Un système de fabrication avec contrôle de la qualité intégré

Jean-Marie Proth

IRIA
CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (3) 954 90 20

Rapports de Recherche

N° 341

UN SYSTÈME DE FABRICATION AVEC CONTRÔLE DE LA QUALITÉ INTÉGRÉ

Jean-Marie PROTH

Octobre 1984

UN SYSTEME DE FABRICATION

AVEC

CONTROLE DE LA QUALITE INTEGRE

J.M. PROTH

ABSTRACT

This paper is devoted to a two levels production system with high production rate and integrated quality control. Only one machine works at the highest level and supplies with parts alternatively the two machines from the lowest level. The quality control is implemented at the input of each machine of the lowest level. This production system is able to produce various types of parts. The goal consists in minimizing the total cost which is the sum of the cost for shifting tools at the highest level and the cost for rejecting batches which contain at leastone defective part. We consider the particular case with indentical machines at the lowest level and the general case.

RESUME.

Ce papier est consacré a un système de production à deux niveaux qui a une cadence de fabrication élevée et un contrôle de la qualité intégré. Le niveau haut comporte une seule machine qui approvisionne alternativement les deux machines du niveau bas. Le contrôle de la qualité est implémenté à l'entrée de chaque machine du niveau bas. Ce système de production est capable de produire des pièces de types différents. L'objectif consiste à minimiser le coût total conséquence du changement d'outils au niveau haut et du rejet des lots qui contiennent au moins une pièce défectueuse. Nous étudions le cas particulier (machines identiques au niveau bas) et le cas général.

I. INTRODUCTION.

Ce rapport étudie un modèle de contrôle de la qualité intégré à la fabrication. Plusieurs types de produits peuvent être obtenus à l'aide du système de fabrication étudié, et chacun d'eux subit deux opérations. La première (dite "de niveau bas") peut être prise en charge indifféremment par deux machines que le niveau haut approvisionne alternativement. Ce fonctionnement implique que la machine de niveau haut ait un débit sensiblement égal à la somme des débits des machines de niveau bas.

On procède par lots homogènes, c'est à dire constitués d'un seul type de produits. Les temps de changement d'outils étant très importants au niveau bas, on impose à tous les lots concernant un même produit de passer sur la même machine et successivement.

Chaque machine du niveau bas est précédée d'un stockeur qui se remplit durant l'approvisionnement et nourrit la machine lorsque le niveau haut se consacre à l'autre machine du niveau bas. Lorsqu'une machine du niveau bas est alimentée, une pièce sur deux est envoyée en stock. Ces pièces stockées sont testées pour déterminer si l'opération du niveau haut s'est déroulée correctement. Si l'une des pièces testées est trouvée défectueuse, toutes les pièces du lot sont rejetées. On comprend que le coût associé au rejet est une fonction croissante de la taille du lot.

A l'opposé, si l'on réduit la taille des lots, on augmente le nombre de changements d'outils du niveau haut, donc le coût associé. Il faut arbitrer entre ces deux tendances.

Notons enfin que les temps de passage des lots sur les machines du niveau bas auxquelles ils sont affectés sont tous égaux : c'est le seul moyen d'utiliser pleinement les machines du niveau bas.

Nous adopterons deux hypothèses simplificatrices :

1. les stockeurs sont vides à l'instant initial et la machine de niveau bas non utilisée d'entrée n'entraîne aucun coût d'immobilisation : on considère qu'elle termine la commande de la période précédente

2. les clients acceptent de se voir livrer un nombre entier de lots, à condition que la quantité livrée reste dans des limites raisonnables.

Cette contrainte se traduit par la donnée d'un maximum de pièces par lot, et ceci pour chaque type de pièces.

