INT. J. PROD. RES., 1996, VOL. 34, NO. 2, 577–591

Scheduling of Automated Guided Vehicles in a Decision Making Hierarchy

M. S. AKTURK†‡ and H. YILMAZ†

Scheduling efforts made without considering the special limitations of the material handling system might lead to infeasible results. An analytical model is proposed, first, to incorporate the automated guided vehicle (AGV) system into the overall decision making hierarchy. A mathematical formulation is developed to include interaction between the AGV module and other modules in the system by considering the restrictions of the material handling system. A micro-opportunistic approach is then proposed to solve the AGV scheduling problem. Finally, the proposed method is compared with a number of dispatching rules.

1. Introduction

Hierarchical planning systems are becoming a very popular approach for solving production planning and control problems. Factory reference models, on the other hand, have linked the hierarchical planning systems to organizational structures. One well known factory reference model is the automated manufacturing research facility (AMRF) decision making hierarchy at the National Institute of Standards and Technology in the USA. There are five levels in the AMRF hierarchy, which are factory, shop, cell, workstation and equipment. Each level in this hierarchy has a defined task and responsible for making its own decisions, using separate mathematical models as discussed by Jackson and Jones (1987).

The material handling system, which is treated as a special workstation in the AMRF hierarchy, can have a significant impact on the overall feasibility of the scheduling decisions since it is the primary source of coupling between the different levels in the hierarchy and can lead to infeasibility if it is overloaded. Automated Guided Vehicles (AGV) are becoming the standard material handling method, especially for FMSs, because of their inherent flexibility and capacity, which also make them more challenging from a control perspective. Proposing two separate AGV systems (AGVS), one for intracell and one for intercell activities, to resolve the coupling may not be a desirable alternative because of their high investment cost, and consequently would be difficult to justify due to their low utilization rates. We propose a new approach to incorporate AGVS into the overall decision making hierarchy, but there are two major problems. As discussed by Jones and Saleh (1991) and McGinnis (1989), the scheduling of AGVS in this type of distributed, integrated architecture has not been addressed. Another problem is due to the rigid supervisor/subordinate relationships found in hierarchical approaches that prevents direct peer-to-peer communication. Therefore, move orders from different cells require a large information flow across levels, which may result in lateness in the system.

The seminal paper in AGVS modelling is by Maxwell and Muckstadt (1982).

Revision received November 1994.
§ Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey.
‡ To whom correspondence should be addressed.

0020–7543/96 $12·00 © 1996 Taylor & Francis Ltd.

In AGVS literature, most of the studies are related to AGV design issues, which include determination of number of vehicles required, flow path design and route planning, such as the studies by Bartholdi and Platzman (1989), Gaskins and Tanchoco (1987), Tanchoco *et al.* (1987), and King and Wilson (1991). Tanchoco and Sinreich (1992) suggest a single closed loop guide path layout configuration and discuss the benefits of using a simple guide path versus more complex guide path designs. On the other hand, few studies are made on the operational issues, such as vehicle dispatching and traffic management, and most of them try to identify the conditions that certain dispatching rules perform well (Egbelu and Tanchoco 1984 and Sabuncuoglu and Hommertzheim 1992). Taghaboni and Tanchoco (1988) describe a LISP driven controller for scheduling free-ranging AGV by using the shortest travel time first dispatching rule and a subroutine is incorporated in their routeing procedure to check if more than one vehicle can pass an intersection simultaneously without crossing each other's paths. Krishnamurthy *et al.* (1993) propose a column-generation-based heuristic to find conflict-free routes for multiple AGVs to minimize the makespan. As Bozer and Srinivasan (1989) state, the experimental conditions studied in these cases and objectives are quite different. Because of this reason, many rules are found to perform well in certain settings but no single rule appears to be the best. Actually, the significant effect of material handling in operating costs suggests using more efficient AGV scheduling methods to minimize the cost. Therefore, there is a need to develop a new solution procedure for the AGV scheduling problem that can consider the interaction of the AGV module with the rest of the decision making hierarchy, the current load of the AGVS and the criticality of the jobs simultaneously.

2. Problem statement

The problems associated with placing the AGV module into the decision hierarchy directly (either as a cell or as a workstation) are noted above. To overcome these problems, we propose a hybrid approach in which the control mechanism for the AGV module is designed using a heterarchical structure, so it can interface both shop and cell

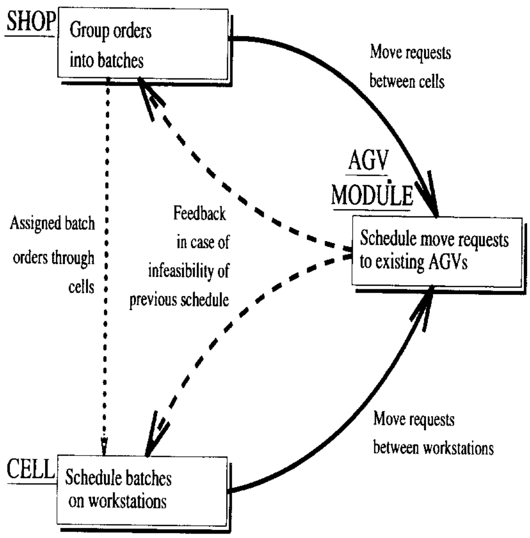

Figure 1. Functional relations in proposed hybrid model.

levels directly as shown in Fig. 1. The proposed hybrid approach still uses the AMRF hierarchical control structure, but we can relax the constraints due to the AGVS, which will be resolved by the AGV module in a hetererchical format. Since each module solves its own problem, the linkages between these modules are very important to have a feasible solution for the overall scheduling problem. In the AMRF hierarchy, in addition to other tasks, the shop level is also responsible for assigning jobs to cells. In the shop level's scheduling problem, the beginning and ending times of jobs in cells are determined with approximate transportation time requirements, which will be passed to the proposed AGV module. Furthermore, the cell level is responsible for scheduling the jobs to workstations. With some approximate time requirements for material movement, each cell prepares an initial schedule. Similar to those of the shop level, a release time and due-date for each move is determined.

