A NEW LOWER BOUND FOR SCHEDULING OF FMS BASED ON AGV MATERIAL HANDLING

P. Lacomme*, A. Moukrim**, 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 85, E-mail: lacomme@isima.fr

** Université de Technologie de Compiègne, HeuDiaSyC, UMR CNRS 6599, BP 20529 60205 Compiègne cedex (France) Tel.: +33 3 44 23 49 52, Fax: +33 3 44 23 44 77, E-mail: Aziz.Moukrim@hds.utc.fr

*** IUP de Management et Gestion des entreprise, Université d’Auvergne, Laboratoire d’Informatique, de Modélisation et d’Optimisation des Systèmes (LIMOS), Campus Universitaire des Cézeaux, 63177 Aubière Cedex France, Tel.: +33 4 73 40 77 71, E-mail: tchernev@sp.isima.fr

Abstract: This paper addresses the simultaneous job input sequencing and vehicle dispatching problems in flexible manufacturing systems (FMS) using a single device Automated Guided Vehicle System (AGVS). A Branch and Bound approach based on simulation has been previously developed. However, efficient easily computable lower bounds are still required to provide solutions for small and medium size problems. In this paper a new lower bound is proposed and benchmarks proved that this lower bound outperforms the lower bound previously published. Copyright © 2002 IFAC

Keywords: Flexible manufacturing systems, Automated guided vehicle systems, scheduling algorithms, Branch and Bound, Lower Bound.

1. INTRODUCTION

Many Flexible Manufacturing Systems (FMS) employ Automated Guided Vehicle Systems (AGVS) as material handling system. The AGVS effectiveness depends on several factors, among them a well designed vehicle management which functions are: (i) dispatching, which is the process of selecting and assigning tasks to material handling devices; (ii) routing, which is the process of selecting specific paths taken by material handling devices; (iii) scheduling, which is the process of determining of the arrival and departure times of material handling devices. Scheduling encompasses the dispatching and routing issues with the introduction of time in order to reduce the impact of blocking and congestion in meeting a material handling workload. The vehicle management has a significant effect on the travel time, the number of devices used, the AGVS response time, the operating expenses and the initial investment costs.

One of the most difficult operational problems in FMS is the proper coordination of the production sequencing and the allocation in the time of required resources. This paper deals with the simultaneous job input sequencing and vehicle dispatching in FMS using a single vehicle AGVS. The objective is the makespan minimization. However, to obtain really profitable solutions the following constraints must be taken into account: (i) the limited input/output buffer capacity; (ii) the limitation on the number of jobs simultaneously allowed in the shop; (iii) the dynamic behaviour of the system under study and thus the impact of the vehicle blocking and congestion as well as the impact of the machine blocking.

This problem is previously studied by Gourgand et al., (1999) who propose a two stages iterative approach for approximately solving the joint job input sequencing and vehicle dispatching. Their approach considers the constraints defined above.
Large size problems with up to 50 jobs to sequence are solved by an iterative approach.

Lacomme et al., (2000) propose a branch and bound approach coupled with a discrete events simulation model. The discrete events simulation model provides evaluation of the makespan taking into account all the managing constraints of the system. Very efficient lower bound are required to ensure performances to branch and bound approaches. Unfortunately, previous lower bounds dedicated to this problem are not enough efficient and the lower bound used in others scheduling problems required many investigations to take into account the problem constraints.

In the following sections, the problem is described and the Branch and Bound approach proposed by Lacomme et al. (2000) and Espinouse et al. (2001) is presented. A new efficient Lower Bound is presented in section 4. The section 5 provides a benchmark test. Lastly, the results of the study are summarised and some suggestions for future researches are identified.

2. PROBLEM FORMULATION

The simultaneous job input sequencing and vehicle dispatching in FMS with finite buffers capacity and a single vehicle AGVS can be stated as follows: given a FMS and dispatching rules, determine the job input sequence which minimizes the makespan. Since in FMS the number of jobs simultaneously allowed into the system is limited and the sum of processing times is constant, the only way to minimize the makespan is to reduce the waiting times due to the blocking and non availability of resources needed. Therefore the objective is to find an order in which the jobs enter in the manufacturing system reducing the waiting times due to the blocking and non availability of resources needed as well as minimizes the deadheading time.