II. FORMULATION DU PROBLEME.

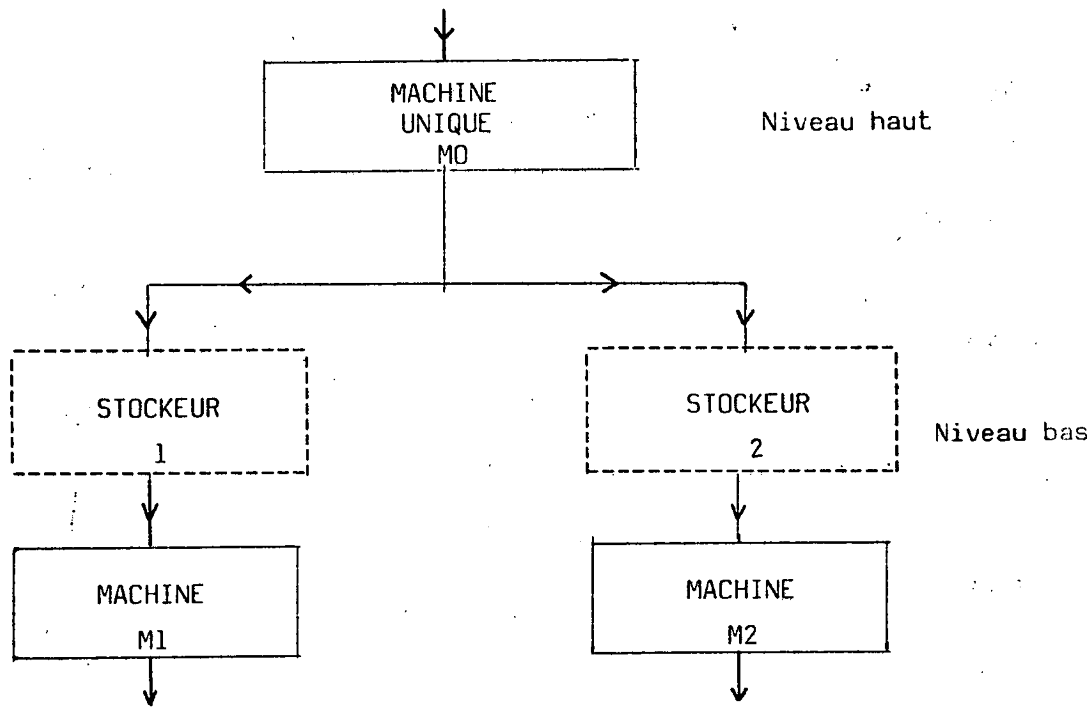

Fig. 1

La figure 1 schématise le système.

A l'instant initial, n produits différents sont à fabriquer.

Pour $i = 1,2,\ldots,n$:

Q_i est la quantité de produit i demandée.

p_i est la probabilité de rejet d'une unité de produit i à la suite du test

d_i^j $(j=0,1,2)$ est le nombre moyen d'unités de produit i que M_j est capable de fabriquer par unité de temps. On suppose que le débit des machines est suffisamment souple pour que ces quantités ne dépendent pas du nombre de changements d'outils

k_i est le coût entraîné par le rejet d'une unité de produit i

x_i, inconnue de notre problème, est le nombre d'unités de produit i par lot

w_i est la taille maximale d'un lot de produit i

z_i est la taille minimale d'un lot de produit i, exigée pour assurer aux tests une fiabilité minimale.

Nous notons L le coût de changement d'outil au niveau haut. Les coûts de changement d'outils au niveau bas ne sont pas pris en considération car la politique adoptée les minimise dans tous les cas de figure.

Nous désignons encore par T_j (j=1,2) l'ensemble des produits traités au niveau bas par la machine M_j.

Nous noterons $r(x_i)$ le nombre de lots de produits i à fabriquer si le nombre d'unités par lot est x_i :

$$r(x_i) = \begin{cases} Q_i/x_i & \text{si } Q_i/x_i \text{ est entier} \\ [Q_i/x_i] + 1 & \text{sinon} \end{cases} \tag{1}$$

où [.] désigne la partie entière.

Le coût résultant des changements d'outils au niveau haut s'écrit, si $X = (x_1, x_2, \ldots, x_n)$ est l'ensemble des tailles des lots :

$$C_1(X) = L \sum_{j=1}^{2} \sum_{i \in I_j} r(x_i) \tag{2}$$

Nous supposons x_i pair. La probabilité pour qu'une pièce au moins prise parmi les $x_i/2$ pièces stockées soit jugée défectueuse est :

$$P(x_i) = 1 - (1-p_i)^{x_i/2} \tag{3}$$

Cette formulation suppose l'indépendance entre les états des pièces testées.