The AGV module receives move orders between and within cells in the form of time windows in which the corresponding move request has to be completed. The AGV module should try to find a feasible schedule so that the overall schedule for the system stays feasible. However, if there is no feasible schedule due to additional constraints imposed by the AGVS, then the AGV module should form a new schedule such that related shops or cells can revise their schedules and conduct to AGV module again. In this respect, the objective of the AGV module's scheduling problem should be the minimum amount of deviation from the given time windows. There are two possible deviations that should be penalized. One is earliness, that is requiring an earlier start time of the time window, and the other deviation is tardiness, which corresponds to a later delivery to the drop-off point.

The following assumptions are made in our study. There are N move requests with given time windows and pick-up drop-off points and M identical vehicles in a planning horizon. The layout of the shop with the pick-up and delivery spurs of each workstation is known and the guide path is assumed to be uni-directional. The loads are unit loads, therefore each request requires a devoted vehicle and one vehicle is sufficient for a load request. For the traffic management problem, the control at intersection points of the uni-directional guide path is sufficient to avoid collisions.

The problem can be modelled as a mixed integer program (MIP), where the parameters are

- R_i release time of job i
- D_i due-date of job i
- t_i time required for moving the ith load from its pick-up point to drop-off point
- $u_{h,i}$ time required for moving from the drop-off point of job h to pick-up point of job i

and the decision variables are

- $X_{i,j,k}$ 0-1 binary variable which is equal to 1 if job i is processed as the kth job of vehicle j
- $Y_{h,i}$ 0-1 binary variable which is equal to 1 if job i is processed by the same vehicle after job h
- $S_{i,j,k}$ starting time of job i processing by vehicle j as its kth job
- $E_{i,j,k}$ earliness associated with starting of job i by vehicle j as its kth job
- $F_{i,j,k}$ finishing time of processing job i by vehicle j as its kth job
- $T_{i,j,k}$ tardiness associated with completion of job i by vehicle j as its kth job

$$\text{Min} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{N} (T_{i,j,k} + E_{i,j,k})$$

$$\text{s.t.} \sum_{j=1}^{M} \sum_{k=1}^{N} X_{i,j,k} = 1 \quad \forall i \tag{1}$$

$$\sum_{i=1}^{N} X_{i,j,k} \leqslant 1 \quad \forall j,k \tag{2}$$

$$S_{i,j,k} + E_{i,j,k} \geqslant R_i \cdot X_{i,j,k} \quad \forall i,j,k \tag{3}$$

$$F_{i,j,k} - S_{i,j,k} \geqslant t_i \cdot X_{i,j,k} \quad \forall i,j,k \tag{4}$$

$$S_{i,j,k} \geqslant F_{h,j,k-1} + Y_{h,i} \cdot u_{h,i} \quad \forall i,h,j,k \tag{5}$$

$$Y_{h,i} \geqslant X_{h,j,k-1} + X_{i,j,k} - 1 \quad \forall i,h,j,k \tag{6}$$

$$F_{i,j,k} - T_{i,j,k} \leqslant D_i \cdot X_{i,j,k} \quad \forall i,j,k \tag{7}$$

$$S_{i,j,k}, E_{i,j,k}, F_{i,j,k}, T_{i,j,k} \geqslant 0 \quad \forall i,j,k \tag{8}$$

The objective is to minimize the total deviation from the time windows. The first and second set of constraints ensure that a single vehicle is assigned for each job and any order of a particular vehicle is not assigned to more than one job, respectively. The third, sixth and seventh constraint sets are the definitions of $E_{i,j,k}$, $Y_{h,i}$ and $T_{i,j,k}$ respectively. The fourth set is offsetting the loaded travel times on the basis of jobs. The fifth set is offsetting the unloaded travel times on the basis of vehicles. The problem is defined similarly to the time constrained vehicle routeing problem (TCVRP), which is proven to be NP-hard by Kolen *et al.* (1987). Actually, by adding the loaded travel time required for a particular job to the unloaded travel time, our problem becomes identical to TCVRP, except the objective functions since TCVRP tries to minimize the distance travelled, whereas our objective is the minimum deviation.

3. The AGV scheduling algorithm

One can approach our problem from two simple points of view. The job-based approach might try to schedule the 'tightly constrained jobs' (i.e. with smaller time windows) first. However, the unloaded movement times of vehicles, which correspond to sequence dependent set-up times in scheduling framework, are ignored. The second view might be a vehicle-based one to minimize the unloaded travel times so that jobs will have more opportunities to be scheduled. So a vehicle-based approach eliminates the disadvantage of a job-based one but it disregards the critical jobs. The idea of micro-opportunistic scheduling would correspond to combining these two perspectives in a single algorithm in which the perspective can be changed each time a job is assigned to a vehicle. As a result, both the critical jobs and the unloaded travel times are considered simultaneously. A similar idea is applied to the job shop scheduling problem by Sadeh (1991), but there are neither time windows nor set-up times. Furthermore, instead of strictly scheduling the jobs to vehicles, we impose intervals for each job in which it will be scheduled in the end to the assigned vehicle. These intervals get narrow in time after each assignment, and at the end, the exact schedule is formed. This gives a large amount of flexibility to our scheduler. The proposed micro-opportunistic scheduling algorithm (MOSA) is outlined below. As explained in § 2, the input parameters of MOSA are passed from the AMRF control hierarchy.