The issue of joint optimization of job and AGV schedules is a complex problem which has been formulated as a non-linear integer program (Gourgand et al., 1998). Due to its intractability the literature contains numerous heuristic approaches. An efficient lower bound is required to ensure performances to branch and bound approaches. Unfortunately, previous lower bounds dedicated to this problem are not enough efficient and the lower bound used in others scheduling problems required many investigations to take into account the problem constraints.

In the next section we present the simulation based on Branch and Bound approach taking into account the following constraints:

- the limited input/output buffer capacity;
- the limitations on the number of jobs simultaneously allowed in the system;
- the dynamic behaviour of the system under study and thus the impact of the vehicle blocking and congestion as well as the impact of the machine blocking.

3. THE BRANCH AND BOUND APPROACH

Let us define the following notations which will be used to describe the Branch and Bound:

- \( n \) number of jobs to sequence
- \( m \) the number of different job types
- \( N \) the number of jobs simultaneously allowed in the FMS
- \( M_0 \) the input station of the system
- \( M_S \) the output station of the system
- \( m_s \) the number of machine in the system
- \( s \) the number of machine in the system except the input/output station
- \( i \) job type index \( i \in I = \{1, 2, \ldots, m\} \)
- \( n_i \) number of jobs of type \( i \) such that
  \[ \sum_{i=1}^{m} n_i = n \]
- \( x_p \) partial jobs input sequence in which jobs in the first \( p \) positions have been fixed
- \( x_n^* \) optimal job input sequence
- \( U_{n-p} \) the set of the unscheduled jobs associated to the partial job input sequence \( x_p \)

The problem is to compute \( x_n^* \) for which

\[ \forall x_n \ H(x_n^*) \leq H(x_n) \]  

The feasible set of solutions of the job input sequencing problem from a combinatorial point of view is given by \( X = \{\text{set of all } n \text{-jobs schedules}\} \). Although complete enumeration would permit to obtain the optimal solution, this approach is unpractical due to computational time problems. For example for 20 jobs to schedule (5 types of jobs and 4 jobs of each type) there is \( (20!/4^44^44^4!)/4!^4 \) different jobs input sequences of 20 jobs. Dominated solutions (which can be excluded from consideration) permit to reduce the search effort.

General framework

The studied problem can be decomposed into two subproblems and solved by an iterative search procedure that tries to accommodate the combinatorial nature in finding the solution. Given a solution for the job input sequencing subproblem by the branch and bound algorithm, it remains to find a vehicle schedule based on a given vehicle dispatching rule and to evaluate the system performance and the makespan using a dedicated discrete events simulation model. In other words at each node of the search tree the current solution is evaluated by a dedicated discrete events simulation model (figure 1). Iterations are carried out in order to improve the initial job input sequence and the vehicle
schedule and to find a better solution. After a given number of iterations the procedure will terminate with the optimal solution, if a feasible solution exists.

Fig.1. Template of the general framework

At the top of the search tree, \( v_0 \), the initial solution \( H(x_0) \) is evaluated using the dedicated discrete events simulation model. In the job sequence \( x_0 \), the jobs are ordered in a cyclic manner. The upper bound, noted \( UB_0 \), for this first iteration is set to \( H(x_0) \). At each iteration \( t \) a node \( v_t \) is chosen, according to a depth-first search, in the search tree. The lower bound for node \( v_t \), noted \( LB_t \), is calculated. Whenever \( LB_t \) is greater than the current upper bound \( UB \), the branch is pruned. At the bottom of the search tree, the upper bound is updated, if the current upper bound \( UB \) is greater than \( LB_t \), a backtrack is performed.

3. PREVIOUS LOWER BOUNDS

To provide more details of the procedure the following additional definitions are introduced. Assume a specific node is reached in the search process where:

\[ H^p(x_n) \] the lower bound of \( x_n \) with only first \( p \) positions which have been fixed. It means that only \( H(x_p) \) of the \( x_p \) partial jobs input sequence has been evaluated by simulation.

\( E(x_p) \) the \( p^{th} \) job entry date in the system.

\( E(x_p) \) is computed by simulation

\( DS(x_p) \) the \( p^{th} \) job end date in the system.

