



Familles de produits et ilots de fabrication: le cas de machines multiples

H. Garcia, B. Mutel, Jean-Marie Proth

► To cite this version:

H. Garcia, B. Mutel, Jean-Marie Proth. Familles de produits et ilots de fabrication: le cas de machines multiples. RR-0469, INRIA. 1985. inria-00076085

HAL Id: inria-00076085

<https://hal.inria.fr/inria-00076085>

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.

IRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt

BP 105

78153 Le Chesnay Cedex
France

Tél. : (1) 39 63 55 11

Rapports de Recherche

N° 469

**FAMILLES DE PRODUITS
ET ILOTS DE FABRICATION :
LE CAS DE MACHINES MULTIPLES**

**Hervé GARCIA
Bernard MUTEL
Jean-Marie PROTH**

Décembre 1985

**FAMILLES DE PRODUITS
ET ILOTS DE FABRICATION :
LE CAS DE MACHINES MULTIPLES**

GARCIA Hervé *

MUTEL Bernard *

PROTH Jean-Marie **

* : LAEI, IUT, Ile du Saulcy, 57000 METZ

** : INRIA, Château du Montet, Rue du Doyen Roubault, 54500 VANDOEUVRE

Ce travail est financé par l'Agence de l'Informatique.

ABSTRACT :

We consider a set of part types and a set of machines.

Every part belonging to a given part type is manufactured following the same sequence of tasks, i.e. the same working process. Some tasks can be performed using several machines which are identical or not.

Each machine is able to perform one and only one task. We are looking for a partition of the set of part types in part families, and for a partition of the set of machines in production subsystems in such a way that :

1. the number of part families and the number of production subsystems are equal,
2. one (and only one) part family corresponds to every production subsystem,
3. one (and only one) production subsystem corresponds to every part family,
4. the previous partitions minimize a criterion which is the weighted sum of the number of times a machine is used to perform a part belonging to a part family which does not corresponds to the subsystem containing the machine, and the number of times a machine is not used to perform a part belonging to the part family which corresponds to its production subsystem.

An heuristic algorithm is proposed and a numerical example is given.

RESUME :

Nous considérons un ensemble de types de pièces et un ensemble de machines. Chaque pièce appartenant à un type de pièces donné est fabriquée en suivant la même séquence de tâches, c'est-à-dire la même gamme de fabrication.

Certaines tâches peuvent être exécutées en utilisant diverses machines identiques ou non.

Chaque machine est capable d'exécuter une tâche et une seule. Nous cherchons une partition de l'ensemble des types de produits en familles de produits, et de l'ensemble des machines en îlots de fabrication de telle sorte que :

1. le nombre de familles de produits et le nombre d'îlots de fabrication soient égaux,
2. une (et une seule) famille de produits correspond à chaque îlot de fabrication,
3. un (et un seul) îlot de fabrication correspond à chaque famille de produits,
4. les partitions précédentes minimisent un critère qui est la somme pondérée du nombre de fois qu'une machine est utilisée pour transformer une pièce appartenant à une famille de pièces qui ne correspond pas à l'îlot auquel appartient la machine, et le nombre de fois qu'une machine n'est pas utilisée pour transformer une pièce appartenant à la famille qui correspond à son îlot.

Une heuristique est proposée et un exemple numérique est donné.

I - INTRODUCTION

Le cas où une tâche donnée ne peut être exécutée que par une machine a déjà été traité (voir [8]). Pour cela, nous avons considéré une matrice de 0 et de 1 comportant autant de lignes que de types de produits et autant de colonnes que de machines, et définie de la manière suivante : l'élément de la ligne i et de la colonne j est égal à 1 si les produits de type i ont à subir une transformation sur la machine j et 0 sinon. Chaque ligne de cette matrice a été complétée par le poids associé au type de produit correspondant. Ce poids représente le nombre moyen de produits de ce type à fabriquer durant une période d'étude. Nous avons alors cherché, par permutation des lignes et des colonnes, à faire apparaître un nombre donné de blocs non empiétants, qui maximisent le nombre de 1 situés à l'intérieur des blocs plus le nombre de 0 situés à l'extérieur, après pondération : il s'agit du critère 4 donné dans le résumé.

Les tests effectués montrent que cette approche est plus performante que celles de Mc Auley [1] et J.R. King [2]. Elle permet en effet de tenir compte de l'importance des séries de produits.

Le présent travail généralise l'approche précédente au cas où une opération peut être exécutée par plusieurs machines différentes.

Ce cas, très proche de la réalité industrielle, n'a été que très peu étudié. Dans les rares études connues, les auteurs se contentent de répartir des machines identiques dans des îlots par essais successifs (voir [3] et [4]). A notre connaissance, seuls les travaux de Y. Lemaire et B. Mutel ([6]) et de L.E. Stanfel ([5]) tentent d'intégrer la répartition des machines dans l'algorithme de classification. Dans la publication [6], on part de la méthode de classification dite des "nuées dynamiques" et l'on contraint le système de telle sorte qu'il y ait ajustement des charges induites par les produits aux capacités des machines. Dans [5], on utilise une méthode de programmation linéaire en 0-1. Le critère consiste à minimiser les trajets inter-îlots pour les produits. Dans cette approche, il faut fixer le nombre maximum de machines par cellules et il n'est pas tenu compte des charges.

Dans ce papier nous proposons une heuristique de recherche d'ilots de fabrication et de familles de produits qui prend en compte les opérations dans l'ordre croissant des machines qui sont en mesure de les exécuter. Nous illustrons cet algorithme par un exemple numérique.

II - POSITION DU PROBLEME ET DEFINITIONS

Soit n le nombre de produits et m le nombre d'opérations.

Nous connaissons :

$$T = [t_{ij}] ; i=1, \dots, n ; j=1, \dots, m \quad (1)$$

où t_{ij} est le temps de passage d'un produit de type i sur une machine de référence capable d'exécuter l'opération j . Lorsqu'un seul type de machine est en mesure d'exécuter l'opération j , c'est une machine de ce type qui est prise comme machine de référence.

Si M_j est la machine de référence pour l'opération j et si $M_j^1, M_j^2, \dots, M_j^{k_j}$ sont les machines susceptibles d'effectuer l'opération j , nous désignons par :

$$\alpha_j^k, k=1, \dots, k_j \quad (2)$$

le rapport des cadences (ou vitesses, ou débits) de M_j^k et M_j .

Soit

$$\theta_{ij}^k ; i=1, \dots, n ; j=1, \dots, m ; k=1, \dots, k_j \quad (3)$$

le temps de passage moyen d'un produit de type i sur machine M_j^k pour y subir l'opération j . En fait : $\theta_{ij}^k = t_{ij} * p_{ij}^k / \alpha_j^k$, où p_{ij}^k est la proportion de produits i passant sur k pour y subir j .

De plus :

$$\mu_i ; i=1, \dots, n \quad (4)$$

est le poids attaché au produit de type i . Il représente le nombre de produits de ce type à fabriquer par période de travail.

La relation suivante exprime que toutes les opérations sont exécutées :

$$\sum_{k=1}^{k_j} \alpha_j^k \theta_{ij}^k = t_{ij} ; i=1, \dots, n ; j=1, \dots, m \quad (5)$$

Si τ est la période de travail considérée, les inégalités suivantes indiquent que la charge attribuée aux différentes machines ne les entraîne pas à travailler au delà de la période τ :

$$\sum_{i=1}^n \mu_i \theta_{ij}^k \leq \tau ; j=1, \dots, m ; k=1, \dots, k_j \quad (6)$$

On suppose en outre connu, pour chaque machine $M_{j,k}^k \in \{1, 2, \dots, k_j\}$ un seuil de rentabilité $x_j^k \in [0, 1]$. Ce seuil de rentabilité est le pourcentage du temps de la période de travail durant laquelle la machine doit être occupée pour être rentable.

Dans un but de rentabilité, les inégalités suivantes doivent être vérifiées :

$$\sum_{i=1}^n \mu_i \theta_{ij}^k \geq x_j^k \tau ; j=1, \dots, m ; k=1, \dots, k_j \quad (7)$$

Compte tenu des définitions précédentes,

$$M = \sum_{j=1}^m k_j \text{ est le nombre total de machines} \quad (8)$$

Soit q le nombre de familles de produits et d'îlots de fabrication souhaités.

Nous recherchons les θ_{ij}^{*k} ; $i=1, \dots, n$; $j=1, \dots, m$; $k=1, \dots, k_j$, vérifiant (5), (6) et (7), une partition $\mathcal{A}^* = \{A_1^*, \dots, A_q^*\}$ de l'ensemble des n types de produits en q familles et une partition $\mathcal{B}^* = \{B_1^*, \dots, B_q^*\}$ de l'ensemble des M machines en q îlots de fabrication de telle sorte que :

$$\begin{aligned} & h \sum_{(i,w) \in \bigcup_{s=1}^q (A_s^* * B_s^*)} \mu_i x_{ij}^k \theta_{ij}^{*k} > 0 \\ & + (1-h) \sum_{(i,w) \notin \bigcup_{s=1}^q (A_s^* * B_s^*)} \mu_i [1-x_{ij}^k] \theta_{ij}^{*k} > 0 \\ & = \text{Max}_{(\mathcal{A}, \mathcal{B}) \in \mathcal{H}} \left\{ \text{Max}_U \left[h \sum_{(i,w) \in \bigcup_{s=1}^q (A_s * B_s)} \mu_i x_{ij}^k \theta_{ij}^k > 0 \right. \right. \\ & \left. \left. + (1-h) \sum_{(i,w) \notin \bigcup_{s=1}^q (A_s * B_s)} \mu_i [1-x_{ij}^k] \theta_{ij}^k > 0 \right] \right\} \end{aligned} \quad (9)$$

où $j \in \{1, \dots, m\}$ et $k \in \{1, \dots, k_j\}$ sont reliés à w par la relation :

$$w = \sum_{r=1}^{j-1} k_r + k \quad \left(\sum_{r=1}^{j-1} k_r = 0 \text{ si } j = 1 \right)$$

$h \in [0,1]$ est le coefficient de pondération entre les deux sommes. Il est donné par l'utilisateur.

U contient les ensembles de θ_{ij}^k qui vérifient (5), (6) et (7).

