

## BORNES INFÉRIEURES POUR L'ORDONNANCEMENT INTÉGRÉ DE LA PRODUCTION ET DU TRANSPORT DANS LES FMS AVEC UN SEUL CHARIOT FILOGUIDE

**M.L. Espinouse, P. Lacomme,**

Université de Technologie de Troyes,  
Laboratoire d'Optimisation des  
Systèmes Industriels (LOSI)  
12, rue Marie Curie, BP 2060 10010  
Troyes Cedex (France), Tel. : +33 3  
25 71 {58 36, 58 60},  
E-mail : {Espinouse,  
Lacomme}@univ-troyes.fr

**A. Moukrim**

Université de Technologie de  
Compiègne  
HeuDiaSyC, UMR CNRS 6599  
BP 20529  
60205 Compiègne cedex (France)  
Tel. : +33 3 44 23 49 52  
E-mail : moukrim@hds.utc.fr

**N. Tchernev**

Université Blaise Pascal Laboratoire  
d'Informatique, de Modélisation et  
d'Optimisation des Systèmes  
(LIMOS)  
Campus Universitaire des Cézeaux  
63177 Aubière Cedex France, Tel. :  
+33 4 73 40 75 14,  
E-mail : tchernev@isima.fr

**RESUME :** *Cet article concerne les problèmes d'ordonnancement des jobs en entrée et de planification des mouvements du moyen de transport/manutention dans les systèmes flexibles de production comportant un chariot sans conducteur automatiquement guidé. La résolution exacte peut être envisagée sous la forme d'une procédure de type Branch and Bound. Cependant, souhaitant prendre en compte toutes les contraintes de fonctionnement de ce type d'ateliers (capacités limitées des buffers d'entrée/sortie, blocage en transfert...), nous avons proposé une approche basée sur une procédure de Branch and Bound couplée avec un modèle de simulation à événements discrets. Pour améliorer l'efficacité de l'approche de résolution lors de l'exploration de l'arbre de recherche, il est nécessaire de connaître ou établir des bornes inférieures de qualité. Ainsi, dans cet article nous réalisons un état de l'art sur les idées principales de la littérature qui permettraient d'établir de nouvelles bornes inférieures dédiées aux systèmes flexibles de production.*

**MOTS-CLES :** *Système flexible de production, chariot automatiquement guidé, Branch and Bound, simulation, borne inférieure*

### 1. INTRODUCTION

Un système flexible de production avec un chariot filoguidé est un atelier automatisé composé de  $m$  stations et d'un système de manutention utilisant un chariot filoguidé qui assure le transport des jobs entre les stations ainsi que le transport entre les stations et la station d'entrée/sortie. Un tel atelier se compose de  $g$  gammes (ou types de jobs) et de  $n$  jobs à ordonnancer en entrée de l'atelier. Chaque gamme définit la liste des stations sur lesquelles la pièce doit être traitée et le temps de traitement de la pièce sur chaque station. Chaque station se compose d'un stock d'entrée, d'un poste d'usinage et d'un stock de sortie. Le système de gestion des chariots ne permet pas d'anticipation.

Les contraintes de fonctionnement à prendre en compte sur ce type de systèmes sont les suivantes :

- la politique de gestion du chariot lorsqu'au moins une pièce se trouve dans un stock de sortie d'une station. Cette politique permet de déterminer quelle pièce doit être traitée par le chariot lorsque plusieurs pièces sont en attente dans les stocks de sorties des stations.
- la politique de gestion du chariot lorsque aucune pièce ne se trouve dans un stock de sortie d'une station.

- la politique de choix d'une pièce dans le stock d'entrée d'une station.

L'étude présentée concerne les FMS avec un chariot :

- les politiques considérées pour la gestion du chariot lorsqu'au moins une pièce se trouve dans un stock de sortie d'une station sont FIFO (First In First Out), STT (Shortest Travel Time), SPT (Shortest Processing Time), MOQS (Maximum Outgoing Queue Size).
- la politique de gestion utilisée lorsque aucune pièce ne se trouve dans un stock de sortie consiste à faire attendre le chariot à l'endroit où il se trouve.
- la politique utilisée pour choisir une pièce dans le stock d'entrée d'une station est FIFO.

Le problème consiste à résoudre de manière conjointe le problème d'ordonnancement des jobs en entrée et la gestion du chariot (job input sequencing and vehicle dispatching) (Blazewicz *et al.* 1991,1994). Pour cela il est possible d'utiliser une approche de type Branch and Bound où chaque nœud de l'arbre de recherche correspond à une solution partielle évaluée par simulation afin de prendre en compte toutes les contraintes de fonctionnement.

Si  $x_p$  est un ordre partiel des jobs en entrée pour lequel les  $p$  premières positions ont été fixées et si  $H(x_p)$  est le makespan associé à l'ordre partiel  $x_p$ , ( $H(x_p)$  est évalué en utilisant un modèle de simulation à événements discrets), alors l'arbre de recherche ressemble à celui présenté sur la figure 1.

