îlots de fabrication".
Thèse de doctorat. Université Louis Pasteur Strasbourg janvier 1990.
Chapitre IV
- [8] R. De Guio; B. Mutel
" Algorithmes et critères de seriation d'une matrice non symétrique.
Application a la formation d'îlots de fabrications"
Colloque International sur les méthodes de bloc sériation
et applications " Strasbourg 3-4-5 avril 1990 Recueil des
conférences.
ENSAIS 24, bd de la victoire 67084 Strasbourg Cedex France
- [9] G. Harhalakis, Nagi R, and J.M Proth
" An efficient heuristic in manufacturing cell formation for
group technology applications"
International Journal of Production Research, Vol
28, No1, pp185-188 1990
- [10] J.R King,
" Machine-component grouping using ROC algorithm"
International Journal of Production Research, Vol 18, pp
213-231, 1980
- [11] K.R Kumar, A. Kusiak, A. Vannelli

- " Grouping of parts and components in flexible manufacturing systems."
European Journal of Operational Research, 24, pp.387-397
- [12] A. Kusiak and W.S Chow
" Efficient Solving of the Group Technology Problem"
Journal of Manufacturing Systems, Vol6, No 24, 1987
- [13] F. Marcotorchino
" Agregation des similarités en classification automatique. "
Thèse d'Etat. Université de Paris VI (1981)
- [14] F. Marcotorchino
" A unified approach of the Block-Seriation Problems"
Journal of Applied Stochastic Models and Data Analysis
Vol 3, N° 2, J. Wiley (1987).
- [15] H. Messetfa
" Partionnement d'un graphe en classes fixées."
Colloque International sur les méthodes de bloc sériation
et applications " Strasbourg 3-4-5 avril 1990 Recueil des
conférences.
ENSAIS 24, bd de la victoire 67084 Strasbourg Cedex France
- [16] B. Mutel, R. De Guio
" Recognition of cellullar manufacturing under management
constraints"
C.I.R.P. annals , vol 38/1 August 1989.
- [17] L.E Stanfel
" A successive approximations method for a cellular
manufacturing problem ".
INRIA annual meeting report , pp 523-536 1987.
- [18] A. Vanelli, and K. Ravi Kumar
"A method for finding minimal bottle-neck cells for grouping
part-machine families"
International Journal of Production Research, Vol
24, No2, pp387-400, 1986.
- [19] J.A. Ventura, F.Frank Chen, and Chih-Lang Wu
" Grouping parts and tools in flexible manufacturing systems
production planning"
International Journal of Production Research, 1990, Vol.
28, No. 6, pp.1039-1056

		J				
		┌───┴───┐				
i \ j		1	2	3	4	5
1	1	1	0	1	0	1
2	1	0	0	1	0	0
3	0	1	1	0	0	0
4	0	1	0	1	1	0
5	1	0	0	0	0	0
6	1	0	0	0	1	0
7	0	0	0	0	0	1
8	0	0	1	1	1	0
9	1	1	0	0	0	0
10	0	1	1	1	0	0

Figure 1:matrix A

		J				
		┌───┴───┐				
i \ j		1	2	3	4	5
1	1	0	0	1	0	1
2	0	0	0	1	0	0
3	0	1	0	0	0	0
4	0	0	0	1	0	0
5	1	0	0	0	0	0
6	0	0	1	0	1	0
7	0	0	0	0	0	0
8	0	0	1	0	1	0
9	0	1	0	0	0	0
10	0	0	0	1	0	0

Figure 2: Matrix Z defines generic terms of A which belong to a diagonal block.

$$\begin{array}{r}
 i \backslash j \\
 7 \\
 1 \\
 6 \\
 8 \\
 2 \\
 4 \\
 10 \\
 3 \\
 9 \\
 5
 \end{array}
 \begin{array}{c}
 \left[\begin{array}{cccc}
 35 & 4 & 2 & 1 \\
 00 & 0 & 0 & 0 \\
 11 & 0 & 0 & 0 \\
 11 & 0 & 0 & 0 \\
 11 & 0 & 0 & 0 \\
 00 & 1 & 0 & 0 \\
 00 & 1 & 0 & 0 \\
 00 & 1 & 0 & 0 \\
 00 & 0 & 1 & 0 \\
 00 & 0 & 1 & 0 \\
 00 & 0 & 0 & 1
 \end{array} \right]
 \end{array}
 \begin{array}{l}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} X_0 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} X_1 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} X_2 \\
 \left. \begin{array}{l} \\ \\ \end{array} \right\} X_3 \\
 \left. \begin{array}{l} \\ \end{array} \right\} X_4
 \end{array}$$