Represent the output station reached date

\( t_{ij} \) the loaded vehicle travel time from machine \( i \) to machine \( j \) including loading and unloading time

\( v_{ij} \) the unloaded vehicle travel time from machine \( i \) to machine \( j \)

\( \alpha \) the evaluation of \( E(x_{p+1}) \)

\( n \) the number of machines that job \( k \) has to visit

\( M(k,i) \) the \( i^{th} \) machine that has to be visited by job \( k \) where \( M(k,0) \) is the system input station and \( M(k,m_k) \) is the system output station

The challenging problem consists in computing a powerful estimation of \( H(x_p) \). This estimation is noted \( H^p(x_n) \) (see above) because only \( p \) jobs have been already sequenced (figure 2).

Fig.2. The Lower Bound with regard to \( H(x_p) \)

A detailed presentation of some lower bounds are proposed in (Espinouse et al., 2001). The authors report:

- a basic lower bound;
- a lower bound dedicated to problem with an important number of jobs based on the same types;
- a lower bound taking into account the limited number of jobs simultaneously allowed in the system;
- a probabilistic estimation of the vehicle unloaded transportation time.

The benchmark of (Espinouse et al., 2001) proves that no lower bound is better than other ones. The basic lower bound presented below provides solutions with lowest computational time requirements see (Espinouse et al., 2001):

\[
H^p(x_n) = E(x_p) + \max_i \sum_{k \in M(n-p)} l_{M(k,i)M(k,i+1)} + \sum_{k \in M(n-p)} p_{i,k} + \alpha_p
\]

is a lower bound of \( H(x_n) \) denoted \( LB_1 \).

4. NEW LOWER BOUND

A lower bound is defined as follows:

\[
H^p(x_n) = \alpha_p + \nabla , \text{where:}
\]

\( \alpha_p \) the evaluation of \( E(x_{p+1}) \)
the evaluation of the time required to treat unscheduled jobs of $U_{n-p}$

4.1 Entry date of the $(p+1)^{th}$ job

The entry date of the $(p+1)^{th}$ job highly depends on the number of jobs simultaneously allowed in the system and indeed of the $p$ value. Two cases should be distinguished:

**First case:** $p < N$

As long as possible and as long as the number of jobs in the system is inferior than $N$ the managing rule of the system consists in serviced first the entry station. So the vehicle serviced the entry station as long as possible when $p < N$.

The entry date of the $(p+1)^{th}$ job is $E(x_p)$ added to the loaded vehicle travel time required to go from the entry station to the first station used by the $p^{th}$ job and the unloaded travel time to go from the first station of the $p^{th}$ job to the entry station. The entry date of the $(p+1)^{th}$ job is delayed (figure 3) due to the transportation times of $p^{th}$ job.

\[ E(x_{p+1}) = E(x_p) + t_{M(p,0),M(p,1)} + v_{M(1),M(0)} \]

**Second case:** $p = N$

Let us consider first that the ordered set of jobs at the system entrance is equal to the ordered set of jobs at the output machine.

The entry date of $(p+1)^{th}$ job is grater than or equal to the end date of the $(p+1-N)^{th}$ job. So $E(x_{p+1}) \geq DS(x_{p+1-N})$. Figure 4 provides an example for $N = 4, p = 5$.

\[ |S_p| = \begin{cases} \min \{ E(x_p) + t_{M(p,0),M(p,1)} + v_{M(1),M(0)} \} & \text{if } |S_p| < N \\ \max \{ E(x_p) + t_{M(p,0),M(p,1)} + v_{M(1),M(0)} \} & \text{otherwise} \end{cases} \]

4.2 Estimation of the time required to treat the remaining jobs of $U_{n-p}$

The total loaded time required to transport one job between the entry machine and the output station

More complex situations can occur because each job has one type defining a list of machines with a processing time associated to the job and the machine. So the ordered set of output jobs could be different than the input ordered set of job at the system entrance.

Fig. 5. Example for $N = 4, p = 9$

So the entry date of the $(p+1)^{th}$ is greater or equal to the lowest end date of the jobs which are in the system. So $DE(x_{p+1}) \geq \min_x DS(x)$.