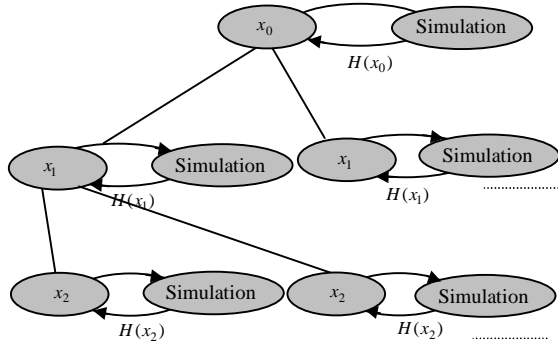


Figure 1. Schéma de principe (Lacomme *et al.*, 2000a)

Le problème concerne la détermination d'une borne inférieure pour élaguer rapidement l'arbre de recherche. Nous introduisons les notations suivantes :

$x_p$  : ordre partiel des jobs pour lequel seulement  $p$  jobs ont été ordonnancés

$U_{n-p}$  : ensemble des jobs non encore ordonnancés associés à l'ordre partiel  $x_p$

$H(x_p)$  : makespan associé à l'ordre partiel  $x_p$

$t_{ij}$  : temps de déplacement du chariot à charge pour transporter un job de la station  $i$  à la station  $j$  (les temps de chargement et déchargement sont compris)

$p_{ik}$  : processing time du job  $k$  sur la station  $i$

$\bar{H}^p(x_n)$  : borne inférieure du makespan associé à  $x_n$  avec seulement  $p$  jobs déjà ordonnancés ; seul  $H(x_p)$  est accessible par simulation puisque l'ordre des  $n-p$  jobs restants n'a pas été déterminé.

$E(x_p)$  : date d'entrée du  $p^{ième}$  job dans le système

$N$  : nombre de jobs simultanément autorisés dans le système

$n_k$  : nombre de jobs de type  $k$  devant être traités

On suppose que tout job  $k$  doit passer par  $m_k + 1$  stations où la première station est la station d'entrée et la dernière est la station de sortie. De plus, on désigne par  $M(k,i)$  : numéro de la  $i^{ième}$  station sur laquelle le job  $k$  passe avec  $M(k,0)$  la station d'entrée du système et  $M(k,m_k)$  la station de sortie du système.

$v_{i,j}$  : temps de déplacement du chariot à vide de la station  $i$  à la station  $j$  (le temps de chargement et le temps de déchargement ne sont pas pris en compte

puisque'ils sont inclus dans  $t_{ij}$ )

Le problème consiste à proposer une évaluation aussi précise que possible de :  $\bar{H}^p(x_n)$

## 2. BORNES INFÉRIEURES EXACTES

### 2.1. Déplacements à charge et temps d'occupation des machines

Nous proposons une borne inférieure très simple du temps de traitement des jobs de  $U_{n-p}$ . Pour traiter les jobs  $k \in U_{n-p}$ , il est nécessaire de transporter chaque job d'une station à l'autre en fonction de sa gamme. Comme on ne dispose que d'un seul chariot, la distance totale parcourue pour le job  $k$  vaut  $\sum_{i=0}^{m_k-1} t_{M(k,i),M(k,i+1)}$ . De

même,  $\sum_{k \in U_{n-p}} \sum_{i=0}^{m_k-1} t_{M(k,i),M(k,i+1)}$  est le temps total de transport à charge. De plus, comme chaque job  $k$  a un processing time  $p_{ik}$  sur la station  $i$ , le temps total de traitement sur la station  $i$  est :  $\sum_{k \in U_{n-p}} p_{i,k}$ . Donc,

$Max_i \sum_{k \in U_{n-p}} p_{i,k}$  donne la somme des processing time sur la station goulet. Il s'ensuit qu'une borne inférieure du temps de traitement des jobs de  $U_{n-p}$  est donnée par le maximum de ces deux quantités :

$$\nabla = Max \left( \sum_{k \in U_{n-p}} \sum_{i=0}^{m_k-1} t_{M(k,i),M(k,i+1)}, Max_i \sum_{k \in U_{n-p}} p_{i,k} \right).$$

La date d'entrée du  $p^{ième}$  job dans le système est  $E(x_p)$ . Cette date est un minorant de la date d'entrée du job numéro  $p+1$  (le job  $p+1$  ne peut pas entrer dans le système avant que le job  $p$  ne soit entré). Ensuite le chariot amène le  $p^{ième}$  job vers la première station qui doit exécuter la première opération du job  $p$ . La durée de transport est  $t_{0,M(p,1)}$ . Pour faire entrer le job suivant, il faut que le chariot soit présent sur la station d'entrée. Ce déplacement dans le meilleur des cas est un déplacement à vide du chariot de  $M(p,1)$  vers la station d'entrée, soit  $v_{M(p,1),0}$ . Donc en additionnant  $E(x_p)$ ,

$t_{0,M(p,1)}$ ,  $v_{M(p,1),0}$  et  $\nabla$  on obtient une borne inférieure de  $H(x_n)$  et on a le théorème suivant.