Step 0: (Initialization.) Identify time slots from release times and due-dates of each job. Determine the individual weights of each job, W_i, using the following expression:

$$W_i = 1/(D_i - R_i - t_i) \quad \text{if} \quad D_i > R_i + t_i$$

Step 1: Let U be the set of un-scheduled jobs, If $U = \emptyset$ then go to Step 10, else determine time slot loads from the set of all un-scheduled jobs as follows:

$$SL_k = \sum_{i \in U} I_{i,k} * W_i$$

where the indicator function $I_{i,k}$ is equal to 1 if $i \in U$ and slot k is included in the interval (R_i, D_i). Find the most contended time slot, $m = \arg\max_{\forall k}\{SL_k\}$, and let a_m be the starting time and b_m be the ending time of slot m denoted by time interval $[a_m, b_m]$.

Step 2: (Schedule the jobs in the most contended time slot.) Let C be the set of un-scheduled jobs in the most contended time slot sorted in descending order of weights and $i \in C$ if $R_i \leqslant b_m$ and $D_i \geqslant a_m$. Let job $k = \arg\max_{i \in C}\{W_i\}$, then $S_{k,1} = R_k$ and compute the 'slack' of the first vehicle, $\epsilon_1 = D_k - S_{k,1} - t_k$.

Step 3: Let job i be the next un-scheduled job in C, if job i has either the same drop-off point with the first pick-up or the same pick-up point with the last drop-off point of a vehicle j on the temporary schedule so that no penalty incurs, then schedule job i on vehicle j and update the 'slack' of the vehicle, $\epsilon_j = \min\{\epsilon_j, D_i - S_{i,j} - t_i\}$. Otherwise schedule job i to a new vehicle at the earliest possible time, compute the 'slack' of the new vehicle. If $C = \emptyset$ then go to Step 6, otherwise select the next unscheduled job in C until all the vehicles are used at least once.

Step 4: For each un-scheduled job $i \in C$, the minimum distance/penalty requirement on each vehicle j is calculated such that $\pi^l_{i,j}$ indicates the desirability of placing job i on vehicle j just before the first job on the vehicle, i.e. to the left of the block of jobs on the vehicle j, in a Gantt chart representation. Since a job can be appended either side of a block, two priority functions, $\pi^l_{i,j}$ and $\pi^r_{i,j}$, for left and right, respectively, are as follows:

$$\pi^l_{i,j} = \frac{1}{t_i + u_{i,p}} * \exp\left(\frac{ST_j + \epsilon_j - (R_i + t_i + u_{i,p})}{2*k*t_{av}}\right)$$

and

$$\pi^r_{i,j} = \frac{1}{t_i + u_{d,i}} * \exp\left(\frac{D_i - (ET_j + u_{d,i} + t_i)}{2*k*t_{av}}\right)$$

where p is the pick-up point of first, d is the drop-off point of last job on the vehicle, R_i and D_i are release time and due-date of job i, ST_j, ET_j and ϵ_j are the starting time, ending time of slack of vehicle j respectively, k is a constant and t_{av} is the average travelling time and jobs. For each $i \in C$, the best vehicle j^* and its side, i.e. l or r, are selected that maximizes the desirability functions.

Step 5: (Select the next job to be scheduled.) For each un-scheduled job $i \in C$, either one of the following functions, $\rho^l_{i,j*}$ and $\rho^r_{i,j*}$, based on the 'direction' selected for that job, i.e. l or r, on the best vehicle j^* are calculated to find the job criticality as follows:

$$\rho^l_{i,j^*} = \frac{1}{t_i + u_{i,p}} * \exp\left(\frac{(R_i + t_i + u_{i,p}) - (ST_{j^*} + \epsilon_{j^*})}{2*k*t_{av}}\right)$$

and

$$\rho^r_{i,j^*} = \frac{1}{t_i + u_{d,i}} * \exp\left(\frac{ET_{j^*} + u_{d,i} + t_i - D_i}{2*k*t_{av}}\right)$$

Schedule job $i = \max_{i \in C}\{\rho^l_{i,j^*}, \rho^r_{i,j^*}\}$ to the vehicle j^*, and update $\epsilon_{j^*} = \min\{\epsilon_{j^*}, D_i - S_{i,j^*} - t_1(i)\}$. If $C \neq \emptyset$ then go to Step 4.

Step 6: (Identify conflicts between temporary schedules.) Let Y be the set of temporary blocks, B_v be the earliest start time of temporary schedule v and F_v be the latest completion time. If either $B_v \geqslant F_w$ or $F_v \leqslant B_w$ where $v \neq w$ and $v, w \in Y$ then go to Step 7. If both $B_v \geqslant F_w$ or $F_v \leqslant B_w$ and $B_v \geqslant F_z$ or $F_v \leqslant B_z$ where $v \neq w \neq z$ and $v, w, z \in Y$ go to Step 9, otherwise go to Step 1.

Step 7: (Merge temporary schedules.) Start with the earlier of these two blocks, i.e. minimum B_v. Identify and append the jobs in between them to the earlier slot. Sort the completion times of vehicles in the earlier block in an ascending order and select the first vehicle in the list. For the selected vehicle, sort the un-merged vehicles of the later block in increasing possible start times including the required unloaded travel time. Merge the first vehicle in the list to the selected vehicle of earlier block.

Step 8: If there is no overlap between the vehicles, schedule the group at the earliest time possible and update slacks to a single slack. Else shift the schedule of the vehicle resulting in a minimum penalty either to the left (if it is the earlier one) or to the right (if it is the later one), while holding the other one fixed. If there are still vehicles to merge, select the next vehicle from the earlier group and go to Step 7. Otherwise go to Step 1.

