

H. P. HILLION

J. M. PROTH

**Analyse de fabrications non linéaires et répétitives  
à l'aide de graphes d'événements temporisés**

*RAIRO. Recherche opérationnelle*, tome 22, n° 2 (1988),  
p. 137-176

[http://www.numdam.org/item?id=RO\\_1988\\_\\_22\\_2\\_137\\_0](http://www.numdam.org/item?id=RO_1988__22_2_137_0)

© AFCET, 1988, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

## ANALYSE DE FABRICATIONS NON LINÉAIRES ET RÉPÉTITIVES A L'AIDE DE GRAPHERS D'ÉVÉNEMENTS TEMPORISÉS (\*)

par H. P. HILLION <sup>(1)</sup> et J. M. PROTH <sup>(2)</sup>

---

**Résumé.** — Nous nous intéressons aux systèmes de production non linéaires avec fabrications répétitives. Nous montrons comment les modéliser à l'aide de graphes d'événements temporisés. Cette modélisation nous permet de mettre en évidence qu'il est toujours possible de saturer la machine la plus chargée, donc de maximiser la productivité du système. Nous montrons comment y parvenir avec un minimum d'en-cours lorsque les séquences d'entrée des pièces dans les machines sont fixées. Cet aspect est particulièrement important dans le cas de systèmes de production flexibles qui utilisent des ressources pour le transport des pièces : minimiser les en-cours est alors équivalent à minimiser le nombre de ressources de transport (chariots, palettes, balancelles, ...), lesquelles sont toujours très coûteuses.

Nous posons ensuite le problème de la recherche de séquences d'entrée des pièces dans les machines qui soient optimales, c'est-à-dire qui permettent d'atteindre la productivité maximale du système avec un minimum d'en-cours.

Une heuristique est proposée.

**Mots clés :** Modélisation, Réseaux de Petri, Graphes d'événements, Systèmes de production, Productivité.

**Abstract.** — This paper studies the functioning of a job-shop-like production system for a repetitive processing of the jobs. A modeling based on deterministic timed Event-Graphs is developed to evaluate the dynamic performance of the system and in particular determine the conditions of functioning at maximal rate (i.e. maximum productivity). It is shown that, given any fixed sequencing of the jobs on the machines, it is always possible to fully utilize the bottleneck machines in steady-state with a finite number of jobs in-process. A heuristic algorithm is further developed to obtain a near-optimal sequencing and scheduling of the jobs, such that the bottleneck machines be fully utilized with the minimum work-in-progress (WIP) inventory. This condition guarantees an optimal functioning in case of Flexible Manufacturing Systems (FMS), since the resources used to transport the parts (such as pallets, carts, etc.) are minimized and the production rate is maximized.

**Keywords :** Job-shop, Event-Graphs, Productivity, In-Process Inventory, Bottleneck Machines.

---

(\*) Reçu septembre 1987.

(1) I.N.R.I.A.-Lorraine, et laboratoire M.A.S.I., Université Paris-VI, France.

(2) I.N.R.I.A.-Lorraine, Technopôle de Nancy-Brabois, Campus Scientifique, boulevard des Aiguillettes, B.P. n° 239, 54506 Vandœuvre-les-Nancy Cedex.

## 1. INTRODUCTION

Nous appelons système de production un ensemble de machines reliées entre elles par des ressources de transport (chariots, balancelles, . . .). Dans la suite, nous supposons que les conflits entre ressources de transport sont inexistantes ou négligeables et que ces ressources sont en nombre limité. En outre, nous considérons que le temps nécessaire pour relier deux machines quelconques est nul, ce qui ne restreint en rien la généralité des résultats, mais allège la présentation.

Une pièce est obtenue par transformations successives sur un sous-ensemble de machines du système. L'ordre dans lequel les machines sont visitées par une pièce est appelé routage. Nous supposons que le routage est unique pour un type de pièce donné. Cette hypothèse n'est pas non plus restrictive : dans la pratique, les routages sont choisis de façon à équilibrer les charges des machines, et ce choix est fait avant de résoudre le problème d'ordonnancement que nous posons ici. Si l'on ajoute au routage les temps de passage des pièces sur les machines, on dit que l'on connaît les gammes de fabrication des pièces.

Lorsque le routage est le même pour toutes les pièces à fabriquer, on dit que l'on a affaire à une production linéaire, ou « flow-shop ». Dans le cas contraire, on parle de production non linéaire, ou « job-shop ». C'est à ce dernier type de production que nous nous intéressons dans la suite. Remarquons au passage qu'un « flow-shop » est un cas particulier du « job-shop » : tous les résultats obtenus par l'étude des « job-shops » seront donc applicables aux « flow-shops ». Nous dégagerons en outre quelques particularités des « flow-shops ».

Dans le « job-shop » que nous étudions, la demande est stationnaire et donnée par le pourcentage de chaque type de pièce à fabriquer. Le contrôle du système se traduit par les ordres de passage des pièces sur les machines. Ces ordres de passage, que nous appelons aussi séquences d'entrée dans les machines, étant donnés, nous montrons qu'il est toujours possible de saturer la machine la plus chargée, à condition de s'autoriser un état initial du système avec des pièces en cours de fabrication. Bien entendu, dans la pratique, ces en-cours initiaux sont virtuels et le régime permanent n'est atteint que lorsque ces pièces virtuelles sont toutes évacuées du système.

La saturation de la machine la plus chargée assure la productivité maximale. Pour l'atteindre, il faut autoriser un nombre suffisant d'en-cours dans le système, donc un nombre suffisant de ressources de transport. On suppose que chaque ressource de transport prend en charge une seule pièce. Cette

hypothèse est cohérente avec ce que nous savons de nombreux Ateliers Flexibles en service qui utilisent comme ressources de transport des chariots filo-guidés, des balancelles, des palettes sur tapis « en écailles », etc. Nous montrons, dans cette communication, qu'il est possible de minimiser le nombre de ressources de transport lorsque les séquences d'entrée dans les machines sont données : il faut pour cela résoudre un problème de programmation linéaire particulier de type 0-1 pour lequel nous fournissons une heuristique efficace.

Le nombre minimum de ressources de transport dépend, bien entendu, des séquences d'entrée des pièces dans les machines. Il est donc naturel de rechercher l'ensemble des séquences d'entrée qui permet d'atteindre la productivité maximale avec un nombre de ressources de transport aussi faible que possible. Nous montrons que l'expression de ce problème d'ordonnancement est simple et nous proposons une heuristique qui nous conduit toujours à une solution proche de l'optimum.

L'importance que nous accordons à la minimisation des ressources de transport s'explique par le fait que celles-ci sont généralement coûteuses. Il convient, par conséquent, d'en limiter le nombre.

Le problème que nous étudions ici a déjà été examiné par G. Cohen et autres [4 et 5] à l'aide de l'algèbre (IR, max, +). L'hypothèse faite par ces auteurs est qu'aucune pièce n'est en cours de transformation à l'instant initial. Dans ce cas, on atteint toujours un régime permanent, mais la machine la plus chargée n'est pas nécessairement saturée lorsque le régime permanent est atteint.

Les auteurs de la présente communication ont mené [1] une étude directe du problème sous les mêmes hypothèses et ont dégagé un certain nombre de propriétés en distinguant les productions de type linéaire (flow-shop) des productions non linéaires (job-shop).

Les résultats que nous présentons ici s'appuient sur une modélisation à l'aide des graphes d'événements temporisés [3].

La communication est divisée en cinq sections principales. La seconde section donne la définition des graphes d'événements temporisés et rappelle les propriétés que nous utiliserons par la suite. Dans la troisième section, nous définissons le système de production étudié et montrons comment le modéliser à l'aide des graphes d'événements temporisés. La section quatre traite du problème de minimisation des en-cours lorsque les séquences d'entrée dans les machines sont connues. Enfin, la section cinq est consacrée à l'optimisation des séquences d'entrée dans les machines, c'est-à-dire à la

recherche des séquences d'entrée qui permettent d'atteindre la productivité maximale avec un minimum d'en-cours.

Ces différents paragraphes sont illustrés d'exemples.

## 2. GRAPHES D'ÉVÉNEMENTS TEMPORISÉS

### 2.1. Définition d'un graphe d'événements temporisé

Nous supposons connue la définition des Réseaux de Petri [2 et 9].

Rappelons simplement qu'un Réseau de Petri est un graphe orienté dont les nœuds sont de deux types : les *places* et les *transitions*. Un arc relie une transition à une place ou réciproquement. Le *marquage*  $M$  d'un Réseau de Petri est une application de l'ensemble  $P$  des places dans  $\{0, 1, 2, \dots\}$ , ensemble des entiers positifs. Elle associe à chaque place un nombre (éventuellement nul) de *jetons*.

Nous appelons *places d'entrée* d'une transition  $t$  l'ensemble  $P_e(t)$  des places qui sont origines des arcs dont l'extrémité est  $t$ . Nous appelons *places de sortie* d'une transition  $t$  l'ensemble  $P_s(t)$  des places extrémités des arcs dont l'origine est  $t$ . Une transition  $t$  peut être *franchie* (on dit encore *mise à feu*) si toutes les places de  $P_e(t)$  contiennent au moins un jeton. Le *franchissement* (ou la *mise à feu*) d'une transition  $t$  consiste à retirer un jeton de chacune des places de  $P_e(t)$  et à ajouter un jeton dans chacune des places de  $P_s(t)$ .

Soit  $N$  un Réseau de Petri et  $M$  un marquage.

Soit  $\sigma$  une séquence de franchissements, c'est-à-dire une suite de transitions à franchir successivement, et  $M_\sigma$  le marquage atteint après franchissement de cette séquence  $\sigma$  :

$$M \xrightarrow{\sigma} M_\sigma.$$

Nous notons  $\mathbf{M}$  l'ensemble des marquages accessibles à partir de  $M$  par franchissement d'une séquence.

On appelle *réseau marqué* un couple  $(N, M)$  où  $N$  est un réseau de Petri et  $M$  un marquage initial. Un réseau marqué est *vivant* si et seulement si, quels que soient la transition  $t$  et  $M^* \in \mathbf{M}$ , il existe une séquence de franchissements qui, partant de  $M^*$ , contient  $t$ .

$(N, M)$  est dit *borné* si et seulement si il existe un entier positif  $k$  tel que, quel que soit  $M^* \in \mathbf{M}$ , le nombre de jetons dans chaque place est inférieur ou égal à  $k$ .

Un graphe d'événements est un Réseau de Petri tel que toute place possède une et une seule transition d'entrée et une et une seule transition de sortie.

Un graphe d'événements temporisé est un couple  $(N, \mu)$  où  $N$  est un graphe d'événement et  $\mu$  une fonction qui, à chaque transition  $t$ , associe un temps de franchissement  $\mu(t)$ .

Nous rappelons les propriétés des graphes d'événements que nous utiliserons dans la suite.

## 2.2. Propriétés des graphes d'événements fortement connexes

Les deux propriétés suivantes, dues à Commoner et autres [6], seront utilisées dans la suite :

**PROPRIÉTÉ 1 :** Le nombre de jetons est invariant dans un circuit élémentaire d'un graphe d'événements. En d'autres termes, si  $M$  est un marquage initial alors, pour tout marquage  $M^* \in M$ , le nombre de jetons dans un circuit élémentaire est le même que pour le marquage  $M$ .

**PROPRIÉTÉ 2 :** Un graphe d'événements est vivant si et seulement si le nombre de jetons contenus dans chaque circuit élémentaire est strictement positif.

Nous ajoutons la propriété suivante, triviale pour tout marquage  $M^* \in M$ .

**PROPRIÉTÉ 3 :** Un graphe d'événements fortement connexe est borné si et seulement si le marquage initial est borné.

Si un graphe d'événements est temporisé, une transition met un certain temps à être franchie, et une transition n'est pas franchissable si un franchissement de cette même transition est déjà en cours. Nous illustrons cette propriété par le graphe simple de la figure 1, dans lequel la durée de  $t$  est égale à 2. L'instant initial est 0 et la place  $P_1$  contient deux jetons à l'instant 0.