**Théorème :**

$$\begin{aligned} \bar{H}^P(x_n) &= E(x_p) + t_{0,M(p,1)} + v_{M(p,1),0} \\ &+ \text{Max} \left( \begin{array}{l} \sum_{k \in U_{n-p}} \sum_{i=0}^{m_k-1} t_{M(k,i),M(k,i+1)} \\ \text{Max}_i \sum_{k \in U_{n-p}} p_{i,k} \end{array} \right) \end{aligned}$$

est une borne inférieure de  $H(x_n)$ .

Même si cette borne n'est pas de très bonne qualité, elle présente l'avantage de pouvoir être facilement calculable.

**2.2. Ordre de traitement des jobs de même type**

Dans la borne inférieure précédente, le fait qu'il existe des jobs de même gamme n'est pas exploité. L'idée que nous proposons consiste à introduire  $l(k,i)$  comme étant une borne inférieure de la date de fin au plus tôt du  $i^{\text{ème}}$  job de gamme  $k$ . Pour évaluer  $l(k,i)$ , nous nous plaçons dans le cas où le système ne traite que des jobs de gamme  $k$ . Aucune attente liée au traitement d'autres jobs n'est prise en compte (que ce soit au niveau des machines ou au niveau du chariot). Nous noterons  $M'(k,i)$  le numéro de la  $i^{\text{ème}}$  station sur laquelle un job de gamme  $k$  passe. En supposant que le premier job de gamme  $k$  entre dans le système à  $t=0$  on a :

$l(k,1) = \sum_{i=0}^{m_k-1} t_{M'(k,i),M'(k,i+1)} + \sum_i p_{i,k}$ . La date de sortie au plus tôt du job  $l(k,1)$  est égale à la somme des temps de transfert du job entre chaque station (temps de déplacement à charge du chariot) plus la somme des processing times du job sur les différentes stations. Le  $2^{\text{ième}}$  job de gamme  $k$  peut sortir au plus tôt à la date :

$$\begin{aligned} l(k,2) &= \sum_{i=0}^{m_k-1} t_{M'(k,i),M'(k,i+1)} + \sum_i p_{i,k} \\ &+ (t_{0,M'(k,1)} + v_{M'(k,1),0}) \end{aligned}$$

En fait le  $r^{\text{ième}}$  job de gamme  $k$  peut sortir au plus tôt à la date :

$$\begin{aligned} l(k,r) &= \sum_{i=0}^{m_k-1} t_{M'(k,i),M'(k,i+1)} + \sum_i p_{i,k} \\ &+ (r-1)(t_{0,M'(k,1)} + v_{M'(k,1),0}) \end{aligned}$$

et ceci pour  $r$  variant de 1 à  $n_k$ . On en déduit le théorème suivant.

**Théorème :**

$$\begin{aligned} \bar{H}^P(x_n) &= E(x_p) + t_{0,M(p,1)} + v_{M(p,1),0} + \\ &\text{Max} \left( \begin{array}{l} \sum_{k \in U_{n-p}} \sum_{i=0}^{m_k-1} t_{M(k,i),M(k,i+1)} \\ \text{Max}_i \sum_{k \in U_{n-p}} p_{i,k}, \text{Max}_k l(k, n_k) \end{array} \right) \end{aligned}$$

est une borne inférieure de  $H(x_n)$ .

Cette borne, basée sur l'ordre de traitement des jobs de même gamme, mène à une modélisation sous forme de graphe de précedence présenté dans le paragraphe suivant.

**2.3. Modélisation sous forme de graphe**

On suppose dans cette section que la matrice des temps de déplacement vérifie l'inégalité triangulaire. Soit  $G=(X,A,c)$  un graphe orienté sans circuit où l'ensemble des sommets  $X$  représente les différents événements relatifs aux dates de début et de fin des différents jobs sur les différentes machines :

$$\begin{aligned} X &= \left\{ O_{M(k,i),k}^D, k=1..n, i=1..m_k \right\} \\ &\cup \left\{ O_{M(k,i),k}^F, k=1..n, i=1..m_k \right\} \cup \{L, U\} \end{aligned}$$

$O_{M(k,i),k}^D$  : sommet représentant le début de traitement du job  $k$  au plus tôt sur la station  $M(k,i)$  (c'est-à-dire le début de traitement du job  $k$  sur la  $i^{\text{ème}}$  station qu'il visite)

$O_{M(k,i),k}^F$  : sommet représentant la fin de traitement au plus tôt du job  $k$  sur la station  $M(k,i)$  (c'est-à-dire la fin de traitement du job  $k$  sur la  $i^{\text{ème}}$  station qu'il visite)  
Nous introduisons également deux autres sommets  $L$  et  $U$  tels que :

- $L$  : entrée des différents jobs dans l'atelier (à la station de chargement)
- $U$  : sortie des différents jobs de l'atelier (à la station de déchargement)

La fonction  $c$  est une fonction de coût associant à chaque arc  $(i,j)$  un coût noté  $c(i,j)$