$\mathcal{A} = \{A_1, \dots, A_q\}$ est une partition de l'ensemble des n types de produits en q familles, $\mathcal{B} = \{B_1, \dots, B_q\}$ est une partition de l'ensemble des M machines en q îlots de fabrication et \mathcal{H} est l'ensemble des couples de partitions $(\mathcal{A}, \mathcal{B})$.

La relation (9) indique donc que le problème consiste à choisir un couple de partitions et un ensemble de θ_{ij}^k qui réalisent le maximum pour une valeur donnée de h , coefficient de pondération entre les sommes choisi par l'utilisateur.

III - ALGORITHME DE RECHERCHE D'UNE BONNE SOLUTION

L'algorithme que nous proposons fait intervenir trois modules :

1. choix des machines pour chaque opération,
2. choix des affectation initiales des machines,
3. affinage du processus d'affectation.

Nous verrons ensuite comment combiner ces modules dans l'algorithme.

1. CHOIX DES MACHINES POUR CHAQUE OPERATION

Soit $j \in \{1, 2, \dots, m\}$ et $M_j^1, M_j^2, \dots, M_j^{k_j}$ les machines disponibles pour effectuer l'opération j .

Soit $\{s_1, \dots, s_{k_j}\} \subset \{1, 2, \dots, k_j\}$

Pour que $\{M_j^u\}, u = s_1, \dots, s_{k_j}$ soit en mesure de satisfaire la demande, il faut qu'il existe au moins un ensemble $\{\theta_{ij}^{s_k}\}, i=1, \dots, n; k=1, \dots, k_j$ qui vérifie les relations (5), (6) et (7).

Partant des relations (5), on somme en i après prémultiplication des deux membres par μ_i . Il vient :

$$\sum_{k=1}^{k_j} \alpha_{j^k}^S \sum_{i=1}^n \mu_i \theta_{i,j^k}^S = \sum_{i=1}^n \mu_i t_{ij}$$

d'où en utilisant (6) :

$$\sum_{k=1}^{k_j} \alpha_{j^k}^S \geq \left(\sum_{i=1}^n \mu_i t_{ij} \right) / \tau \quad (10)$$

De même en utilisant (7) :

$$\sum_{k=1}^{k_j} \alpha_{j^k}^S x_{j^k}^S \leq \left(\sum_{i=1}^n \mu_i t_{ij} \right) / \tau \quad (11)$$

Enfin, pour que l'ensemble $\{M_j^u\}$, $u=s_1, \dots, s_{k_j}$, soit en mesure d'absorber la tâche j dans de bonnes conditions de rentabilité, il faut que (voir (10) et (11)) :

$$\sum_{k=1}^{k_j} \alpha_{j^k}^S x_{j^k}^S \leq \left(\sum_{i=1}^n \mu_i t_{ij} \right) / \tau \leq \sum_{k=1}^{k_j} \alpha_{j^k}^S \quad (12)$$

Dans un premier temps, nous explorerons tous les groupes de machines possibles. Si aucun d'entre eux ne vérifie (12), le problème sera réputé impossible. Si plusieurs d'entre eux vérifient (12), on retient le groupe choisi par l'utilisateur.

On fournira à celui-ci, pour se guider, la valeur du rapport $(b-a) / (c-a)$ pour chacun des groupes, où :

$$\begin{aligned} a &= \sum_{k=1}^{k_j} \alpha_{j^k}^S x_{j^k}^S & b &= \left(\sum_{i=1}^n \mu_i t_{ij} \right) / \tau \\ c &= \sum_{k=1}^{k_j} \alpha_{j^k}^S \end{aligned} \quad (12')$$

Ce rapport mesure le remplissage des machines par rapport au seuil économique.

2. CHOIX DES AFFECTATIONS INITIALES DES MACHINES

Dans toute la suite, nous supposons qu'un prétraitement a été effectué. Ce prétraitement est l'application de l'algorithme décrit dans [8], où les produits sont en lignes et les opérations en colonnes.

Si le groupe retenu se réduit à une seule machine, la colonne représentant cette machine sera la colonne correspondante de la matrice initiale et le processus d'affectation est terminé. Sinon, l'affectation des machines aux classes s'effectue comme suit.

Soit q le nombre de familles de produits et k_j le nombre de machines à affecter.

Nous envisageons deux cas :

a. $k_j < q$

Dans ce cas, certaines familles de produits ne seront rattachées à aucune machine. La figure 1, où $k_j = 2$ et $q = 3$, représente une telle situation :

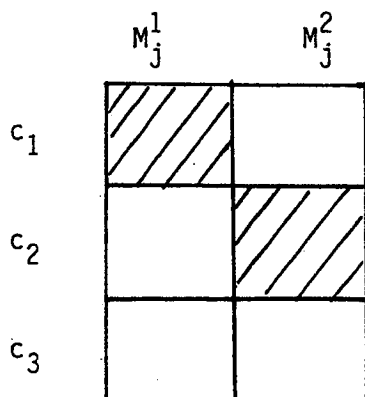


Fig. 1

M_j^1 est affectée à c_1 , M_j^2 est affectée à c_2 et la famille c_3 ne reçoit aucune affectation.

Dans ce cas, on commencera par déterminer les familles de produits qui ne reçoivent aucune affectation : ce sont celles pour lesquelles il est possible de minimiser les utilisations de machines (i.e. les 1) pour un choix judicieux des machines qui vont se charger des produits de cette famille. Ce choix judicieux est le choix des machines dans l'ordre décroissant de leur ca-

dence, c'est-à-dire dans l'ordre décroissant des α_j^k . La suite des opérations se déroulera alors comme dans le cas où $k_j \geq q$.

b. $k_j \geq q$

Nous admettons que les capacités des machines ont été corrigées lorsqu'elles ont été utilisées pour satisfaire les familles de produits qui ne reçoivent aucune affectation (cas $k_j < q$).

Pour chaque famille de produits, nous recherchons l'ensemble des machines qui réalise "au mieux" l'opération.

Soit c_r l'une des familles non encore traitées ($r \in \{1, 2, \dots, q\}$).

Nous retenons de toutes les combinaisons possibles de machines celles qui autorisent la transformation de la famille c_k tout en respectant le critère de rentabilité.

Si $G \subset \{s_1, \dots, s_{k_j}\}$ est une de ces combinaisons, elle vérifiera donc :

$$\sum_{k \in G} \alpha_j^k \theta_{ij}^k = t_{ij}, \quad i \in c_r \quad (13)$$

(voir relation (5)). Cette relation indique que les machines choisies satisfont la demande des produits de c_r pour une certaine répartition des tâches exprimée par les $\theta_{i,j}^k$.

De plus :

$$\tau_k - (1-x_j^k) \tau \leq \sum_{i \in c_r} \mu_i \theta_{ij}^k \leq \tau_k, \quad k \in G \quad (14)$$

où $\tau_k \leq \tau$ est le temps de travail disponible sur la machine k après d'éventuelles affectations dues au cas $k_j < q$. Lorsque $\tau_k = \tau$, nous retrouvons les relations (6) et (7).

En multipliant les membres de (14) par α_j^k et en sommant sur $k \in G$ il vient compte tenu de (13) :

$$\begin{aligned} \sum_{k \in G} \alpha_j^k \tau_k - \sum_{k \in G} \alpha_j^k (1-x_j^k) \tau &\leq \sum_{i \in c_r} \mu_i t_{ij} \\ &\leq \sum_{k \in G} \alpha_j^k \tau_k \end{aligned} \quad (15)$$

Si a, b et c sont les trois membres de (15), pris dans l'ordre croissant, nous retiendrons l'ensemble des machines qui est utilisé au maximum, c'est-à-dire qui minimise :

$$(c-b) / c$$

Soit G_r l'ensemble des machines retenues pour la famille de produits c_r , et w_r la valeur du critère $(c-b) / c$.

Nous retenons r_1 qui minimise w_r :

$$w_{r_1} = \text{Min}_r w_r \quad (16)$$

Le minimum est pris sur l'ensemble des familles non encore traitées.

Les machines sont alors examinées dans un ordre quelconque. Pour chaque machine, on affecte un maximum de produits compte tenu de (13) et de la seconde inégalité de (14). Si la première inégalité de (14) n'est pas vérifiée :

1. s'il reste des produits de C_{r_1} non affectés, on affecte partiellement un de ces produits à la machine.
2. sinon, on réaffecte partiellement un (ou des) produits déjà affectés de façon à respecter (13) et (14).

Les détails sont donnés dans le programme.

On reprend alors le processus après avoir modifié les périodes maximales d'activité τ_k en fonction des affectations précédentes.

3. AFFINAGE DU PROCESSUS D'AFFECTATION

L'affectation des machines étant effectuée, l'affinage de cette affectation consiste à optimiser le critère (9) à partir de l'état initial obtenu. Cette optimisation s'effectue en utilisant l'algorithme décrit dans [8].

Comme nous allons le montrer dans l'algorithme qui suit, le processus d'affinage s'effectue chaque fois qu'ont été affectées les machines correspondant à un ensemble d'opérations qui nécessitent chacune le même nombre de machines. En outre, les ensembles d'opérations sont examinés dans l'ordre croissant des machines qu'ils nécessitent.

4. L'ALGORITHME

Il se déroule comme suit :

1. Initialisation du problème

Cette étape consiste à appliquer l'algorithme décrit dans [8] à la matrice

$T^1 = [x_{t_{ij}} > 0]$ déduite de T. On obtient ainsi q familles de produits qui serviront de base à la suite de l'algorithme.

2. Pour chacune des opérations, c'est-à-dire pour $j = 1, \dots, m$:

2.1. Pour chaque groupe de machines possible

2.1.1. Si le groupe de machines vérifie (12), calcul de l'indicateur (12') et passage au groupe suivant

2.1.2. sinon, passage au groupe suivant

2.2. Si aucun groupe de machines ne vérifie (12), on donne à l'utilisateur la possibilité de décider. Il pourra décider de violer la première inégalité de (12) ou d'abandonner.

2.3. Si au moins un groupe de machines vérifie (12) on choisit le groupe qui réalisera l'opération considérée. Ce choix peut être automatique (critère (12') maxi) ou interactif.