La transition est franchie une première fois à l'instant 2 et une seconde fois à l'instant 4.

On peut modéliser la contrainte de non réentrance des franchissements d'une même transition en associant à chaque transition une place qui empêche le franchissement de la transition simultanément par les deux jetons (voir fig. 2).

Un tel graphe d'événements est vivant. Introduisant la notion d'ordonnement infini du réseau, on montre qu'il existe un ordonnancement au plus tôt, et  $n_0$ ,  $\lambda$  et  $K$  entiers, vérifiant :

$$S_t(n+K) = S_t(n) + K\lambda \text{ pour tout } n \geq n_0 \quad (1)$$

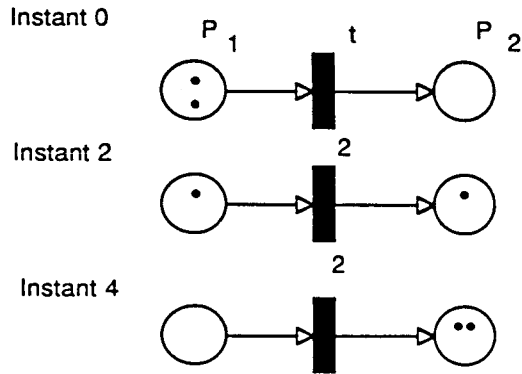


Figure 1

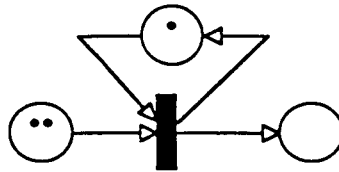


Figure 2

pour toute transition  $t$ , où  $S_t(n)$  est l'instant du début du  $n$ -ième franchissement de la transition  $t$  (voir [3]).

$K$  est appelé facteur de périodicité et  $1/\lambda$  est la fréquence moyenne de franchissement de la transition  $t$ .

Bien entendu,  $K$  est le plus petit entier qui réalise (1).

On montre que [3] :

$$\lambda = \max_{\gamma \in \Gamma} (\mu(\gamma) / M(\gamma)) \quad (2)$$

où :

$\Gamma$  est l'ensemble des circuits élémentaires du graphe.

$\mu(\gamma)$  est la somme des temps de franchissement des transitions de  $\gamma$ .

$M(\gamma)$  est la somme des nombres de jetons contenus dans les places appartenant à  $\gamma$ .

$C(\gamma) = \mu(\gamma)/M(\gamma)$  est appelé temps de cycle du circuit élémentaire  $\gamma$ .  $\lambda$  est la période attachée au système.

Un circuit  $\gamma^*$  qui réalise le maximum de (2), c'est-à-dire tel que :

$$\lambda = \frac{\mu(\gamma^*)}{M(\gamma^*)}$$

est appelé *circuit critique*.

*Remarque* : Si tous les circuits critiques d'un graphe d'événements temporisé fortement connexe ne contiennent qu'un seul jeton, alors  $K$  de la relation (1) est égal à 1 : le régime stationnaire est alors simplement périodique (voir [3]).

### 3. DÉFINITION DU « JOB-SHOP » ET MODÉLISATION

#### 3.1. Définitions et notations

Nous notons  $\mathcal{M} = \{M_1, \dots, M_m\}$  l'ensemble des machines de l'atelier et  $\mathcal{P} = \{P_1, \dots, P_n\}$  l'ensemble des types de pièce à fabriquer. A chaque type de pièce  $P_i$  ( $i=1, \dots, n$ ) correspond un routage  $R_i = \{M_1^i, \dots, M_{k_i}^i\}$  avec  $M_s^i \in \mathcal{M}$  pour  $s=1, \dots, k_i$ . La gamme associée au type de pièce  $P_i$  s'obtient en donnant, pour chaque machine de la séquence  $R_i$ , le temps de passage du produit  $P_i$ . Nous notons  $\mu(M_s^i)$ ;  $s=1, \dots, k_i$ ;  $i=1, \dots, n$ , le temps de passage d'une pièce de type  $P_i$  sur la machine  $M_s^i$ .

Pour  $i=1, 2, \dots, n$ , nous désignons par  $a_i$  la proportion de pièces de type  $P_i$  à fabriquer, prise par rapport au nombre total de pièces. Les  $a_i$  sont des données du problème et  $\sum_{i=1}^n a_i = 1$ .

Le contrôle du système est la donnée, pour chaque machine  $M_j$  ( $j=1, \dots, m$ ), de la séquence d'entrée des types de pièce dans la machine.

Nous notons  $U_j = \{P_1^j, \dots, P_{l_j}^j\}$  la séquence d'entrée dans la machine  $M_j$ . Dans une telle séquence, on peut retrouver plusieurs fois le même type de pièce. Par contre, si  $M_j \notin R_i$ , alors  $P_i \notin U_j$ : en d'autres termes, la séquence d'entrée dans  $M_j$  ne comporte le type de pièce  $P_i$  que si  $M_j$  fait partie des machines que doit visiter toute pièce de type  $P_i$ .



Pour satisfaire les taux relatifs de production, les séquences d'entrée doivent satisfaire la propriété suivante établie dans [1] :

Si  $P_i \in U_j$  alors, si  $n_i^j$  est le nombre d'occurrences de  $P_i$  dans  $U_j$  et  $l_j$  la longueur de la séquence  $U_j$  :

$$n_i^j/l_j = a_i / \sum_{s/M_j \in R_s} a_s \quad (3)$$

pour  $i = 1, \dots, n$  et  $M_j \in R_i$ .

On montre, toujours dans [1], que si la propriété (3) n'est pas vérifiée, alors le système se bloque après un temps fini. Un ensemble  $\{U_j\}_{j=1, \dots, m}$  dont tous les éléments vérifient (3) sera dit *admissible*.

*Remarque.* — Considérons une séquence  $U^* = \{P_1^*, \dots, P_n^*\}$  qui contient au moins une fois chacun des types de pièce  $P_1, \dots, P_n$  et telle que :

$n_i^*/l^* = a_i$  pour  $i = 1, \dots, n$ , où  $n_i^*$  est le nombre d'occurrences de  $P_i$  dans  $U^*$  et  $l^*$  la longueur de la séquence  $U^*$ .

Si, pour  $j = 1, \dots, m$ , nous construisons la séquence  $U_j$  en éliminant de  $U^*$  tout  $P_i$  tel que  $M_j \notin R_i$ , alors l'ensemble des séquences ainsi obtenues est admissible. L'ensemble des séquences reste admissible pour toute permutation appliquée aux  $U_j$ .

Nous donnons un exemple simple pour illustrer ce qui précède.

On considère trois types de pièce notés  $P_1$ ,  $P_2$  et  $P_3$ . Le système de production est composé de trois machines notées  $M_1$ ,  $M_2$  et  $M_3$ .

Les gammes associées aux différents types de pièce sont les suivantes (les quantités entre parenthèses sont les temps de passage sur les machines) :

$$P_1 : M_1(1), M_2(3), M_3(3)$$

$$P_2 : M_3(1), M_2(2)$$

$$P_3 : M_1(2), M_3(1).$$

Les ratios des demandes sont donnés :

$$a_1 = 1/4; \quad a_2 = 1/4; \quad a_3 = 1/2.$$

Nous retenons les séquences admissibles :

$$U_1 = \{P_1, P_3, P_3\}; \quad U_2 = \{P_1, P_2\}; \quad U_3 = \{P_1, P_2, P_3, P_3\}.$$

En utilisant les notations précédentes :

$$\begin{aligned}\mathcal{M} &= \{M_1, M_2, M_3\}; & \mathcal{P} &= \{P_1, P_2, P_3\} \\ R_1 &= \{M_1, M_2, M_3\}; & R_2 &= \{M_3, M_2\}; & R_3 &= \{M_1, M_3\} \\ \mu(M_1^1) &= 1; & \mu(M_2^1) &= 3; & \mu(M_3^1) &= 3 \\ \mu(M_1^2) &= 1; & \mu(M_2^2) &= 2 \\ \mu(M_1^3) &= 2; & \mu(M_2^3) &= 1.\end{aligned}$$

On vérifie la propriété (3) :

$$\begin{aligned}n_1^1/l_1 &= 1/3 = a_1/(a_1 + a_3); & n_3^1/l_1 &= 2/3 = a_3/(a_1 + a_3) \\ n_1^2/l_2 &= 1/2 = a_1/(a_1 + a_2); & n_2^2/l_2 &= 1/2 = a_2/(a_1 + a_2) \\ n_1^3/l_3 &= 1/4 = a_1/(a_1 + a_2 + a_3); & n_2^3/l_3 &= 1/4 = a_2/(a_1 + a_2 + a_3) \\ n_3^3/l_3 &= 1/2 = a_3/(a_1 + a_2 + a_3).\end{aligned}$$

Nous conservons cet exemple tout au long du présent exposé pour illustrer nos propos.

### 3.2. Modélisation à l'aide d'un graphe d'événements temporisé

Le modèle est construit en trois étapes : la modélisation des gammes de production, la modélisation des ressources de transport, et enfin la modélisation du contrôle.

Nous illustrons ces trois étapes à l'aide de l'exemple que nous venons d'introduire.

#### 3.2.1. Modélisation des gammes de production

Considérons la séquence  $U^*$  telle qu'elle a été définie dans le paragraphe précédent. Nous associons à chaque élément de  $U^*$  un graphe d'événements dans lequel chaque transition représente une opération (*i. e.* une machine de la gamme) et chaque place un stock d'en-cours. Les transitions sont placées dans l'ordre des machines dans la gamme de l'élément de  $U^*$  considéré et temporisées par les temps de passage sur les machines correspondantes. Les places sont appelées *places tampons*. On suppose, bien entendu, qu'un type de pièce passe toujours au moins sur une machine. Dans le cas contraire, on écarte ce type de pièce du système étudié.

Soit par exemple  $R_i = \{M_1^i, \dots, M_{k_i}^i\}$  le routage correspondant à tout produit de type  $P_i$ . Alors, chaque fois que  $P_i$  se trouvera dans  $U^*$ , il sera modélisé comme l'indique la figure 3.

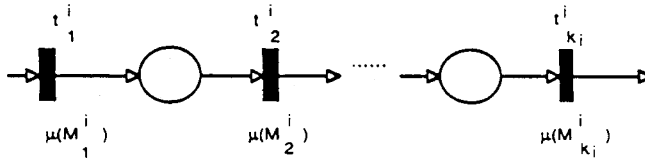


Figure 3

Dans ce modèle,  $t_s^i$  représente  $M_s^i$  et est temporisé par  $\mu(M_s^i)$ ;  $s = 1, \dots, k_i$ .

Revenons à l'exemple du paragraphe précédent en prenant  $U^* = U_3$ .

La première étape de la modélisation nous conduira au graphe d'événements temporisé de la figure 4. Derrière chaque nom de transition, nous avons fait figurer entre parenthèses le nom de la machine correspondante. Sous chaque transition figure sa temporisation.

Dans un tel modèle, chaque jeton représentera une pièce, et nous supposons que chaque ressource de transport (chariot, balancelle, ...) porte au maximum une pièce.

### 3.2.2. Modélisation des ressources de transport

Rappelons que, compte tenu des hypothèses précédentes, le nombre de ressources de transport majore le nombre de pièces en cours de fabrication. Pour pouvoir gérer ce nombre, nous complétons chaque sous-graphe correspondant à un élément  $P_i$  de  $U^*$  en le transformant en circuit par adjonction d'une place supplémentaire que nous appelons *place de ressources*. Le sous-graphe de la figure 3 est ainsi transformé en un circuit élémentaire (fig. 5) appelé *circuit de fabrication*.

A ce stade de la modélisation, l'exemple que nous avons choisi (voir fig. 4) se représente comme l'indique la figure 6.

En considérant que chaque jeton représente une ressource de transport, nous voyons que les circuits élémentaires que nous venons de construire permettent de réguler la fabrication en n'autorisant l'entrée d'une nouvelle pièce que lorsqu'une pièce est terminée.