L'ensemble des arcs  $A$  représente les contraintes de précedence entre les différents événements :

Pour chaque job  $k$  :

$L$  précède la première opération de chaque job. Donc  $(L, O_{M(k,1),k}^D) \in A$  pour tout  $k$ . De plus, il faut  $t_{L,M(k,1)}$  unités de temps pour effectuer le déplacement à charge du chariot entre la station d'entrée  $L$  (de chargement) et la machine  $M(k,1)$  :

$$c(L, O_{M(k,1),k}^D) = t_{L,M(k,1)}$$

$U$  succède à la dernière opération de chaque job. Donc  $(O_{M(k,1),k}^F, U) \in A$ . De plus, il faut  $t_{M(k,m_k),U}$  unités de temps pour effectuer le déplacement à charge du chariot entre la dernière machine traitant le job  $k$ ,  $M(k,m_k)$ , et la station de sortie  $U$  (de déchargement) :

$$c(O_{M(k,m_k),k}^F, U) = t_{M(k,m_k),U} \cdot$$

Pour tout job  $k$  et tout  $i$  dans  $[1..m_k - 1]$ ,

$$(O_{M(k,i),k}^F, O_{M(k,i+1),k}^D) \in A \text{ et}$$

$$c(O_{M(k,i),k}^F, O_{M(k,i+1),k}^D) = t_{M(k,i),M(k,i+1)}$$

$$(O_{M(k,i),k}^D, O_{M(k,i),k}^F) \in A \text{ et}$$

$$c(O_{M(k,i),k}^D, O_{M(k,i),k}^F) = p_{M(k,i),k}$$

**Règle de dominance** : Soient  $k_1$  et  $k_2$  deux jobs de même gamme tels que  $k_1$  est introduit avant  $k_2$  dans le système, alors pour toute station  $i$ , l'opération de  $k_1$  sur  $i$  s'effectue avant celle de  $k_2$  sur  $i$ . Plus généralement, pour un ensemble de jobs tous de même type, on peut imposer un ordre d'introduction arbitraire entre ces différents jobs, définissant ainsi un ordre total entre ces jobs. Ceci permet d'enrichir le graphe  $G = (X, A, c)$  de relations de précédence. Si par exemple,  $k$  et  $k'$  sont deux jobs de même type, on peut décider arbitrairement d'imposer d'introduire  $k$  avant  $k'$  et on obtient pour tout  $i$  dans  $[1..m_k]$  :

$$(O_{M(k,i),k}^D, O_{M(k',i),k'}^D) \in A \text{ et}$$

$$c(O_{M(k,i),k}^D, O_{M(k',i),k'}^D) = v_{M(k,i),M(k',i-1)} + t_{M(k',i-1),M(k',i)}$$

$$(O_{M(k,i),k}^F, O_{M(k',i),k'}^D) \in A \text{ et}$$

$$c(O_{M(k,i),k}^F, O_{M(k',i),k'}^D) = 0$$

Le chemin le plus long dans le graphe  $G = (X, A, c)$  représente alors une borne inférieure.

#### 2.4. Borne liée à la station goulet

La proposition faite dans ce paragraphe est issue de (Ulusoy *et al.*, 1997). L'idée est de s'intéresser de manière spécifique au cas où un grand nombre de jobs suivent la même gamme et de proposer une borne inférieure exploitant cette caractéristique. Nous introduisons les notations suivantes :

$R(k, i)$  : borne inférieure du temps nécessaire pour le traitement du job  $k$  avant d'être présent sur la station numéro  $i$ . Soit  $S(i)$  la date d'arrivée du premier job devant être traité par la station  $i$ . Le job  $k$  ne peut être présent sur la première station au plus tôt qu'à la date :  $R(k, M(k,1)) = t_{0,M(k,1)}$ . De même le job  $k$  sera sur la  $i^{\text{ème}}$  station visitée au plus tôt à la date :

$$R(k, M(k, i)) = R(k, M(k, i-1)) + p_{M(k,i-1),k} + t_{M(k,i-1),M(k,i)}$$

avec  $i > 1$ , c'est-à-dire que le job  $k$  arrive au plus tôt sur la  $i^{\text{ème}}$  station à la date  $R(k, M(k, i))$  qui est égale à  $R(k, M(k, i-1))$  date d'arrivée du job  $k$  sur la  $(i-1)^{\text{ème}}$  station + le processing time  $p_{M(k,i-1),k}$  du job  $k$  sur la  $(i-1)^{\text{ème}}$  station + le temps de transport du job  $k$  de la  $(i-1)^{\text{ème}}$  station à la  $i^{\text{ème}}$  station. La date d'arrivée  $S(i)$  du premier job devant être traité par la station  $i$  est :  $S(i) = \min_k R(k, i)$ . A la date d'entrée du premier job sur

la station  $i$  on ajoute la somme des processing times des jobs devant passer sur la station  $i$ . Donc  $S(i) + \sum_k p_{ik}$

représente une borne inférieure de la date de fin de traitement des jobs sur la station  $i$ . Donc, la date de fin

de traitement sur la station goulet est  $\max_i \left( S(i) + \sum_k p_{ik} \right)$

et on obtient le résultat suivant.

$$\text{Théorème : } \bar{H}^P(x_n) = E(x_p) + \max_{i \in M} \left( S(i) + \sum_k p_{i,k} \right),$$

avec  $M$  ensemble des machines utilisées, est une borne inférieure de  $H(x_n)$ .