Step 9: Start with the earlier conflicting slot, identify and schedule the jobs between this slot and current slot. Merge the current slot to existing schedule. Identify and append the jobs between the current slot and the later conflicting slot. Finally merge the later slot to this combination. Go to Step 1.

Step 10: (Collision avoidance.) Identify the crossing times of each vehicle from the intersection points. If there is an overlap between two vehicles at time t^c at any intersection point, select the one with greater slack, $j^c = \arg\max_j\{\epsilon_j\}$. For all jobs i, whose $S_{i,j^c} + t_i \geqslant t^c$, then update $S_{i,j^c} = S_{i,j^c} + \Delta$.

The release time and due-date of each job can be viewed as points in time. Eliminating repetitions (e.g. one job's due-date may be equal to one other's release time), every interval formed between consecutive such points is called a 'time slot'. These slots will be used to determine the time intervals that are more likely to result in early or tardy jobs due to high contention among the jobs, since jobs with a larger time window are assigned a lower weight. Our algorithm selects these intervals as a starting point. Thus instead of giving priority to particular jobs or vehicles, we give priority to time intervals. An example of a slot load profile is given in Fig. 2. The first slot to start scheduling is the one with highest load. If there is a tie then the time slot with the maximum number of jobs is chosen. In further iterations, some of the jobs will be scheduled and these load profiles will be re-computed at each iteration.

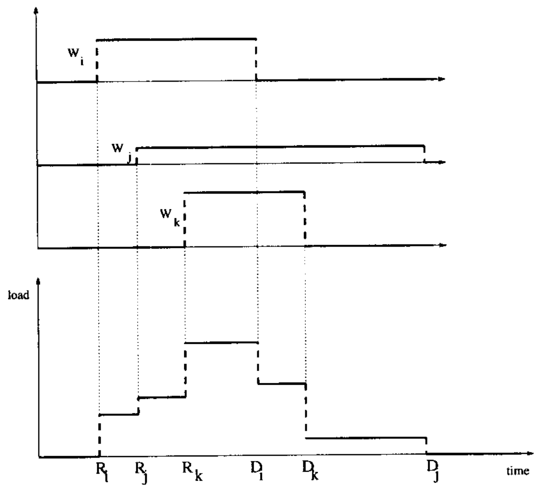

Figure 2. Time slot load profile.

In Step 1, the most contended time slot for the current iteration is determined, whereas in Step 2, the jobs in this slot will be scheduled to M temporary vehicles. In Step 4, a vehicle with a larger feasibility interval and smaller unloaded travel time is given higher priority. Therefore, for each job a particular vehicle and its side is selected that maximizes the desirability functions among the 2^*M possibilities. In Step 5, on the other hand, a higher priority is given to the more critical job. These assignments are preserved in each iteration, but the schedules made in Step 2 are not fixed. Every job is scheduled to start in a certain interval, rather than a fixed point, in which it can be shifted to the right in a Gantt chart representation without incurring penalty. For a block of jobs on a vehicle, there is an interval that the block can be placed anywhere within. These intervals are held in the form of constraints rather than fixed schedules when joining two temporary vehicle blocks in further steps.

In Step 2, the previous vehicle schedules were not taken into consideration. The possible conflicts between the new block and those formed before are resolved in Steps 6–9. In this decomposition procedure, we first ignore the other loads of vehicles and solve the scheduling problem, then we try to impose the previous assignments in the form of constraints to the existing solution. In the case of a conflict, each temporary vehicle is assigned to a vehicle from the previously scheduled block and two vehicles are 'merged'. Two blocks can be in a conflict if the latest completion time of all jobs in the earlier block exceed the earliest starting time of the later block. This corresponds to overlapping of partial schedules, in other words, non-interference constraints for vehicles are violated. Thus there can be three possibilities, which are no conflict, one conflict and two conflict cases. In case of a conflict, not only the two blocks are merged, but also the jobs that are between the two conflicting blocks are scheduled in Step 7. Because there can be jobs that were not included in either of these blocks, which have a release time later than the earlier block's ending time but a due-date earlier than the later block's starting time. If these jobs are not taken into account at the moment, they are likely to create a very high penalty in further steps. Meanwhile two conflicting blocks are not differentiated in Step 9, therefore we utilize Steps 7 and 8 twice to solve the two-way conflict case.

In a uni-directional layout, the control of nodes at the intersection points is sufficient for collision avoidance. The traffic management problem is handled in a two-phase decision making process. In the first phase, we schedule the move orders into the vehicles. In Step 10, we reschedule the starting times of conflicting move orders by perturbing them by a fixed Δ amount.

4. A numerical example

Consider the following 20-job problem with the corresponding R_i, D_i, t_i, pick-up (PU) and drop-off (DO) points with the individual weights as given in Table 1. There are two vehicles and the uni-directional layout is shown in Fig. 3. We first identify time slots and calculate their loads as summarized in Table 2. As a result the most contended time slot is [19, 27] and the set $C = \{20, 14, 7, 12, 6\}$ is sorted in descending order of weights. We start by assigning job 20 to the temporary vehicle 1 at $R_{20} = 14$. As explained in Step 3, job 14 is scheduled to the temporary vehicle 2 at time $R_{14} = 16$. Since all the vehicles are used at least once, we go to Step 4 and the corresponding slacks of each vehicle are $\epsilon_1 = 22-18 = 4$ and $\epsilon_2 = 28-22 = 6$.