A son stade actuel, la modélisation entraîne cependant deux remarques :

- l'exclusion mutuelle de l'utilisation d'une machine n'est pas respectée : deux transitions appartenant à deux circuits élémentaires différents mais qui représentent la même machine peuvent être franchies en même temps. Dans la pratique, cela signifierait qu'une machine est capable de transformer simultanément au moins deux pièces, ce qui est impossible;

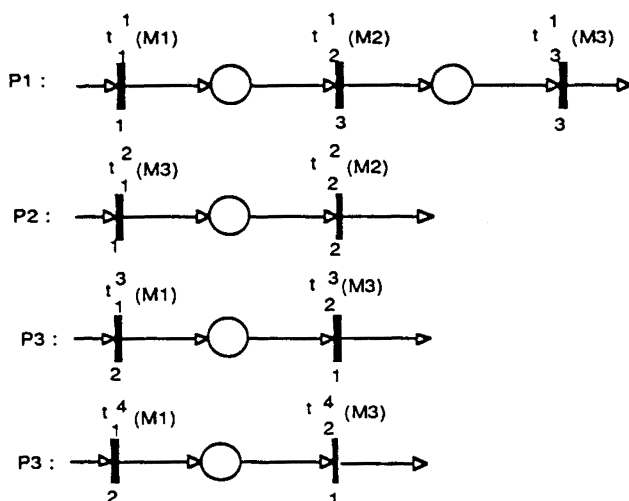


Figure 4

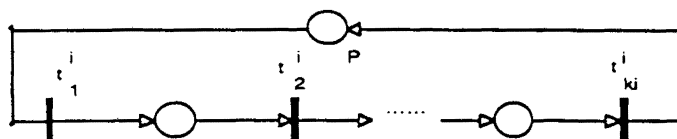


Figure 5

— les ratios de fabrication peuvent ne pas être respectés. Considérons, pour illustrer cette remarque, l'exemple de la figure 6, et supposons que chaque circuit de fabrication contienne un jeton. Alors, le système produirait une pièce de type  $P_1$  toutes les 7 unités de temps et une pièce de type  $P_2$  toutes les 3 unités de temps, ce qui ne correspond pas aux ratios  $a_1 = 1/4$ ,  $a_2 = 1/4$  de la demande.

Il convient donc de compléter la modélisation précédente.

### 3.2.3. Modélisation du contrôle

Nous avons souligné que le système est commandé (*i. e.* contrôlé) en particulier par le biais des séquences d'entrée dans les machines. Pour modéliser ces séquences, nous intégrons les transitions représentant la même machine, mais situées dans des circuits de fabrication différents, dans un même circuit élémentaire, en prenant soin que les circuits de fabrication soient visités dans l'ordre de la séquence d'entrée dans la machine. Les circuits ainsi obtenus

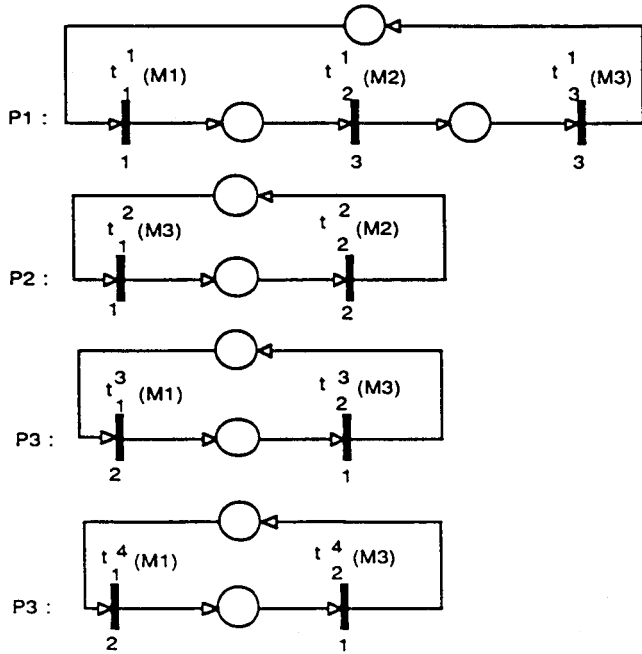
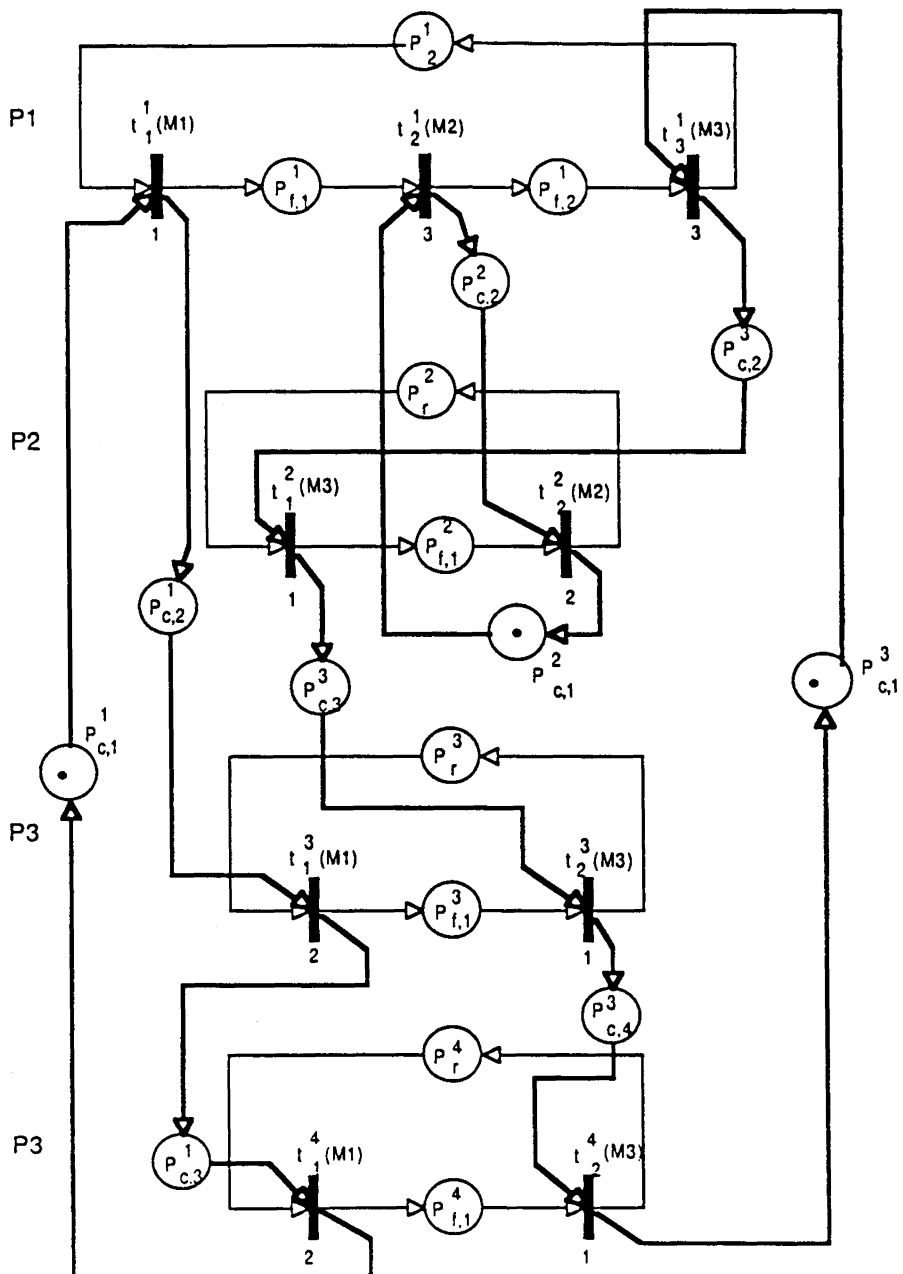


Figure 6

seront qualifiés de *circuits élémentaires de contrôle (ou de commande)*. La figure 7 intègre ce dernier stade de la modélisation pour l'exemple qui nous sert à illustrer notre propos (voir fig. 6).

Dans la figure 7, les arcs des circuits de fabrication sont représentés en traits fins et les arcs des circuits de commande en traits gras. Les circuits de commande respectent les ordres  $U_1, U_2, U_3$  (voir § 3.1.) à une permutation circulaire près. Chaque circuit élémentaire de commande ne contient qu'un seul jeton, ce qui assure qu'à chaque instant au plus une transition est en cours de franchissement dans chaque circuit de commande. On est ainsi assuré que, à chaque instant, au plus une pièce est en cours de transformation sur chaque machine.

Sur la figure 7, nous avons placé un jeton dans une des places de chacun des circuits de commande. Ce jeton est placé de sorte que la transition qui suit la place choisie corresponde au type du premier élément de  $U_i$  ( $i = 1, 2, 3$ ). En conséquence, nous admettons qu'au démarrage de la fabrication certains produits sont déjà en cours de fabrication. Dans notre exemple, la pièce de type  $P_1$  qui est prise en compte par  $M_2$  à l'instant initial est supposée être



passée par  $M_1$  avant l'instant initial. En pratique, ces pièces sont des pièces virtuelles qui n'interviennent que durant le régime transitoire. Le régime permanent est atteint lorsque toutes les pièces virtuelles sont évacuées du système.

Nous verrons, dans le paragraphe suivant, qu'un circuit de fabrication peut contenir plus d'un jeton et que le nombre de jetons contenus dans chaque circuit élémentaire ainsi que leur répartition déterminent la productivité du système et la possibilité pour celui-ci de se bloquer ou non après un temps fini.

Notons que le graphe finalement obtenu est fortement connexe. On y distingue trois sortes de circuits élémentaires :

- les circuits de fabrication qui peuvent contenir, chacun, plusieurs jetons. Ces jetons représentent les ressources de transport utilisées, mais aussi le nombre maximal d'en-cours qui peuvent se trouver dans le système à un instant donné (rappelons qu'une ressource de transport prend en charge au plus une pièce);

- les circuits de commande qui contiennent, chacun, un jeton et un seul. Ils garantissent :

- α. que chaque machine ne produira pas plus d'une pièce à la fois,

- β. que les ordres de passage sur les machines seront respectés,

- les circuits mixtes, composés pour partie d'arcs appartenant à des circuits de fabrication et pour partie d'arcs appartenant à des circuits de commande.

Donnons quelques exemples de circuits élémentaires mixtes pour le graphe de la figure 7 :

$$\gamma_1 = (p_{c,1}^1, t_1^1, p_{f,1}^1, t_2^1, p_{c,2}^2, t_2^2, p_1^2, t_1^2, p_{c,3}^3, t_2^3, p_{c,4}^3, t_2^4, p_r^4, t_1^4)$$

$$\gamma_2 = (t_2^1, p_{f,2}^1, t_3^1, p_{c,2}^3, t_1^2, p_{f,1}^2, t_2^2, p_{c,1}^2)$$

$$\gamma_3 = (t_1^1, p_{f,1}^1, t_2^1, p_{f,2}^1, t_3^1, p_{c,2}^3, t_1^2, p_{c,3}^3, t_2^3, p_{c,4}^3, t_2^4, p_r^4, t_1^4, p_{c,1}^1).$$

Le graphe de la figure 7 comporte 18 circuits mixtes. Nous disposons, pour détecter ces circuits, d'un logiciel développé au M.I.T. <sup>(3)</sup> à partir des algorithmes de recherche des invariants d'un Réseau de Petri [7, 8].

---

<sup>(3)</sup> Ce logiciel a été développé par Monsieur H. P. Hillion durant son séjour au « Laboratory for Information and Decision Systems » du M.I.T. Le travail a été financé par le « US Office of Naval Research » sous le contrat numéros N00014-85-K-0782 et N00014-84-K-0519.

Parmi les circuits mixtes, nous distinguons :

- les circuits *externes*, qui contiennent au moins une place de ressources (indiquée  $r$ ).  $\gamma_1$  est un circuit mixte externe car il contient  $p_r^2$  et  $p_r^4$ .  $\gamma_3$  appartient également à ce type de circuit car il contient  $p_r^4$ .
- les circuits *internes*, qui ne contiennent pas de place de ressources.  $\gamma_2$  est de ce type.