3. Classement des opérations dans l'ordre croissant de nombre de machines utilisées.

Soit S le nombre maximal de machines utilisées pour une opération

4. Pour $kk = 1, S$

4.1. Pour chaque opération utilisant kk machines, choix des affectations initiales (voir III-2)

4.2. Affinage sur la matrice de 0 et de 1 déduite de ces affectations (voir III-3).

5. Fin de processus.

On a observé que l'intégration des opérations dans le système se fait dans l'ordre croissant du nombre de machines qu'elles nécessitent : on cherche ainsi à placer en priorité les machines les plus difficiles à affecter

à un îlot de fabrication parce que les moins nombreuses pour effectuer une opération.

L'algorithme que nous proposons n'est évidemment pas optimal. Il possède cependant la propriété suivante.

5. UNE PROPRIETE DE L'ALGORITHME

Nous allons montrer le résultat suivant :

THEOREME :

S'il existe, pour une opération j et des familles de produits données, une affectation des k_j machines telle que :

1. une machine donnée ne soit jamais utilisée pour transformer un produit appartenant à une famille autre que celle à laquelle elle est affectée,
 2. toutes les machines soient saturées,
- alors le processus précédent permet d'atteindre cette solution.

DEMONSTRATION :

- a. Le fait que le critère puisse être égal à $n.k_j$ pour la famille de produits considérée entraîne que si $k_j < q$ il existe au moins $q-k_j$ familles ne demandant pas la transformation j . Donc le processus de choix de ces familles, qui consiste à minimiser les utilisations des machines pour cette opération, conduira effectivement à décider que ces familles ne reçoivent aucune affectation de machine.
- b. Passons maintenant au cas $k_j \geq q$. En suivant le processus d'affectation donné en III-2-b, nous voyons que nous sommes conduits, à chaque étape, à affecter à une famille de produits un ensemble de machines qui permet d'effectuer la transformation et que cet ensemble de machine est saturé. D'où la conclusion. □

Ce résultat montre que, dans certains cas de figure (la saturation), notre algorithme minimise le nombre de zéros situés en dehors des blocs diagonaux dans la matrice de 0 et de 1 déduite de T (cf. (1)), ce qui est évidemment une situation de départ favorable.

ANNEXE I

LES PROGRAMMES

Nous donnons d'abord une courte documentation concernant les logiciels disponibles.

Les textes des différents programmes suivent cette courte documentation. On n'y trouvera pas le texte de ALBLOC, sans intérêt ici. Son seul rôle est de générer automatiquement une matrice à blocs non empiétants en contrôlant le pourcentage de 0 situés à l'extérieur des blocs et le pourcentage de 1 situés à l'intérieur. Ce programme nous a servi à tester la méthode et à la comparer aux méthodes existantes. Les résultats de ces tests sont en cours de publication.

```
*****  
*                                                                 *  
*          DOCUMENTATION DU PROGRAMME TGB                       *  
*                                                                 *  
*****
```

I/ PROGRAMME DE RECHERCHE D'ILOTS DE FABRICATIONS :TGB

Recherche d'ilots de fabrications, machines multiples.

Contraintes actuelles du programme

Nombre total de machines : 100

Nombre total de types differents de pieces : 2000

Nombre maximal de types de machines par operation : 9

Nombre maximal de machines par type : 20

Nombre maximal d'ilots de fabrication : 50

Fichiers:

42 : resultats matrices reordonnees [fichier texte]

51 : resultats classes des produits [fichier texte]

53 : affectations des machines [fichier texte]

55 : resultats intermediaires [fichier texte]

81,83,87 : fichiers de travail

II/ PROGRAMME DE CHARGEMENT : CHTG2

Memes contraintes que pour tgb

Fichiers:

60 : donnees operations/produits

82 : donnees operations/machines

84 : donnees operations/produits [fichier texte]

85 : donnees operations/machines [fichier texte]

III/ PROGRAMME DE MANIPULATION DES FICHIERS 60 & 84 : MANIP

Fichiers:

59 : resultats d'editions [fichier texte]

Ce programme permet:

- de melanger la matrice operations/ produits par echange de lignes et colonnes. Il est possible de melanger simultanement le fichier operations machines.
- d'editer le fichier operations/produits sous forme d'un tableau, de calculer le cumul des charges par operation et le nombre de machines de reference (cadence relative=1) necessaires.

IV/ PROGRAMME DE SIMULATION DE DONNEES : ALBLOC

Fichiers:

60 : fichier operations/produits

Ce programme genere pseudo-aleatoirement des donnees operations/produits.

```
c *****
c ** PROGRAMME TG **
c *****
c
c Contraintes actuelles du programme
c-----Nombre total de machines : 100
c-----Nombre total de types differents de pieces : 2000
c-----Nombre maximal d'ilots de fabrication : 50
c
c Ce programme est precede du programme de chargement CHTG2
c
c Fichiers:
c 42 : resultats matrices reordonnees [fichier texte]
c 51 : resultats classes des produits [fichier texte]
c 53 : affectations des machines [fichier texte]
c 55 : resultats intermediaires [fichier texte]
c
c 81,83,87 : fichiers de travail
c
c
c
c
c logical clv,result
c dimension tt(100),ity(100),iop(100),ttl(100),iaf(100)
c dimension icomb(20,20),coef(20)
c dimension nb(20),g(20),nx(100),p(20),ij(20)
c dimension a(101),na(100),izl(2000),iz(2000),c(50,100),iu(100),iw(100)
c dimension clv(50),ntim(50),ntin(50)
c dimension iwl(100)
c common n,ic,h,a,c,tot,iop,ity,kzl
c
c open(81,access="direct",form="formatted",recl=1500)
c open(60,access="direct",form="formatted",recl=1500)
c open(82,access="direct",form="formatted",recl=160)
c open(83,access="direct",form="formatted",recl=300)
c open(87,access="direct",form="formatted",recl=300)
c open(42,form="formatted")
c open(51,form="formatted")
c open(53,form="formatted")
c eps=1.e-4
c read(60,77,rec=1,err=188)n,m
c write(81,77,rec=1,err=9)n,m
77 format(2i3)
c write(83,77,rec=1,err=11)n,m
c write(0,1018)
1018 format(/,2x,"PROGRAMME TG",/)
c write(0,1030)
1030 format(2x,"Resultats intermediaires sur fichier (file55) oui=1,non=0")
c read(0,16)i
c result=(i.eq.1)
```

```
if(result)open(55,form="formatted")
if(result)write(55,1018)
write(87,77,rec=1,err=112)n,m
write(0,1)n,m
if (result)write(55,1)n,m
1 format(4x,"Nombre de types de produits:",i4,4x,"Nombre d'operations:",i4)
```

C
C
C
C
C
C
C

```
*****
**      PRETRAITEMENT      **
*****
```

CONSTITUTION DE LA MATRICE A ELEMENTS BINAIRES

```
write(0,200)
200 format(2x,"Entrez la periode de travail")
read(0,16)to
if(result)write(55,1019)to
1019 format(/,2x,"Periode de travail consideree=",f8.3)
do 2 i=2,n+1
read(60,8,rec=i,err=188)(a(j),j=1,m+1)
write(81,8,rec=i,err=9)(a(j),j=1,m+1)
do 3 j=1,m
na(j)=0
if(a(j).gt.eps)na(j)=1
3 continue
x=a(m+1)
write(83,10,rec=i,err=11)x,(na(j),j=1,m)
2 write(87,10,rec=i,err=112)x,(na(j),j=1,m)
10 format(E14.7,100i1)
8 format(101E14.7)
```

C
C
C

INFORMATIONS ET CHOIX DES PRODUITS INITIAUX

```
1017 if(result)write(55,1010)
write(0,15)
15 format(2x,"Nombre maximal de classes souhaitees?")
read(0,16)ic
if(result)write(55,1020)ic
1020 format(2x,"Nombre maximal de classes souhaitees =",i4)
16 format(v)
write(0,905)
905 format(2x,"Coef. de ponderation entre 0 et 1")
read(0,16)h
if (result)write(55,1021)h
1021 format(2x,"Coefficient de ponderation =",f8.3)
write(0,22)
22 format(2x,"Choix des produits initiaux par l'utilisateur =0",/,
& 2x,"Determination automatique des produits initiaux =1")
read(0,16)i
if (i.eq.1)goto 19
```

C

c CHOIX DES POINTS INITIAUX PAR L'UTILISATEUR

```
c
c
  if(result)write(55,1022)
1022 format(2x,"Choix des produits initiaux : Manuel")
  do 25 j=1,ic
    write(0,26)
  26 format(2x,"Produit suivant?")
    read(0,16)i
    if(result)write(55,1023)j,i
1023 format(2x,"Classe:",i3,",Produit No:",i4)
    k=i+1
    read(83,10,rec=k,err=11)x,(na(kk),kk=1,m)
    do 900 kk=1,m
  900 c(j,kk)=na(kk)
  25 continue
  goto 50
```

c DETERMINATION AUTOMATIQUE DES PRODUITS INITIAUX

```
c
c
  19 write(0,18)
  18 format(2x,"Distance minimale entre points initiaux?")
    read(0,16)w
    if(result)write(55,1024)w
1024 format(/,2x,"Choix initial des representants : Automatique",
  &      /,2x,"Distance mini entre points initiaux=",f8.3)
```

c RECHERCHE DES PRODUITS

```
c
c
  do 21 j=1,n
21 iz(j)=0
  jj=1
  call random_$uniform(x)
  kk=x*n+2
  if(kk.gt.n+1)kk=n+1
31 iz(kk-1)=1
  read(83,10,rec=kk,err=11)y,(na(k),k=1,m)
  do 901 k=1,m
901 c(jj,k)=na(k)
  do 28 i=1,n
  if(iz(i).eq.1)goto 28
  k1=i+1
  read(83,10,rec=k1,err=11)y,(na(j),j=1,m)
  u=0
  do 30 k=1,m
30 u=u+(c(jj,k)-na(k))**2
  u=u**0.5
  if (u.gt.w)goto 28
  iz(i)=1
28 continue
  if(jj.ge.ic)goto 50
  jj=jj+1
```

```
do 29 i=1,n
  kk=i+1
  if(iz(i).eq.0)goto 31
29 continue
  write(0,4)
  if(result)write(55,4)
  4 format(2x,"Nombre de points insuffisant")
  write(0,5)
  if(result)write(55,5)
  5 format(2x,"Distance minimale entre points initiaux trop forte",/)
  write(0,1016)
1016 format(2x,"Abandon=0, Reprise=1")
  read(0,16)i
  if(i.eq.1)goto 1017
  goto 1000
```