#### 2.5. Proposition d'une borne inférieure prenant en compte le nombre de jobs limité dans le système

Il est intéressant de remarquer que les bornes inférieures précédentes ne permettent pas de prendre en compte le fait que le nombre de jobs simultanément autorisés dans le système est limité. Or ceci est un élément déterminant pour la performance du système puisqu'il a une influence sur le nombre d'opérations effectivement traitées dans le système. Sans perte de généralité, les jobs appartenant à  $U_{n-p}$  sont re-numérotés  $\pi_1, \pi_2 \dots$

selon l'ordre croissant de :

$$\sum_{j=1}^{m_{\pi_k}-1} p_{\pi_k j} + \sum_{i=0}^{m_{\pi_k}-1} t_{M(\pi_k, i)M(\pi_k, i+1)} \cdot$$

Le premier job ordonnancé dans le système noté  $k_1$  termine au plus tôt à l'instant :

$$\sum_{j=1}^{m_{k_1}-1} p_{k_1 j} + \sum_{i=0}^{m_{k_1}-1} t_{M(k_1,i)M(k_1,i+1)} \cdot$$

Le deuxième job ordonnancé dans le système noté  $k_2$  termine au plus tôt à l'instant :

$$\sum_{j=1}^{m_{k_2}-1} p_{k_2 j} + \sum_{i=0}^{m_{k_2}-1} t_{M(k_2,i)M(k_2,i+1)} \cdot$$

Le  $N^{\text{ième}}$  job ordonnancé dans le système noté  $k_N$  termine au plus tôt à l'instant :

$$\sum_{j=1}^{m_{k_N}-1} p_{k_N j} + \sum_{i=0}^{m_{k_N}-1} t_{M(k_N,i)M(k_N,i+1)} \cdot$$

Le  $(N+1)^{\text{ième}}$  job ordonnancé dans le système noté  $k_{N+1}$  pour commencer doit attendre que le premier job ordonnancé sorte du système, il termine donc au plus tôt à l'instant :

$$\begin{aligned} & \sum_{j=1}^{m_{k_1}-1} p_{k_1 j} + \sum_{i=0}^{m_{k_1}-1} t_{M(k_1,i)M(k_1,i+1)} \\ & + \sum_{j=1}^{m_{k_{N+1}}-1} p_{k_{N+1} j} + \sum_{i=0}^{m_{k_{N+1}}-1} t_{M(k_{N+1},i)M(k_{N+1},i+1)} \end{aligned} \cdot$$

Le  $(N+2)^{\text{ième}}$  job ordonnancé dans le système noté  $k_{N+2}$  pour commencer doit attendre que le deuxième job ordonnancé sorte du système, il termine donc au plus tôt à l'instant :

$$\begin{aligned} & \sum_{j=1}^{m_{k_2}-1} p_{k_2 j} + \sum_{i=0}^{m_{k_2}-1} t_{M(k_2,i)M(k_2,i+1)} + \\ & \dots \\ & \sum_{j=1}^{m_{k_{N+2}}-1} p_{k_{N+2} j} + \sum_{i=0}^{m_{k_{N+2}}-1} t_{M(k_{N+2},i)M(k_{N+2},i+1)} \end{aligned}$$

Le  $(2N)^{\text{ième}}$  job ordonnancé dans le système noté  $k_{2N}$  pour commencer doit attendre que le  $N^{\text{ième}}$  job ordonnancé sorte du système, il termine donc au plus tôt à l'instant :

$$\begin{aligned} & \sum_{j=1}^{m_{k_N}-1} p_{k_N j} + \sum_{i=0}^{m_{k_N}-1} t_{M(k_N,i)M(k_N,i+1)} \\ & + \sum_{j=1}^{m_{k_{2N}}-1} p_{k_{2N} j} + \sum_{i=0}^{m_{k_{2N}}-1} t_{M(k_{2N},i)M(k_{2N},i+1)} \end{aligned}$$

Le  $(2N+1)^{\text{ième}}$  job ordonnancé dans le système noté  $k_{2N+1}$  pour commencer doit attendre que le  $(N+1)^{\text{ième}}$

job ordonnancé sorte du système, il termine donc au plus tôt à l'instant :

$$\begin{aligned} & \sum_{j=1}^{m_{k_1}-1} p_{k_1 j} + \sum_{i=0}^{m_{k_1}-1} t_{M(k_1,i)M(k_1,i+1)} \\ & + \sum_{j=1}^{m_{k_{N+1}}-1} p_{k_{N+1} j} \\ & + \sum_{i=0}^{m_{k_{N+1}}-1} t_{M(k_{N+1},i)M(k_{N+1},i+1)} \\ & + \sum_{j=1}^{m_{k_{2N+1}}-1} p_{k_{2N+1} j} \\ & + \sum_{i=0}^{m_{k_{2N+1}}-1} t_{M(k_{2N+1},i)M(k_{2N+1},i+1)} \end{aligned}$$