Ces définitions nous serviront dans la suite. Nous étudions maintenant les performances d'un système lorsque les séquences d'entrée des pièces dans les machines sont fixées.

#### 4. ÉTUDE DES PERFORMANCES LORSQUE LES SÉQUENCES D'ENTRÉE DANS LES MACHINES SONT FIXÉES

Le modèle obtenu étant un graphe d'événements temporisé fortement connexe, nous pouvons utiliser les résultats du paragraphe 2. Nous tirons toutes les conséquences de ces résultats en 4.1. En 4.2, nous proposons une heuristique pour le calcul du nombre minimal de jetons et de leur répartition dans le graphe afin d'atteindre la productivité maximale. Nous examinons brièvement le cas du « flow-shop » dans le paragraphe 4.3. Enfin, en 4.4, nous donnons les résultats obtenus en appliquant l'heuristique à l'exemple qui illustre notre exposé.

##### 4.1. Comportement et performances du « job-shop »

Supposons choisie une répartition initiale quelconque des jetons. Les propriétés suivantes caractérisent une telle répartition.

**PROPRIÉTÉ 4 :** Il y aura blocage du système au bout d'un temps fini si et seulement si l'un des circuits élémentaires (de fabrication, de commande ou mixte) ne contient aucun jeton à l'instant initial.

Cette propriété est la transcription de la propriété 2.

**PROPRIÉTÉ 5 :** La condition nécessaire est suffisante pour que la productivité maximale soit atteinte est qu'au moins un des circuits de commande soit critique. La productivité est maximale pour un ordonnancement (*i.e.* un ensemble de séquences d'entrée dans les machines) et une répartition initiale des jetons donnés s'il n'est pas possible de l'améliorer en modifiant l'ordonnancement et/ou la répartition initiale des jetons.

*Démonstration :* (a) La condition est suffisante. En effet, si un circuit de commande est critique, cela signifie que la machine correspondante est utilisée



à plein (cette machine est évidemment la plus chargée), et il est impossible de faire mieux en respectant les proportions de la demande.

(b) La condition est nécessaire. Supposons en effet qu'aucun circuit de commande ne soit critique et que la production soit maximale. Soit  $\Gamma_c$  l'ensemble des circuits élémentaires de commande du système. Alors, d'après (2) :

$$\lambda = \text{Max} \left\{ \text{Max}_{\gamma \in \Gamma_c} [\mu(\gamma)/M(\gamma)], \text{Max}_{\gamma \in \Gamma - \Gamma_c} [\mu(\gamma)/M(\gamma)] \right\} \quad (4)$$

et le maximum se réalise pour  $\gamma^* \in \Gamma - \Gamma_c$ .

Or tout circuit  $\gamma \in \Gamma - \Gamma_c$  contient au moins une place qui n'appartient pas à un circuit élémentaire de commande. Nous pouvons donc ajouter des jetons dans ces places jusqu'à ce que le maximum du second membre de (4) soit obtenu pour  $\gamma^{**} \in \Gamma_c$ .

Nous aurons alors fait décroître  $\lambda$ , qui est la période, et donc fait croître  $1/\lambda$ , qui est la fréquence moyenne de tirage d'une quelconque transition. D'où une augmentation de la productivité, ce qui est contraire à l'hypothèse.

**PROPRIÉTÉ 6 :** Au moins un circuit de commande est critique si toutes les places tampons et toutes les places de ressources contiennent un jeton à l'instant initial.

*Démonstration :* Nous notons  $\mu(t)$  le temps de franchissement de la transition  $t$ .

Nous désignons par :

$\Gamma_f$  l'ensemble des circuits élémentaires de fabrication,

$\Gamma_m$  l'ensemble des circuits élémentaires mixtes.

Alors :  $\Gamma_f \cup \Gamma_m \cup \Gamma_c = \Gamma$

(a) Chaque circuit de commande ne contient qu'un seul jeton, donc pour tout  $\gamma^* \in \Gamma_c$  :

$$\mu(\gamma^*)/M(\gamma^*) = \mu(\gamma^*) = \sum_{t \in \gamma} \mu(t) \leq \text{Max}_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t). \quad (5)$$

(b) La présence, à l'instant initial, d'un jeton dans chaque place des circuits de fabrication permet d'écrire, pour tout  $\gamma^* \in \Gamma_f$  :

$$\mu(\gamma^*)/M(\gamma^*) \leq \text{Max}_{t \in \gamma} \mu(t) \leq \text{Max}_{\gamma \in \Gamma_f} \text{Max}_{t \in \gamma} \mu(t). \quad (6)$$

(c) Considérons maintenant l'ensemble  $\Gamma_m$  des circuits élémentaires mixtes. Soit  $\gamma^* \in \Gamma_m$ .  $\gamma^*$  est composé :

- de  $z$  ( $z \geq 2$ ) chemins appartenant chacun à un circuit élémentaire de commande, notés  $\alpha(1), \alpha(2), \dots, \alpha(z)$ ;
- de  $z$  chemins appartenant chacun à un circuit élémentaire de fabrication, notés  $\beta(1), \beta(2), \dots, \beta(z)$ .

Les chemins alternent pour former le circuit élémentaire mixte  $\gamma$ . En les numérotant de manière adéquate, on a :

$$\gamma^* = \{\alpha(1) \oplus \beta(1) \oplus \alpha(2) \oplus \beta(2) \oplus \dots \oplus \alpha(z) \oplus \beta(z)\},$$

où  $\oplus$  représente la concaténation. Les transitions situées aux extrémités des chemins  $\alpha(i)$  et  $\beta(i)$ , ou  $\beta(i)$  et  $\alpha(i+1)$ , sont comptabilisées comme appartenant aux chemins  $\alpha(i)$  ou  $\alpha(i+1)$  respectivement. Avec cette convention, il y a dans  $\beta(i)$  ( $i=1, \dots, z$ ) une place de plus que de transitions. Remarquons qu'un chemin de type  $\beta(i)$  peut ne contenir qu'une place, alors que le chemin de type  $\alpha(i)$  est formé au moins d'une place entourée de deux transitions.

Si  $w(i)$  est le nombre de places de  $\beta(i)$  ( $i=1, \dots, z$ ), et si l'on se souvient qu'il y a un jeton dans chaque place des circuits de fabrication, alors :

$$\mu(\gamma^*)/M(\gamma^*) \leq \left\{ \sum_{i=1}^z \mu[\alpha(i)] + \mu[\beta(i)] \right\} / \left[ \sum_{i=1}^z w(i) \right]. \quad (7)$$

[Il y a égalité lorsqu'aucun chemin  $\alpha(i)$  ne contient de jeton].

Soit  $B$  l'ensemble des chemins  $\beta(i)$  ( $i=1, \dots, z$ ) qui ne se réduisent pas à une place. Bien entendu,  $\text{card}(B) \leq z$ .

Alors (7) peut se réécrire :

$$\mu(\gamma^*)/M(\gamma^*) \leq \left\{ \sum_{i=1}^z \mu[\alpha(i)] + \sum_{i \in B} \mu[\beta(i)] \right\} / \left[ \sum_{i=1}^z w(i) \right]$$

que l'on peut encore écrire :

$$\mu(\gamma^*)/M(\gamma^*) \leq \left( \sum_{i=1}^z \mu[\alpha(i)] + \sum_{i \in B} \mu[\beta(i)] \right) / \left( z + \sum_{i \in B} [w(i) - 1] \right)$$

D'où :

$$\mu(\gamma^*)/M(\gamma^*) \leq \text{Max} \left\{ \sum_{i=1}^z \mu[\alpha(i)] / z, \sum_{i \in B} \mu[\beta(i)] / \sum_{i \in B} [w(i) - 1] \right\}. \quad (8)$$

Or :

$$\sum_{i=1}^z \mu[\alpha(i)]/z \leq \max_{i=1, \dots, z} \mu[\alpha(i)] \leq \max_{i=1, \dots, z} \mu[\gamma(i)]$$

où  $\gamma(i)$  est le circuit élémentaire de commande contenant  $\alpha(i)$  et, *a fortiori* :

$$\sum_{i=1}^z \mu[\alpha(i)]/z \leq \max_{\gamma \in \Gamma_c} \mu(\gamma) = \max_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t) \quad (9)$$

et :

$$\sum_{i \in B} \mu[\beta(i)] / \sum_{i \in B} [w(i) - 1] \leq \max_{i \in B} \max_{t \in \beta(i)} \mu(t)$$

soit :

$$\sum_{i \in B} \mu[\beta(i)] / \sum_{i \in B} [w(i) - 1] \leq \max_{\gamma \in \Gamma_f} \max_{t \in \gamma} \mu(t). \quad (10)$$

Finalement (8) se réécrit, en tenant compte de (9) et de (10) :

$$\mu(\gamma^*)/M(\gamma^*) \leq \max_{\gamma \in \Gamma_f} \{ \max_{t \in \gamma} \mu(t), \max_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t) \} \quad (11)$$

pour tout  $\gamma^* \in \Gamma_m$ .

(d) Partant de (5), (6) et (11), nous voyons que :

$$\max_{\gamma \in \Gamma} [\mu(\gamma^*)/M(\gamma^*)] \leq \max_{\gamma \in \Gamma_f} \{ \max_{t \in \gamma} \mu(t), \max_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t) \} \quad (12)$$

Mais toute transition appartenant à un circuit de fabrication appartient aussi, par construction, à un circuit de commande, si bien que la relation (12) se réduit à :

$$\max_{\gamma^* \in \Gamma} [\mu(\gamma^*)/M(\gamma^*)] \leq \max_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t)$$

et, puisque  $\Gamma_c \subset \Gamma$  :

$$\max_{\gamma^* \in \Gamma} [\mu(\gamma^*)/M(\gamma^*)] = \max_{\gamma \in \Gamma_c} \sum_{t \in \gamma} \mu(t)$$

ce qui achève la démonstration.

Nous venons de mettre en évidence le fait qu'il existe un état initial qui assure la productivité maximale, laquelle est caractérisée par l'occupation permanente de la machine la plus chargée. A cet état initial correspond un nombre total de jetons dans les circuits de fabrication. Ce nombre est le nombre de ressources nécessaires pour atteindre la productivité maximale. La question est maintenant de savoir s'il est possible d'atteindre cette productivité maximale avec un nombre plus réduit de ressources : c'est l'objet du paragraphe suivant.

#### 4.2. Minimisation des ressources pour un ordonnancement fixé

Nous allons montrer que le nombre minimal de ressources nécessaires est solution d'un programme de programmation linéaire en variables 0, 1.

Notons :

$$C_0 = \max_{\gamma \in \Gamma_c} [\mu(\gamma)/M(\gamma)] = \max_{\gamma \in \Gamma_c} \mu(\gamma). \quad (13)$$

Le problème à résoudre s'écrit :

étant donné un graphe d'événements temporisé modélisant un job-shop répétitif tel que le jeton de chaque circuit de commande soit initialisé, trouver un marquage initial tel que :

- la somme des jetons soit minimale,
- au moins un circuit de commande [celui (ou ceux) qui réalise(nt) le maximum dans (13)] soit critique, ce qui s'écrit :

$$M(\gamma) \geq \mu(\gamma)/C_0, \quad \forall \gamma \in \Gamma - \Gamma_c.$$

Si  $[y]$  désigne l'entier immédiatement supérieur ou égal à  $y$  alors, comme  $M(\gamma)$  est entier, l'inégalité précédente se réécrit :

$$M(\gamma) \geq [\mu(\gamma)/C_0] = a(\gamma). \quad (14)$$

Dans la suite,  $s$  représente le nombre total de circuits du graphe d'événements temporisé et  $p$  le nombre de places. Nous supposons les circuits numérotés de 1 à  $s$  et les places de 1 à  $p$ .