c
c
c

CONSTITUTION DES CLASSES

```
50 icont=0
51 do 32 i=1,n
  k=i+1
  read(83,10,rec=k,err=11)y,(na(j),j=1,m)
  u=m+1
  do 33 jj=1,ic
  uu=0
  do 34 j=1,m
34 uu=uu+(na(j)-c(jj,j))**2
  if(uu.gt.u) goto 33
  iz(i)=jj
  u=uu
33 continue
32 continue
  if(icont.eq.0)goto 42
  if(icont.gt.50)goto 60
  do 44 j=1,n
  if(iz(j).ne.izl(j))goto 42
44 continue
  goto 60
42 do 43 j=1,n
43 izl(j)=iz(j)
```

c
c
c

RECHERCHE DES CENTRES D'INERTIE SUIVANTS

```
do 35 i=1,ic
do 37 j=1,m
37 c(i,j)=0
35 a(i)=0
do 36 i=1,n
  k=i+1
  jj=iz(i)
  read(83,10,rec=k,err=11)y,(na(j),j=1,m)
```

```
      a(jj)=a(jj)+1
      do 38 j=1,m
38    c(jj,j)=c(jj,j)+na(j)
36    continue
      do 39 i=1,ic
      do 40 j=1,m
40    c(i,j)=c(i,j)/a(i)
39    continue
      icont=icont+1
      goto 51
c
c  FIN DE RECHERCHE DES FAMILLES DE PRODUITS INITIALES
c
c  GROUPEMENT DES PRODUITS ET DES OPERATIONS
c
60  do 90 i=1,n
90  izl(i)=i
      do 91 j=1,m
      iop(j)=0
      ity(j)=0
91  iw(j)=j
      kzl=m
      call apo(m,83,na,iu,iw,iz,izl,iwl)
c
c  AFFICHAGE DES RESULTATS DU GROUPEMENT PRODUITS-OPERATIONS
c
      write(0,1010)
1010 format(/,2x,"Groupement initial des produits et des operations",/,
      &      2x,"-----")
      write(0,1031)tot
      if(result)write(55,1031)tot
1031 format(2x,"Valeur finale du critere=",f8.3)
      do 1050 i=1,ic
1050 clv(i)=.true.
      do 1051 i=1,m
1051 clv(iu(i))=.false.
      do 1052 i=1,n
1052 ntin(i)=0
      do 1053 i=1,m
1053 ntim(i)=0
      do 1054 i=1,n
1054 ntin(iz(i))=ntin(iz(i))+1
      do 1055 i=1,m
1055 ntim(iu(i))=ntim(iu(i))+1
      iuc=0
      do 1058 i=1,ic
      if(clv(ic))goto 1058
      iuc=iuc+1
1058 continue
      write(0,1011)iuc
      if(result)write(55,1011)iuc
```



```
1011 format(2x,i4," Classe(s)",/,2x,"Taille des groupements:")
      do 1057 i=1,ic
        write(0,1012)i
        if(result)write(55,1012)i
1012 format(2x,"Classe:",i4)
      if(.not.clv(i))goto 1056
      write(0,1013)
      if(result)write(55,1013)
1013 format(2x,"VIDE")
      goto 1057
1056 write(0,1014)ntim(i),ntin(i)
      if(result)write(55,1014)ntim(i),ntin(i)
1014 format(2x,i4," operation(s)",,i5," Produit(s)")
1057 continue
      write(0,1015)
1015 format(2x,"Abandon=0, Suite=1, Reprise=2")
      read(0,16)i
      if(i.eq.2)goto 1017
      if(i.eq.0)goto 1000
      nx(1)=0
      y=0
      do 14 i=1,n
        k=i+1
        read(83,10,rec=k,err=11)y,(na(il),il=1,m)
14 write(87,10,rec=k,err=112)y,(nx(1),il=1,m)
```

```
c
c *****
c **      FIN DU PRETRAITEMENT      **
c *****
c
c *****
c **      TRAITEMENT PRINCIPAL      **
c *****
c
c
c
c-----Recherche du nombre maximum de machines
c
```

```
      if(result)write(55,548)
548 format(/,2x,"Traitement principal",
      &      /,2x,"-----")
      maxm=0
      do 911 j=1,m
        read(82,153,rec=j,err=186)nt,(nb(i),g(i),p(i),i=1,nt)
        nn=0
        do 910 il=1,nt
910 nn=nn+nb(il)
          if (nn.gt.maxm)maxm=nn
911 continue
      write(0,912)maxm
912 format(2x,"Nombre maximum de machines=",I3)
```

c

c-----BOUCLE PRINCIPALE D'ITERATIONS SUR LE NB DE MACHINES

```
c
    kz=0
    kz1=0
    im=1
325 ind=0
    do 154 j=1,m
        read(82,153,rec=j,err=186)nt,(nb(i),g(i),p(i),i=1,nt)
153 format(i1,9(i2,e14.7,e14.7))
        nn=0
        do 155 il=1,nt
155 nn=nn+nb(il)
            if(nn.ne.im)goto 154
            ind=ind+1
            if(im.ne.1)goto 156
```

```
c
c-----ETUDE DU CAS OU L'OPERATION EST EFFECTUEE PAR
c----- UNE SEULE MACHINE.
```

```
c
    kz=kz1+1
    kz1=kz1+1
    write(0,549)j
    if(result)write(55,549)j
549 format(/,2x,"Operation:",i4,/,2x,"*****")
543 write(0,547)j
    if(result)write(55,547)j
547 format(2x,"L'operation ",i4," sera prise en charge par une machine")
    do 160 i=1,n
        k=i+1
        read(83,10,rec=k,err=11)y,(na(il),il=1,m)
        iop(kz)=j
        ity(kz)=1
        j5=1
        if(kz.gt.1)j5=kz-1
        read(87,10,rec=k,err=112)y,(nx(il),il=1,j5)
        nx(kz)=na(j)
        write(87,10,rec=k,err=112)y,(nx(il),il=1,kz)
160 continue
    go to 154
```

```
c
c-----ETUDE DU CAS OU L'OPERATION EST EFFECTUEE
c----- PAR PLUSIEURS MACHINES
```

```
c
c      Introduction d'un compteur
```

```
c
156 write(0,549)j
    if(result)write(55,549)j
    kz=kz1+1
    do 164 i=1,nt
164 ij(i)=0
```

```
c
```

c-----Recherche des groupes de machines possibles

```
c
    iq=0
    i=1
167  ij(i)=ij(i)+1
    if(ij(i).le.nb(i)) go to 125
    i=i+1
    if(i.le.nt) go to 167
    go to 1544
125  if(i.eq.1) go to 166
    do 126 il=1,i-1
126  ij(il)=0
    i=1
166  bi=0
    bs=0
    do 168 k=1,nt
    bi=bi+g(k)*p(k)*ij(k)
168  bs=bs+g(k)*ij(k)
    som=0
    do 173 k=1,n
    kl=k+1
    read(81,8,rec=kl,err=9)(a(il),il=1,m+1)
173  som=som+a(m+1)*a(j)
    som=som/to
    if((som.gt.bs).or.(som.lt.bi))goto 167
    xx=(som-bi)/(bs-bi)
    iq=iq+1
    if (iq.gt.20) goto 1544
    do 179 k=1,nt
179  icomb(iq,k)=ij(k)
    coef(iq)=xx
    go to 167
1544 write(0,180)j
    if(result)write(55,180)j
180  format(2x,"Solution(s) possible(s) pour l'operation ",i4)
    if(iq.ge.1) go to 69
    write(0,66)
    if(result)write(55,66)
66  format(6x,"** Aucune solution ** ")
    go to 1000
69  do 181 k=1,iq
    write(0,182)k,coef(k)
    if(result)write(55,182)k,coef(k)
182  format(2x,"* Solution numero:",i3," : COEFF=",f8.3)
    do 184 il=1,nt
    write(0,185)icomb(k,il),il
    if(result)write(55,185)icomb(k,il),il
185  format(6x,i3," machine(s) de type ",i3)

184  continue
181  continue
```

c-----Choix du groupe de machines qui va etre utilise

```
c
  if(iq.gt.1)goto 544
  write(0,545)
  if(result)write(55,545)
545 format(2x,"La seule solution possible est retenue")
  is=1
  goto 546
544 write(0,500)
500 format(2x,"Quelle solution choisissiez-vous?")
  read(0,16)is
  if(result)write(55,1026)is
1026 format(2x,"Solution choisie:",i4)
```

c
c mac = nombre de machines

```
c
546 mac=0
  do 501 j5=1,nt
501 mac=mac+icomb(is,j5)
```

c
c-----Recherche des temps de travail disponibles et reperage des
c-----machines utilisees dans la matrice finale

```
c
  kz1=kz1+mac
  if(mac.eq.1)goto 543
  do 513 i=1,n
  k=i+1
  j5=1
  if(kz.gt.1) j5=kz-1
  read(87,10,rec=k,err=112)y,(nx(il),il=1,j5)
  do 514 j5=kz,kz1
  nx(j5)=0
  iop(j5)=j
514 write(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
513 continue
  read(82,153,rec=j,err=186)nt,(nb(i),g(i),p(i),i=1,nt)
  kk=kz-1
  do 515 i=1,nt
  i5=icomb(is,i)
  if (i5.eq.0)goto 515
  do 516 kg=1,i5
  kk=kk+1
  tt(kk)=to*g(i)
  tt1(kk)=tt(kk)
516 ity(kk)=i
515 continue
```

c
c Calcul du nb de classes kc

```
c
kc=0
do 502 j5=1,ic
```

```
502 nb(j5)=0
    do 503 j5=1,m
        kk=iu(j5)
503 nb(kk)=1
    do 504 j5=1,ic
504 kc=kc+nb(j5)
    ix=0
```

c

c-----Nombre de classes strictement superieur au nb de machines

c

c

c-----RECHERCHE DES CLASSES SANS AFFECTATION DE MACHINE

c

```
c    do 45 i5=1,n
c    k=i5+1
c    read(81,8,rec=k,err=9)(a(j5),j5=1,m+1)
    do 505 j5=1,ic
505 nb(j5)=0
    do 506 i5=1,n
        k=izl(i5)+1
        read(83,10,rec=k,err=11)y,(na(il),il=1,m)
        kk=iz(i5)
        ij(kk)=kk
506 nb(kk)=nb(kk)+(1-na(j))
```