In Step 4, the desirability function for each unscheduled job $i \in C$ are calculated as summarized in Table 3. These calculations for job 7 are given below for each vehicle as an example, where $t_{av} = (\Sigma_{i=1}^{20} t_i)/20 = 7{\cdot}8$ and $k = 2$

$$\pi^l_{7,1} = \frac{1}{8+4} * \exp\left(\frac{14+4-(16+8+4)}{2*2*7{\cdot}8}\right) = 0{\cdot}0605$$

$$\pi^r_{7,1} = \frac{1}{8+0} * \exp\left(\frac{32-(18+8+0)}{2*2*7{\cdot}8}\right) = 0{\cdot}1515$$

$$\pi^l_{7,2} = \frac{1}{8+8} * \exp\left(\frac{16+6-(16+8+8)}{2*2*7{\cdot}8}\right) = 0{\cdot}0454$$

$$\pi^r_{7,2} = \frac{1}{8+10} * \exp\left(\frac{32-(22+8+10)}{2*2*7{\cdot}8}\right) = 0{\cdot}0429$$

So job 7 should be scheduled on vehicle 1 right after job 20, which means to the right of the vehicle 1 in a Gantt chart representation, and the following function is used to calculate the criticality of job 7 based on the previous selection as explained in Step 5.

$$\rho^r_{7,1} = \frac{1}{8+0} * \exp\left(\frac{18+8+0-32}{2*2*7{\cdot}8}\right) = 0{\cdot}103$$

The maximum $\rho_{i,j*}$ is given by job 7 as shown in Table 3 so it is scheduled to the right of the vehicle 1 at time 18. The new slack time of vehicle 1 is $\epsilon_1 = \min\{4, 32-26\} = 4$. For the remaining jobs 12 and 6, their priorities on vehicle 2 stay same and the new $\pi^r_{i,1}$ values after scheduling job 7 are given in Table 3. So we select job 6 and schedule it just before job 14 to the left of the vehicle 1. The slack time of vehicle 2 is still 4. For the remaining job 12, $\pi^l_{i,1} = -\infty$ due to the starting time of the planning horizon restriction. Since only one job left there is no need to calculate $\rho_{i,j*}$, and job 12 is scheduled to the right of the vehicle 1 at time 32. The Gantt chart after the first iteration is shown in Fig. 4, where the unloaded travel time is denoted by UT and the upper numbers indicate the stations.

Since there is no conflict we determine the next most contended time slot which is [85, 89] and $C = \{1, 18, 15, 11\}$. Job 1 is scheduled to the temporary vehicle 1 at $R_1 = 85$ and $\epsilon_1 = 4$. Job 18 cannot be scheduled to the vehicle 1 without incurring any penalty

Job #	R_i	D_i	PU	DO	t_i	W_i
1	85	93	4	3	4	0·25
2	126	142	1	6	8	0·125
3	46	54	4	6	4	0·25
4	97	117	2	6	10	0·1
5	136	152	6	1	8	0·125
6	3	27	3	6	12	0·083
7	16	32	6	1	8	0·125
8	113	129	2	5	8	0·125
9	108	124	3	4	8	0·125
10	38	58	5	1	10	0·1
11	72	92	2	6	10	0·1
12	22	42	5	1	10	0·1
13	137	157	3	5	10	0·1
14	16	28	6	2	6	0·167
15	83	99	3	4	8	0·125
16	62	70	6	3	4	0·25
17	52	64	4	2	6	0·167
18	73	89	5	2	8	0·125
19	130	150	2	6	10	0·1
20	14	22	4	6	4	0·25

Table 1. Job set parameters.

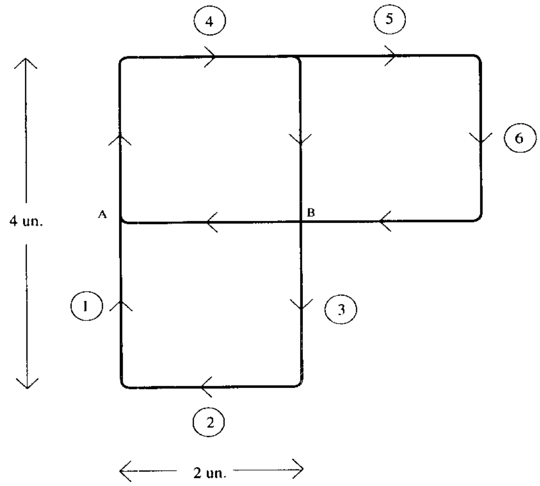

Figure 3. Guide path layout.

Time slot	Jobs	SL_k
(19, 27)	(6, 14, 7, 20, 12)	0·725
(38, 42)	(12, 10)	0·200
(52, 54)	(10, 3, 17)	0·517
(62, 64)	(17, 16)	0·417
(85, 89)	(11, 18, 15, 1)	0·600
(97, 99)	(15, 4)	0·225
(113, 117)	(4, 9, 8)	0·350
(136, 152)	(2, 19, 5, 3)	0·450

Table 2. Time slot loads.

Order	Job #	$\pi^l_{i,1}$	$\pi^r_{i,1}$	$\pi^l_{i,2}$	$\pi^r_{i,2}$	Max	$(j^*, \text{Dir.})$	ρ^l_{i,j^*}	ρ^r_{i,j^*}
	7	0·0605	0·1515	0·0454	0·0429	0·1515	$(1, r)$	—	0·103
1	12	0·0401	0·0568	0·0312	0·0592	0·0592	$(2, r)$	—	0·0521
	6	0·0426	0·0499	0·1043	0·0264	0·1043	$(2, l)$	0·0666	—
2	12	0·0401	0·0625	0·0312	0·0592	0·0625	$(1, r)$	—	0·625
	6	0·0426	0·0272	0·1043	0·0264	0·1043	$(2, l)$	0·0666	—
3	12	$-\infty$	0·0625	0·0312	0·0592	0·0625	$(1, r)$		

Table 3. Summary of the first iteration.