Obviously, after the job $p$ the unloaded and loaded vehicle travel time must be taken into account. Hence:

\[ DE(x_{p+1}) \geq \max \{ E(x_p) + t_{M(p,0),M(p,1)} + v_{M(1),M(0)} \} \]

Figure 5 provides an example for $N = 4, p = 9$. The entry date of the job number 10 is the end date of the job number 4.

Therefore, depending on the values of $N$ and $p$, one can use one of two expressions presented above to the entry date of the job number $(p+1)$. A more concise expression for $\alpha_p$ is available taking into account the number of jobs in the system at the date $E(x_p)$. Let us note: $S_p = \frac{1}{2} DS(x_4) > DE(x_3)$.

\[ |S_p| = \begin{cases} \min_{x \in S_p} DS(x) & \text{if } |S_p| < N \\ \max_{x \in S_p} DS(x) & \text{otherwise} \end{cases} \]
is: \( \sum_{i=0}^{m-1} t_{M(k,i+1)} \). The total processing time required to treat the job is: \( \sum_{i=0}^{m_k} p_{i,k} \). So the minimal time required to treat the job is: \( D_k = \sum_{i=0}^{m} p_{i,k} + \sum_{i=0}^{m-1} t_{M(k,i+1)} \). The following characterization of the makespan can be given. The time required to treat all the jobs of \( U_{n-p} \) can not be less than or equal to \( \max_{i \in U_{n-p}} D_i \). The makespan is superior or equal to the total amount of time required to treat all the jobs of \( U_{n-p} \) (including loaded transportation and processing time) divided by the maximal number of tasks executable in parallel. The maximal number of parallel tasks depends of \( m_s \) and \( N \). But the transportation time have been included in the \( D_k \) expression. So the vehicle can be considered as a machine providing also operations in parallel. If \( N > m_s + 1 \) then the number of machines (increased of one) limits the number of executable jobs in parallel, \( N \) limits the number of jobs executable in parallel. Therefore the makespan is greater than or equal to \( \sum_{i \in U_{n-p}} D_i \) and

\[
\nu = \max \left( \frac{\sum_{i \in U_{n-p}} D_i}{\min(N, m_s + 1)} , \max_{i \in U_{n-p}} D_i \right)
\]

is an estimation of the time required to treat the jobs of \( U_{n-p} \). It follows that:

**Lower Bound Theorem:**

\[
\overline{H}(x_n) = \alpha_p + \nu
\]

where

\[
\alpha_p = \begin{cases} E(x_p) + t_{M(p,0),M(p,0)} + v_{M(p,1),M(p,0)} & \text{if } |S_p| < N \\ \min_{i \in S_p} DS(x_i) & \text{otherwise} \end{cases}
\]

and

\[
\nu = \max \left( \frac{\sum_{i \in U_{n-p}} D_i}{\min(N, m_s + 1)} , \max_{i \in U_{n-p}} D_i \right)
\]

is a lower bound of \( H(x_n) \). This lower bound is noted \( LB_2 \).

5. **COMPUTATIONAL EXPERIMENTS**

### 5.1 Network and data

For the procedure evaluation, four different layouts and five job sets taken from (Ulusoy and Bilge, 1993) have been studied.

For each layout four machines and one Input/Output station are used. Each machine has an input and output buffers with a limited capacity. In order to limit the magnitude of this study it is assumed that there are an equal number of places at each buffer. The buffer capacity is limited to two places. The Input/Output machine is assumed to have a sufficient capacity to store all jobs to be scheduled. The other characteristics of these systems are available in (Espinouse et al., 2001).

### 5.2 Benchmark

Only results with the FIFO rule, Job-Set 1 and Topology 1 are details hereafter. The number of jobs simultaneously allowed in the system is 4, 6 and 8 jobs. The results are provided for 8 and 10 jobs to schedule at the system entrance. To evaluate the performances of the lower bounds, the following criteria have been taken into account:

- the total number of nodes generated;
- the number of nodes pruned;
- the level in the branch and bound process in which the nodes have been pruned;
- the computational times required for optimal solution computation.

All experiments have been performed on a Pentium III 600 MHz personal computer under Windows 95 Operating System using Delphi 5.0 programming environment.