$Y_1 \quad Y_2 \quad Y_3 \quad Y_4$

Figure 3: Matrix Z permuted in order to point out diagonal blocks full with ones and non diagonal blocks full with zeroes. ($Y_0 = \emptyset$)

$$\begin{array}{r}
 i \backslash j \\
 7 \\
 6 \\
 8 \\
 2 \\
 4 \\
 10 \\
 3 \\
 9 \\
 5
 \end{array}
 \begin{array}{c}
 \left[\begin{array}{cccc}
 Y_1 & Y_2 & Y_3 & Y_4 \\
 35 & 4 & 2 & 1 \\
 01 & 0 & 0 & 0 \\
 11 & 0 & 0 & 1 \\
 01 & 0 & 0 & 1 \\
 11 & 1 & 0 & 0 \\
 00 & 1 & 0 & 1 \\
 01 & 1 & 1 & 0 \\
 10 & 1 & 1 & 0 \\
 10 & 0 & 1 & 0 \\
 00 & 0 & 1 & 1 \\
 00 & 0 & 0 & 1
 \end{array} \right]
 \end{array}
 \begin{array}{l}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} X_0 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} X_1 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} X_2 \\
 \left. \begin{array}{l} \\ \\ \end{array} \right\} X_3 \\
 \left. \begin{array}{l} \\ \end{array} \right\} X_4
 \end{array}$$

Figure 4: Matrix A permuted like Z in figure 3.