Nous définissons :

– la matrice  $D = [d_{ij}]$ ;  $i = 1, \dots, s$ ;  $j = 1, \dots, p$ , où  $d_{ij} = 1$  si la place  $j$  appartient au circuit  $i$ , et 0 sinon :

$$- A = \begin{bmatrix} a_1 \\ \vdots \\ a_s \end{bmatrix}, \quad \text{où } a_i = [\mu(i)/C_0] = a(i) \quad [\text{voir (14)}] \quad (15)$$

$$- X = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \quad \text{où } x_j = 1 \text{ si la place } j \text{ contient un jeton et 0 sinon.}$$

Le vecteur  $X$  est l'inconnue de notre problème qui s'écrit :

$$\text{Min} \left( \sum_{j=1}^p x_j \right)$$

sous les contraintes :

(P)  $DX \geq A$ , qui exprime qu'au moins un circuit de commande est critique.

$$x_j \in \{0, 1\} \quad \text{pour } j = 1, \dots, p.$$

$x_j$  est donné si la place  $j$  appartient à un circuit de commande ( $j \in \Gamma_c$ ).

*Remarque* : Bien entendu, on peut éliminer du critère les  $x_j$  tels que  $j \in \gamma$  lorsque  $\gamma \in \Gamma_c$ , car ils sont connus. Dans ce cas, il faut bien entendu modifier en conséquence la contrainte  $DX \geq A$ .

#### 4.3. Une heuristique pour la résolution de (P)

Le nombre de circuits élémentaires à prendre en compte dans les modèles que nous venons de proposer croît très rapidement, surtout du fait de l'existence des circuits mixtes. Dans l'exemple qui illustre ce papier, nous avons relevé 18 circuits élémentaires. Il est facile de voir que le nombre de places n'est pas, lui non plus, négligeable.

D'où l'intérêt de disposer d'une heuristique efficace.

Nous supposons que, dans le problème (P), le marquage des places de commande a été éliminé du critère et que la contrainte  $DX \geq A$  a été modifiée en conséquence. Nous conservons cependant les mêmes notations pour ne pas compliquer l'exposé.

Il s'agit donc de résoudre :

$$\text{Min} \left( \sum_{j=1}^p x_j \right)$$

(P') sous les contraintes :

$$DX \geq A \quad \text{et} \quad x_j \in \{0, 1\} \quad \text{pour } j = 1, \dots, p.$$

Dans la suite, nous utiliserons la notion de *degré de liberté*.

DÉFINITION : On appelle *degré de liberté* d'un circuit élémentaire  $i \in \{1, \dots, s\}$  la quantité  $f_i$  définie par :

$$f_i = \sum_{j=1}^p d_{ij} - a_i.$$

Cette quantité est la différence entre le nombre de places du circuit  $i$  et le nombre minimal de jetons qu'il doit contenir pour qu'un circuit de commande soit critique.  $f_i$  est donc d'autant plus grand que le nombre de possibilités de placement de cet ensemble minimal de jetons est grand, d'où son nom. A la limite, si  $f_i = 0$  il est nécessaire de placer un jeton dans chaque place, ce qui résout le problème pour ce circuit.

L'algorithme que nous proposons est basé sur une idée simple : mettre les jetons dans les places qui les valorisent le plus, c'est-à-dire dans les places qui appartiennent simultanément au plus grand nombre de circuits élémentaires, mais, dans le même temps, remplir d'abord les places qui appartiennent aux circuits élémentaires qui ont le plus faible degré de liberté.

L'idée qui précède est traduite par l'algorithme que nous proposons maintenant.

1. Calculer les degrés de liberté  $f_i$  pour  $i = 1, 2, \dots, s$ .
2. Classer les lignes de  $D$  et de  $X$  dans l'ordre croissant des  $f_i$ .
3. Pour  $j = 1, 2, \dots, p$ , calculer :

$$k_j = \text{Min} \left\{ q \text{ tel que } \prod_{i=1}^q d_{ij} = 0; q \in [1, 2, \dots, s] \right\}.$$

Si  $q$  n'existe pas, on pose  $q = s + 1$ .

$k_j - 1$  représente le nombre d'éléments consécutifs de la  $j$ -ième colonne de  $D$ , pris à partir du premier élément, qui sont égaux à 1. Notons qu'il existe

au moins une colonne  $j$  telle que  $k_j > 1$ , sinon le circuit correspondant à la première ligne ne contiendrait aucune place.

4. Calculer :

$$k = \max_{j=1, \dots, p} k_j.$$

5. Calculer  $J_s = \left\{ j \text{ tel que } \prod_{i=1}^{k-1} d_{ij} = 1 \right\}$ , choisir  $j(s) \in J_s$ , et faire  $x_{j(s)} = 1$ .

6. Pour  $i = 1, 2, \dots, s$ , faire :

$$a_i = a_i - d_{ij(s)}.$$

7. Supprimer la colonne  $j(s)$  de  $D$ , l'élément  $j(s)$  de  $X$ , et faire  $p = p - 1$ .

8. Supprimer les lignes  $i$  de  $D$  telle que  $a_i \leq 0$  et supprimer les éléments correspondants de  $A$ . Soit  $n(s)$  le nombre de circuits supprimés. Faire  $s = s - n(s)$ .

9. Si  $s > 0$ , aller en 1, sinon fin.

Nous voyons que, à chaque itération, nous retenons parmi les circuits qui ont le plus petit degré de liberté ceux qui ont au moins une place en commun, et nous attribuons un jeton à cette place. Nous appliquons donc les deux règles suivantes de manière hiérarchique à chaque itération :

1. satisfaire les circuits pour lesquels les possibilités de choix sont les plus restreintes.

2. affecter un jeton à une place qui appartient au plus grand nombre possible de circuits parmi ceux sélectionnés en 1.

Cet algorithme conduit en général à de bonnes solutions, sans que l'on puisse affirmer que ces solutions sont optimales. Le résultat 7 présente une condition suffisante pour que la solution donnée par l'algorithme soit optimale.

**RÉSULTAT 7 :** Soit  $X = (x_1, \dots, x_p)$  une solution admissible du problème  $(P')$ .

Alors, si  $\sum_{j=1}^p d_{ij} x_j = a_i$  pour tout  $i \in \Gamma_f$ ,  $X$  est optimale. Rappelons que  $\Gamma_f$

est l'ensemble des circuits de fabrication.

*Démonstration :* Dans ce cas, tous les circuits de fabrication contiennent le plus petit nombre de jetons qui assure qu'au moins un circuit de commande est critique.

#### 4.4. Le cas du « flow-shop »

Nous examinons maintenant le cas particulier du « flow-shop ». Rappelons que, dans un « flow-shop », tous les produits visitent toutes les machines dans le même ordre; seuls les temps de passage sur les machines diffèrent. La figure 8 donne un exemple de « flow-shop » qui fait intervenir deux machines  $M_1$  et  $M_2$  sur lesquelles passent trois types de produits  $P_1$ ,  $P_2$  et  $P_3$  dans les proportions  $1/4$ ,  $1/4$ ,  $1/2$  respectivement. Les gammes sont, avec les conventions d'écriture du paragraphe 3.1 :

$$P_1 : M_1(1), M_2(2)$$

$$P_2 : M_1(3), M_2(1)$$

$$P_3 : M_1(2), M_2(2).$$

Les ratios des demandes sont :

$$a_1 = 1/4; \quad a_2 = 1/4; \quad a_3 = 1/2.$$

Nous utiliserons les séquences admissibles suivantes :

$$U_1 = \{P_1, P_2, P_3, P_3\}; \quad U_2 = \{P_1, P_3, P_2, P_3\}.$$

Enfin, toujours avec les mêmes notations :

$$\mathcal{M} = \{M_1, M_2\}; \quad \mathcal{P} = \{P_1, P_2, P_3\}$$

$$R_1 = R_2 = R_3 = \{M_1, M_2\}$$

$$\mu(M_1^1) = 1, \quad \mu(M_2^1) = 2, \quad \mu(M_1^2) = 3$$

$$\mu(M_2^2) = 1, \quad \mu(M_1^3) = 2, \quad \mu(M_2^3) = 2.$$

La figure 8 modélise ce « flow-shop » (nous utilisons les mêmes conventions que pour la figure 7).

On observe que, dans le modèle d'un « flow-shop », *tous les circuits élémentaires mixtes contiennent au moins une place de ressources.*

A titre d'exemple, nous donnons trois circuits mixtes pour l'exemple de la figure 8 en soulignant la place de ressources.

$$\gamma_1 = (t_1^1, p_{f,1}^1, t_2^1, p_{c,2}^2, t_2^3, p_r^3, t_1^3, p_{c,4}^1, t_1^4, p_{c,1}^1)$$

$$\gamma_2 = (t_2^2, p_r^2, t_1^2, p_{c,3}^1, t_1^3, p_{f,1}^3, t_2^3, p_{c,3}^2)$$

$$\gamma_3 = (t_1^4, p_{f,1}^4, t_2^4, p_{c,1}^2, t_2^1, p_r^1, t_1^1, p_{c,2}^1, t_1^2, p_{c,3}^1, t_1^3, p_{c,4}^1).$$



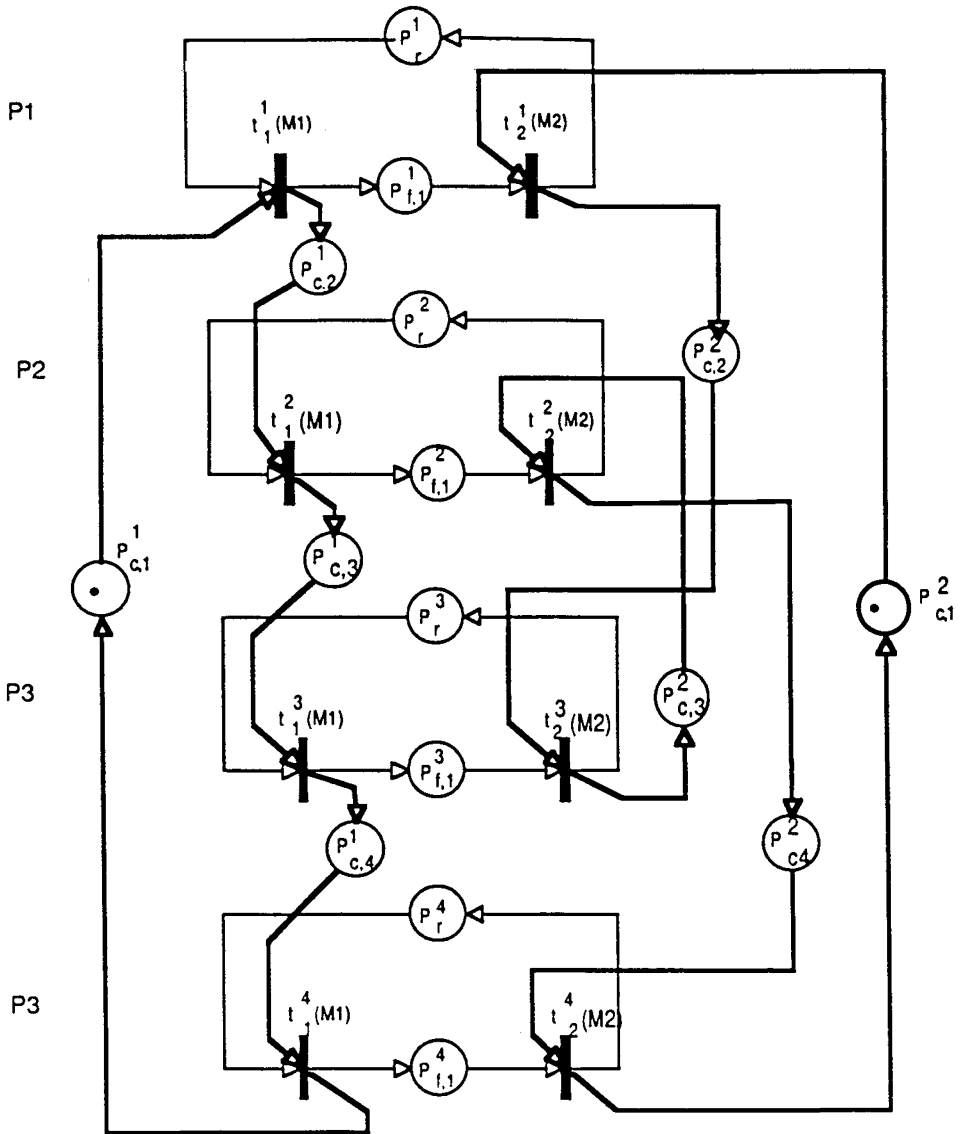


Figure 8

Si bien que tous les circuits autres que les circuits de commande contiennent au moins une place de ressources. Il en résulte qu'il est toujours possible de rendre critique au moins un circuit de commande en utilisant un marquage



TABLEAU

	$P_r^1$	$P_{f,1}^1$	$P_{f,2}^1$	$P_r^2$	$P_{f,1}^2$	$P_r^3$	$P_{f,1}^3$	$P_r^4$	$P_{f,1}^4$	$P_{c,1}^1$	$P_{c,2}^1$	$P_{c,3}^1$	$P_{c,1}^2$	$P_{c,2}^2$	$P_{c,1}^3$	$P_{c,2}^3$	$P_{c,3}^3$	$P_{c,4}^3$		D
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2
2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		1
3	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1
4	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0		2
5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0		1
6	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0		1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1		1
8	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1		2
9	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1		1
10	0	1	1	0	0	1	0	0	0	1	0	1	0	0	0	1	1	0		3
11	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0		2
12	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0		1
13	1	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0		2
14	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	0		2
15	1	1	0	1	0	1	0	0	1	0	0	1	0	1	1	0	1	0		3
16	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1		2
17	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	1	1		2
18	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	1		2

Notons que, pour ce cas particulier, le nombre de chariots obtenus est *optimal*. On peut en effet facilement vérifier que cinq chariots (*i.e.* jetons) représentent le nombre minimal qui permet de rendre les circuits de fabrication non critiques.