<table>
<thead>
<tr>
<th>n</th>
<th>N</th>
<th>Optimal solution</th>
<th>Generated</th>
<th>Pruned</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>505</td>
<td>92 781</td>
<td>62 277</td>
<td>0'53</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>496</td>
<td>105 218</td>
<td>67 742</td>
<td>1'58</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>514</td>
<td>112 968</td>
<td>71 430</td>
<td>3'54</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>558</td>
<td>138 983</td>
<td>88 225</td>
<td>4'03</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>614</td>
<td>1 191 183</td>
<td>793 938</td>
<td>13'00</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>566</td>
<td>715 239</td>
<td>483 179</td>
<td>8'13</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>598</td>
<td>1 211 421</td>
<td>774 174</td>
<td>13'06</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>634</td>
<td>1 398 407</td>
<td>875 062</td>
<td>14'08</td>
</tr>
</tbody>
</table>

Table 1 and 2 prove the efficiency of \( LB_2 \) which permit to compute optimal solution in lowest computation time than \( LB_1 \). The total number of generated nodes is less important using \( LB_2 \). Table 3 provides a comparison of the Branch and Bound process under the two lower bounds. 12 jobs have to be scheduled and only 4 jobs are simultaneously allowed in the system.
Table 2 Results obtained with $LB_1$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$</th>
<th>Optimal solution</th>
<th>Number of nodes generated</th>
<th>Pruned nodes</th>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>505</td>
<td>116 388</td>
<td>74 617</td>
<td>1'50s</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>496</td>
<td>105 827</td>
<td>67 823</td>
<td>2'04s</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>514</td>
<td>112 968</td>
<td>71 430</td>
<td>3'54s</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>558</td>
<td>138 983</td>
<td>88 225</td>
<td>5'04s</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>614</td>
<td>1 719 281</td>
<td>1 065 834</td>
<td>27'50</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>566</td>
<td>812 022</td>
<td>535 594</td>
<td>10'48s</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>598</td>
<td>1 222 788</td>
<td>777 256</td>
<td>12'47s</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>634</td>
<td>1 399 509</td>
<td>875 129</td>
<td>15'12s</td>
</tr>
</tbody>
</table>

Table 3. Comparative study of $LB_2$ and $LB_1$ with 12 jobs to schedule.

<table>
<thead>
<tr>
<th>Level</th>
<th>$LB_2$</th>
<th>$LB_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>932</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>21 991</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>107 948</td>
<td>5 765</td>
</tr>
<tr>
<td>10</td>
<td>195 995</td>
<td>298 090</td>
</tr>
<tr>
<td>11</td>
<td>145 437</td>
<td>221 185</td>
</tr>
<tr>
<td>12</td>
<td>10 876</td>
<td>10 554</td>
</tr>
</tbody>
</table>

$LB_2$ is highly efficient because nodes are pruned in highest level during the branch and bound process. In table 3 $LB_2$ prune nodes in level 7 (only 6 jobs have been scheduled before pruning) and $LB_1$ prunes nodes only at level 9. The lower bounds efficiency highly depends on the level at which the prune process is performed. The lower bound $LB_2$ locate efficiently the optimal solution after a few number of nodes. Table 4 highlights that a great percent of computational time is consumed to prove the solution optimality.

Table 4. Results obtained with $LB_2$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$</th>
<th>Total number of nodes generated</th>
<th>Localisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>92 781</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>105 218</td>
<td>14 582</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>112 968</td>
<td>87 225</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>138 983</td>
<td>19 043</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1 191 183</td>
<td>64</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>715 239</td>
<td>87 633</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1 211 421</td>
<td>887 184</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>1 398 407</td>
<td>573 074</td>
</tr>
</tbody>
</table>

6. CONCLUDING REMARKS

The problem of integrated job input sequencing and device dispatching in a general job-shop, where both buffer and resource capacity are limited, has been addressed using the branch and bound approach proposed by (Lacomme et al., 2000). The proposed approach permits to take into account all the constraints of the problem. Previous lower bounds dedicated to this problem are not enough efficient and the lower bounds used in others scheduling problems required many investigations to take into account the problem characteristics.

The work presented here is a step to optimally solve large scale problems. This work has future useful extensions including:

- Heuristic and metaheuristic utilisation for upper bound computation
- Determination of lower bounds for 2 vehicles.

REFERENCES