Figure 5

Number of intra-block

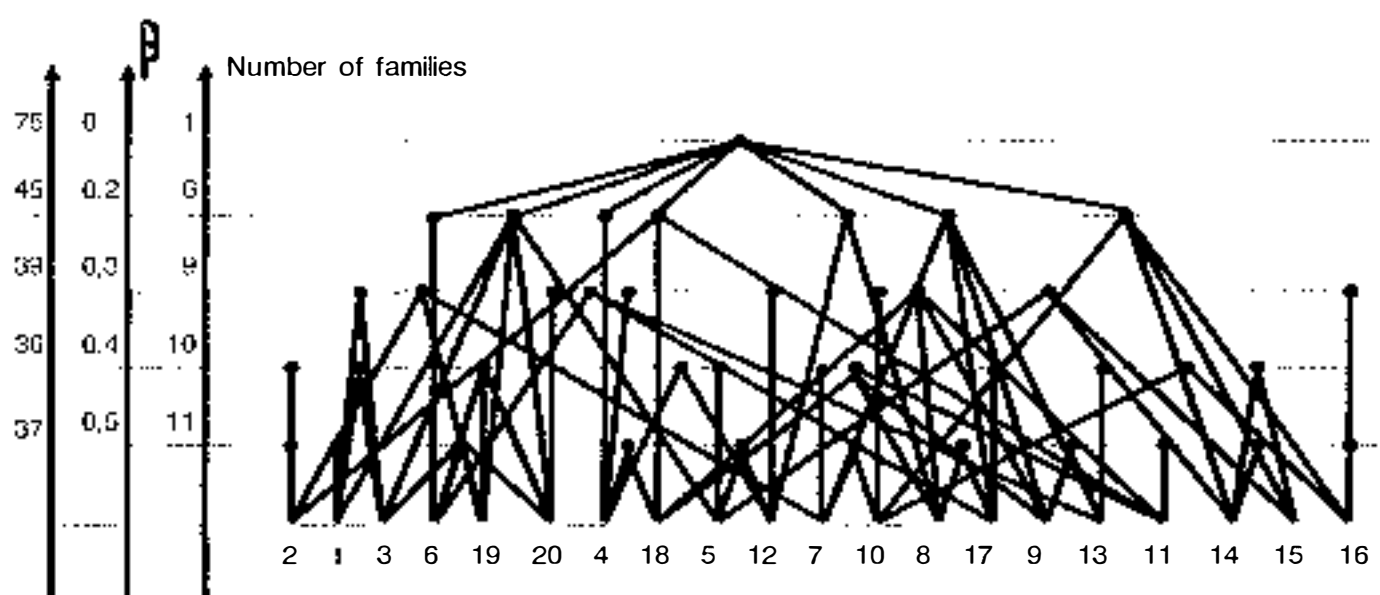


Figure 7

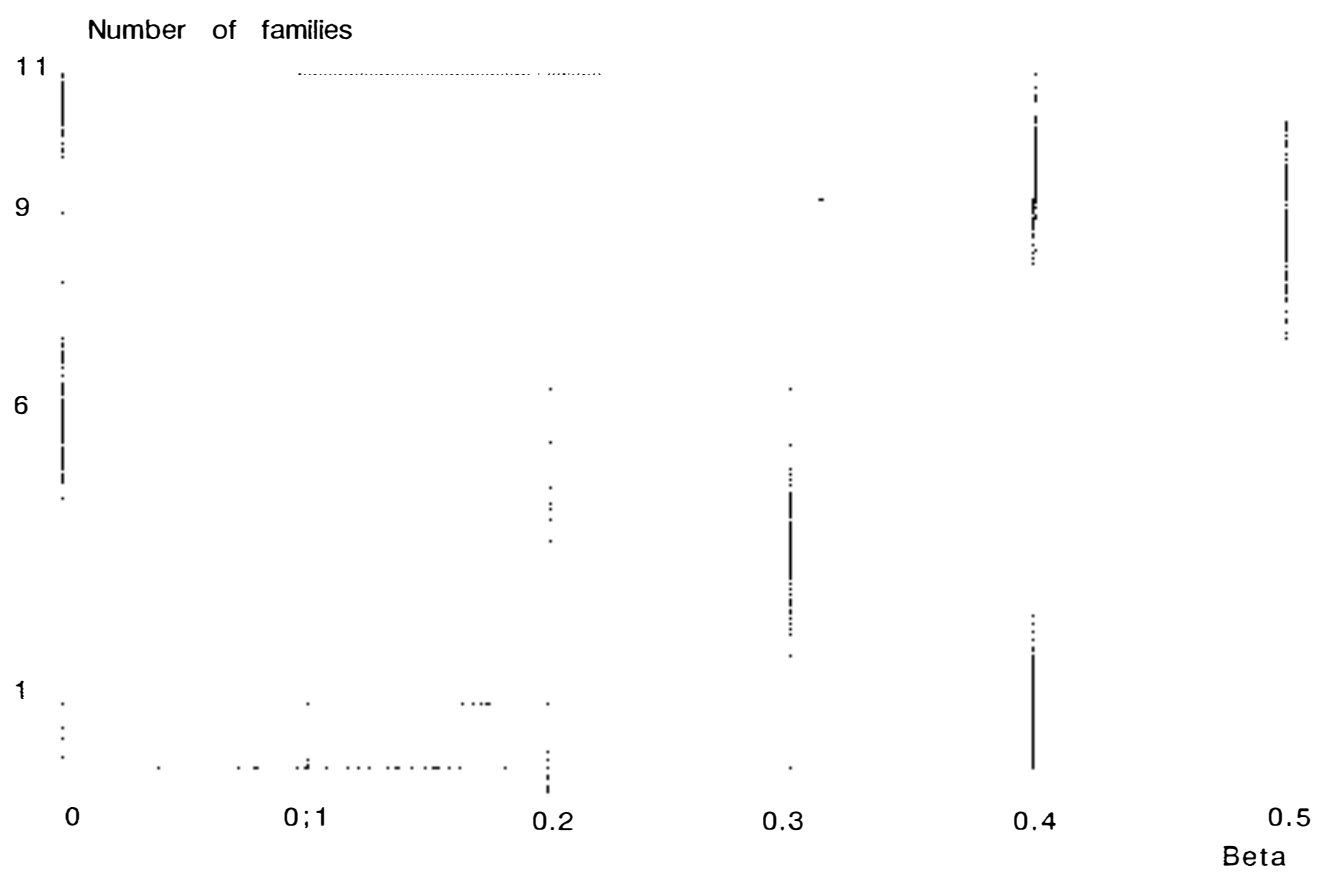


Figure 10: Application of the seriation algorithm
to matrix of figure 9

	1	2	3	3	4	5	6	6	7	8	9	No. column families
	1:	2:	3	5:	4:	6:	7	1:	8:	9:	0:	No. columns
1 1	5:	:	:	:	:	:	:	:	:	:	:	
1 11	1:	:	:	:	:	:	1:	:	:	:	:	
2 2	:	5:	:	:	:	:	:	:	:	:	:	
2 5	1:	3:	2:	1:	:	1	:	:	:	:	:	
3 3	2:	2:	4	2:	:	:	:	1:	1:	1:	1:	
4 4	1:	1:	:	3:	:	1	:	:	:	:	:	
5 6	1:	1:	:	:	3:	1	:	:	:	:	:	
6 7	:	1:	1	:	1:	:	6	1:	1:	1:	:	
7 8	1:	:	:	:	1:	1	:	2:	1:	:	:	
8 9	:	1:	:	:	:	1	:	:	2:	:	:	
9 10	1:	1:	:	1:	:	1	:	:	:	:	4:	