## 5. OPTIMISATION DES SÉQUENCES D'ENTRÉE DANS LES MACHINES

Dans cette dernière section, nous cherchons à définir les séquences d'entrée dans les machines permettant d'atteindre la productivité maximale avec un minimum de ressources de transport (*i. e.* d'en-cours, ou de jetons dans les circuits de fabrication du modèle).

Nous savons maintenant qu'il est toujours possible de saturer la machine la plus chargée et que, dans ce cas, la période  $\lambda$  attachée au système est égale à la somme des temps de passage des produits d'une séquence sur cette machine.

Nous donnons, en 5.1, un algorithme approché pour la recherche des séquences optimales. Cet algorithme approché est appliqué à notre exemple en 5.2.

### 5.1. Recherche des séquences optimales d'entrée dans les machines

#### 5.1.1. Le problème

Considérons un ordonnancement tel que chaque transition soit franchie une fois et une seule sur une période d'amplitude  $\lambda$  et pour un marquage initial donné.  $\lambda$  est la période attachée au système.

Si  $k$  est le nombre de jetons contenus dans un circuit de fabrication, les  $k$  jetons franchissent, au cours de la période  $\lambda$ , des séquences distinctes de transitions dont l'union est l'ensemble des transitions du circuit. La séquence de transitions franchies par un jeton donné est dans l'ordre des machines que ces transitions représentent dans la gamme du type de pièce correspondant au circuit de fabrication considéré.

L'exemple de la figure 9 illustre ce qui précède.

Dans cet exemple,  $\lambda=8$  et nous avons considéré 3 jetons dans le circuit de fabrication. Leur position initiale est donnée dans la figure (places  $p_r$ ,  $p_2$  et  $p_5$ ). Nous avons ensuite représenté une position possible dans le temps des premiers franchissements de  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ ,  $t_5$  et  $t_6$  (en supposant que cet ordonnancement obéit aux autres contraintes du problème). On observe que l'ordre global de la gamme, soit  $(t_1, t_2, t_3, t_4, t_5, t_6)$ , a été remplacé par trois ordres partiels  $(t_1, t_2)$ ,  $(t_3, t_4, t_5)$  et  $(t_6)$ . Dans la pratique, cela signifie que les machines peuvent ne pas être utilisées dans l'ordre de la gamme à condition qu'elles soient utilisées pour fabriquer des pièces différentes (représentées par des jetons différents). Chaque axe des temps de la figure 9 est attaché à un jeton différent.

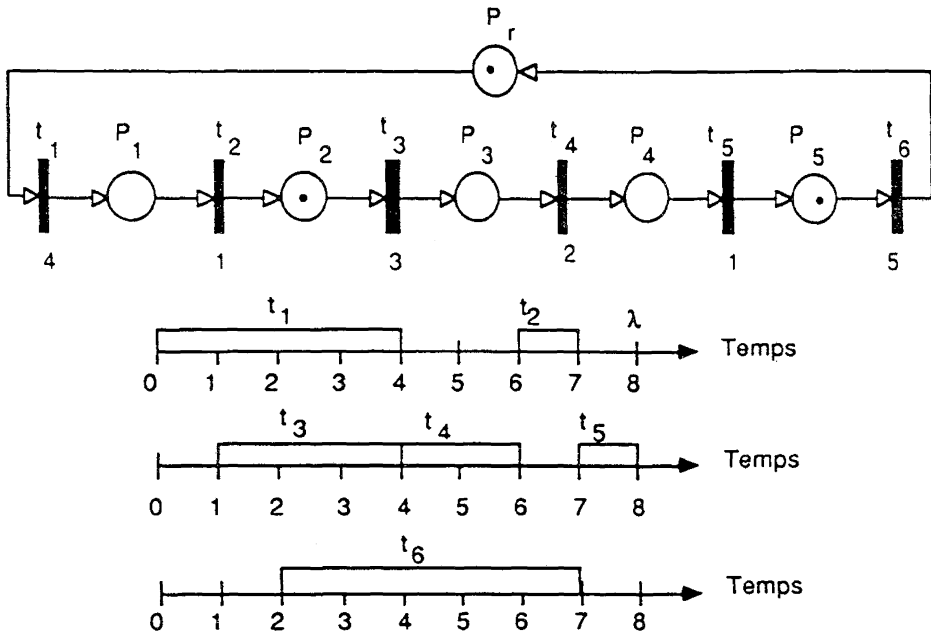


Figure 9

Si nous considérons maintenant l'ensemble des circuits de fabrication du système, nous savons que les périodes de franchissement de transitions représentant la même machine sont disjointes.

Le problème que nous cherchons à résoudre s'écrit :  
déterminer un marquage initial comportant un nombre minimal de jetons tel qu'il existe un ordonnancement du type précédent sur une période d'amplitude  $\lambda$ .

### 5.1.2. Présentation informelle de l'algorithme approché

L'idée de base de l'algorithme est d'affecter à chaque circuit de fabrication autant de périodes d'amplitude  $\lambda$  que de jetons qu'il doit contenir au minimum pour qu'un circuit de commande soit critique, puis d'essayer de placer sur ces périodes les périodes de franchissement des transitions sous les contraintes données au paragraphe précédent : disjonction des périodes de franchissement de transitions représentant une même machine et séquençement correspondant à la gamme sur chaque période d'amplitude  $\lambda$  (voir fig. 9).

S'il est possible de trouver un ordonnancement sur les périodes d'amplitude  $\lambda$  initialement affectées aux circuits de fabrication, on est certain d'avoir

obtenu un marquage initial. Sinon, on est contraint d'ajouter des périodes supplémentaires à un ou plusieurs circuits de fabrication pour pouvoir trouver un ordonnancement satisfaisant les contraintes. Le nombre total de périodes d'amplitude  $\lambda$  finalement utilisées est le nombre de jetons utilisés.

L'heuristique procède de manière itérative. Supposons ordonnancées les périodes de franchissement correspondant aux franchissements des  $k$  premières transitions d'un circuit de fabrication donné. Cet ordonnancement occupe un certain nombre de périodes d'amplitude  $\lambda$ .

Si ce nombre est inférieur au nombre de périodes initialement affectées, le temps disponible pour placer les périodes de franchissement restantes est la somme :

- du temps séparant l'instant de fin du premier franchissement de la  $k$ -ième transition de  $\lambda$ ;
- de  $r\lambda$ , où  $r$  est le nombre de périodes d'amplitude  $\lambda$  non encore utilisées.

Si le nombre de périodes occupées est supérieur au nombre de périodes initialement affectées, nous conservons seulement le premier terme de la somme.

Le rapport entre la somme des temps de franchissement des transitions non encore examinées et le temps disponible que nous venons de définir est noté  $\bar{d}_i(k)$ , où  $i$  est le circuit de fabrication considéré.

Nous définissons encore  $d_i(k) = 1 - \bar{d}_i(k)$ .

A l'itération suivante, on retient le circuit de fabrication correspondant à la plus petite valeur de  $d_i(k)$ .

Si cette valeur est négative, on est assuré de ne pas pouvoir placer la période de franchissement suivante sur l'intervalle de temps restant. On affecte donc une période supplémentaire d'amplitude  $\lambda$  à ce circuit de fabrication.

Dans tous les cas, la transition dont on va placer la période de franchissement est choisie comme suit :

- si une transition a déjà été examinée pour le circuit de fabrication considéré, on choisit la transition qui suit dans l'ordre des machines correspondantes dans la gamme du type de produit;
- sinon, on choisit la transition qui conduira à la valeur maximale pour  $d_i(1)$ .

On cherche ensuite à placer la période de franchissement du tirage ainsi défini conformément aux contraintes. Ce placement peut conduire à affecter au circuit de fabrication une période d'amplitude  $\lambda$  supplémentaire.

Nous illustrons cet algorithme par un exemple.

Nous nous rapportons à l'exemple du paragraphe 3.1 qui comporte (voir fig. 7) quatre circuits de fabrication que nous examinerons dans l'ordre dans lequel ils sont représentés dans la figure.

Dans cet exemple,  $\lambda = 6$  et la machine  $M_3$  est la plus chargée.

*Première itération :*

$$\begin{aligned} d_1(0) &= 1 - 7/12 = 5/12; & d_2(0) &= 1 - 3/6 = 1/2; \\ d_3(0) &= 1 - 3/6 = 1/2; & d_4(0) &= 1 - 3/6 = 1/2. \end{aligned}$$

Nous retenons donc le circuit de fabrication 1 car

$$d_1(0) < d_2(0) = d_3(0) = d_4(0).$$

Nous essayons successivement les placements des franchissements de  $t_1^1$ ,  $t_2^1$  et  $t_3^1$  et calculons  $d_1(1)$  dans chaque cas.

$d_1(1)$  est maximal pour  $t_2^1$  et  $t_3^1$  ( $5/9$ ). Nous plaçons le franchissement de  $t_2^1$ .

La situation après placement de ce premier franchissement est donnée par la figure 10.

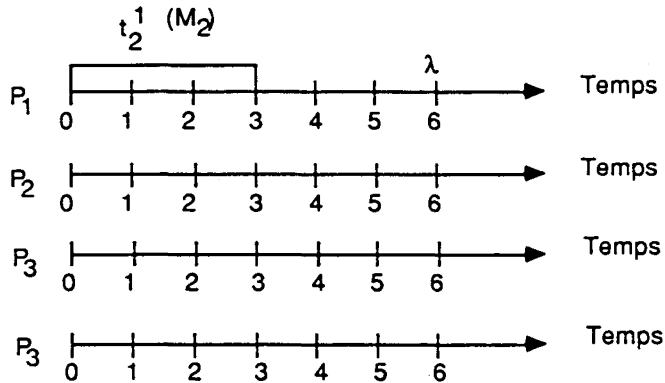


Figure 10

*Seconde itération :*

Après placement du franchissement de  $t_2^1$ , nous avons :

$$d_1(1) = 1 - 4/9 = 5/9, \quad d_2(0) = d_3(0) = d_4(0) = 1/2.$$

Comme  $d_2(0) = d_3(0) = d_4(0) < d_1(1)$ , nous avons le choix entre les circuits élémentaires de fabrication 2, 3 et 4. Nous choisissons 2.

Nous essayons de placer les franchissements de  $t_1^2(M_3)$  et de  $t_2^2(M_2)$ .