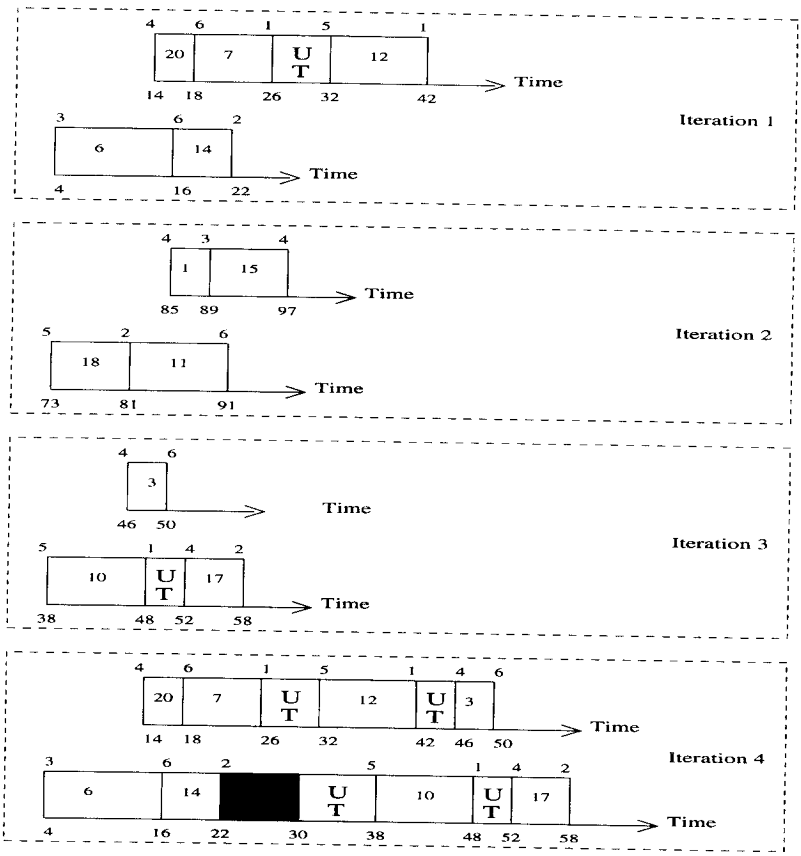

Figure 4. Gantt charts for iterations 1–4.

Order	Job #	$\pi^l_{i,1}$	$\pi^r_{i,1}$	$\pi^l_{i,2}$	$\pi^r_{i,2}$	Max	$(j^*, \text{Dir.})$
1	11	0·0505	0·0312	0·0351	0·1033	0·1033	$(2, r)$

Table 4. Summary of the second iteration.

Temporary vehicle	Earliest possible starting time
1	$\max\{B_v, c_2 + u_{2,4}\} = \{46, 28\} = 46$
2	$\max\{B_v, c_2 + u_{2,5}\} = \{38, 30\} = 38$

Table 5. Summary of merging, iteration 4.

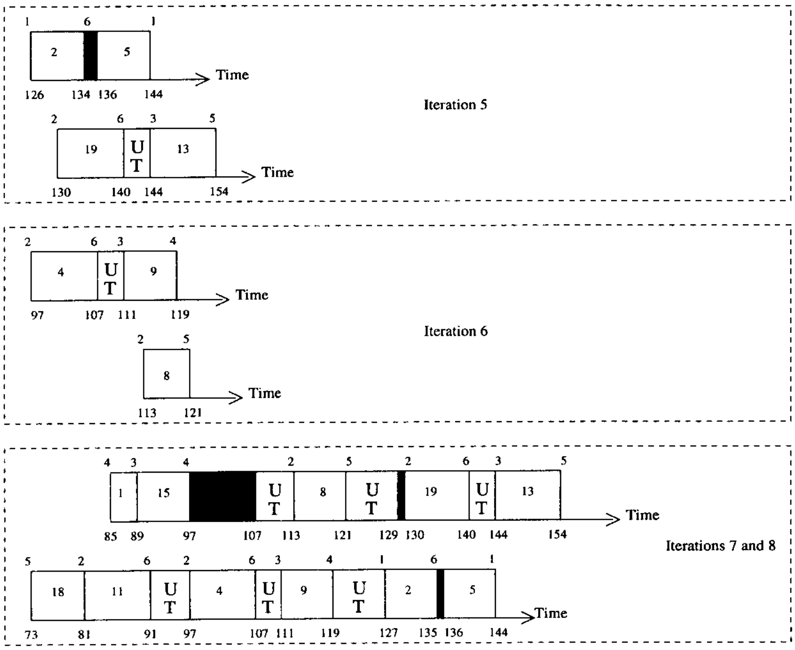

Figure 5. Gantt charts for iterations 5–8.

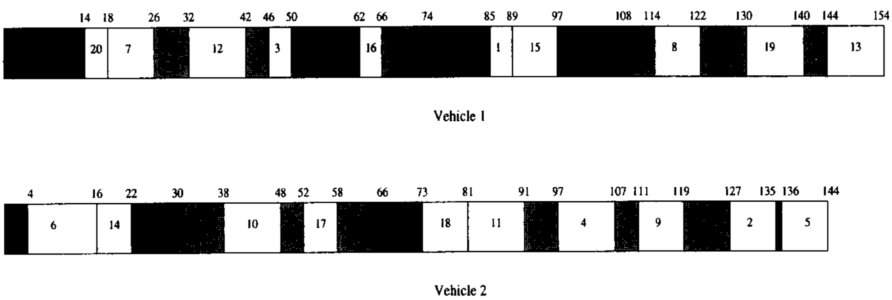

Figure 6. Final Gantt chart for the Example problem.