Figure 12: Hierarchic classification of matrix A with the seriation algorithm

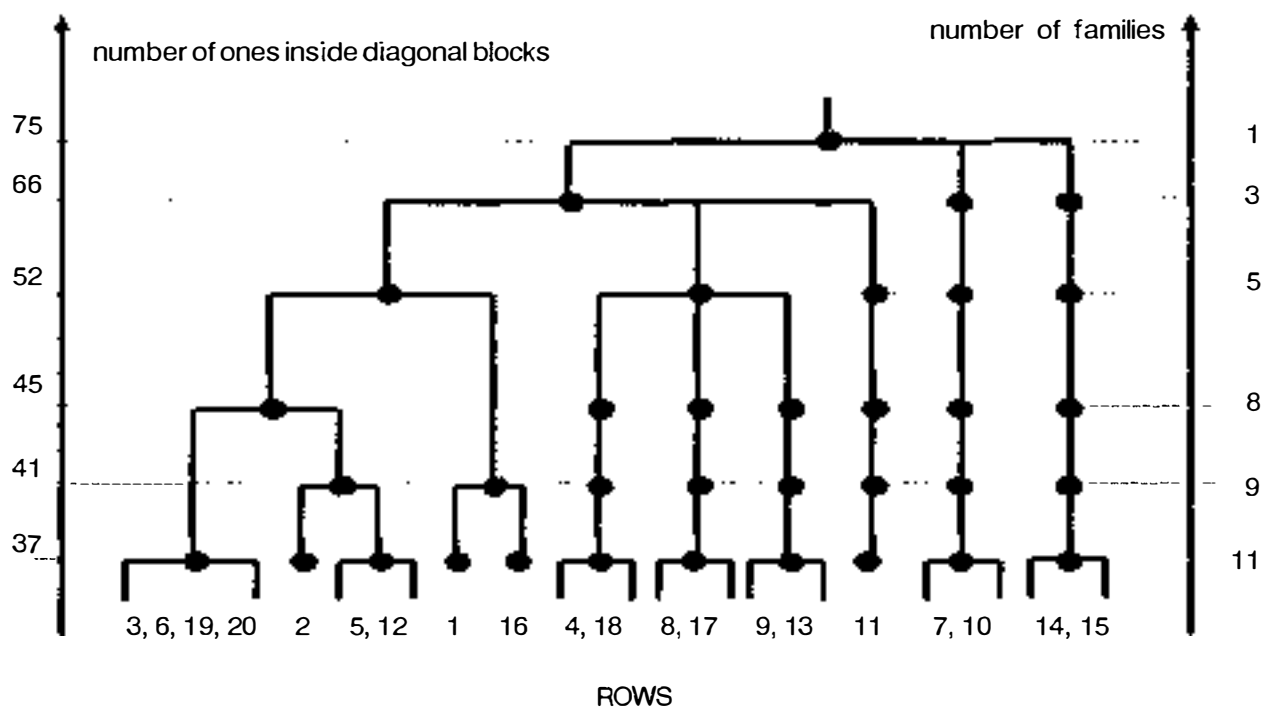