Le coût associé au rejet des pièces défectueuses s'écrit finalement, compte tenu de (3) :

$$C_2(X) = \sum_{i=1}^{n} r(x_i) \quad [1 - (1-p_i)^{x_i/2}] \; k_i$$

Note : Si x_i est impair, il faut remplacer $x_i/2$ par $(x_i+1)/2$ dans la formulation ci-dessus. Nous ferons l'approximation de toujours garder $x_i/2$: les x_i étant importants, cela ne modifie par sensiblement le résultat.

Soit :

$$C_2(X) = \sum_{j=1}^{2} \sum_{i \in I_j} k_i \; r(x_i) \; [1 - (1-p_i)^{x_i/2}] \tag{4}$$

Le coût à minimiser s'écrit finalement :

$$\begin{aligned} C(X) &= C_1(X) + C_2(X) \\ &= \sum_{j=1}^{2} \sum_{i \in I_j} r(x_i) \; \{L + k_i \; [1 - (1-p_i)^{x_i/2}]\} \end{aligned} \tag{5}$$

Pour que les machines de niveau bas fonctionnent sans interruption, il faut que tous les lots aient le même temps de passage sur la machine de niveau bas qui les traite.

Cette contrainte s'écrit :

$$\frac{x_i}{d_i^j} = K, \quad \forall \; i \in I_j, \; j = 1,2 \tag{6}$$

où K est une constante positive.

Compte tenu de (6), le coût (5) se réécrit :

$$C(K) = \sum_{j=1}^{2} \sum_{i \in I_j} r_i^*(K) \; \{L+k_i \; [1 - (1-p_i)^{K \; d_i^j/2}]\} \tag{7}$$

avec (c.f. (1)) :

$$r_i^*(K) = \begin{cases} Q_i/(K \; d_i^j) & \text{si } i \in I_j \text{ et } Q_i/(K \; d_i^j) \text{ est entier} \\ [Q_i/(K \; d_i^j)] + 1 & \text{si } i \in I_j \text{ et } Q_i/(K \; d_i^j) \text{ n'est pas entier.} \end{cases} \tag{8}$$

Nous savons que la taille des lots est bornée pour chaque type de pièce :

$$z_i \leq x_i \leq w_i, \quad i = 1,2,\ldots,n$$

Cette contrainte se réécrit, compte tenu de (6) :

$$z_i/d_i^j \leq K \leq w_i/d_i^j, \quad \forall\, i \in I_j, \quad j = 1,2$$

d'où

$$\underset{j=1,2}{\text{Max}} \left[\underset{i \in I_j}{\text{Max}} \; z_i/d_i^j \right] \leq K \leq \underset{j=1,2}{\text{Min}} \left[\underset{i \in I_j}{\text{Min}} \; w_i/d_i^j \right] \tag{9}$$

La minimisation de (7) sous la contrainte (9) doit se faire en retenant, pour chaque valeur de K, la partition $\{I_1(K), I_2(K)\}$ de $\{1,2,\ldots,n\}$ qui minimise l'écart entre les temps de fonctionnement des deux machines, ce que nous écrivons :

$$\left| \sum_{i \in I_1(K)} r_i^*(K) - \sum_{i \in I_2(K)} r_i^*(K) \right| \tag{10}$$

Le problème s'énonce finalement :

chercher K_1 tel que :

$$C(K_1) = \underset{K}{\text{Min}}\; C(K)$$

où : (11)

1. pour chaque valeur de K, la partition $\{I_1(K), I_2(K)\}$ retenue est celle qui minimise (10)
2. K vérifie (9).

Les $r_i^*(K)$ $(i=1,2,\ldots,n)$ sont donnés par (8).

Nous traitons d'abord le cas particulier où les deux machines de niveau bas sont identiques.