Si nous plaçons le franchissement de  $t_1^2$ ,  $d_2(1) = 1 - 2/5 = 3/5$ .

Si nous plaçons le franchissement de  $t_2^2$ , nous ne pouvons le placer qu'entre 3 et 5 pour éviter de chevaucher le franchissement de  $t_2^1$  déjà placé et qui correspond à la même machine. Dans ce cas,  $d_2(1) = 1 - 1 = 0$ .

Nous choisissons donc de placer le franchissement de  $t_1^2$  comme l'indique la figure 11.

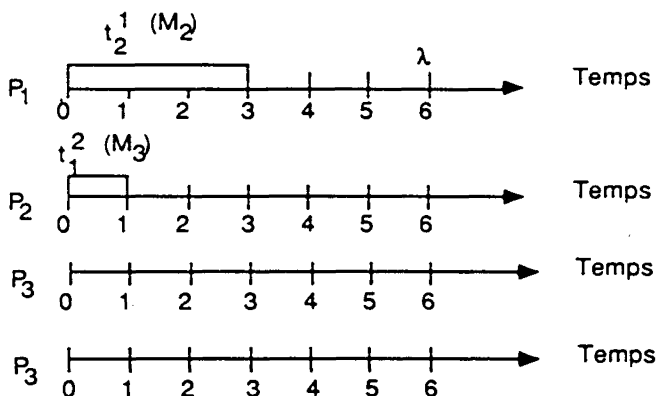


Figure 11

*Troisième itération :* Au début de la troisième itération, nous avons :

$$d_1(1) = 5/9; \quad d_2(1) = 3/5; \quad d_3(0) = d_4(0) = 1/2.$$

Nous avons le choix entre les circuits élémentaires de fabrication 3 et 4 car  $dl(0) = d_4(0) < \min(d_1(1), d_2(1))$ . Nous choisissons le circuit 3.

Si nous plaçons le franchissement de  $t_1^3$ , il débutera à l'instant 0 et se terminera à l'instant 2, car il n'entre en conflit avec aucun des tirages déjà placés.

Si nous plaçons le franchissement de  $t_2^3$ , il ne pourra débiter qu'à l'instant 1, car il entre en conflit avec le franchissement de  $t_1^2$ . Dans ce cas :  $d_3(1) = 1 - 2/4 = 1/2$ . Nous plaçons le franchissement de  $t_1^3$  comme l'indique la figure 12.

*Quatrième itération :* Après les placements que nous venons de faire :

$$d_1(1) = 5/9; \quad d_2(1) = 3/5; \quad d_3(1) = 3/4; \quad d_4(0) = 1/2.$$

Nous choisissons donc le circuit 4 car  $d_4(0) < \min(d_1(1), d_2(1), d_3(1))$ .



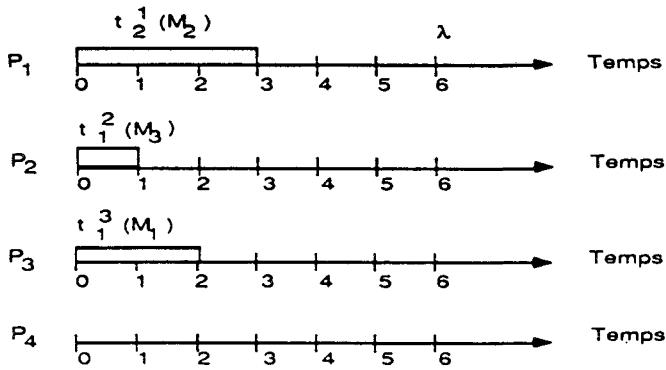


Figure 12

Nous avons le choix entre le placement du franchissement de  $t_4^1(M_1)$  et celui de  $t_2^4(M_3)$ .

Si nous choisissons  $t_4^1$ , nous le plaçons entre 2 et 4 pour éviter le conflit avec le franchissement de  $t_1^3$  qui concerne la même machine et  $d_4(1) = 1 - 1/2 = 1/2$ .

Si nous choisissons le franchissement de  $t_2^4$ , nous le placerons entre 1 et 2 pour éviter le conflit avec le franchissement de  $t_1^2$  et  $d_4(1) = 1 - 1/2 = 1/2$ .

Nous pouvons donc placer indifféremment le franchissement de  $t_4^1$  ou celui de  $t_2^4$ . Nous choisissons de placer le premier, ce qui nous conduit à la figure 13.

*Cinquième et sixième itérations :* Au début de la cinquième itération, nous avons :

$$d_1(1) = 5/9; \quad d_2(1) = 3/5; \quad d_3(1) = 3/4; \quad d_4(1) = 1/2.$$

Comme  $d_4(1) < \min(d_1(1), d_2(1), d_3(1))$ , nous placerons le franchissement de  $t_2^4$ , ce qui réglera le problème du circuit 4.

Comme ensuite,  $d_1(1) < \min(d_2(1), d_3(1))$ , nous placerons le franchissement de  $t_1^3$  sur un second intervalle  $(0, \lambda)$  (i.e. en introduisant un second jeton dans le circuit 1) entre 1 et 4.

La figure 14 donne la situation à la fin de la sixième itération.

*Septième, huitième et neuvième itérations :* Au début de cette itération :

$$d_1(2) = 1/2; \quad d_2(1) = 3/5; \quad d_3(1) = 3/4.$$

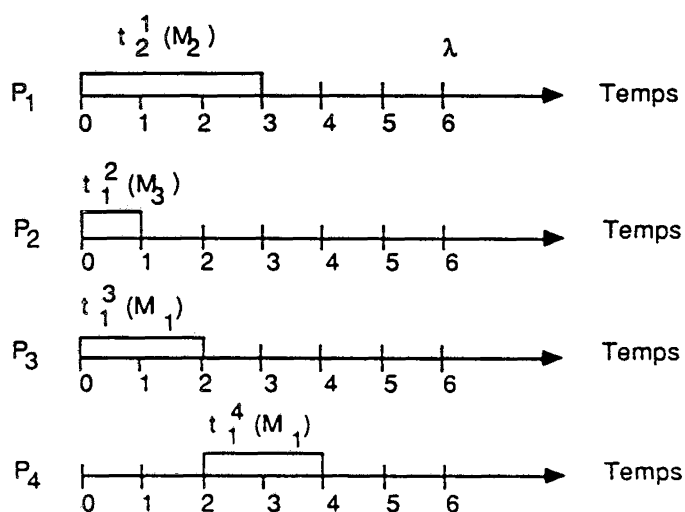


Figure 13

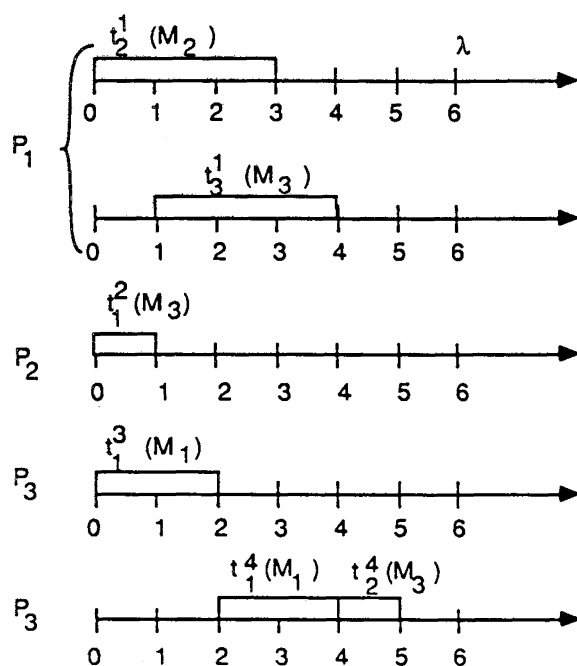


Figure 14

Nous choisissons donc de placer le franchissement de  $t_1^1(M_1)$  sur le second intervalle  $(0, \lambda)$  entre 4 et 5.

Nous choisissons ensuite le second circuit car  $d_2(1) < d_3(1)$  et nous plaçons le franchissement de  $t_2^2(M_2)$  entre 3 et 5.

Enfin, nous plaçons  $t_2^3(M_3)$  entre 5 et 6.

Le placement complet est donné par la figure 15.

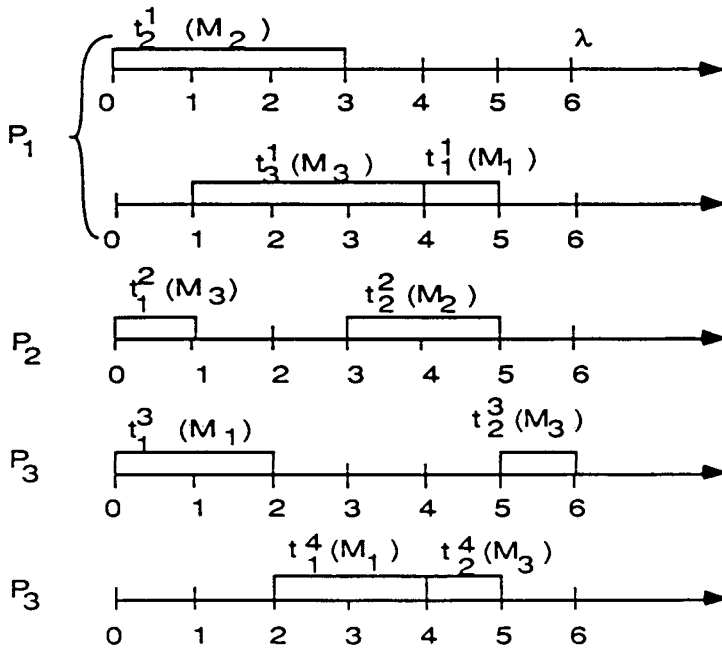


Figure 15

Nous avons obtenu un ordonnancement qui requiert le minimum de jetons dans chacun des circuits élémentaires de fabrication: nous sommes donc assurés qu'il est optimal.

La figure 15 nous montre que les séquences d'entrée sont :

pour  $M_1$ :  $P_3, P_3, P_1$ ; pour  $M_2$ :  $P_1, P_2$ ; pour  $M_3$ :  $P_2, P_1, P_3, P_3$ .

Ces séquences d'entrée sont optimales pour notre exemple et ne correspondent pas aux circuits de commande de la figure 7.

### 5.1.3. Description formelle de l'algorithme approché

Pour tout  $i \in \Gamma_f$ , nous désignons par :

- $T_i$  l'ensemble des transitions de  $i$ ;
- $T_i(k)$  l'ensemble des  $k$  premières transitions à franchir lorsqu'on les considère dans l'ordre des machines qu'elles représentent dans la gamme du type de pièce  $P_i$ ;
- $\varphi_i(k)$  l'instant de fin de premier franchissement de la  $k$ -ième transition de  $i$  si les transitions sont prises dans l'ordre des machines correspondantes dans la gamme de  $P_i$ ;
- $N_i(k)$  représente le nombre de jetons qui serait nécessaire pour atteindre la solution optimale si le circuit de fabrication  $i$  se limitait aux  $k$  premières transitions et aux places correspondantes;
- Nous notons :

$$N_i^*(k) = \mathcal{M}ax[N_i(k), a_i], \text{ où } a_i \text{ est défini dans (15).}$$

$N_i^*(k)$  est le maximum entre le nombre de jetons utilisés si le circuit élémentaire se limitait aux  $k$  premières transitions et la borne inférieure du nombre de jetons à placer dans ce même circuit pour être assuré qu'un circuit de commande sera critique.

— Nous définissons enfin le *degré de liberté attaché* à  $i \in \Gamma_f$  connaissant l'ordonnancement des franchissements des  $k$  premières transitions de  $i$  par :

$$d_i(k) = 1 - \overline{d}_i(k)$$

où (16)

$$\overline{d}_i(k) = \left[ \sum_{t \in T_i - T_i(k)} \mu(t) \right] / [(\lambda - \varphi_i(k)) + (N_i^*(k) - N_i(k))\lambda].$$

Le numérateur de  $\overline{d}_i(k)$  représente la charge en temps restant à placer sur la période  $[0, \lambda]$  pour effectuer l'ordonnancement des franchissements des transitions appartenant à  $T_i - T_i(k)$ .