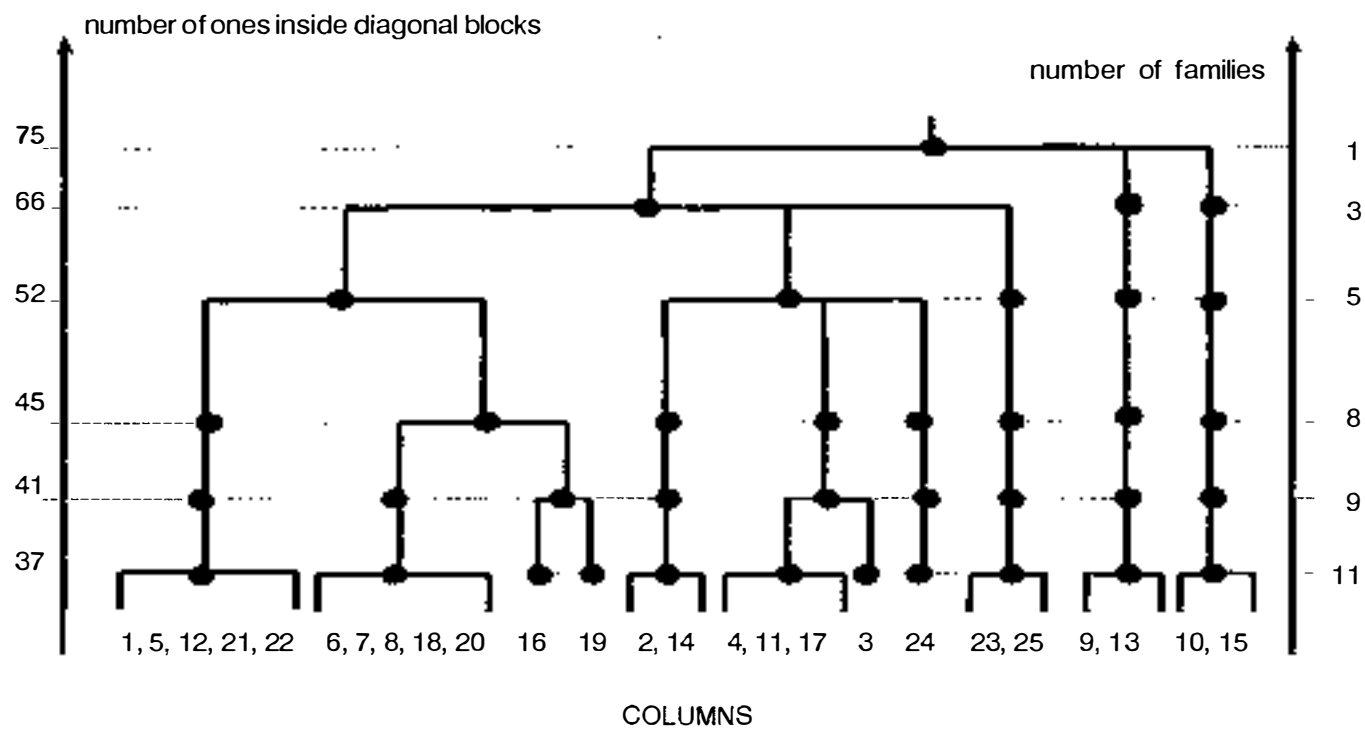
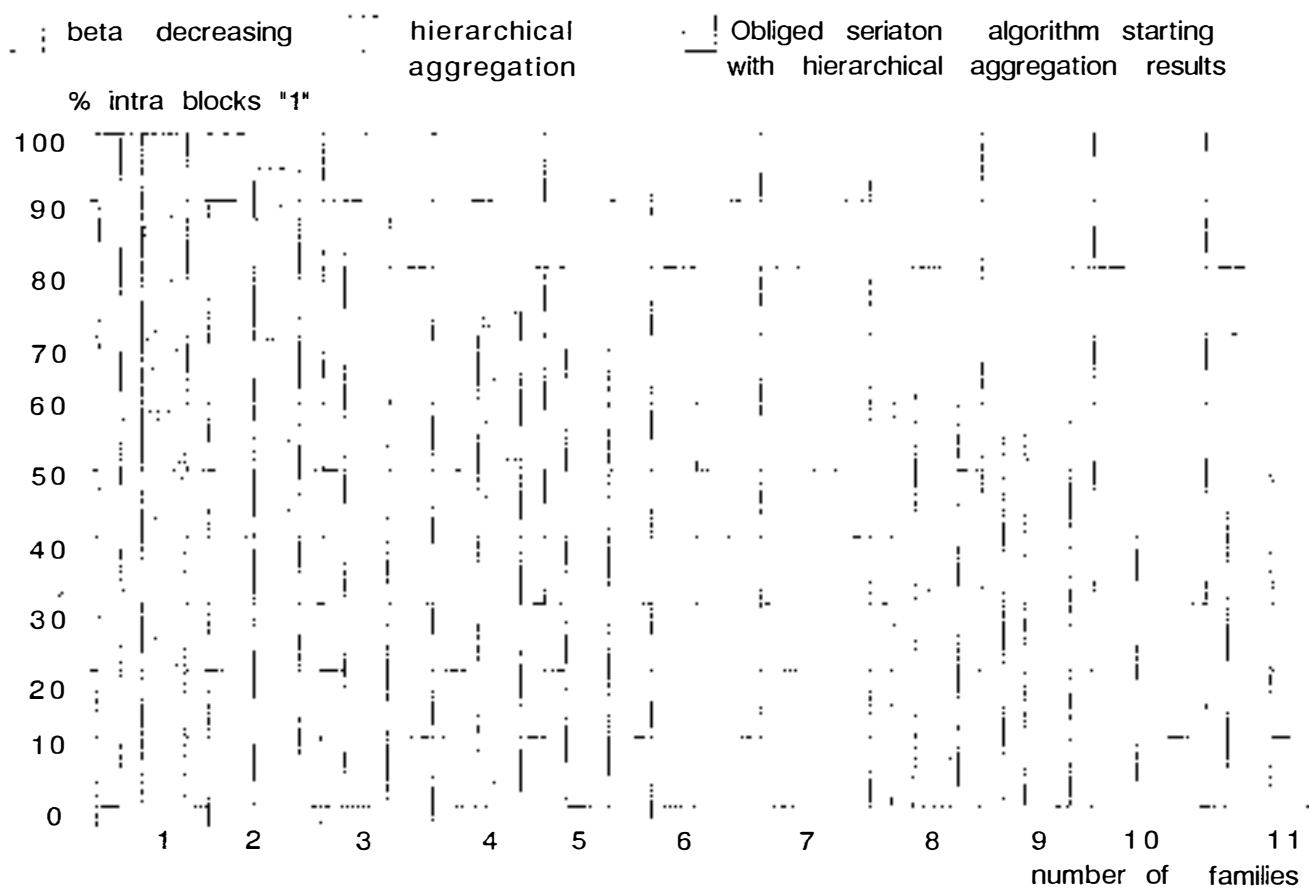