III. CAS PARTICULIER : LES DEUX MACHINES DE NIVEAU BAS SONT IDENTIQUES.

Dans ce cas, le nombre de produits i (i=1,2,...,n) qu'une machine de niveau bas est capable de fabriquer par unité de temps ne dépend pas de la machine :

$$d_i^1 = d_i^2 = d_i \quad \text{pour } i = 1,2,\ldots,n \tag{12}$$

Le problème s'énonce alors :

(A) $$\text{minimiser} \sum_{i=1}^{n} r_i^*(K) \left\{L + k_i \left[1 - (1-p_i)^{Kd_i/2}\right]\right\} = C(K) \tag{13}$$

sachant que :

1. $$\max_{i=1,\ldots,n} (z_i/d_i) \leq K \leq \min_{i=1,\ldots,n} (w_i/d_i) \tag{14}$$

2. $$r_i^*(K) = \begin{cases} Q_i/(K\ d_i) & \text{si } Q_i/(K\ d_i) \text{ est entier} \\ \left[Q_i/(K\ d_i)\right] + 1 & \text{sinon} \end{cases} \tag{15}$$

(B) choisir alors la partition $\{I_1(K),\ I_2(K)\}$ qui minimise (10).

La simplification consiste en la séparation de la recherche de la valeur optimale de C(K) et de la recherche de la partition $\{I_1(K),\ I_2(K)\}$.

III.1. Recherche de l'optimum de C(K).

Nous montrons d'abord que l'optimum est à rechercher dans un ensemble fini de valeurs de K.

Pour simplifier les notations nous noterons :

$$\alpha = \max_{i=1,\ldots,n} (z_i/d_i)$$

$$\beta = \min_{i=1,\ldots,n} (w_i/d_i)$$

et nous supposons bien entendu que $\alpha \leq \beta$.

Résultat 1.

C(K) atteind son minimum :

1. soit pour $K = \alpha$
2. soit pour $K \in (\alpha,\beta]$ et tel que $Q_i/(K\ d_i)$ est entier pour au moins un $i \in \{1,2,...,n\}$.

Démonstration.

Quel que soit $i \in \{1,2,...,n\}$:

a) $L + k_i\ [1 - (1-p_i)^{K\ d_i/2}]$ est une fonction croissante de K

b) $r_i^*(K)$ est une fonction constante par morceaux et décroissante de K.

Donc :

$$f_i(K) = r_i^*\ (K)\ \{L + k_i\ [1 - (1-p_i)^{K\ d_i/2}]\} \tag{16}$$

est une fonction croissante par morceaux de K.

Si $K_1^i, K_2^i, ..., K_{h(i)}^i$ sont les valeurs de K appartenant à $(\alpha,\beta]$ et telles que $Q_i/(K\ d_i)$ est entier, alors $f_i(K)$ est donné par la figure 2.

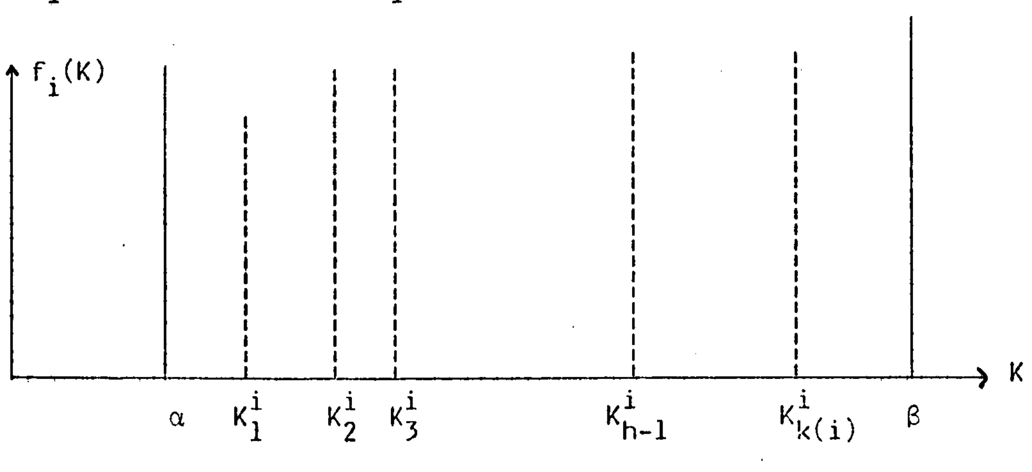

Fig. 2

On voit donc que chaque fonction $f_i(K)$ atteind son minimum en un point de l'ensemble :

$$\{\alpha, K_1^i, K_2^i, \ldots, K_{h(i)}^i\}$$

Considérons maintenant

$$C(K) = \sum_{i=1}^{n} f_i(K)$$

et supposons que le minimum soit obtenu pour $K^* \in [\alpha,\beta]$,

avec

$$K^* \notin \{\alpha, \{K_\ell^i\}, i = 1,\ldots,n \text{ et } \ell = 1,\ldots, h(i)\} = E \tag{16}$$

Soit K^0 le plus grand élément de E inférieur à K^*.