En numérotant  $\pi_1, \pi_2 \dots$  les jobs appartenant à  $U_{n-p}$  selon l'ordre croissant de

$$\sum_{j=1}^{m_{\pi_k}-1} p_{\pi_k j} + \sum_{i=0}^{m_{\pi_k}-1} t_{M(\pi_k,i)M(\pi_k,i+1)}$$

on peut énoncer le théorème suivant.

**Théorème :**

$$\begin{aligned} & \bar{H}^p(x_n) = E(x_p) \\ & + \sum_{k=\pi_1}^{\pi_{\max}} \left( \sum_{j=1}^{m_k-1} p_{k j} + \sum_{i=0}^{m_k-1} t_{M(k,i)M(k,i+1)} \right) \text{ avec} \\ & \pi_{\max} = \left\lfloor \frac{n-p+1}{N} \right\rfloor \text{ est une borne inférieure} \end{aligned}$$

### 3. BORNES INFÉRIEURES APPROCHÉES

Nous proposons dans ce paragraphe des éléments permettant la mise au point de bornes inférieures approchées c'est-à-dire de bornes dont on ne peut pas démontrer qu'elles soient des bornes inférieures mais qui en pratique permettent d'obtenir de très bons résultats avec le Branch and Bound.

Nous avons identifié trois voies principales :

Ils s'agit de travaux proposant une estimation des déplacements à vide et une estimation des déplacements à charge (avec les temps de traitement des jobs) en prenant en compte le nombre de jobs simultanément autorisés dans le système. Ensuite, nous présentons une borne inférieure approchée qui a été testée sur des jeux d'essais représentatifs.

#### 3.1. Evaluation probabiliste des déplacements à vide

Les travaux de (Egbelu, 1997) sont représentatifs du type d'évaluation des déplacements à vide du chariot que l'on peut faire sur ce type de système. Ces déplacements sont

particulièrement difficiles à prendre en compte car ils ont lieu la plupart du temps en temps masqué. L'évaluation proposée est une évaluation probabiliste des déplacements. Pour présenter cette évaluation des déplacements à vide nous allons introduire les notations suivantes :

$f_{ij}$  : flux de jobs entre la station  $i$  et la station  $j$  c'est-à-dire le nombre de jobs parmi ceux restant à ordonnancer qui dans leur gamme possèdent la séquence : station  $i \rightarrow$  station  $j$ . Cela représente le nombre de fois qu'un trajet à charge va être effectué par le chariot entre la station  $i$  et la station  $j$ .

$P_w(j)$  : probabilité qu'un chariot se trouve devant la station  $j$

$P_c(i)$  : probabilité qu'un job appelle un chariot sur la station  $i$

$P_p(j,i)$  : probabilité que le parcours à vide de  $j$  vers  $i$  soit effectué

$d_{ji}$  : distance pour aller de la station  $j$  à la station  $i$

$d(j,i)$  : distance totale parcourue par le chariot à vide entre la station  $j$  et la station  $i$  (le chariot peut passer plusieurs fois).

$D_{vide}$  : évaluation des déplacements à vide

Si un chariot se trouve sur la station  $j$ , cela signifie qu'il a effectué un parcours à charge d'une autre station vers cette station  $j$ . Ainsi on trouve la somme des flux entrant dans la station  $j$  et qui vaut  $\sum_{i=0}^m f_{ij}$  et la somme des flux circulant dans le système qui est  $\sum_{i=0}^m \sum_{j=1}^{m+1} f_{ij}$ . On en déduit :

$$P_w(j) = \frac{\sum_{i=0}^m f_{ij}}{\sum_{i=0}^m \sum_{j=1}^{m+1} f_{ij}} .$$

On peut calculer ensuite la probabilité qu'un job appelle le chariot sur la station  $i$ . Cette probabilité dépend de la somme des flux sortant de la station  $i$  ( $\sum_{j=1}^{m+1} f_{ij}$ ) et de la somme totale des flux circulant dans le système. On obtient alors :

$$P_c(i) = \frac{\sum_{j=1}^{m+1} f_{ij}}{\sum_{i=0}^m \sum_{j=1}^{m+1} f_{ij}} .$$

Remarquons que  $P_c(i) = P_w(i)$  car ce qui entre au niveau d'une station est égal à ce qui en sort. Donc la probabilité que le parcours à vide de  $j$  vers  $i$  (avec  $i \neq j$ ) soit effectué est:

$$P_p(j,i) = P_w(j) \times P_c(i)$$

en supposant que les événements soient indépendants. La proportion de flux effectués à vide de la station  $j$  vers la station  $i$  est :  $P_p(j,i) \times \sum_k \sum_l f_{lk}$ . La distance totale parcourue à vide entre la station  $j$  et la station  $i$ ,  $d(j,i)$ , est telle que :

$$d(j,i) = P_p(j,i) \times d_{ji} \times \sum_k \sum_l f_{lk}$$

La distance totale parcourue à vide est :

$$D_{vide} = \sum_i \sum_j \left[ P_p(j,i) \times d_{ji} \times \sum_k \sum_l f_{lk} \right] .$$

On obtient la proposition suivante.