Factor	Levels		
Number of jobs	100	150	200
Layout	Small	Medium	Large
Tightness factor	2·0	4·0	6·0
Number of vehicles	2	2	8

Table 6. Factors and levels in the experimental design.

so it is scheduled to vehicle 2 and $\epsilon_2 = 8$. Job 15, on the other hand, can be scheduled right after job 1 to the right of vehicle 1 and $\epsilon_1 = \min\{4, 2\} = 2$. We calculate the priorities for the only remaining job 11 as summarized in Table 4. The Gantt chart after the second iteration is shown in Fig. 4. There is still no overlap between these two temporary assignments and the Gantt chart for the next most contended time slot [52, 54] for which $C = \{3, 17, 10\}$ is also shown in Figure 4.

There is an overlap between the first iteration and the third iteration because the latest completion time of the earlier block, i.e. iteration 1, is greater than the earliest starting time of the later block, i.e. iteration 3. There is no jobs between these two blocks. The completion times, c_j, of the vehicles of the first iteration is sorted in an ascending order and vehicle 2, i.e., $c_2 = 22$, is selected. The possible earliest starting times of the later block including the unloaded travel times are given in Table 5. Both assignments are feasible, but the temporary vehicle 2 has the earliest starting time. Therefore we merge vehicle 2 of the earlier block with the temporary vehicle 2 of the later block. Furthermore, the remaining two temporary vehicles are also merged as shown in Fig. 4, Iteration 4. New slack times are $\epsilon_1 = \min\{0, 4\} = 0$ and $\epsilon_2 = \min\{6, 6\} = 6$. The conflict is resolved and the next time slot is [136, 152]. The Gantt chart of the fifth iteration along with the iterations 6, 7 and 8 are shown in Fig. 5.

After finding an initial schedule in Phase I, we check the crossing times of each vehicle from two intersection points A and B as indicated in Fig. 3. Let Z_1^A and Z_1^B be the crossing times of vehicle 1 from intersection points A and B, respectively such that $Z_1^A = \{27, 43, 71, 94, 116, 133, 149\}$ and $Z_1^B = \{21, 37, 65, 88, 110, 126, 143\}$. Similarly $Z_2^A = \{10, 33, 49, 61, 84, 100, 116, 128\}$ and $Z_2^B = \{19, 43, 55, 78, 94, 110, 122, 139\}$. Since both vehicles use the intersection point B at time 110, we delay job 8 on vehicle 1, which has a greater slack value, by 1 time unit. The final schedule is summarized in the Gantt chart of Fig. 6, which is both feasible, i.e. the total deviation is equal to zero, and also free of collisions.

5. Experimental analysis

In order to span different experimental settings, any variable that is likely to be significant on the performance of the algorithm should be included. The experimental factors are listed in Table 6. The number of jobs to be scheduled is likely to affect the performance of different methods since it directly affects the load of the system. The studies in the literature show that the layout can affect the performance of different methods, since the distances between points are a function of layout. The third factor determines the tightness of time windows by utilizing the total work content rule by Baker (1984), which assigns due-dates proportional to the product of the loaded travel time of a job and the tightness factor, K, such that $D_i = R_i + K \cdot t_i$. The fourth factor is the number of vehicles in the system which affects the average load of each vehicle. In order to observe any quadratic relation between the factors and performance

Replication no.		ER	EDD	STTF	RM	MOSA
Rep. 1	Aver.	1353·2	1690·0	353·3	348·3	227·9
	Min.	0	0	0	0	0
	Max.	8666	8881	2201	2944	1391
Rep. 2	Aver.	1401·2	1751·3	353·0	345·1	263·8
	Min.	0	0	0	0	0
	Max.	9604	9305	2239	2509	2206
Rep. 3	Aver.	1437·4	1832·6	389·5	381·5	261·8
	Min.	5	0	1	0	0
	Max.	12973	10250	2716	2790	1531
Rep. 4	Aver.	1504·8	2164·4	403·1	365·4	267·3
	Min.	0	0	0	0	0
	Max.	17096	24831	3859	2571	1432
Rep. 5	Aver.	1461·8	1657·9	446·1	360·7	268·5
	Min.	0	0	0	0	0
	Max.	20065	15047	6532	4461	2381
Overall	Aver.	1431·7	1819·2	389·0	360·2	257·9

Table 7. Total deviation comparison.

	CPU times (sec.)				
	ER	EDD	STTF	RM	MOSA
Aver.	0·10	0·09	0·11	0·19	8·59
Min.	0·04	0·05	0·03	0·06	1·04
Max	0·17	0·17	0·22	0·28	39·00

Table 8. Computation times comparison.

	Significance levels		
Factors	Linear	Quadratic	Total
Number of jobs	1%	1%	1%
Layout	1%	1%	1%
Time window tightness	1%	*	1%
Number of vehicles	1%	*	1%
Layout-time window tightness	1%	*	1%

Table 9. Significant factors for total deviation.

measures, every factor has three levels in the design. Since there are four factors and three levels, our experiment is 3^4 full-factorial design, which corresponds to eighty-one treatment combinations. The number of replications of each combination is taken as five, that gives 405 different runs.

Other variables in the system are treated as fixed parameters. One of them is the release time distribution. Release times of jobs are assumed to be distributed uniformly in the interval between zero and expected makespan, Uniform $(0, 2*t_{av}*N/M)$. The distribution of departments for the jobs is also assumed to be uniform, that is every department is equally likely to be the pick-up or drop-off point of a move order. The number of departments in the layout is considered as part of the layout factor, and thus is not independently manipulated. In most of the studies, the transfer of loads between stations is managed by dispatching rules. Therefore, four heuristic methods are selected for comparison purposes, which are earliest release time (ER), shortest travel time first (STTF), earliest due-date (EDD), and Rachamadagu-Morton (RM). STTF is resource-based, while ER and EDD are job-based rules. The fourth alternative is the adaptation of RM rule discussed in Vepsalainen and Morton (1987) to the AGV scheduling problem, in which the unloaded travel times are added to the processing times. Since these are all forward dispatching rules, no job can be assigned before its release time.