Alors, pour $i = 1,2,\ldots,n$;

a) $r_i^*(K^0) = r_i^*(K^*)$

b) $k_i\ [1 - (1-p_i)^{K^0 d_i/2}] \leq k_i\ [1 - (1-p_i)^{K^* d_i/2}]$

et $C(K^0) \leq C(K^*)$, ce qui achève la démonstration. □

Le résultat que nous venons de montrer passe par la recherche de E (voir(16)).

Nous donnons l'algorithme qui aboutit à E. Les éléments de E sont rangés dans le tableau K. Cet algorithme ne donne pas la suppression éventuelle des éléments multiples, ni le classement.

1. Faire M = 0
2. <u>Pour i = 1,2,...,n</u>
    - <u>2.1 pour N = 1,2,3,...</u>
        - 2.1.1. Faire $X = Q_i/(N\ d_i)$
        - <u>2.1.2. Si $X > \beta$</u>
            - 2.1.2.1. Aller en 2.2

2.1.3. Sinon

2.1.3.1. Si $X < \alpha$

2.1.3.1.1. Aller en 3.

2.1.3.2. Sinon

2.1.3.2.1. Faire $M = M + 1$

2.1.3.2.2. Faire $K(M) = X$

2.1.3.2.3. Aller en 2.2

2.2. Fin de boucle N

3. Fin de boucle i

4. Faire $M = M + 1$

5. Faire $K(M) = \alpha$

Il suffit alors de calculer (voir (13) et (15))

$$C(K(\ell)) \quad \text{pour } \ell = 1,2,\dots,M$$

et de conserver ℓ_1 tel que :

$$C(K(\ell_1)) = \min_{\ell=1,2,\dots,M} C(K(\ell))$$

$K(\ell_1)$ représente alors le temps optimal de passage d'un lot quelconque sur une machine de niveau bas et $C(K(\ell_1))$ est le coût associé.

Les tailles de lots sont donc, à l'optimum, les suivantes :

$$x_i = K(\ell_1) \cdot d_i \quad \text{pour } i = 1,2,\dots,n.$$

III.2. Recherche de la partition $\{I_1(K(\ell_1)),\ I_2\ (K(\ell_1))\}$

La connaissance de $K(\ell_1)$ entraîne celle de

$$r_i^*\ (K(\ell_1)) \quad \text{pour } i = 1,2,\dots,n.$$

Le problème consiste maintenant à rechercher $I_1(K(\ell_1))$ et $I_2(K(\ell_1))$ tels que :

$$\left| \sum_{i \in I_1(K(\ell_1))} r_i^* \,(K(\ell_1)) - \sum_{i \in I_2(K(\ell_1))} r_i^* \,(K(\ell_1)) \right|$$

soit minimal.

Ce problème est combinatoire est nécessite en théorie 2^n essais de répartition des n produits sur les deux machines. Dans la pratique, il est possible de réduire considérablement ce nombre d'essais. Il reste cependant élevé. Nous ne développerons pas cet aspect. On voit cependant l'importance d'avoir pu dissocier la recherche de l'optimum de C(X) de la répartition des produits sur les machines. Cela ne sera plus possible dans le cas général.

III.3. Un exemple numérique.

Nous donnons un exemple numérique simple pour illustrer l'algorithme précédent.

Nous considérons 7 produits.

Les données sont les suivantes :

$Q_1 = 9000 \quad Q_2 = 20000 \quad Q_3 = 60000 \quad Q_4 = 8000 \quad Q_5 = 12000 \quad Q_6 = 24000 \quad Q_7 = 30000$

$p_i = 10^{-3}$ pour $i = 1,2,\dots,7$

$d_1 = 1000 \quad d_2 = 4000 \quad d_3 = 6000 \quad d_4 = 2000 \quad d_5 = 4000 \quad d_6 = 6000 \quad d_7 = 6000$

$k_1 = 7 \quad k_2 = 2 \quad k_3 = 4 \quad k_4 = 3 \quad k_5 = 2 \quad k_6 = 1 \quad k_7 = 1$

$z_i = 1000$ et $w_i = 9000$ pour $i = 1,2,\dots,7$.