**Proposition :**  $D_{vide} = \sum_i \sum_j \left[ P_p(j,i) \times d_{ji} \times \sum_k \sum_l f_{lk} \right]$  est une évaluation des déplacements à vide.

### 3.2. Borne inférieure approchée

Récemment (Lacomme *et al.*, 2000b) ont proposé une borne inférieure approchée qui fait intervenir le nombre de jobs simultanément autorisés dans le système. L'idée de base est similaire à celle des autres bornes mais elle fait intervenir les relations entre  $H(x_p)$  et  $E(x_p)$ .

L'approche proposée a été testée avec les politiques FIFO, STT, SPT et MOQS pour la gestion du chariot et les différents job-sets et topologies proposés dans (Ulusoy et Bilge, 1993). Cette borne inférieure permet de résoudre des problèmes de grandes tailles. Elle a été comparée avec la solution optimale obtenue en utilisant un Branch and Bound et une borne exacte sur une batterie de tests comprenant 720 exemples. Les tests réalisés montrent que dans la majorité des cas, cette borne permet de trouver une solution optimale. Le tableau 1 montre la qualité des résultats obtenus.

De/Vers	E/S	M1	M2	M3	M4
E/S	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

Tableau 2. Temps de transport entre les machines [min]

Nombre total de solutions non optimales	Pourcentage de solutions optimales	Nombre de solutions comparées	Nombre total de solutions non optimales >1%	Pourcentage par rapport aux solutions comparées
199	72%	720	115	15,97%

Tableau 1. Qualité des résultats obtenus lors de l'utilisation de la borne inférieure approchée

#### 4. EVALUATION DES BORNES PROPOSEES

Dans ce paragraphe nous proposons une étude de la qualité des bornes proposées. Pour l'évaluation de ces bornes nous avons repris les exemples proposés par Ulusoy et Bilge (Ulusoy et Bilge, 1993). Chacune des topologies proposées se compose d'une station d'entrée/sortie et de quatre stations de traitement des pièces. On suppose que la station d'entrée/sortie a une capacité suffisante pour que l'ensemble des pièces à traiter et celles déjà traitées puissent être stockées. Les gammes utilisées et les temps de déplacement du chariot sont ceux proposés dans (Ulusoy et Bilge, 1993). A titre d'exemple, on considère la topologie proposée sur la figure 2.

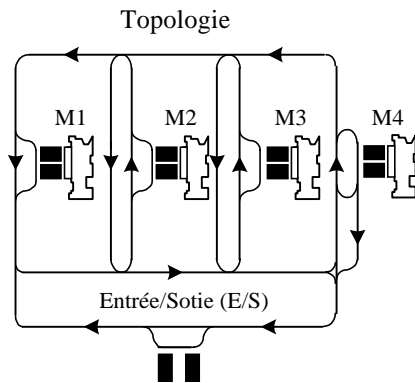


Figure 2. Topologie du système étudié

Les temps de transport entre les machines, les différents types de pièces, les gammes et les durées opératoires sont donnés respectivement par les tableaux 2 et 3.

Gamme	G1	G2	G3	G4	G5
Opération	1-2-3	1-2-3	1-2-3	1-2	1-2
Machine	1-2-4	1-3-2	3-4-1	4-2	3-1
Temps usinage	8-16-12	20-10-18	12-8-15	14-18	10-15

Tableau 3. Gammes et durées opératoires [min]

Les tableaux 4 et 5 montrent les résultats comparatifs. Les bornes (LB<sub>i</sub>) sont obtenues par les méthodes présentées précédemment et prennent en compte :  
 LB1 : les déplacements à charge et le temps d'occupation des machines (paragraphe 2.1.) ;  
 LB2 l'ordre de traitement des jobs de même type (paragraphe 2.2) ;  
 LB3 la station goulet (paragraphe 2.4) ;  
 LB4 le nombre de jobs limité dans le système (paragraphe 2.5).

LB<sub>i</sub>+TV borne exacte LB<sub>i</sub> plus le temps de déplacement à vide (TV) calculé avec l'évaluation probabiliste présentée dans le paragraphe 3.1.

Les résultats ont été obtenus respectivement pour deux (tableau 4) et quatre jobs (tableau 5) simultanément autorisés dans les systèmes. Les valeurs de  $E(x_p)$ , date d'entrée du  $p^{ième}$  job dans le système, ont été évaluées par la simulation pour  $p=5$ .