The performance measure is the total deviation from time windows. Five sets of 81 problems are generated using the parameters as stated and run on PC-486. A summary of total deviation results is given in Table 7, which indicates that the proposed micro-opportunistic scheduling approach (MOSA) outperforms all the others at the 0·5% significance level due to a paired t-test. The average penalty of MOSA is around 72% of the second best, which is RM. Furthermore, as far as the ranges of these methods are concerned, our algorithm's maximum is about 53% of the second best, which gives a good indication that our algorithm is also robust to changes in settings. Furthermore, no rule is dominated, i.e. each rule performs best in at least one run.

Time requirements, which are given in Table 8, indicate that our method requires considerably larger time than the other rules. However, this should be expected since the other four rules are only dispatching rules. Although the computation time is relatively large, the maximum time requirement of 39 seconds for a problem with 200 jobs is much less than the planning horizon for such a planning decision. Finally, an ANOVA model is performed to observe the effects of factors on the performance measure, which is summarized in Table 9. As was expected, all factors are found to be significant on the performance of our method. For combination of factors, only the layout-time window tightness interaction is found to be significant.

6. Conclusion

In this paper, the problem of incorporating the AGVS module to the decision making hierarchy is analysed by stating the reasons and consequences of this difficulty. In an attempt to overcome this difficulty, a hybrid model for AGV module is proposed. A micro-opportunistic heuristic method is developed, then, to solve the AGV scheduling problem. Comparison of the proposed method with a number of AGV dispatching rules showed that MOSA performed better than other methods in an acceptable computation time.

For further research, there are mainly two areas that need further analysis. First, as was noted, the number of vehicles is assumed to be an input for an AGVS module. This is because of the fact that the decision on the number of vehicles is a high level decision and it should be made at the design stage of an AGVS system. However, the changing conditions in a shop may justify buying new vehicles or shifting some vehicles across different shops. In order to facilitate these high level decisions, the average deviation incurred and the utilization rate of vehicles should be fed back to the plant level. Another future research topic would be developing a mechanism to estimate the average transportation time requirements using the statistical information passed from the AGV module, such as the actual realization of transportation times between stations due to vehicle contention and traffic effects.

References

BARTHOLDI, J. J., and PLATZMAN, L. K., 1989, Decentralized control of automated guided vehicles on a single loop. *IEE Transactions*, **21** (1), 76–81.

BAKER, K. R., 1984, Sequencing rules and due-date assignments in a job shop. *Management Science*, **30** (9), 1093–1104.

BOZER, Y. A., and SRINIVASAN, M. M., 1989, Tandem configurations for AGV systems offer simplicity and flexibility. *Industrial Engineering*, **21** (2), 23–27.

EGBELU, P. J., and TANCHOCO, J. M. A., 1984, Characterization of Automatic Guided vehicle Dispatching Rules. *International Journal of Production Research*, **22** (3), 359–374.

GASKINS, R. J., and TANCHOCO, J. M. A., 1987, Flow Path Design for Automated Guided Vehicle Systems. *International Journal of Production Research*, **25** (5), 667–676.

JACKSON, R. H. F., and JONES, A. W. T., 1987, An architecture for decision making in the factory of the future. *Interfaces*, **17** (6), 15–20.

JONES, A., and SALEH, A., 1991, The cell as a part of a manufacturing system. *Manufacturing Cells: Control, Programming and Integration*, P. Rogers, and D. J. Williams (eds) (London: Butterworth-Heinemann).

KING, R. E., and WILSON, C., 1991, A review of automated guided vehicle systems design and scheduling. *Production Planning and Control*, **2** (1), 44–51.

KOLEN, A. W. J., RINNOOY KAN, A. H. G., and TRIENEKENS, H. W. J. M., 1987, Vehicle routeing with time windows. *Operations Research*, **35** (2), 266–274.

KRISHNAMURTHY, N. N., BATTA, R., and KARWAN, M. H., 1993, Developing conflict-free routes for automated guided vehicles. *Operations Research*, **41** (6), 1077–1090.

MAXWELL, W. L., and MUCKSTADT, J. A., 1982, Design of automated guided vehicle systems. *IIE Transactions*, **14** (2), 114–124.

MCGINNIS, L. F., 1989, Some Issues in Controlling Integrated WIP Material Handling Requirements. *Advanced Information Technologies for Industrial Material Flow Systems* S. N. Nof and C. L. Moodie, eds, Nato ASI, **F53** (Berlin: Springer-Verlag).

SADEH, N., 1991, MICRO-BOSS: a micro-opportunistic factory scheduler. Technical Report CMU-RI-TR-91-22, Carnegie Mellon University, USA.

SABUNCUOGLU, I., and HOMMERTZHEIM, D. L., 1992, Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion. *International Journal of Production Research*, **30** (7), 1617–1635.

TAGHABONI, F., and TANCHOCO, J. M. A., 1988, a LISP based controller for free-ranging automated guided vehicle systems. *International Journal of Production Research*, **26** (2), 173–188.

TANCHOCO, J. M. A. and SINREICH, D., 1992, OSL–optimal single-loop guide paths for AGVS. *International Journal of Production Research*, **30** (3), 665–681.

TANCHOCO, J. M. A., EGBELU, P. J., and TAGHABONI, F., 1987, Determination of the total number of vehicles in an AGV-based system. *Material Flow*, **4,** 33–51.

VEPSALAINEN, A. P. J., and MORTON, T. E., 1987, Priority rules for job shops with weighted tardiness costs. *Management Science*, **33** (8), 1035–1047.