c

c

c

c-----Classement par ordre decroissant

c

```
    do 507 i5=1,ic-1
    do 507 j5=i5+1,ic
    if(nb(i5).gt.nb(j5))goto 507
    jj=nb(i5)
    nb(i5)=nb(j5)
    nb(j5)=jj
    jj=ij(i5)
    ij(i5)=ij(j5)
    ij(j5)=jj
507 continue
    kk=kc-mac
    if(kc.le.mac) go to 600
```

c

c

c-----Recherche du rang de la machine la plus disponible

c-----du type de la machine et du rang dans le type

c

c

```
528 xx=0
    kk=kz-1
    k=0
    do 517 i=1,nt
```

```
i5=icomb(is,i)
if(i5.eq.0)goto 517
do 518 k=1,i5
kk=kk+1
if(tt(kk).lt.eps) go to 518
if(tt(kk).lt.(xx+eps))goto 518
xx=tt(kk)
jty=i
jrt=k
irt=kk
518 continue
517 continue
ix=ix+1
kk=ij(ix)
```

```
c
c-----Recherche du temps a consacrer a la classe retenue
```

```
c
c
c
tokk=0
do 523 i=1,n
k=izl(i)+1
read(81,8,rec=k,err=9)(a(i5),i5=1,m+1)
i5=iz(i)
if(i5.ne.kk)goto 523
tokk=tokk+a(j)*a(m+1)
523 continue
if(tokk.lt.eps) go to 7
```

```
c
c-----Prise en compte de la machine choisie
```

```
c
do 525 i=1,n
k=izl(i)+1
read(81,8,rec=k,err=9)(a(i5),i5=1,m+1)
if(a(j).lt.eps) go to 525
jj=iz(i)
if(jj.ne.kk)goto 525
w=a(j)*a(m+1)
if((w-eps).le.tt(irt)) goto 526
```

```
c
c-----Temps disponible sur la machine
c-----insuffisant pour satisfaire l'operation
```

```
c
a(j)=(1.-tt(irt)/w)*a(j)
tt(irt)=0
write(81,8,rec=k,err=9)(a(i5),i5=1,m+1)
read(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
nx(irt)=1
write(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
ix=ix-1
goto 528
```

```
c
c-----Temps disponible sur la machine suffisant pour satisfaire
c-----      l'operation.
c
  526 tt(irt)=tt(irt)-w
      a(j)=0
      write(81,8,rec=k,err=9)(a(il),il=1,m+1)
      read(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
      nx(irt)=1
      write(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
  525 continue
  57  format(2x,10(" ",f5.1))
c
c-----Fin du traitement d'une classe sans affectation
c
  7  if(ix.lt.(kc-mac)) goto 528
c
c
c
c      FIN DE TRAITEMENT DES CLASSES SANS AFFECTATION MACHINE
c
c      AFFECTATION DES MACHINES AUX CLASSES
c
c
c
  600 ix=ix+1
      kk=ij(ix)
      jz=jz+1
c
c-----kk est la classe pour laquelle on recherche une affectation
c
c-----Recherche du temps a consacrer a cette classe
c
      tokk=0
      do 530 i=1,n
      k=iz1(i)+1
      read(81,8,rec=k,err=9)(a(i5),i5=1,m+1)
      i5=iz(i)
      if(i5.ne.kk)goto 530
      tokk=tokk+a(j)*a(m+1)
  530 continue
      if(tokk.lt.eps) go to 6
c
c-----Recherche d'une machine capable d'absorber cette tache dans
c-----des conditions economiques acceptables.
c
c
c
      jn1=0
      jn2=0
      jn11=0
      jn22=0
```

```
k=kz-1
read(82,153,rec=j,err=186)nt,(nb(i),g(i),p(i),i=1,nt)
do 531 i=1,nt
i5=icomb(is,i)
if(i5.eq.0)goto 531
do 532 jj=1,i5
k=k+1
if(tt(k).lt.eps) go to 532
aa=ttl(k)*p(i)
bb=tt(k)
if(tokk.gt.bb)goto 531
irt=k
jrt=jj
jty=i
jnl=k
if(k-kz+1.ne.jz) go to 97
jnll=k
irtl=k
jrtl=jj
jtyl=i
97 if(tokk.le.aa)goto 531
jn2=k
if(k-kz+1.eq.jz) jn22=k
532 continue
531 continue
if(jn22.eq.0) go to 95
jn2=jn22
irt=irtl
jrt=jrtl
jty=jtyl
95 if(jn2.eq.0)goto 533
```

c
c-----Prise en compte de la machine qui est utilisee au dela de
c-----son seuil de rentabilite et en deca de sa capacite residuelle
c

```
tt(jn2)=tt(jn2)-tokk
k=jn2
536 do 535 i=1,n
kl=izl(i)+1
read(81,8,rec=kl,err=9)(a(i5),i5=1,m+1)
i5=iz(i)
if (i5.ne.kk)goto 535
if(a(j).lt.eps) go to 535
a(j)=0
write(81,8,rec=kl,err=9)(a(i5),i5=1,m+1)
read(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)
nx(k)=1
write(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)
535 continue
iaf(kk)=ity(k)
```


goto 6

c

c-----Prise en compte de la machine qui est utilisee en deca de son
c-----seuil de rentabilite et en deca de sa capacite residuelle

c

533 if(jn11.eq.0) go to 96

 jnl=jn11

 irt=irt1

 jrt=jrt1

 jty=jty1

96 if(jnl.eq.0)goto 534

 tt(jnl)=tt(jnl)-tokk

 k=jnl

 goto 536

c

c-----Cas ou la machine ne sera pas suffisante pour satisfaire
c----- la demande.

c

534 do 537 jj=kz,kz1

 if(tt(jj).lt.eps)goto 537

 k=jj

 goto 538

537 continue

538 do 539 i=1,n

 kl=iz1(i)+1

 read(81,8,rec=kl,err=9)(a(i5),i5=1,m+1)

 i5=iz(i)

 if(i5.ne.kk)goto 539

 if(a(j).lt.eps) go to 539

 w=a(j)*a(m+1)

 if(w.lt.tt(k))goto 540

 a(j)=(1-tt(k)/w)*a(j)

 tt(k)=0

 write(81,8,rec=kl,err=9)(a(i5),i5=1,m+1)

 read(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)

 nx(k)=1

 write(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)

 ix=ix-1

 goto 600

540 a(j)=0

 tt(k)=tt(k)-w

 write(81,8,rec=kl,err=9)(a(i5),i5=1,m+1)

 read(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)

 nx(k)=1

 write(87,10,rec=kl,err=112)y,(nx(il),il=1,kz1)

539 continue

6 if(ix.lt.kc) go to 600

c

c ***** FIN DE LA BOUCLE PRINCIPALE SUR LE NB DE MACHINES *****

c

154 continue

```
    im=im+1
    if (im.le.maxm)goto 325
c
c-----Groupement des produits et des operations
c
1001 do 92 j=1,kz1
    92 iw1(j)=100*iop(j)+ity(j)
    do 570 j=1,kz1
    570 na(j)=0
        do 571 i=1,n
            k=i+1
            read(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
            do 572 j=1,kz1
                if(nx(j).gt.0) na(j)=1
    572 continue
    571 continue
        j1=0
        do 575 il=1,kz1
            if(na(il).eq.0) goto 575
            j1=j1+1
            iw1(j1)=iw1(il)
            iop(j1)=iop(il)
            ity(j1)=ity(il)
    575 continue
        do 573 i=1,n
            k=i+1
            read(87,10,rec=k,err=112)y,(nx(il),il=1,kz1)
            j1=0
            do 574 il=1,kz1
                if(na(il).eq.0) go to 574
                j1=j1+1
                nx(j1)=nx(il)
    574 continue
            write(87,10,rec=k,err=112)y,(nx(il),il=1,j1)
    573 continue
        kz1=j1
        write(51,1400)(iz(i),i=1,n)
1400 format(2000i3)
        write(53,1401)(iw1(i),i=1,kz1)
1401 format(50i4)
c
    write(0,12)
    12 format(2x,"Tableau resultant")
        do 541 i=1,n
            k1=iz1(i)+1
            read(87,10,rec=k1,err=112)y,(nx(il),il=1,kz1)
    541 write(0,542)(nx(il),il=1,kz1)
    542 format(30(1x,il))
c
c Lecture du coefficient
c
```

```
581 write(0,905)
    read(0,16)h
c
    rewind 51
    rewind 53
    read(51,1400)(iz(i),i=1,n)
    read(53,1401)(iwl(i),i=1,kz1)
    call apo(kz1,87,nx,iu,iw,iz,iz1,iwl)
    if(result)write(55,586)h
586 format(/,2x,"Groupement final des machines",/,2x,
    & "-----",/,2x,"coefficient=",f8.3)
    write(0,1031)tot
    if(result)write(55,1031)tot
c
c   Matrice finale
c
    write(0,590)(iop(i),i=1,kz1)
590 format(2x,"Operation",7x,50(1x,i2))
    write(0,595)(ity(i),i=1,kz1)
595 format(2x,"Machine",9x,50(1x,i2))
    write(0,596)(iw(i),i=1,kz1)
596 format(2x,"Position",8x,50(1x,i2))
    write(0,597)(iu(i),i=1,kz1)
597 format(2x,"Classe",10x,50(1x,i2))
    do 591 i=1,n
        k=iz1(i)+1
        read(87,593,rec=k,err=112)x,(na(j),j=1,kz1)
593 format(E14.7,100i1)
        write(0,592)x,iz1(i),iz(i),(na(iw(j))),j=1,kz1)
592 format(1x,f7.2,1x,i3,1x,i4,2x,50(1x,i1,1x))
591 continue
c
    write(0,580)
580 format(2x,"Fin=0,Modification du coef.=1")
    read(0,16)i5
    if(i5.eq.1) go to 581
    go to 1000
c
112 write(0,273)
273 format(2x,"ERREUR I/O FICHER 87")
    go to 1000
    9 write(0,105)
105 format(2x,"ERREUR I/O FICHER 81")
    goto 1000
    11 write(0,106)
106 format(2x,"ERREUR I/O FICHER 83")
    goto 1000
186 write(0,187)
187 format(2x,"ERREUR I/O FICHER 82 (ou chargement non effectue)")
    goto 1000
188 write(0,189)
```

```
189 format(2x,"ERREUR I/O FICHER 60 (ou chargement non effectue)")
```