Les deux tableaux donnent les makespan obtenus (en minutes) selon les différentes méthodes. La comparaison des makespan obtenus par les bornes exactes avec les résultats de la simulation montre qu'aucune des bornes inférieures ne peut être considérée comme étant meilleure. Cependant, l'utilisation de l'évaluation probabiliste des déplacements à vides donne des résultats prometteurs. Toutefois cette utilisation ne permet pas de garantir l'optimalité.

Type de jobs et ordre d'entrée dans le système	LB1	LB2	LB3	LB4	Simulation (FIFO)	Transport à vide (TV)	LB1+TV	LB2+TV	LB3+TV	LB4+TV
G1, G1, G1, G1, G1, G1, G1, G1, G1, G1	351	355	307	299	526	128	479	483	435	427
G4, G4, G4, G4, G4, G4, G4, G4, G4, G4	347	347	303	365	572	63	410	410	366	428
G1, G2, G3, G4, G5, G1, G2, G3, G4, G5	397	397	297	438	586	123	520	520	420	561
G1, G2, G3, G1, G2, G1, G2, G3, G1, G2	419	419	339	464	616	142	561	561	481	606
G4, G5, G4, G5, G4, G4, G5, G4, G5, G4	363	363	283	374	510	90	453	453	373	464

Tableau 4. Résultats pour 2 jobs autorisés simultanément dans le système, politique de gestion du chariot pour traiter les demandes de transport : FIFO (First In First Out)

Type de jobs et ordre d'entrée dans le système	LB1	LB2	LB3	LB4	Simulation (FIFO)	Temps de transport à vide (TV)	LB1+TV	LB2+TV	LB3+TV	LB4+TV
G1, G1, G1, G1, G1, G1, G1, G1, G1, G1	347	351	303	263	544	128	475	479	431	391
G4, G4, G4, G4, G4, G4, G4, G4, G4, G4	327	327	283	283	692	63	390	390	346	346
G1, G2, G3, G4, G5, G1, G2, G3, G4, G5	369	369	269	337	569	123	492	492	392	460
G1, G2, G3, G1, G2, G1, G2, G3, G1, G2	363	363	283	335	584	142	505	505	425	477
G4, G5, G4, G5, G4, G4, G5, G4, G5, G4	335	335	255	284	530	90	425	425	345	374

Tableau 5. Résultats pour 4 jobs autorisés simultanément dans le système, politique de gestion du chariot pour traiter les demandes de transport : FIFO (First In First Out)

## 5. CONCLUSION

Nous nous sommes intéressés à la résolution conjointe du problème d'ordonnancement des jobs en entrée du système et du problème de gestion de la ressource de transport. Nous avons proposé un algorithme de Branch and Bound couplé avec un modèle de simulation à événements discrets pour prendre en compte le maximum de contraintes de fonctionnement. Dans ce type d'approche, il est nécessaire de disposer de bornes inférieures performantes pour élaguer au maximum l'arbre de recherche. Dans la littérature peu de travaux abordent ce problème dans le cadre des systèmes flexibles de production avec un chariot filoguidé. Notre objectif à terme est de proposer des bornes inférieures efficaces pour les systèmes à un chariot et de nous intéresser ensuite aux systèmes flexibles de production composés de plusieurs chariots.

## REFERENCES

- Blazewicz, J., H.A. Eiselt, G. Finke, G. Laporte and J. Weglarz, 1991, Scheduling Tasks and Vehicles in a Flexible Manufacturing System, *International Journal of Flexible Manufacturing System*, 4 (1), p. 5-16.
- Blazewicz, J., R.E. Burkard, G. Finke and G.J. Woeginger, 1994, Vehicle Scheduling in Two-Cycle Flexible Manufacturing Systems, *Mathematical Computer Modelling*, 20(2), p. 19-31.
- Egbelu P.J., 1997, The use of Non Simulation Approaches in Estimating Vehicle Requirements in an Automatic Guided Vehicle Based Transport System, *Material Flow*, 4(4), p. 17-32.
- Lacomme P., A. Moukrim and N. Tchernev, 2000a, Résolution conjointe de problèmes d'ordonnancement et de la gestion du chariot filoguidé : couplage Branch and Bound - simulation à événements discrets. *Congrès ROADEF*, Ecole des Mines de Nantes, France, p. 128-129.
- Lacomme P., A. Moukrim and N. Tchernev, 2000b, Résolution exacte et approchée du problème de planification dans les FMS avec chariot automatiquement guidé : Branch and Bound, simulation, nouvelles bornes inférieures, *Groupe*

*Bermudes*, Bulletin de liaison n°17, HeuDiaSyc / Université de Technologie de Compiègne, France.

Ulusoy G. and Ü. Bilge, 1993, Simultaneous Scheduling of Machines and Material Handling System in an FMS. *International Journal of Production Research*, 31, p. 2857-2873.

Ulusoy G., F. Sivrikaya-Serifoglu, and Ü. Bilge, 1997, A genetic algorithm approach to the simultaneous scheduling of stations and automated guided vehicles, *Computers Ops Res.*, 24(4), p. 335-351.