Figure 13: Comparison of three aggregation methods



i\j	1	2	3	4	5	6	7
1	7	16	7	0	5	0	0
2	0	0	2	0	0	10	2
3	0	0	0	29	0	0	19
4	76	10	0	0	48	5	0
5	0	0	0	76	0	0	43
6	5	0	14	71	0	10	0
7	14	24	29	0	10	0	0
8	14	0	0	0	0	24	0
9	3	0	0	38	0	0	7
0	8	0	29	0	0	10	0
1	86	10	0	0	0	0	0
2	0	8	0	0	5	0	0
3	5	0	86	100	0	33	0

Figure 14: Machine parts loads matrix.

Type of seriation demanded	β	fr(Z)	br(Z)	solution	limit
quasi seriation	0.1	537.0	0.60	S1	10
seriation	0.1	535.0	0.60	S2	10
quasi seriation	0.5	159.0	0.32	S3	50
seriation	0.5	-208.0	-0.42	S4	50

Table 2: Obtained solutions for figure matrix of figure 14 initialized with $X_0=Y_0=\emptyset$.

$$\begin{aligned}
 S1 & \left\{ \begin{array}{l} X_0 = \{2, 10, 12\} ; X_1 = \{1, 7\} ; X_2 = \{3, 5, 6, 9, 13\} ; X_3 = \{4, 11\} ; X_4 = \{8\} ; \\ Y_0 = \{\} ; Y_1 = \{2\} ; Y_2 = \{3, 4, 7\} ; Y_3 = \{1, 5\} ; Y_4 = \{6\} \end{array} \right. \\
 S2 & \left\{ \begin{array}{l} X_0 = \{\} ; X_1 = \{1, 7, 12\} ; X_2 = \{3, 5, 6, 9, 13\} ; X_3 = \{4, 11\} ; X_4 = \{2, 8, 10\} \\ Y_0 = \{\} ; Y_1 = \{2\} ; Y_2 = \{3, 4, 7\} ; Y_3 = \{1, 5\} ; Y_4 = \{6\} \end{array} \right. \\
 S3 & \left\{ \begin{array}{l} X_0 = \{1, 2, 3, 7, 8, 9, 10, 12\} ; X_1 = \{4, 11\} ; X_2 = \{5, 6, 13\} \\ Y_0 = \{2, 3, 5, 6, 7\} ; Y_1 = \{1\} ; Y_2 = \{4\} \end{array} \right. \\
 S4 & \left\{ \begin{array}{l} X_0 = \{\} ; X_1 = \{4, 11\} ; X_2 = \{3, 5, 6, 9, 13\} ; X_3 = \{1, 12\} ; X_4 = \{7, 10\} ; \\ X_5 = \{2, 8\} \\ Y_0 = \{\} ; Y_1 = \{1, 5\} ; Y_2 = \{2\} ; Y_3 = \{4\} ; Y_4 = \{3\} ; Y_5 = \{6, 7\} \end{array} \right.
 \end{aligned}$$