c

```
1000 close(81)  
      close(82)  
      close(83)  
      close(87)  
      close(42)  
      close(51)  
      close(53)  
      close(60)  
      stop  
      end
```



```
15 write(60,8,rec=i,err=152)(a(j),j=1,m+1)
   goto 12
```

c
c
c

ADJONCTION D'UN ELEMENT NOUVEAU

```
5 read(60,77,rec=1,err=152)n,m
  write(0,102)
102 format(2x,"Nombre de produits a adjoindre?")
    read(0,2)n1
    i=n+1
    n=n+n1
20 write(0,16)i
16 format(2x,"Produit numero ",i4)
    read(0,2)(a(j),j=1,m)
    write(0,18)
18 format(2x,"Poids attache au produit?")
    read(0,2)a(m+1)
    write(60,8,rec=i+1,err=152)(a(j),j=1,m+1)
    write(0,19)
19 format(2x,"-----")
    i=i+1
    if(i.le.n)goto 20
    write(60,77,rec=1,err=152)n,m
    goto 12
```

c
c
c

NOUVELLES DONNEES

```
6 write(0,22)
22 format(2x,"Nombre de produits ?")
    read(0,2)n
    write(0,21)
21 format(2x,"Nombre d'operations?")
    read(0,2)m
    write(0,100)m
100 format(/,2x,"Pour chaque produit, tapez les",i3," valeurs ",
  &      /,2x,"separees par des virgules.",/)
    i=1
    goto 20
```

c
c
c

IMPRESSION

```
200 read(60,77,rec=1,err=152)n,m
    write(84,201)
201 format(12x,"Fichier produits-operations",/,12x,27("*"))
    write(84,202)n,m
202 format(2x,"Nombre de produits:",i4," Nombre d'operations:",i4)
    do 205 i=1,n
      k=i+1
      read(60,8,rec=k,err=152)(a(j),j=1,m+1)
      write(84,203)i,a(m+1)
203 format(4x,"Produit numero:",i4,3x,"Poids",f10.2)
```

```
    write(84,204)(j,a(j),j=1,m)
204 format(5(2x,i3,"/",f10.2))
205 continue
    goto 12
c
c *****
c * FICHER OPERATIONS-MACHINES *
c *****
c
50 write(0,51)
51 format(2x,"Voulez-vous considerer le fichier operations-machines 0/1?")
    read(0,2)i
    if(i.eq.0)goto 1000
    read(60,77,rec=1,err=152)n,m
    write(0,299)
299 format(2x,"Modification=1, Nouvelles donnees=2, Impression=3")
    read(0,2)k5
    goto(104,105,300),k5
c
c MODIFICATIONS
c
104 write(0,52)
52 format(2x,"Donnez le numero de l'operation concernee ou -1")
    read(0,2)i
    if (i.eq.-1) goto 50
    read(82,53,rec=i,err=150)nt,(nb(j),a(j),p(j),j=1,nt)
53 format(i1,9(i2,E14.7,E14.7))
    do 54 j=1,nt
    write(0,55)j,nb(j),a(j)
55 format(2x," Type: ",i2,2x,"Nombre: ",i2,2x,"Cadences relatives:",f8.2)
54 write(0,24)p(j)
24 format(2x,"Seuil economique:",f8.2)
    write(0,26)
26 format(" Voulez-vous changer toute l'operation?0/1")
    read(0,2)j
    if(j.eq.0) go to 62
    write(0,112)
    read(0,2)nt
    write(0,113)
    do 27 j=1,nt
27 read(0,2)nb(j),a(j),p(j)
    go to 61
62 write(0,58)
58 format(2x,"Donnez le type de machine concernee ou -1")
    read(0,2)j
    if (j.eq.-1) goto 61
    if(j.gt.nt+1)goto 62
    if(j.eq.nt+1)nt=nt+1
    write(0,59)
59 format(2x,"Nombre de machines de ce type?")
    read(0,2)nb(j)
```

```
write(0,60)
60 format(2x,"Cadence relative?")
read(0,2)a(j)
write(0,25)
25 format(2x,"Seuil economique?")
read(0,2)p(j)
goto 62
61 write(82,53,rec=i,err=150)nt,(nb(j),a(j),p(j),j=1,nt)
goto 104
```

c

c NOUVELLES DONNEES

c

```
105 do 110 i=1,m
write(0,111)i
111 format(2x,"Operation ",i3)
write(0,112)
112 format(2x,"Nombre de types de machines?")
read(0,2)nt
write(0,113)
113 format(2x,"Pour chaque type de machine donnez le nombre",/,
& 2x,"la cadence relative et le seuil economique")
do 114 j=1,nt
114 read(0,2)nb(j),a(j),p(j)

110 write(82,53,rec=i,err=150)nt,(nb(j),a(j),p(j),j=1,nt)
goto 50
```

c

c IMPRESSION

c

```
300 read(60,77,rec=1,err=152)n,m
write(85,301)
301 format(12x,"Fichier operations-machines",/,12x,27(" "))
do 305 i=1,m
read(82,53,rec=i,err=150)nt,(nb(j),a(j),p(j),j=1,nt)
write(85,302)i,nt
302 format(2x,"Operation",i4,2x,"Nombre de types de machines:",i3)
do 303 j=1,nt
303 write(85,304)j,nb(j),a(j),p(j)
304 format(2x,"Type:",i4,3x,"Nombre:",i4,3x,"Cadence relative:",f8.2,/,
& 3x,"Seuil economique:",f8.2)
305 continue
goto 50
150 write(0,151)
151 format(2x,"Erreur lecture/ecriture fichier operations-machines")
goto 1000
152 write(0,153)
153 format(2x,"Erreur lecture/ecriture fichier produits-operations")
1000 close(60)
close(82)
close(84)
```



```
C *****
C ** PROGRAMME DE MANIPULATION **
C ** DES FICHIERS 60 & 82 **
C *****
C
C fichier de resultats file59
C
integer pl,pc
logical sim
dimension a(150,35),cum(34),pl(150),pc(34),t(35),al(150),ac(34)
dimension nb(101),aa(101),pp(20)
open(60,access="direct",form="formatted",recl=1500)
open(82,access="direct",form="formatted",recl=160)
open(69,access="direct",form="formatted",recl=160)
open(59,form="formatted")
open(56,form="formatted")
800 write(0,35)
35 format(5x,"1.Nouveau fichier",/,5x,"2.Modification",/,5x,"3.Impression",
&/,5x,"4.Melange",/,5x,"5.Fin")
read(0,2)ind
go to (500,600,700,1600,9),ind
C
C *****
C ** NOUVEAU FICHIER **
C *****
C
500 write(0,1)
1 format(2x,"_Nombre d'individus ? ")
read(0,2)n
2 format(v)
write(0,3)
3 format(2x,"_Nombre de parametres?(sans compter le poids) ")
read(0,2)m
m=m
write(60,77,rec=1,err=1000)n,m
77 format(2i3)
C
do 10 i=2,n+1
write(0,11)i-1
11 format(/,15x,24("="),/,15x,"I individu numero",i4," I",/,15x,24("="),/)
write(0,30)
30 format(2x,"_Le poids se tape en derniere position ! ")
read(0,2)(a(i-1,j),j=1,m+1)
write(60,12,rec=i,err=1000)(a(i-1,j),j=1,m+1)
12 format(10le14.7)
10 continue
goto 800
C
C *****
C ** MODIFICATION **
```

```
c *****
c
600 read(60,4,rec=1,err=1000)n,m
    write(0,601)
601 format(2x,"Individu a modifier ou 0")
    read(0,2)num
    if(num.eq.0) go to 800
    read(60,12,rec=num+1,err=1000)(a(num,j),j=1,m+1)
    write(0,603)num
603 format(1x,"Individu No ",i4)
    write(0,602)(a(num,j),j=1,m+1)
602 format(2x,35(f4.2,1x))
    write(0,900)
900 format(1x,"Numero du parametre=")
    read(0,2)j
    write(0,902)a(num,j)
902 format(1x,"ancienne valeur=",f4.2,/,1x,"Donnez la nouvelle valeur:")
    read(0,2)a(num,j)
    write(60,12,rec=num+1,err=1000)(a(num,j),j=1,m+1)
61 format(i4," ",35(f4.2,1x))
950 continue
    goto 800
```

```
c
c *****
c **      IMPRESSION      **
c *****
c
700 read(60,4,rec=1,err=1000)n,m
    4 format(2i3)
    write(59,60)n,m
60 format(2x,"Nombre de types de pieces =",i4,
& /,2x,"Nombre d'operations      =",i4)
951 write(59,953)(j/10,j=1,m)
    write(59,953)(j-10*(j/10),j=1,m)
953 format(4x,35(i4,1x))
    do 952 i=2,n+1
        read(60,12,rec=i,err=1000)(a(i-1,j),j=1,m+1)
        write(59,61)i-1,(a(i-1,j),j=1,m+1)
952 continue
```

```
c
c CUMUL DES CHARGES PAR OPERATION
c
    write(0,1504)
1504 format(2x,"Calcul des charges (oui=1,non=0)")
    read(0,2)i
    if(i.ne.1)goto 1102
    write(0,1505)
1505 format(2x,"Valeur de To (calcul du nb de machines)?")
    read(0,2)to
    do 1200 j=1,m
1200 cum(j)=0
```

```
do 1100 i=1,n
  do 1100 j=1,m
    cum(j)=cum(j)+a(i,j)*a(i,m+1)
1100 continue

write(59,1500)to
1500 format(/,2x,"Cumul des charges par operation (to=",f8.3,")")
write(59,1503)
1503 format(2x,"OPERATION | CHARGE | NMAC.")
eps=1.e-4
do 1102 j=1,m
  nmac=cum(j)/to
  x=abs(float(nmac)-cum(j)/to)
  if(x.gt.eps)nmac=nmac+1
  write(59,1501)j,cum(j),nmac
1501 format(5x,i4,4x,f8.2,3x,i4)
1102 continue
write(0,*)"Resultats de l'impression dans file59"
goto 800

c
c MELANGE
c
1600 read(60,77,rec=1,err=1000)n,m
write(56,1612)n,m
1612 format(2x,"Melange de file60",/,2x,i4," individus, ",i4," parametres.")
write(0,*)"Voulez-vous melanger simultanement le fichier op/machines?"
write(0,*)" (0/1)"
read(0,*)i
sim=(i.eq.1)
if(sim)then
  do 5001 i=1,m
    read(82,53,rec=i,err=5000)nt,(nb(j),aa(j),pp(j),j=1,nt)
    write(69,53,rec=i,err=5005)nt,(nb(j),aa(j),pp(j),j=1,nt)
53 format(i1,9(i2,E14.7,E14.7))
5001 continue
  endif
do 1610 i=1,n
  pl(i)=i
1610 continue

do 1611 j=1,m
  pc(j)=j
1611 continue

do 1613 i=1,n
1613 read(60,12,rec=i+1,err=1000)(a(i,j),j=1,m+1)

do 1601 i=1,n
  call random_$uniform(x)
  al(i)=x
1601 continue
```

```
do 1602 j=1,m
  call random_$uniform(x)
  ac(j)=x
1602 continue
```

```
do 1603 i=1,n-1
  do 1603 il=i+1,n
    if(al(i).ge.al(il))goto 1603
    x=al(i)
    al(i)=al(il)
    al(il)=x
    k=pl(i)
    pl(i)=pl(il)
    pl(il)=k
1603 continue
```

```
do 1604 j=1,m-1
  do 1604 jl=j+1,m
    if(ac(j).ge.ac(jl))goto 1604
    x=ac(j)
    ac(j)=ac(jl)
    ac(jl)=x
    k=pc(j)
    pc(j)=pc(jl)
    pc(jl)=k
1604 continue
```

```
write(56,1615)(pc(j)/10,j=1,m)
write(56,1615)(pc(j)-10*(pc(j)/10),j=1,m)
1615 format(6x,35(i4,1x))
```

```
if(sim)then
  do 1607 j=1,m
    read(69,53,rec=pc(j),err=5000)nt,(nb(i),aa(i),pp(i),i=1,nt)
    write(82,53,rec=j,err=5000)nt,(nb(i),aa(i),pp(i),i=1,nt)
1607 continue
endif
```

```
do 1605 i=1,n
  do 1606 j=1,m
    t(j)=a(pl(i),pc(j))
1606 continue
  t(m+1)=a(pl(i),m+1)
  write(60,12,rec=i+1,err=1000)(t(j),j=1,m+1)
  write(56,1614)pl(i),(t(j),j=1,m+1)
1614 format(2x,i4," ",35(f4.2,1x))
1605 continue
```

```
write(0,1616)
1616 format(2x,"Melange: les positions initiales sont dans file56")
```

goto 800

```
C *****  
1000 write(0,7)  
   7 format("* ERREUR I/O FICHIER 60 *")  
     goto 9  
5000 write(0,*)"ERREUR I/O FICHIER 82"  
     goto 9  
5005 write(0,*)"ERREUR I/O FICHIER 69"  
   9  close(60)  
     close(59)  
     close(56)  
     close(82)  
     close(69)  
     stop  
     end
```

Considérons le premier tableau. Il concerne 30 types de pièces et 12 opérations. La première colonne donne les numéros des pièces et les deux premières lignes les numéros des opérations. La dernière colonne est le poids associé à chaque type de pièce. Les autres éléments du tableau sont les temps nécessaires aux produits pour subir les opérations. C'est ce tableau qu'il convient de traiter.

Nous trouvons ensuite le contenu du fichier opérations-machines. On y trouve pour chaque opération, le nombre de types de machines différents. L'un de ces type sert de base. Pour lui, la cadence relative est 1. Cela signifie que les temps que l'on trouve dans le tableau d'entrée concernent cette machine. Les autres cadences relatives donnent le rapport entre les temps figurant dans le tableau d'entrée et le temps que chaque pièce du type concerné passera sur cette machine. Le seuil économique indique la proportion minimale de temps durant lequel la machine doit fonctionner pour être rentable.

La séquence qui débute par "Programme TG" est la suite des informations fournies par le système tout au long du traitement. Dans le cas où plusieurs solutions existent, l'utilisateur choisit celle qu'il veut voir appliquer. Le logiciel prend à sa charge les conséquences de cette décision.

On donne ensuite le tableau intermédiaire qui est le classement de la matrice d'entrée par utilisation de l'algorithme donné dans [8]. C'est le prétraitement. Les poids sont donnés dans la première colonne.

On fournit enfin les îlots de fabrication et les familles de produits correspondantes. La première ligne est le numéro de l'opération. La seconde indique le type de machine concernée. La troisième ligne indique l'ordre dans lequel la machine a été prise en compte au cours de la procédure décrite précédemment.

Nombre de types de pieces = 30

Nombre d'operations = 12

	0	0	0	0	0	0	0	0	0	1	1	1	
	1	2	3	4	5	6	7	8	9	0	1	2	
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.09	0.00	3.26	0.00	0.00	3.00
2	0.00	5.84	0.00	6.70	0.00	7.70	0.00	0.00	0.00	0.00	1.38	0.00	2.00
3	0.00	0.00	8.27	0.00	0.00	0.00	0.00	8.27	0.00	6.89	0.00	0.00	5.00
4	4.51	5.94	0.00	6.71	0.00	5.14	0.00	0.00	0.00	0.00	4.78	5.51	2.00
5	1.92	0.00	0.00	0.00	6.34	0.00	0.09	0.00	1.50	0.00	0.00	0.00	1.00
6	0.00	5.79	0.00	7.30	0.00	2.86	0.00	8.33	0.00	0.00	8.82	1.42	2.00
7	0.00	0.42	0.00	7.61	0.00	5.74	0.00	0.00	0.00	6.83	8.17	0.74	3.00
8	2.22	1.11	0.35	0.96	0.00	0.00	0.00	2.59	0.00	6.49	0.00	0.00	3.00
9	0.00	2.52	0.00	0.61	0.00	0.55	0.00	0.00	0.00	0.00	7.69	0.00	1.00
10	0.00	0.00	2.34	0.00	0.00	0.00	0.00	1.93	0.00	3.40	0.00	0.00	1.00
11	0.23	0.00	0.00	0.00	2.22	0.00	7.34	0.00	2.35	0.00	0.00	0.00	2.00
12	0.00	0.00	2.27	0.00	0.00	0.00	2.72	3.63	0.00	7.07	0.00	0.00	5.00
13	0.00	0.00	8.93	0.00	0.00	0.00	0.00	8.92	0.00	6.86	0.00	0.00	4.00
14	0.86	0.00	1.79	0.00	0.00	0.00	0.00	3.51	0.00	8.40	0.00	0.00	2.00
15	0.00	1.38	0.00	4.70	0.00	0.97	0.00	0.00	0.00	0.00	7.49	0.00	1.00
16	0.00	0.00	7.04	0.00	0.00	8.78	0.00	6.05	5.59	5.12	0.00	0.00	2.00
17	0.00	4.69	0.00	5.68	0.00	1.29	0.00	0.00	0.00	2.66	2.14	1.23	3.00
18	4.48	0.00	0.00	0.00	7.84	0.00	3.43	0.00	0.99	0.00	0.00	0.00	1.00
19	0.29	6.61	0.00	0.00	1.90	0.00	8.09	6.81	7.31	0.00	0.00	0.00	4.00
20	0.00	0.00	0.73	0.00	0.00	0.00	0.00	1.49	0.00	3.80	0.00	0.00	1.00
21	0.00	3.25	0.00	3.12	0.00	7.04	0.00	0.00	0.00	0.00	3.40	3.04	3.00
22	3.26	3.07	0.00	5.26	0.00	3.47	0.00	0.00	0.00	0.00	0.26	1.27	2.00
23	0.00	0.00	6.52	0.00	0.00	0.00	0.00	6.23	0.00	0.23	1.57	0.00	4.00
24	0.00	3.85	0.00	1.83	0.00	6.76	0.00	0.00	0.00	0.00	1.26	4.49	1.00
25	2.91	0.00	0.00	0.00	6.20	7.28	5.14	0.00	7.17	0.00	0.00	0.00	1.00
26	5.66	2.66	0.00	0.00	5.20	0.00	0.00	0.00	7.92	0.00	0.42	0.00	5.00
27	7.83	0.00	4.22	0.00	0.00	0.00	0.00	0.06	0.00	5.55	0.00	0.00	3.00
28	0.00	0.00	5.51	0.00	0.00	0.00	0.00	0.00	0.00	5.48	0.00	0.00	4.00
29	2.13	0.00	0.00	0.00	6.15	0.00	5.15	0.00	1.90	0.00	0.00	0.00	1.00
30	0.00	0.00	3.46	0.00	0.00	1.49	0.00	5.96	0.00	4.95	0.00	0.00	4.00

DONNEES INITIALES

Fichier operations-machines

Operation 1 Nombre de types de machines: 1
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.80
Operation 2 Nombre de types de machines: 2
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.40
Type: 2 Nombre: 3 Cadence relative: 1.30
Seuil economique: 0.60
Operation 3 Nombre de types de machines: 3
Type: 1 Nombre: 2 Cadence relative: 1.00
Seuil economique: 0.30
Type: 2 Nombre: 1 Cadence relative: 1.10
Seuil economique: 0.60
Type: 3 Nombre: 1 Cadence relative: 1.60
Seuil economique: 0.50
Operation 4 Nombre de types de machines: 1
Type: 1 Nombre: 3 Cadence relative: 1.00
Seuil economique: 0.50
Operation 5 Nombre de types de machines: 2
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.30
Type: 2 Nombre: 1 Cadence relative: 1.50
Seuil economique: 0.60
Operation 6 Nombre de types de machines: 2
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.30
Type: 2 Nombre: 1 Cadence relative: 1.50
Seuil economique: 0.50
Operation 7 Nombre de types de machines: 1
Type: 1 Nombre: 2 Cadence relative: 1.00
Seuil economique: 0.40
Operation 8 Nombre de types de machines: 3
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.40
Type: 2 Nombre: 1 Cadence relative: 1.50
Seuil economique: 0.70
Type: 3 Nombre: 1 Cadence relative: 1.80
Seuil economique: 0.75
Operation 9 Nombre de types de machines: 3
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.50
Type: 2 Nombre: 1 Cadence relative: 1.20
Seuil economique: 0.60
Type: 3 Nombre: 1 Cadence relative: 1.40
Seuil economique: 0.70
Operation 10 Nombre de types de machines: 3
Type: 1 Nombre: 1 Cadence relative: 1.00
Seuil economique: 0.50

Type: 2 Nombre: 1 Cadence relative: 1.20
Seuil economique: 0.70
Type: 3 Nombre: 1 Cadence relative: 2.00
Seuil economique: 0.60
Operation 11 Nombre de types de machines: 1
Type: 1 Nombre: 3 Cadence relative: 1.00
Seuil economique: 0.65
Operation 12 Nombre de types de machines: 1
Type: 1 Nombre: 2 Cadence relative: 1.00
Seuil economique: 0.30

PROGRAMME TG : Suite des messages qui apparaissent à l'écran lors de l'exécution.
Les chiffres encadrés sont fournis par l'utilisateur.

Nombre de types de produits: 30 Nombre d'operations: 12

Periode de travail consideree= 100.000

Groupement initial des produits et des operations

Nombre maximal de classes souhaitees = 3
Coefficient de ponderation = 0.500

Choix initial des representants : Automatique
Distance mini entre points initiaux= 1.700
Valeur finale du critere= 417.000

3 Classe(s)

Taille des groupements:

Classe: 1
3 operation(s), 13 Produit(s)
Classe: 2
5 operation(s), 10 Produit(s)
Classe: 3
4 operation(s), 7 Produit(s)

Traitement principal

Operation: 1

L'operation 1 sera prise en charge par une machine

Operation: 5

Solution(s) possible(s) pour l'operation 5

* Solution numero: 1 : COEFF= 0.494

1 machine(s) de type 1

0 machine(s) de type 2

La seule solution possible est retenue

L'operation 5 sera prise en charge par une machine

Operation: 6

Solution(s) possible(s) pour l'operation 6

* Solution numero: 1 : COEFF= 0.595

0 machine(s) de type 1

1 machine(s) de type 2

* Solution numero: 2 : COEFF= 0.101

1 machine(s) de type 1

1 machine(s) de type 2

Solution choisie: 1
L'operation 6 sera prise en charge par une machine

Operation: 7

Solution(s) possible(s) pour l'operation 7
* Solution numero: 1 : COEFF= 0.574
1 machine(s) de type 1
La seule solution possible est retenue
L'operation 7 sera prise en charge par une machine

Operation: 12

Solution(s) possible(s) pour l'operation 12
* Solution numero: 1 : COEFF= 0.085
1 machine(s) de type 1
La seule solution possible est retenue
L'operation 12 sera prise en charge par une machine

Operation: 4

Solution(s) possible(s) pour l'operation 4
* Solution numero: 1 : COEFF= 0.112
2 machine(s) de type 1
La seule solution possible est retenue

Operation: 8

Solution(s) possible(s) pour l'operation 8
* Solution numero: 1 : COEFF= 0.872
1 machine(s) de type 1
1 machine(s) de type 2
0 machine(s) de type 3
* Solution numero: 2 : COEFF= 0.587
1 machine(s) de type 1
0 machine(s) de type 2
1 machine(s) de type 3
Solution choisie: 1

Operation: 9

Solution(s) possible(s) pour l'operation 9
* Solution numero: 1 : COEFF= 0.926
1 machine(s) de type 1
0 machine(s) de type 2
0 machine(s) de type 3
* Solution numero: 2 : COEFF= 0.506
0 machine(s) de type 1
1 machine(s) de type 2
0 machine(s) de type 3
Solution choisie: 1

L'operation 9 sera prise en charge par une machine

Operation: 10

Solution(s) possible(s) pour l'operation 10

* Solution numero: 1 : COEFF= 0.604

1 machine(s) de type 1

0 machine(s) de type 2

1 machine(s) de type 3

* Solution numero: 2 : COEFF= 0.384

0 machine(s) de type 1

1 machine(s) de type 2

1 machine(s) de type 3

Solution choisie: 1

Operation: 11

Solution(s) possible(s) pour l'operation 11

* Solution numero: 1 : COEFF= 0.898

1 machine(s) de type 1

La seule solution possible est retenue

L'operation 11 sera prise en charge par une machine

Operation: 2

Solution(s) possible(s) pour l'operation 2

* Solution numero: 1 : COEFF= 0.753

0 machine(s) de type 1

1 machine(s) de type 2

La seule solution possible est retenue

L'operation 2 sera prise en charge par une machine

Operation: 3

Solution(s) possible(s) pour l'operation 3

* Solution numero: 1 : COEFF= 0.892

2 machine(s) de type 1

0 machine(s) de type 2

0 machine(s) de type 3

* Solution numero: 2 : COEFF= 0.779

1 machine(s) de type 1

1 machine(s) de type 2

0 machine(s) de type 3

* Solution numero: 3 : COEFF= 0.320

2 machine(s) de type 1

1 machine(s) de type 2

0 machine(s) de type 3

* Solution numero: 4 : COEFF= 0.499

1 machine(s) de type 1

0 machine(s) de type 2

1 machine(s) de type 3

* Solution numero: 5 : COEFF= 0.204
2 machine(s) de type 1
0 machine(s) de type 2
1 machine(s) de type 3
* Solution numero: 6 : COEFF= 0.313
0 machine(s) de type 1
1 machine(s) de type 2
1 machine(s) de type 3
* Solution numero: 7 : COEFF= 0.045
1 machine(s) de type 1
1 machine(s) de type 2
1 machine(s) de type 3
Solution choisie: 4

Groupement final des machines

coefficient= 0.500

Valeur finale du critere= 504.500

Operation			0	0	0	0	0	0	0	0	0	0	0
Machine			0	0	0	0	0	0	0	0	0	0	0
Position			10	8	3	12	11	6	4	2	9	7	5
Classe			1	1	1	2	2	2	2	2	3	3	3
4.00	30	1	1	1	1	0	0	1	0	0	0	0	0
4.00	28	1	1	0	1	0	0	0	0	0	0	0	0
3.00	27	1	1	1	1	0	0	0	0	0	0	0	1
4.00	23	1	1	1	1	0	1	0	0	0	0	0	0
1.00	20	1	1	1	1	0	0	0	0	0	0	0	0
2.00	16	1	1	1	1	0	0	1	0	0	1	0	0
2.00	14	1	1	1	1	0	0	0	0	0	0	0	1
4.00	13	1	1	1	1	0	0	0	0	0	0	0	0
5.00	12	1	1	1	1	0	0	0	0	0	0	1	0
1.00	10	1	1	1	1	0	0	0	0	0	0	0	0
3.00	8	1	1	1	1	0	0	0	1	1	0	0	1
5.00	3	1	1	1	1	0	0	0	0	0	0	0	0
3.00	1	1	1	1	0	0	0	0	0	0	0	0	0
1.00	24	2	0	0	0	1	1	1	1	1	0	0	0
2.00	22	2	0	0	0	1	1	1	1	1	0	0	1
3.00	21	2	0	0	0	1	1	1	1	1	0	0	0
3.00	17	2	1	0	0	1	1	1	1	1	0	0	0
1.00	15	2	0	0	0	0	1	1	1	1	0	0	0
1.00	9	2	0	0	0	0	1	1	1	1	0	0	0
3.00	7	2	1	0	0	1	1	1	1	1	0	0	0
2.00	6	2	0	1	0	1	1	1	1	1	0	0	0
2.00	4	2	0	0	0	1	1	1	1	1	0	0	1
2.00	2	2	0	0	0	0	1	1	1	1	0	0	0
1.00	29	3	0	0	0	0	0	0	0	0	1	1	1
5.00	26	3	0	0	0	0	1	0	0	1	1	0	1
1.00	25	3	0	0	0	0	0	1	0	0	1	1	1
4.00	19	3	0	1	0	0	0	0	0	1	1	1	1
1.00	18	3	0	0	0	0	0	0	0	0	1	1	1
2.00	11	3	0	0	0	0	0	0	0	0	1	1	1
1.00	5	3	0	0	0	0	0	0	0	0	1	1	1

PRETRAITEMENT : CLASSIFICATION CROISEE DES
 PRODUITS ET DES OPERATIONS

Operation			3	10	8	7	2	11	4	4	12	6	3	10	9	8	5	1
Machine			2	3	3	1	1	1	1	1	1	1	1	2	1	1	1	1
Position			16	12	9	4	14	13	7	6	5	3	15	11	10	8	2	1
Classe			1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
5.00	12	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
1.00	10	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3.00	8	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1
5.00	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3.00	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00	24	2	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0
2.00	22	2	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	1
3.00	21	2	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0
3.00	17	2	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0
1.00	15	2	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
1.00	9	2	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
3.00	7	2	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0
2.00	6	2	0	0	1	0	1	1	0	1	1	1	0	0	0	0	0	0
2.00	4	2	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	1
2.00	2	2	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0
4.00	30	3	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0
1.00	29	3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1
4.00	28	3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
3.00	27	3	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
5.00	26	3	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1
1.00	25	3	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1
4.00	23	3	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0
1.00	20	3	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
4.00	19	3	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1
1.00	18	3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1
2.00	16	3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
2.00	14	3	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
4.00	13	3	1	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0
2.00	11	3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1
1.00	5	3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1

ILOTS CONSTITUES

BIBLIOGRAPHIE

- [1] : Mc. AULEY
"Machine grouping for efficient production",
Production Engineering, février 72.
- [2] : J.P. KING,
"Machine component grouping in production flow analysis. An approach
using order clustering algorithm",
Int. J. Prod. Res., vol. 18, n°2, 1980.
- [3] : J.L. LUBRIDGE,
"Production flow analysis. Production engineering",
n° 42, 1963.
- [4]: G. DOUMEINGTS and all
"Méthodologie de conception de systèmes flexibles de fabrication"
Revue de l'ADETTA, 1984.
- [5] : L.E. STANFEL,
"Partitioning problems in cellular manufacturing",
Conférence Systèmes de Production, Approche scientifique, économique,
stratégique, 23-26 avril 1985.
- [6] : Y. LEMAIRE et B. MUTEL,
"Automatic recognition of production cells and part families",
Colloque PROLAMAT, LENINGRAD, 1982.
- [7] : A. BROYAND and all,
"La TGAO : problèmes d'implantation et d'utilisation",
Colloque des outils de la productique, Paris, décembre 84.
- [8] : H. GARCIA et J.M. PROTH,
"Group technology in production management",
Applied stochastic models and data analysis, vol. 1, n° 1, July 85, p 25-34

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