Considérons le dénominateur de  $\overline{d}_i(k)$ . Le premier terme de la somme,  $\lambda - \varphi_i(k)$ , représente le temps disponible sur le dernier axe des temps utilisé pour l'ordonnancement des franchissements des  $k$  premières transitions.

Le premier facteur du second terme,  $[N_i^*(k) - N_i(k)]$ , représente le nombre de périodes  $(0, \lambda)$  encore entièrement disponibles après l'ordonnancement du franchissement de la  $k$ -ième transition si l'on se réfère au minimum  $a_i$  de jetons [*i.e.* de période  $(0, \lambda)$ ] nécessaire pour être assuré qu'un circuit de

commande sera critique. Cette quantité peut être nulle si ce minimum est déjà atteint.

Finalement,  $\bar{d}_i(k)$  est le rapport entre la période minimale de temps nécessaire pour ordonnancer les franchissements des transitions non encore considérées et le temps encore disponible.

Si  $\bar{d}_i(k) > 1$ , cela signifie que, dans le meilleur des cas, il n'est plus possible d'ordonnancer les franchissements des transitions de  $T_i - T_i(k)$  dans la période de temps restante. Il faut alors ajouter  $g_i(k)$  périodes  $(0, \lambda)$  [i. e.  $g_i(k)$  jetons],  $g_i(k)$  étant le plus petit entier tel que :

$$\sum_{t \in T_i - T_i(k)} \mu(t) / [(\lambda - \varphi_i(k)) + (N_i^*(k) - N_i(k))\lambda + g_i(k)\lambda] \leq 1 \quad (17)$$

$d_i(k)$  est alors d'autant plus proche de 0 qu'il sera plus difficile de placer les franchissements restants.

C'est à partir de ces idées qu'est construit l'algorithme approché que nous décrivons maintenant.

## 1. INITIALISATION

### 1.1. Pour chaque circuit élémentaire $i$ ( $i = 1, \dots, n$ )

1.1.1. Faire  $k_i = 0$ .  $k_i$  sera, dans la suite, le nombre de franchissements de transitions de  $i \in \Gamma_f$  déjà traitées à l'itération considérée.

1.1.2. Faire  $N_i(k_i) = N_i(0) = 1$  : C'est le nombre de périodes d'amplitude  $\lambda$  disponibles pour placer les périodes de franchissement des transitions suivantes.

1.1.3. Faire  $\varphi_i(k_i) = \varphi_i(0) = 0$  :  $\varphi_i(k_i)$  contiendra l'instant au plus tôt où le franchissement de la transition de  $i \in \Gamma_f$  pourra être effectué.

1.1.4. Calculer  $d_i(k_i) = d_i(0)$  [voir (16)].

1.2. Faire  $m = 1$  :  $m$  contiendra le rang de l'itération.

1.3. Faire  $E_m = \{1, 2, \dots, n\}$  :  $E_m$  contient l'ensemble des circuits non encore complètement traités à l'itération  $m$ .

## 2. ORDONNANCEMENT

2.1. Calculer  $i_i$  tel que :

$$d_{ii}(k_{ii}) = \min_{i \in E_m} d_i(k_i)$$

$i_i$  est le circuit élémentaire considéré dans la suite de l'itération.

2.2. Si  $d_{ii}(k_{ii}) < 0$ .

2.2.1. Ajouter  $g_{ii}(k_{ii})$  périodes d'amplitude  $\lambda$  pour l'ordonnancement des périodes de franchissement des transitions non encore examinées dans le circuit élémentaire  $i_i$  [voir (17)].

2.2.2. Faire  $N_{ii}(k_{ii}) = N_{ii}(k_{ii}) + g_{ii}(k_{ii})$ .

2.3. Choix de la transition à franchir, notée  $t_{ii}$ .

Si  $k_{ii} > 0$ , on choisit la transition suivant immédiatement la dernière transition placée dans l'ordre des machines de la gamme du type de produit correspondant au circuit élémentaire considéré; si  $k_{ii} = 0$ , on choisit la transition qui conduit à la valeur maximale de  $d_{ii}(1)$ .

2.4. Placement de la période de franchissement.

Les explications sont fournies à la fin de cet algorithme.

2.5. Faire  $T_{ii}(k_{ii} + 1) = T_{ii}(k_{ii}) \cup \{t_{ii}\}$ .

2.6. Calculer  $\varphi_{ii}(k_{ii} + 1)$ .

Les explications sont fournies à la fin de l'algorithme.

2.7. Calculer  $d_{ii}(k_{ii} + 1)$  [voir (16)].

2.8. Faire :

$$k_{ii} = k_{ii} + 1.$$

2.9. Faire :

$$m = m + 1.$$

2.10. Calculer :

$$E_m = \{i/i \in \{1, \dots, n\} \text{ et } T_i - T_i(k_i) \neq \emptyset\}.$$

2.11. Si  $E_m \neq \emptyset$ , aller en 2.1, sinon fin.

Nous faisons quelques remarques concernant le placement du franchissement de la transition  $t$  choisi pour être placé au rang  $k_{ii} + 1$ .

Plusieurs cas peuvent se présenter ( $t$  est la transition à placer) :

(a) les contraintes du problème (voir § 5.1.1) autorisent le placement à partir de  $\varphi_{ii}(k_{ii})$ , et  $\lambda - \varphi_{ii}(k_{ii}) \geq \mu(t)$ .

Dans ce cas, le placement est effectué à partir de  $\varphi_{ii}(k_{ii})$  et  $\varphi_{ii}(k_{ii} + 1) = \varphi_{ii}(k_{ii}) + \mu(t)$ .

(b) les contraintes du problème n'autorisent le placement qu'à partir d'un instant  $h_{ii}(k_{ii}) > \varphi_{ii}(k_{ii})$ , et  $\lambda - h_{ii}(k_{ii}) \geq \mu(t)$ .

Dans ce cas, le placement est effectué à partir de  $h_{ii}(k_{ii})$  et  $\varphi_{ii}(k_{ii} + 1) = h_{ii}(k_{ii}) + \mu(t)$ .

(c) enfin,  $\lambda - \varphi_{ii}(k_{ii}) < \mu(t)$  si le placement du tirage suivant est autorisé à partir de  $\varphi_{ii}(k_{ii})$  ou  $\lambda - h_{ii}(k_{ii}) < \mu(t)$  si le placement du tirage suivant est autorisé à partir de  $h_{ii}(k_{ii}) > \varphi_{ii}(k_{ii})$ .

Dans ce cas on est obligé de rajouter un intervalle de temps supplémentaire, ce qui autorise le placement conformément aux cas (a) et (b) précédents, mais à partir de l'instant 0 [en d'autres termes,  $\varphi_{ii}(k_{ii})$  devient égal à 0].

#### 5.1.4. Discussion de l'algorithme approché

L'algorithme que nous venons de présenter est basé sur une idée simple : placer au plus tôt les périodes de franchissement *a priori* les plus difficiles à placer, en respectant les contraintes.

Il est facile de voir que, dans tous les cas, il ne conduira pas à plus de jetons par circuit de fabrication qu'il n'y a de places. Alors tout ordonnancement conduit à une productivité maximale, comme nous l'avons vu (propriété 6).

C'est la seule majoration que nous pouvons proposer.

Nous avons par ailleurs effectué 100 applications numériques sur des exemples comportant entre 5 et 15 machines et 5 à 15 produits ayant des gammes comportant de 2 à 5 machines et avec des ratios de production exigeant entre 5 et 20 circuits de fabrication.

Les résultats obtenus sont les suivants :

- nombre minimal de jetons : 71 cas.
- dépassement de 1 du nombre minimal de jetons : 18 cas.
- dépassement de 2 du nombre minimal de jetons : 11 cas.

Sauf dans le cas où le nombre minimal de jetons est atteint, on ne peut décider si la solution est ou non optimale.

Il faut cependant souligner le nombre important de cas (71) qui permettent d'affirmer l'optimalité.

La recherche d'un « plus mauvais cas » se poursuit en s'inspirant des approches utilisées dans [10] et [11]. Bien entendu, l'existence des contraintes complique considérablement l'étude.

## 6. CONCLUSION

Cette communication a montré l'utilité des graphes d'événements dans la modélisation des « job-shops » lorsque les fabrications sont répétitives. L'utilité des graphes d'événements temporisés tient au fait qu'il permet de séparer la modélisation du contrôle de celle du processus.

Un résultat important et nouveau a été trouvé : il est toujours possible de saturer la (ou les) machine(s) la (ou les) plus chargée(s). Ce résultat, bien connu pour les « flow-shops », est nouveau dans le cas général des « job-shops ».

Le problème de la minimisation du nombre de ressources de transport (*i. e.* de l'en-cours maximal) lorsque les séquences d'entrée sont connues a été examiné. Nous avons montré que la solution s'obtient en résolvant un problème de programmation linéaire avec variables binaires. Malheureusement ce problème est, dans la pratique, de très grande taille, et il n'est pas raisonnable d'espérer pouvoir utiliser les algorithmes classiques. C'est pourquoi nous avons proposé une heuristique qui, après de nombreuses utilisations, s'est révélée performante. Elle passe par la recherche préalable de tous les circuits élémentaires du graphe : nous pensons qu'il est possible de trouver des heuristiques qui ne seraient pas soumises à cette contrainte qui reste pénalisante.

Enfin, l'heuristique de recherche de « bonnes » séquences d'entrée dans les machines est satisfaisante. La voie reste cependant ouverte à la recherche. Notons que cette dernière heuristique donne à la fois de bonnes réponses d'entrée, le nombre de jetons correspondants et leur position initiale.

## BIBLIOGRAPHIE

1. J. BARRE, H. HILLION et J. M. PROTH, *Conditions de fonctionnement et propriétés d'optimalité d'un atelier flexible*, Rapport de Recherche I.N.R.I.A., n° 247, 1983, I.N.R.I.A., B.P. n° 105, 78153 Le Chesnay, France.
2. G. W. BRAMS, *Réseaux de Petri : théorie et pratique*, Masson, Paris, 1983.
3. P. CHRETIENNE, *Les réseaux de Petri Temporisés*, Thèse d'Etat, Université de Paris-VI, Paris.



4. G. COHEN, D. DUBOIS, J. P. QUADRAT et M. VIOT, A Linear-System Theoric View of Discrete Event Processes and its View for Performance Evaluation in Manufacturing. *I.E.E.E. Trans. on Automatic Control*, vol. AC-30, n° 3, 1985, p. 210-220.
5. G. COHEN, P. MOLLER, J. P. QUADRAT et M. VIOT, Linear System Theory for Discrete Event System, *Proceedings on 23rd Conference on Decision and Control*, vol. 1, 1985, p. 539-544.
6. F. COMMONER, A. W. HOLT, S. EVEN et A. PHUELI, Marked Directed Graphs, *Journal of Computer and System Sciences*, vol. 5, n° 5, 1971, p. 511-523.
7. J. MARTINEZ et M. SILVA, A Simple and Fast Algorithm to Obtain all Invariants of a Generalized Petri Net, *Lecture notes in Computer Science*, n° 52, Springer-Verlag, Berlin, 1980, p. 302-310.
8. H. ALAIWAN et J. M. TOUDIC, Recherche des semi-flots, des verrous et des trappes dans les Réseaux de Petri, *Techniques et Sciences Informatiques*, vol. 4, n° 11, 1985, p. 103-112, Dunod, Paris.
9. J. L. PETERSON, *Petri Nets Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs, N.J., 1981.
10. E. G. COFFMAN Jr., H. R. GAREY, P. S. JOHNSON et R. E. TARJAN, Performance Bounds for Level Oriented Two-Dimensional Packing Algorithms, *S.I.A.M. J. Comput.*, vol. 9, n° 4, novembre 1980.
11. B. S. BAKER, E. G. COFFMAN Jr. et R. L. RIVEST, Orthogonal Packings in Two Dimension, *S.I.A.M. J. Comput.*, vol. 9, n° 4, novembre 1980.