La contrainte (14) s'écrit ici :

$$1 \leqq K \leqq 1.5$$

L'ensemble E obtenu est, après classement de ses éléments :

$$E = \{1,\ 10/9,\ 9/8,\ 5/4,\ 9/7,\ 4/3,\ 10/7,\ 3/2\}$$

A chacune de ces valeurs correspond une séquence r_1^*, $r_2^*,\dots,r_7^*$. Le tableau suivant regroupe ces résultats, calculés à l'aide de (15).

K	$r_1(K)$	$r_2(K)$	$r_3(K)$	$r_4(K)$	$r_5(K)$	$r_6(K)$	$r_7(K)$
1	9	5	10	4	3	4	5
10/9	9	5	9	4	3	4	5
9/8	8	5	9	4	3	4	5
5/4	8	4	8	4	3	4	4
9/7	7	4	8	4	3	4	4
4/3	7	4	8	3	3	3	4
10/7	7	4	7	3	3	3	4
3/2	6	4	7	3	2	3	4

Il suffit alors de chercher les 8 valeurs de C(K) à partir des 8 lignes de ce tableau et de conserver la valeur minimale. Il restera ensuite à chercher la partition des $r_i(K)$ correspondants qui minimisent (10). Cette seconde phase, rappelons le, nécessite l'utilisation d'une heuristique, compte tenu de son aspect combinatoire.

. LE CAS GENERAL.

Le cas particulier que nous venons d'étudier permettait de dissocier la recherche du minimum de C(K) de la recherche de la partition $\{I_1, I_2\}$. Ce n'est plus possible dans le cas général car la relation (12) n'est plus vérifiée.

Tel qu'il est donné en (11), le problème se révèle extrêmement complexe car il demande que soit résolu un problème combinatoire pour différents ensembles de valeurs de K.

Nous choississons de transformer le problème en :

1. choisissant d'abord la partition $\{I_1, I_2\}$ de façon à minimiser une combinaison linéaire de deux critères
2. minimisant ensuite (7), les critères précédents étant connus.

IV.1. Choix de la partition.

Soit $\lambda \in (0,1)$ donné par l'utilisateur.

Nous choissons de minimiser

$$\begin{aligned} &\lambda \left\{ \left| \sum_{i=1}^{n} \delta_i \; Q_i/d_i^1 - \sum_{i=1}^{n} (1-\delta_i) \; Q_i/d_i^2 \right| \right\} \\ &+ (1-\lambda) \left\{ \sum_{i=1}^{n} \delta_i \; Q_i/d_i^1 + \sum_{i=1}^{n} (1-\delta_i) \; Q_i/d_i^2 \right\} \end{aligned} \tag{17}$$

où δ_i = 0 ou 1 pour i = 1,2,...,n.

Si le produit i passe sur la machine 1 lorsque $\delta_i = 1$ et sur la machine 2 lorsque $\delta_i = 0$:

a) l'expression en facteur de λ exprime l'écart entre les temps de fonctionnement des deux machines
b) l'expression en facteur de $(1-\lambda)$ est le temps total d'utilisation de l'ensemble des deux machines (i.e. la somme des temps de fabrication de l'ensemble des produits).

Nous cherchons à minimiser une somme pondérée de ces critères, la pondération (obtenue par la valeur donnée à λ) dépendant de l'utilisateur.

Si n est relativement petit ($n \leqq 20$) on peut évidemment calculer la valeur de (17) pour chacune des 2^n valeurs possibles du n-uple :

$$\{\delta_1, \delta_2, \ldots, \delta_n\}$$

Sinon, seule une heuristique peut donner une solution satisfaisante en un temps raisonnable.

Nous donnons une heuristique possible :

1. Choix d'une partition initiale

   1.1 Pour $i = 1,2,\ldots,n$

      1.1.1. Si $d_i^1 < d_i^2$, affecter le produit i à I_2^0

      1.1.2. Sinon, affecter i à I_1^0

   1.2. Fin de boucle i.

2. Amélioration de la partition initiale.

   2.1. Soit C^0 la valeur de (17) pour la partition $\{I_1^0, I_2^0\}$

   2.2. Pour $i = 1,2,\ldots,n$

      2.2.1. On construit la partition $\{J_1^i, J_2^i\}$ de la manière suivante :

         2.2.1.1. Si $i \in I_1$

$$\begin{cases} J_1^i = I_1^{i-1} - \{i\} \\ J_2^i = I_2^{i-1} + \{i\} \end{cases}$$

         2.2.1.2. Si $i \in I_2$

$$\begin{cases} J_1^i = I_1^{i-1} + \{i\} \\ J_2^i = J_2^{i-1} - \{i\} \end{cases}$$

      2.2.2 Calcul de H^i, valeur de (17) pour la partition $\{J_1^i, J_2^i\}$

   2.3. Fin de boucle i

   2.4. Recherche de H^* t.q. :

$$H^* = \underset{i=1,2,\ldots,n}{\mathrm{Min}} \quad H^i$$

soit i_1 t.q. $H^* = H^{i_1}$.

2.5. Test sur H*

2.5.1. Si $H^* < C^{i-1}$

2.5.1.1. Faire $C^i = H^*$

2.5.1.2. Faire

$$I_1^i = J_1^i 1$$
$$I_2^i = J_2^i 1$$

2.5.1.3. Aller en 2.2.

2.5.2. Si $H^* \geq C^{i-1}$

2.5.2.1. On choisit la partition :

$$I_1 = I_1^{i-1}$$
$$I_2 = I_2^{i-1}$$

2.5.2.2. Fin de processus.

En résumé, cet algorithme effectue une répartition initiale des produits sur les machines de façon à minimiser le temps total de fabrication.

Il tente ensuite de diminuer la valeur du critère (17) en faisant passer certains produits du sous-ensemble I_1 au sous-ensemble I_2 et/ou réciproquement.

On observe que :

1. la recherche de la partition ne tient pas compte de la taille des lots, mais simplement des quantités à produire. La valeur du critère (17) obtenue à l'aide de l'algorithme précèdent ne sera donc probablement pas atteinte.
2. l'algorithme améliore la valeur du critère (17) en faisant passer, à chaque pas d'itération, un produit de l'ensemble I_j à l'ensemble I_{3-j} $(j=1,2)$. En général, on obtiendra donc pas la valeur minimale de (17).

IV.2. Minimisation du coût (c.f. (7)).

La partition $\{I_1, I_2\}$ est celle que nous avons trouvée à l'aide de l'algorithme précédent.

Dans l'expression du coût (7), nous pouvons donc remplacer d_i^j par d_i^1 si

$i \in I_1$ et d_i^j par d_i^2 si $i \in I_2$ $(i=1,2,\dots,n)$.

Si nous appelons d_i la valeur d_i^j ainsi définie, la recherche de l'optimum de C(K) est alors identique à la recherche développée en III.1.

Remarques :

1. Il est possible, lorsque la valeur de (17) obtenue à la sortie de IV.2. est mauvaise, d'envisager une pénalisation des d_i^j et d'itérer. Nous ne développerons pas cet aspect ici.
2. La valeur de λ est fournie par l'utilisateur. Proche de 1, elle favorise la recherche d'un faible écart entre les temps de fonctionnement des deux machines. Proche de 0, elle favorise un faible temps de fabrication total.

CONCLUSION.

On aura observé que, lorsque les machines sont identiques, on commence par rechercher la taille des lots qui optimise la fonction de coût, puis on effectue la répartition des produits sur les machines en fonction de la valeur trouvée pour K. Lorsque les machines sont différentes, on introduit une hypothèse simplificative supplémentaire qui permet de se libérer de la taille des lots pour définir la répartition des produits sur les machines. C'est seulement après avoir effectué cette répartition que l'on optimise le coût.

On pourrait améliorer cette heuristique en pénalisant les nombres d'unités de produits que les machines du niveau bas sont capables de produire par unité de temps. Cette démarche sort du cadre de ce papier.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique