

Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system

IHSAN SABUNCUOGLU[†] and DON L. HOMMERTZHEIM[‡]

In this paper, an on-line dispatching algorithm is proposed for the FMS scheduling problem. The algorithm uses various priority schemes and relevant information concerning the load of the system and the status of jobs in the scheduling process. This information is organized into hierarchical levels. The scheduling decision process is hierarchical in the sense that different decision criteria are applied sequentially to identify the most appropriate part and the machine to be served. The algorithm schedules the jobs on a machine or an automated guided vehicle (AGV) one at a time as the scheduling decision is needed (or as the status of the system changes). Performance of the proposed algorithm is compared with several machine and AGV scheduling rules by using the mean flow-time and the mean tardiness criteria. Simulation results indicate that the proposed algorithm produces significant mean flow-time and mean tardiness improvements over existing scheduling rules for a variety of experimental conditions.

1. Introduction

Flexible manufacturing systems (FMSs) can be described as batch manufacturing systems which consist of a group of numerically controlled (NC) machines connected by an automated materials handling system under computer control. These systems are used to process a wide variety of different parts with low-to-medium demand volume. Since a large number of manufacturing companies operate within the mid-variety and mid-volume range, FMSs cover a large number of application areas. Therefore, there are many types of FMSs as well as a variety of classifications used to describe them. For example, Groover (1987) divided FMSs into dedicated and random systems. Dedicated systems consist of special machine tools arranged in a flow configuration to produce a limited variety of identical parts; whereas, random FMSs are capable of producing a higher variety of parts in a random order. The system for which the on-line dispatching algorithm proposed in this paper is a random FMS.

In many respects, an FMS can be viewed as an automated job shop. The major difference between an FMS and a conventional job shop is that the human functions are automated in the FMS. Furthermore, because of its integrated nature, a scheduling task for an FMS requires additional considerations of tools, fixtures, automated guided vehicles (AGVs), pallets, etc. In general, machine tools are more versatile (capable of performing various operations) and are usually equipped with sensors, tool magazines, and automatic tool changing mechanisms so that a wide variety of different parts can be machined automatically with minimum changeover time. The materials handling system (usually an AGV system) is also more flexible and connects various workcentres of the FMS. However, all this built-in flexibility in FMSs complicates operational problems due to a large number of alternative operations and material handling routes to be considered during scheduling decisions.

Received August 1991.

[†] Department of Industrial Engineering, Bilkent University, Ankara, Turkey 06572.

[‡] Department of Industrial Engineering, Wichita State University, Wichita, KS 67208, USA.

This paper proposes a dispatching algorithm for such an FMS scheduling problem. The algorithm uses information concerning the load of the system and the status of jobs in the scheduling process. This information is organized in hierarchical levels. It is an on-line scheduling algorithm which schedules the jobs on a machine or an AGV one at a time as the scheduling decision is needed (or as the status of the system changes). Performance of the proposed algorithm is also compared with several machine and AGV scheduling rules under various experimental conditions using the mean flow-time and the mean tardiness measure.

The rest of the paper is organized as follows. In Section 2, definitions and characteristics of the FMS scheduling problem are presented. This is followed by a survey of relevant literature in Section 3. In Section 4, structure of the proposed algorithm, system considerations, and experimental conditions are discussed. Performance of the proposed algorithm and comparisons with scheduling rules are presented in Section 5. Finally, concluding remarks are made in Section 6.

2. Scheduling problems of FMS

An FMS scheduling problem is considered to be a detailed minute-by-minute scheduling of the machines, materials handling system, and other support equipment. Given the actual shop conditions and a set of parts with known processing requirements, it is concerned with accomplishing the following tasks:

- Schedule actual job release times.
- Sequence the jobs and determine the start and completion times of each operation on a wide variety of resources.
- Monitor the execution of the schedule and provide effective contingency handling.

Although scheduling refers to the time-phased allocation of all the system resources such as machines, tools, materials handling system, etc., it is most often applied to the scheduling of jobs on the machines. However, for a dynamic and highly integrated system such as an FMS, the scheduling of the materials handling system in real time and considerations of limited input/output buffer capacities are also equally important. These points will be further discussed in the following sections.

2.1. Scheduling of machines and the materials handling system

From a system perspective, an FMS basically consists of two interrelated subsystems: a machining subsystem and a materials handling subsystem. The machining subsystem is a typical job shop in which parts with different processing steps and routing plans are processed. On the other hand, the materials handling system is usually an AGV system in which mobile vehicles move the parts from one workstation to another along a variable or a pre-defined path.

These two subsystems are so closely integrated that the performance of one affects the other. While each job completion at a machining centre generates an arrival to the AGV subsystem, completion of a transportation service by an AGV determines job potentials for workstations. In a typical FMS, the system status changes so frequently that at one time the machining subsystem can be a 'constrained resource' or a 'driving force' for the material flow whereas at some other time the AGV system becomes a critical resource and eventually dominates the schedule. Because of this two-way interaction, both subsystems must be taken into account simultaneously in scheduling an FMS.

2.2. Considerations of limited queue capacities

One dimension of the FMS scheduling problem is that both the input and output queue capacities at workstations are limited. Therefore, there is always a possibility that a particular machine can be blocked or the system can be locked due to limited queue spaces. Blocking occurs when a machine cannot move its part to a buffer if the buffer is full. Whereas, locking occurs when the system is totally prevented from functioning, i.e. when no part movement can be achieved in the system. Therefore, limited queue capacities must be taken into account in scheduling an FMS.

The scheduling problem described above is a more general class of the job shop problem. As compared with traditional job shop problems in which only machines are considered to be constrained resources, in scheduling of FMSs both the materials handling system and the limited in-process buffer spaces are also constrained resources and should be modelled directly.

Even though the important characteristics of an FMS are considered in this study, some of the other typical FMS resources such as tools, pallets/fixtures, and their availabilities are not modelled for the following reasons. First, as discussed in Looveren *et al.* (1986), allocation of these resources can be made at the planning stage when the batching and the loading decisions are made, so it may be reasonable to assume the continuous availability of these types of resources. Second, the tool management itself is an important problem which would benefit from separate research. Third, in this paper, the FMS scheduling problem is considered as an extension of the job shop problem. Thus, the materials handling (AGV subsystem) and limited input/output buffer space aspects of an FMS are added to the machining aspects of a job shop in this study.

2.3. Real-time scheduling

Today, as a result of advances in data collection and process technology, scheduling decisions can be made on-line and close to real-time. Therefore, in the recent literature of FMSs, the terms 'on-line scheduling', 'real-time scheduling', and 'dynamic scheduling' are widely used as a substitute for the term 'scheduling'. However, there is, to some extent, a misunderstanding of these terms in relation to on-line and off-line scheduling methods.

Off-line scheduling refers to scheduling all operations of available jobs for the entire scheduling period, whereas, on-line scheduling attempts to schedule operations one at a time when they are needed. On the other hand, real-time scheduling is a short-term decision-making process which generates and updates the schedule based on the current status of the system and the overall system requirements. The definition for real-time scheduling implicitly assumes contingency handling, therefore, it is a broader term than either on-line or off-line. In this context, dynamic scheduling is used to emphasize the dynamic nature of the real-time scheduling problem.

Real-time scheduling can be employed by either an off-line or an on-line method or a combination of methods. If off-line scheduling methods are utilized, the scheduling process becomes scheduling and rescheduling; whereas, according to the on-line scheduling approach, the scheduling decision is made when the state of the system changes, such as job completion, arrival of parts, etc. There are advantages and disadvantages with each. Scheduling can be a very tedious task with off-line methods due to both the difficulty in generating the schedule, and updating it frequently in a dynamic environment. On the other hand, scheduling decisions made by on-line methods may not provide the best results due to a lack of a broad system view. A better

way may be to use a hybrid approach (both off- and on-line) to achieve improved overall system performance.

According to the above classification, the scheduling procedure proposed in this paper can be considered as an on-line algorithm.

3. Relevant literature

There are a number of approaches and proposed solution procedures in the FMS scheduling literature. These can be divided into the following categories:

- (1) development of analytical tools (static scheduling algorithms);
- (2) development of intelligent scheduling systems using expert systems (ES) and/or artificial intelligence (AI) techniques; and
- (3) investigation of the performances of scheduling rules using simulation models and the design of on-line dispatching algorithms.

Analytical approaches formulate the FMS scheduling problem as a constrained optimization model in terms of an explicit objective function and explicit constraints, and then solve the model by using an appropriate solution algorithm (Kusiak and Cyrus 1985, Kusiak 1986, Chang and Sullivan 1984, Nakamura and Shingu 1985, Raman *et al.* 1986, Shanker and Tzen 1985, Hutchinson *et al.* 1989, Dietrich and Escudero 1989). In general, scheduling of the machining subsystem (job shop) and AGV subsystem (vehicle scheduling problem) are both NP-complete. Thus heuristic algorithms are usually proposed for the solution of their scheduling problems. These types of scheduling algorithms are usually of the off-line type and can only be implemented in a rolling horizon scheme in which the whole process resorts to scheduling/rescheduling. Furthermore, most of the analytical formulations available in FMS scheduling literature consider only the scheduling of parts on the machines. Even though some of them include part transfer times, they still assume uninterrupted availability of the materials handling equipment. While this assumption is valid for a conveyorized production system, it is not reasonable for systems which use AGV-based materials handling. Besides, each scheduling algorithm was developed for a different type of FMS under different assumptions and therefore it is very difficult to compare them. Finally, there is little known about their application in a dynamic FMS environment.

The second track, in which numerous publications have emerged in recent years, is applications of ES and AI to scheduling problems of FMSs and other automated manufacturing systems. The main idea behind applications of ES to scheduling problems is that each scheduling system is unique to the given environment and, therefore, a wide variety of technical knowledge and expertise should be taken into account in solving these scheduling problems. As stated by Sauve and Collinot (1987), today's scheduling systems, which are based on optimization techniques, misunderstand the role of the expert scheduler and underestimate the importance of qualitative factors involved in scheduling problems. Therefore scheduling problems are prime candidates for applications of artificial intelligence technology. Among the numerous publications for applications of ES and AI to scheduling of FMSs or other automated manufacturing systems, a reader can refer to the following for more detailed information on this subject: Kusiak (1989), Kusiak and Chen (1988), Park *et al.* (1989).

The third approach to scheduling FMSs is the experimental investigation of FMS scheduling problems using simulation models and the design of easy-to-use on-line dispatching algorithms. Research in this area can be also further classified into: (1)

testing the performance of several machine and AGV scheduling rules against various scheduling criteria, and (2) design of dynamic dispatching algorithms or other on-line scheduling techniques.

Even though most of the industrial scheduling problems are dynamic and stochastic, the majority of the scheduling techniques available in the literature assume static and deterministic conditions (Baker 1974, French 1982). This is partially due to the difficulty in formulating the dynamic problems analytically. Except in relatively simple cases, determination of the optimum schedule by analytical means is extremely difficult due to the combinatorial nature of scheduling problems (King 1979). Because of these reasons, the third approach takes the form of testing scheduling rules (or queue disciplines) and has received ample attention from researchers and practitioners, beginning from the earlier job shop studies (Conway *et al.* 1967) to FMS studies (Stecke and Solberg 1981, Egbelu and Tanchoco 1984, Denzler and Boe 1987, Choi and Malstrom 1988, Montazeri and Wassenhove 1990, Sabuncuoglu and Hommertzheim 1989, 1990 a, b, c). Research in this area is important because the scheduling rules are widely used in practice, ranging from direct use as a scheduling scheme to indirect use as part of knowledge bases of expert systems and static schedule generation schemes. Furthermore, results of these simulation-based experiments help researchers towards a better understanding of the dynamic nature of FMS scheduling problems.

Finally, as a natural extension of simple scheduling rules, the last approach, i.e. development of easy-to-use dispatching algorithms or on-line scheduling heuristics, was taken by many researchers including Egbelu (1987) and Slomp *et al.* (1988). Sometimes, these techniques can be as simple as an augmented scheduling rule. Furthermore, these types of scheduling methods can also be implemented in an ES environment where the decision logic can be embedded in the knowledge base of expert systems so that scheduling decisions can be made on-line in a dynamic manufacturing environment such as FMSs.

In conclusion, there is a wide variety of research activities presented in the literature for FMS scheduling. While the research in each direction is necessary for better understanding and solving the scheduling problems of FMSs, this paper focuses on the development of a dispatching algorithm for on-line scheduling of machines and an AGV-based materials handling system of an FMS.

4. Dispatching algorithm

Sabuncuoglu and Hommertzheim (1989, 1990 a, b) tested various machine and AGV scheduling rules against different scheduling criteria. These rules were simple rules which prioritized the jobs for resources (i.e. machines or AGVs) upon their availability. And by their nature, these rules are very suitable for on-line scheduling implementations. One of the important characteristics of the scheduling rules is that they act as an independent local mechanism in scheduling the jobs. For example, machine scheduling rules do not consider the availability of AGVs when the priorities of jobs are set for any workstation. Similarly, AGV scheduling rules do not directly take into account availability of workstations (or machines) for jobs to be served. Therefore, in implementation, these rules form a dispatching mechanism consisting of two independent sets of rules, one for each type of resource (i.e. machining and AGV subsystems). Thus, because of their myopic nature, a job having the highest priority on the current workstation may wait in the output queue for a long time while a lower priority job in the system is being processed in the next workcentre.

The basic rationale behind the proposed algorithm is to eliminate some of the deficiencies of these scheduling rules and develop a better dispatching mechanism (or algorithm) by considering important interactions between machines and AGVs during the scheduling process. The algorithm proposed in this paper is based on the idea that a job should not be scheduled on a machine (or an AGV) if it will have to wait for an AGV (or a machine) in the next activity. The algorithm uses various priority schemes (or rules) and relevant information concerning the load of the system and the status of jobs in the scheduling process. All this information is organized in hierarchical levels. Since the algorithm has a dispatching nature, it does not produce one complete schedule identifying the expected start and completion times of all operations for all jobs. But rather, it schedules the jobs on a machine or an AGV one at a time as the scheduling decision is needed (or as the status of the system changes). In other words, it is an on-line scheduling algorithm rather than off-line.

4.1. Structure of the dispatching algorithm

In developing the dispatching algorithm a number of assumptions were made. These assumptions are: (a) continuous availability of tools, (b) zero setup time for the operations, and (c) no tooling and pallet/fixture considerations.

The algorithm uses various information in the scheduling process. Some of this information is job related, such as operation times, number of operations, and due-dates. Others are system related and depend upon the status of machines and AGVs, queue levels, the number of parts in the system, current locations of AGVs, etc.

Since the machines and AGVs are two critical resource types in the FMS studied, the algorithm consists of two main parts: (1) a set of procedures to schedule jobs on the machines, and (2) a set of procedures to schedule jobs on AGVs.

Logic associated with scheduling jobs on the AGV

This part of the algorithm is applied whenever an AGV completes its current operation and becomes available for the next task assignment. It consists of the following four hierarchical levels.

Level 1. Push logic (checking the critical stations). It identifies workstations which are either blocked or their respective queues are full. When this happens, not only can the workstation not accept any part from the other workstations, but also may not perform the operation because of blocking. Therefore, one of the outgoing parts at this workstation has to be delivered to its next workstation. This decision is made by using a hierarchical rule. The rule is hierarchical in the sense that different decision criteria are applied sequentially to identify the most appropriate part and the machine to be served. That is, at the top level of the hierarchy one decision criterion is applied. If there is a tie, the next decision criterion is used to break the tie. In the case where there is still a tie at the bottom level, one of the workstations and a part is selected on a FCFS (first come first served) basis. These five decision criteria are as follows:

- (1) A part at a critical workstation which has finished all its operations but is waiting to be delivered to the unload station, will obtain the transportation service first.
- (2) A part at the most demanded critical workstation (i.e. total number of parts waiting at the different workstations or receiving station to be delivered to the blocked workstation) will be serviced first.

- (3) A part which has the smallest queue level at the next workstation has the highest priority.
- (4) A part which is closest to the current location of idle AGV has the highest priority (i.e. shortest distance travelled rule).
- (5) A part with the least amount of work remaining or the earliest due-date has the highest priority to obtain service from the AGV depending upon the scheduling criterion employed (i.e. mean flow-time, mean tardiness, etc.).

As can be seen, the first decision criterion ensures that no part is held in the system unnecessarily if it has already completed its operations. The second criterion identifies the most demanded workstation. Other criteria determine part priorities based on either job-related or system-related information. At each level described above, availability of the next workstation is also checked. Therefore, if queue spaces of the destination stations are full for all parts at the blocked machines, a part at one of the most demanded workstations is taken to the nearest central buffer area. Since the parts have to be delivered from the congested workstation to other workstations or a central buffer area, parts are pushed in the system. Therefore, the dispatching strategy can be considered to be 'push' type.

Level 2. Buffer logic (checking the parts in the central buffer area). If there are some parts in the central buffers, a part with the most available destination queue space is served first. In the case of a tie, a part which has the least amount of work remaining or the earliest due-date is selected for the mean flow-time and the due-date criterion, respectively.

Level 3. Pull logic (checking the idle workstations). As the third step, the algorithm checks to see if there are any idle workstations at the present time. If there are some idle machines, then it searches other workstation queues to locate a workstation which can immediately deliver a part to this idle workstation. In the case where there is more than one idle workstation and more than one station which can deliver the parts to one of these stations, the following hierarchical decision criteria is employed to schedule the AGV for the next journey:

- (1) A workstation which is nearest to the current location of the idle AGV has the highest priority (or SDT rule).
- (2) A part which has the least amount of work remaining or earliest due-date has the highest priority to obtain service from the AGV.

Since the part transfer is derived by the demand of an idle workstation, parts are pulled in the system. Therefore, the dispatching strategy can be considered to be a 'pull' type.

Level 4. Push-pull logic (identification of the most appropriate workstation and part to be serviced). If the central buffer areas are empty and there are no critical or idle stations then the system can be considered to be stable. In these circumstances, the objective is to find a part which has the highest chance of being processed earliest at the next workstation. As was discussed earlier, servicing a part which has a longer expected waiting time at the next workstation reduces the system efficiency. Therefore, at this step of the algorithm, parts waiting for one or more AGVs, are evaluated based on their expected waiting time at their next workstations. In the implementation of this step, the algorithm works like a machine scheduling routine and calculates the expected waiting time using the priority scheme employed in scheduling of the machines. In conclusion, a part which has the lowest expected waiting time is selected to be served by the AGV.

Logic associated with scheduling jobs on the machine

This part of the algorithm is employed whenever a workstation completes processing a part and becomes available for other parts in the queue. It computes the priority index for each candidate job waiting in the workstation queue. For mean flow-time criterion, the priority index is a function of the operation time and the expected waiting time of the job on the next workstation. Both variables are also weighted with respect to a congestion level (or factor) as shown below:

$$\text{priority index} = \text{weight}_1 * \text{operation time} + \text{weight}_2 * \text{expected waiting time on the next operation} \quad (1)$$

The congestion level at a particular workstation is expressed as the ratio of the number of incoming parts to the maximum queue capacity available at the respective workstation. These weights are not constant but vary as the status of the system changes. As can be noted in equation (1), the priority index degenerates to the SPT (shortest processing time) rule if the second weight and/or the expected waiting time is zero. For the due-date based scheduling criteria, MOD (modified operation due-date) is used as a substitute for the operation time in equation (1). After the priority index is calculated for each part, the part with the smallest priority index value is loaded on the machine first. Formalization of the algorithm is given in the Appendix.

5. Performance of the algorithm and comparisons with machine and AGV scheduling rules

In this section, relative performance of the scheduling algorithm will be measured against the different scheduling criteria. Comparison of the algorithm with the machine and AGV scheduling rules will also be presented. But, the system considerations will be discussed first.

The system under study is a hypothetical FMS (Fig. 1). This system has been used in previous studies to test machine and AGV scheduling rules (Sabuncuoglu and Himmertzhaim 1989, 1990 a, b, c). In this system, there are eight workstations and one input/output carousel. There are also two central buffer areas at which parts are

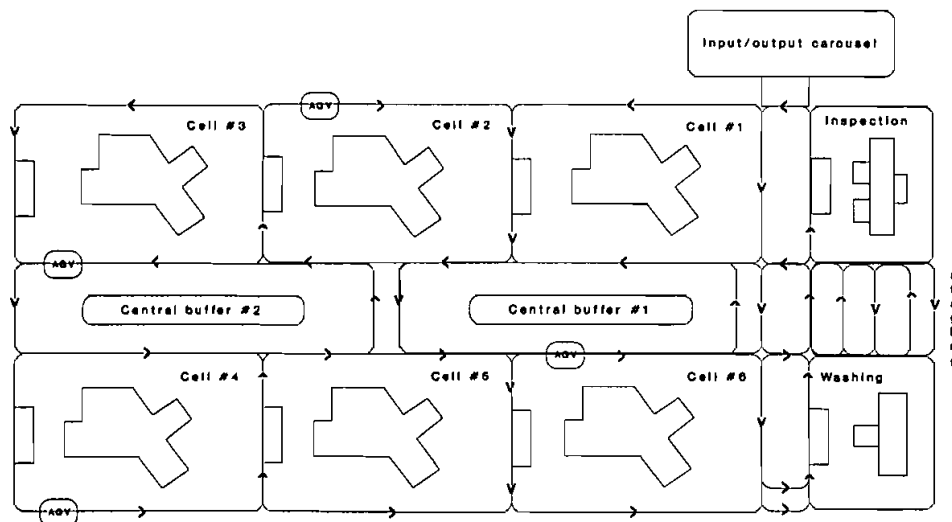


Figure 1. Schematic view of an FMS.

temporarily stored to prevent system locking. Materials and parts are transferred in the system by an AGV system. Each AGV moves a part between the workcentres along a predetermined and unidirectional path. Upon the completion of a part transfer, an idle AGV stays at the destination station. This policy was implemented based on pilot simulation runs.

A simulation model of the FMS described above was developed using SIMAN (Pegden 1985). Data for the simulation runs are generated by a FORTRAN subprogram. In general, job interarrival time is exponentially distributed. Each job is processed by a series of workcentres. Number of operations (number of machines to visit) is determined by a discrete uniform distribution between one and six. Machine assignment is random and no job is allowed to visit the same machine more than once. As a result, machine loads are kept equal. Besides the workcentres, all jobs visit the washing station. However, only 50% of the jobs are processed by the inspection station. The tooling system is not modelled and an infinite number of pallets are assumed. Buffer capacity at each workcentre is limited and equal. Thus, a part completing a current operation waits in the queue at the workcentre until the next station queue and an AGV are available. There are no capacity restrictions on the central buffer area and the input/output carousel. An AGV transfers only one part at a time (i.e. unit load is one). At intersections in the AGV path network (Fig. 1) an AGV moving a part has priority over other AGVs travelling empty. In case of a tie, the right of passing at the intersection is determined on a FCFS basis. Finally, two AGVs are employed in the current model.

The proposed algorithm was tested under various experimental conditions. These are as follows:

- varying machine load levels;
- different queue capacities;
- varying levels of due-date allowances;
- different scheduling criteria;
- different types of processing time distributions and their parameters (i.e. exponential and normal distributions).

As stated above, the proposed algorithm was tested against different scheduling criteria. Baker (1984) states that two measures are of primary interest: flow-time and meeting due-dates. Mean flow-time (or average time in the system) is a good indicator to measure the system responsiveness. It also measures the average work-in-process level in the system. Knowing that flexibility is one of the great assets of flexible manufacturing systems, the degree of responsiveness in production (or flow-time) is the most important factor to assess the degree of flexibility built into the FMS. On the other hand, every manufacturing system has due-dates associated with its orders. Thus, meeting due-dates is also very important for the company. In this study, mean flow-time and mean tardiness criteria are used as the primary measures.

For the operation time distribution, both normal and exponential distributions are tested separately. But only positive values were considered for the normally distributed operation times.

5.1. Testing performance of the algorithm under different machine load levels using mean flow-time measure

Mean flow-time performances of the proposed scheduling algorithm under varying machine load levels are depicted in Fig. 2 along with the results of the scheduling rule

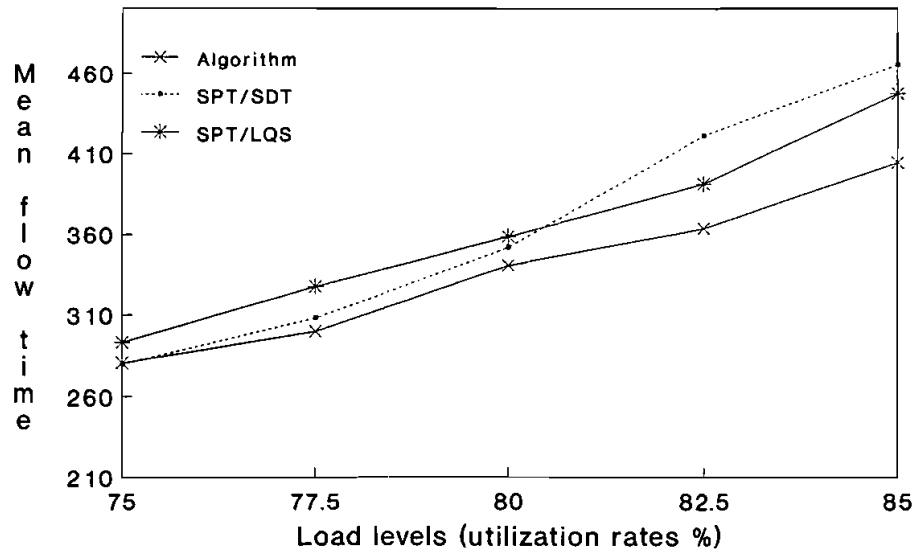


Figure 2. Mean flow-time performance of the proposed algorithm *versus* SPT/SDT and SPT/LQS (exponential case).

combinations, SPT/LQS (shortest processing time/largest queue size) and SPT/SDT (shortest processing time/shortest distance travelled). These two machine and AGV scheduling rule combinations were found to be the best rule combinations against the mean flow-time criterion in the previous studies (Sabuncuoglu and Hommertzheim 1990 a). These results were obtained using exponential processing times. Machine load levels were determined by varying the mean processing time.

In general, the proposed scheduling algorithm improved the mean flow time as the machine loads were increased. At low machine utilizations, there was no difference between algorithm and scheduling rules. But, as the system load was increased (or the congestion in the system increased), the algorithm performed better than these two scheduling rule combinations. For example, at 85% machine utilization rate, approximately 12 and 15% reductions in the mean flow-time were achieved by the algorithm over the SPT/LQS and SPT/SDT rule combinations, respectively. As far as the performances of two AGV scheduling rules are concerned, LQS began to show better performance than SDT when the machine load increased (i.e. above 80% utilization level). Because, in a system where there are limited queue capacities, buffer spaces at workstations became the scarce resources as the load was increased, eventually the LQS rule which was very sensitive to the queue levels began showing better performance than SDT. As shown in Fig. 3, similar mean flow-time improvements were obtained when normally distributed processing times were used. These results indicate that improved scheduling of machines and AGVs can result in considerable mean flow-time advantages in FMSs.

In fact, at the beginning of the simulation experiments, it was expected that the proposed scheduling algorithm would achieve much higher mean flow-time reductions. However, as shown in Figs 2 and 3, its performance is not much different to these scheduling rules for low utilization levels. This can be explained as follows:

- (1) As observed in many real-life scheduling problems, the scheduling rules are relatively robust.

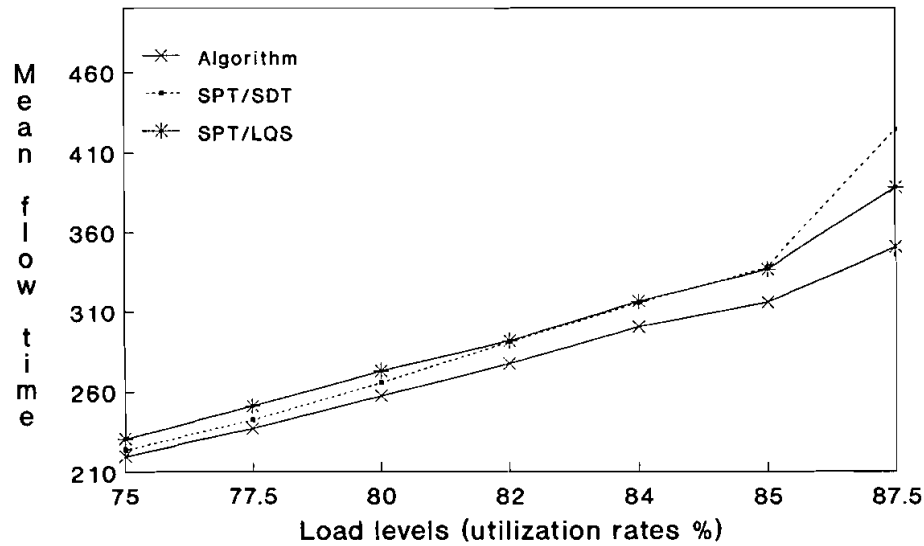
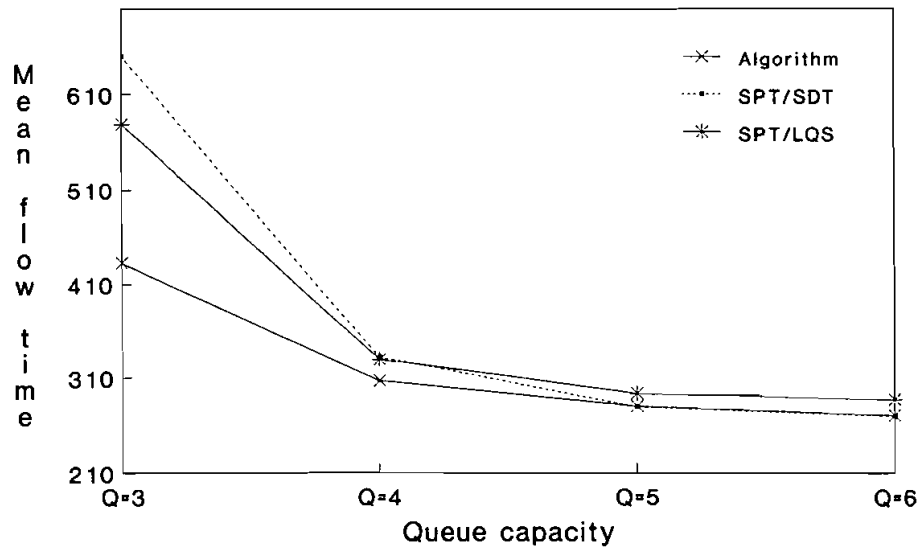


Figure 3. Mean flow-time performance of the proposed algorithm *versus* SPT/SDT and SPT/LQS (normal distribution case).



• Exponential distribution is used for processing times

Figure 4. Mean flow-time performance of the proposed algorithm *versus* SPT/SDT and SPT/LQS at varying queue capacities (with exponential distribution).

- (2) At low utilization levels, there are comparatively less jobs in the queues. Therefore, the chance of improving the system performance is less due to the fewer opportunities for scheduling decisions.
- (3) In the simulation model which had scheduling rules embedded, there were already some complicated logic or decision routines to prevent blocking and to increase the system performance. Therefore, the enhanced features of the scheduling rules resulted in close performances to the dynamic scheduling algorithm at low utilization levels. If they were used alone without any enhancement, the system might have been saturated with these rules.

5.2. Testing the performance of the algorithm under varying queue capacities using the mean flow-time measure

Mean flow-time performance of the proposed scheduling algorithm under varying queue sizes was also studied. As shown in Fig. 4, the scheduling algorithm outperformed the scheduling rules as the capacity of queue spaces at workstations were reduced. In particular, when the queue capacity was three, the algorithm accomplished more than a 26% mean flow-time reduction. Similar results were also obtained for the normally distributed processing times (Fig. 5). These improved performances of the scheduling algorithm at the low queue sizes further emphasize the importance of scheduling decisions in short queue length systems such as FMSs.

5.3. Testing the performance of the algorithm under varying due-date tightness against the mean tardiness criterion

Due-date performance of the algorithm was measured against the mean tardiness criterion under different experimental conditions. Since the due-date allowance (or due-date tightness) is a major factor in comparing the rules, simulation experiments were repeated at varying levels of the due-date tightnesses. Sabuncuoglu and Hommertzheim (1990 b,c) found MOS/SDT (modified operation due-date/shortest distance travelled) and MOD/LQS (modified operation due-date/largest queue size) to be the best machine and AGV scheduling rule combinations for the due-date criterion. Therefore, their performances were also included in this analysis. Results are presented only for exponentially distributed processing times.

In experiments, due-dates of jobs were determined by allocating flow allowances to a job to perform its various activities (i.e. processing, materials handling, waiting, etc.). This type of due-date assignment is also known as the endogenous due-date assignment. Sabuncuoglu and Hommertzheim (1990 b,c) found TWK (total work

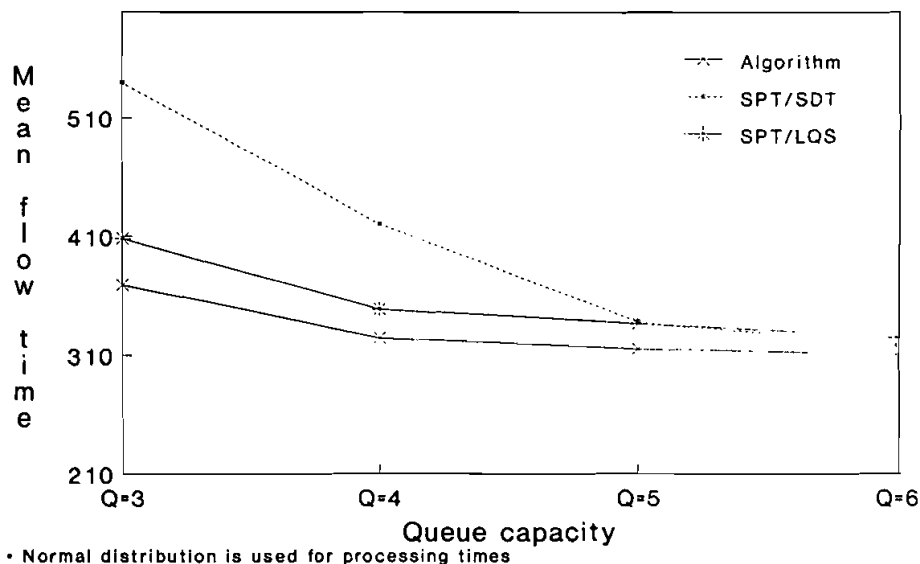
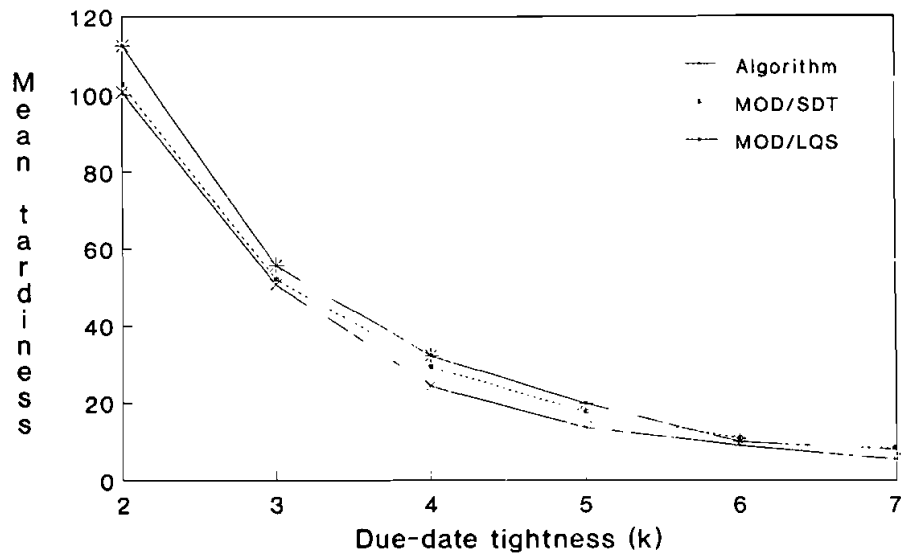


Figure 5. Mean flow-time performance of the proposed algorithm versus SPT/SDT and SPT/LQS at varying queue capacities (with normal distribution).



• Machine load level is 75% and $Q=5$

Figure 6. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS at varying due-date tightnesses.

content) rule to be the best rule. Both its job and operation-based due-date assignment versions given below were used to set the due-dates.

Variable definitions:

$$D_i = R_i + A_i$$

$$d_{i,j} = d_{i,j-1} + p_{i,j}(A_i/P_j) \text{ where,}$$

$$A_i = k * P_j \text{ (flow allowance)}$$

R_i ready time or arrival time of job i

D_i due-date of job i

$d_{i,j}$ due-date of job i for operation j

P_j total remaining operation time of job i at its j th operation

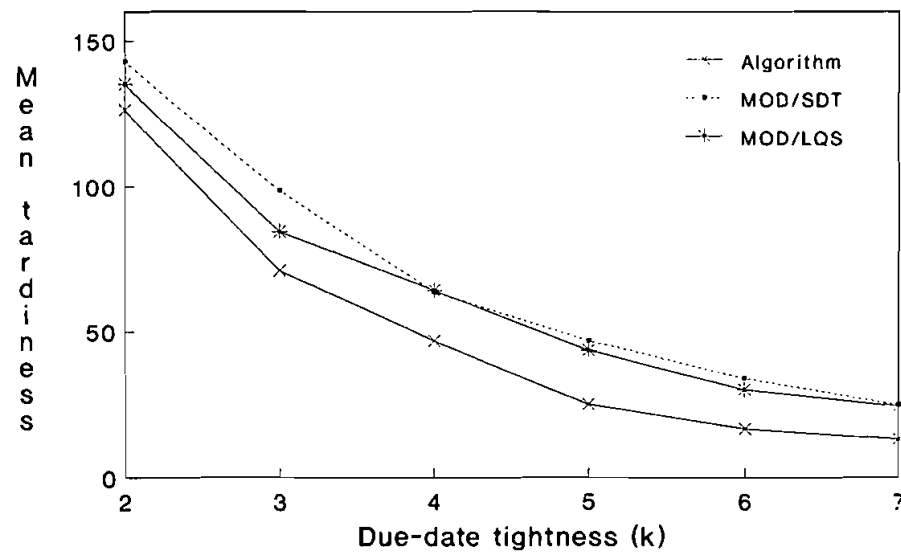
$p_{i,j}$ operation time for j th operation of job i

k due-date tightness parameter

Figure 6 shows mean tardiness performances of the scheduling algorithm together with the MOD/SDT and MOD/LQS rules under the experimental conditions where average machine and AGV utilizations are both 75%. As can be seen, differences between the algorithm and rules are not very apparent even though the algorithm always produced lower tardiness. However, as depicted in Fig. 7, the difference is accentuated when the machine load levels were increased to 80%. At this load level, the average mean flow-time reduction achieved by the algorithm was at least 25% over the varying values of the tightness parameter.

Figure 8 also shows the behaviour of the algorithm and rules at varying levels of machine loads when the due-date tightness parameter is four (i.e. due-date of a job is at least four times greater than total job processing time). Again, the proposed algorithm outperformed the scheduling rule combinations at varying machine load levels.

Relative performance of the algorithm and the rules were also measured at varying levels of queue capacities. As can be seen in Fig. 9, the algorithm performed slightly better than the scheduling rules when queue capacities were four. However, it achieved



* Queue capacity is four (Q=4)

Figure 7. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS at varying due-date tightnesses.

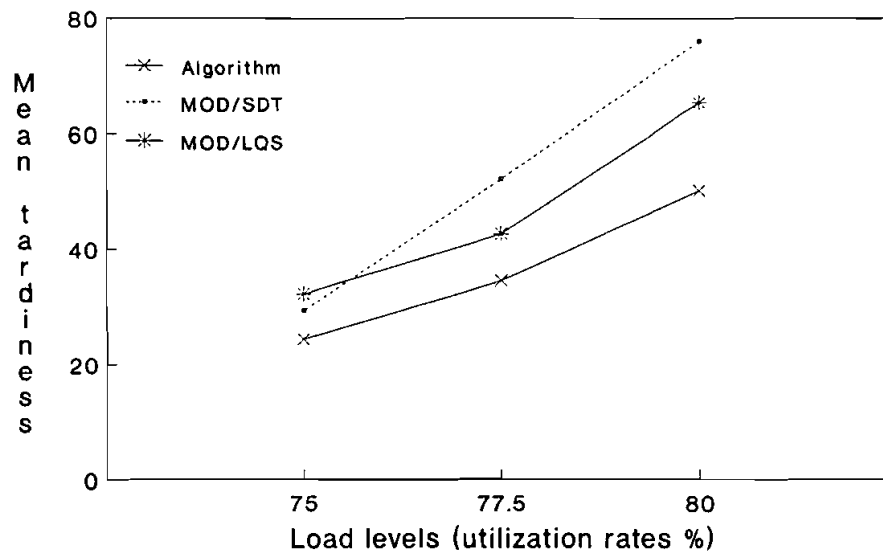
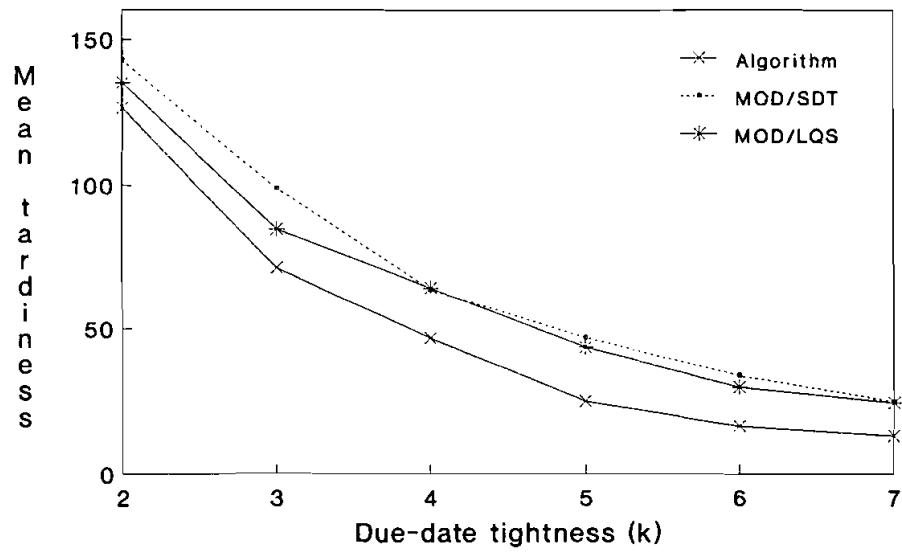


Figure 8. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS under varying machine load levels (due-date tightness parameter (k) is four).

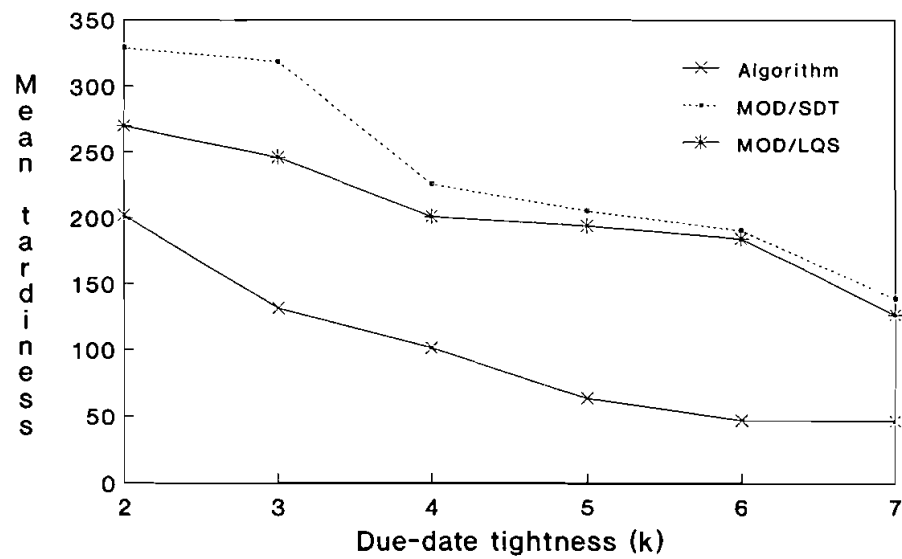
significant reductions in mean tardiness as the queue capacities were reduced from four to three at varying due-date tightnesses (Fig. 10). In particular, as depicted in Fig. 10, the reduction was more than 50% when the queue capacity was three and tightness parameter, four. As can be noted, the range of mean tardiness in Fig. 9 was taken from 0 to 150 to depict these slight differences between the algorithm and scheduling rules for the purpose of showing that the algorithm is at least as good as the scheduling rules.

Figure 11 also shows the superior performance of the proposed algorithm over the scheduling rules when the queue capacity was reduced from five to three.



• Queue capacity is four ($Q=4$)

Figure 9. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS at varying due-date tightness (queue capacity (Q) is four).



• Queue capacity is three ($Q=3$)

Figure 10. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS at varying due-date tightness (queue capacity (Q) is three).

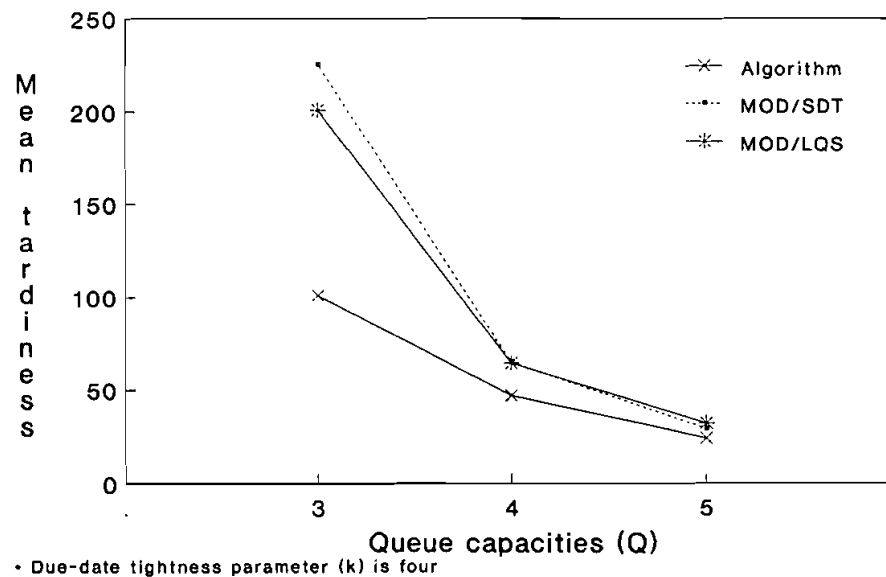


Figure 11. Mean tardiness performance of the proposed algorithm *versus* MOD/SDT and MOD/LQS under varying queue capacities (due-date tightness parameter (k) is four).

6. Concluding remarks

In this paper, scheduling problems of flexible manufacturing systems were analysed. The problem was viewed basically as a class of the dynamic job-shop problem and an on-line dispatching algorithm was developed. Proposed scheduling algorithm was compared with existing scheduling rules against different scheduling criteria. In general, the dynamic scheduling algorithm performed better than the standard scheduling rules. It improved the system performance significantly under high utilization rates or load levels. Its impact on reducing the mean system flow-time was greatest when the queue capacities were low. It also outperformed the scheduling rules when the mean tardiness was the scheduling criterion. Results presented in this paper should be interpreted with reference to the hypothetical FMS and the experimental conditions described earlier. Thus, there is a need for further research to test the algorithm under different FMS configurations and experimental conditions. These could include the consideration of tooling and limiting the number of pallets in the system.

In Section 2, on- and off-line scheduling were discussed in conjunction with real-time scheduling. Results presented in this research are based on the on-line scheduling approach (i.e. scheduling activities one at a time when they are needed). As mentioned earlier, however, there are some benefits of using off-line scheduling. Another future research effort could be the development of an off-line scheduling algorithm and testing its performance relative to the on-line algorithm.

Acknowledgment

The authors wish to thank the referees for their detailed comments and constructive criticisms of the initial draft.

References

- BAKER, K. R., 1974, *Introduction to Sequencing and Scheduling* (New York: John Wiley).
- BAKER, K. R., 1984, Sequencing rules and due-date assignments in a job shop. *Management Science*, **30** (9), 1093–1104.
- CHANG, Y. L., and SULLIVAN, R. S., 1984, Real-time scheduling of FMS. Paper presented at TIMS/ORSA meeting, San Francisco, May, 1984.
- CHOI, R. H., and MALSTROM, E. M., 1988, Evaluation of traditional work scheduling rules in a flexible manufacturing system with a physical simulator. *Journal of Manufacturing Systems*, **7** (1), 33–45.
- CONWAY, R. W., 1963, Some tactical problems in digital simulation. *Management Science*, **10** (1), 47–61.
- CONWAY, R. W., MAXWELL, W. L., and MILLER, L. W., 1967, *Theory of Scheduling* (Reading, Mass.: Addison-Wesley).
- DENZLER, D. R., and BOE, W. J., 1987, Experimental investigation of flexible manufacturing system scheduling rules. *International Journal of Production Research*, **25** (7), 979–994.
- DIETRICH, B. L., and ESCUDERO, L. F., 1989, On solving a 0–1 model for workload allocation on parallel unrelated machines with set-ups. In *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 181–186.
- EGBELU, P. J., 1987, Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system. *Journal of Manufacturing Systems*, **6** (3), 209–221.
- EGBELU, P. J., and TANCHOCO, J. M. A., 1984, Characterization of automated guided vehicle dispatching rules. *International Journal of Production Research*, **22** (3), 359–374.
- FRENCH, S., 1982, *Sequencing and Scheduling* (New York: John Wiley).
- GROOVER, M. P., 1987, *Automation, Production Systems and Computer Integrated Manufacturing* (New York: Prentice-Hall).
- HUTCHINSON, J., LEONG, K., SYNDER, D., and WARD, F., 1989, Scheduling for random job shop flexible manufacturing system. In *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 161–166.
- LOOVEREN, A. J. V., GELDERS, L. F., and WASSENHOVE, L. N. V., 1986, A review of FMS planning problems. In *Modelling and Design of Flexible Manufacturing Systems*, edited by A. Kusiak (Amsterdam: Elsevier), pp. 3–31.
- KING, J. R., 1979, Scheduling and the problem of computational complexity. *Omega*, **7** (3), 233–240.
- KUSIAK, A., 1986, Scheduling flexible machining and assembly system. In *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 521–526.
- KUSIAK, A., 1989, Scheduling automated manufacturing systems: Knowledge-based approach. In *Proceedings of the Third ORSA/TIME Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 377–382.
- KUSIAK, A., and CHEN, M., 1988, Expert systems for planning and scheduling manufacturing systems. *European Journal of Operations Research*, **34** (3), 113–130.
- KUSIAK, A., and CYRUS, P., 1985, Routing and scheduling of automated guided vehicles. In *Towards the Factory of Future*, edited by H. J. Bullinger and H. J. Warnecke (Berlin: Springer-Verlag), pp. 247–251.
- MONTAZERI, M., and WASSENHOVE, L. N. V., 1990, Analysis of scheduling rules for an FMS. *International Journal of Production Research*, **28** (4), 785–802.
- NAKAMURA, N., and SHINGU, T., 1985, Scheduling of flexible manufacturing systems. In *Towards the Factory of Future*, edited by H. J. Bullinger and H. J. Warnecke (Berlin: Springer-Verlag), pp. 147–152.
- PARK, S. C., RAMAN, N., and SHAW, M. J., 1989, Heuristic learning for pattern directed scheduling in a flexible manufacturing system. In *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 369–376.
- PEGDEN, C. D., 1985, *Introduction to SIMAN* (State College, Pennsylvania: Systems Modeling Corporation).

- RAMAN, N., TALBOT, F. B., and RACHAMADUGU, R. V., 1986, Simultaneous scheduling of machines and material handling devices in automated manufacturing. In *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 321–332.
- SABUNCUOĞLU, I., and HOMMERTZHEIM, D., 1989, An investigation of machine and AGV scheduling rules in an FMS. In *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K. E. Stecke and R. Suri (Amsterdam: Elsevier), pp. 261–266.
- SABUNCUOĞLU, I., and HOMMERTZHEIM, D., 1990 a, Experimental investigation of the FMS scheduling problem: evaluation machine and AGV scheduling rules against the mean flow-time criterion. Working paper: 90-01, Industrial Engineering Department, The Wichita State University.
- SABUNCUOĞLU, I., and HOMMERTZHEIM, D., 1990 b, Experimental investigation of the FMS due-date scheduling problem: evaluation of due-date assignment rules. Working paper: 90-02, Industrial Engineering Department, The Wichita State University.
- SABUNCUOĞLU, I., and HOMMERTZHEIM, D., 1990 c, Experimental investigation of the FMS due-date scheduling problem: evaluation of machine and AGV scheduling rules. Working paper: 90-03, Industrial Engineering Department, The Wichita State University.
- SAUVE, B., and COLLINOT, A., 1987, An expert system for scheduling in a flexible manufacturing system. *Robotics and Computer-Integrated Manufacturing*, **3** (2), 227–233.
- SHANKER, K., and TZEN, Y. J., 1985, A loading and dispatching problem in a random flexible manufacturing system. *International Journal of Production Research*, **23** (5), 579–595.
- SLOMP, J., GAALMAN, G. J. G., and NAWIJN, W. M., 1988, Quasi on-line scheduling procedures for flexible manufacturing systems. *International Journal of Production Research*, **26** (4), 585–598.
- STECKE, K. E., and SOLBERG, J., 1981, Loading and control policies for a flexible manufacturing system. *International Journal of Production Research*, **19** (5), 481–490.

Appendix: Formalization of the algorithm

In developing the algorithm, the following variables and parameters are used:

- N number of workstations
- J number of parts in the system
- i workstation identifier, $i = 1, \dots, N$
- j part (or job) identifier, $j = 1, \dots, J$
- t time at which the scheduling decision is made
- $P_j(t)$ priority index of part j at time t
- LO_j location type of part j in the system ($L_j = 1$ if it is in a central buffer area, otherwise, $L_j = 0$)
- Q_i total queue capacity of workstation i
- q_i input queue capacity of workstation i
- $IQ_i(t)$ input queue level of workstation i at time t
- $OQ_i(t)$ output queue level of workstation i at time t
- $A_i(t)$ number of parts scheduled to arrive at workstation i
- $U_i(t)$ operational status of workstation i at time t ($U_i(t) = 1$ if it is busy. Otherwise, $U_i(t) = 0$)
- $B_i(t)$ blocking status of workstation i at time t ($B_i(t) = 1$ if it is blocked. Otherwise, $B_i(t) = 0$)
- $OP_{i,j}$ operation time of the part j at workstation i
- $OD_{i,j}$ operation due-date of the part j at the workstation i
- $EWAT_{i,j}(t)$ expected waiting time of job j for the next operation at workstation i in time t

- $NR_j(t)$ remaining number of operations of the part j at time t
 $D_{k,l}(t)$ number of parts waiting at workstation k to be delivered to workstation l at time t
 $C_p(t)$ status of the central buffer p at time t ($C_p(t)=1$ if there are some parts, otherwise, $C_p(t)=0$)
 $CT(t)$ total number of parts in central buffer areas at time t
 E event type ($E=1$ if AFV scheduling, otherwise, $E=2$ for machine scheduling)
 S_1 set which consists of the parts demanding AGV transportation service
 S_2 set which consists of workstations that are either blocked or their input queues are full
 S_3 subset of S_2 which consists of all jobs j such that jobs at the blocked workstations have already completed their last operations
 S_4 set of jobs (parts) that are in the central buffer areas and their destination stations have available queue spaces
 S_5 set which consists of idle workstations
 S_6 set consisting of workstation which can send some parts to workstations in the set S_5
 V_i set which consists of the parts demanding machining operations at the work station i
 $X(t)$ part number to be serviced at time t
 $Y(t)$ workstation number to be serviced at time t
 $Z(t)$ AGV number to be selected at time t

Dispatching algorithm steps for scheduling machines and AGVS are as follows.

Step 1. Identify the event type. If E is equal to 1 (i.e. an AGV is completing the current assignment) then go to Step 2. Otherwise, if E is equal to 2 (i.e. a part is completing its current operation) then go to Step 9.

Step 2. Search set S_1 to see if there is a job demanding a transportation service. If the set S_1 is empty then stop. Otherwise, go to Step 3.

Step 3. Construct set S_2 as all workstations that are either blocked or their input queues are full

$$S_2 = \{i | B_i(t) = 1 \text{ or } IQ_i(t) + OQ_i(t) + A_i(t) = q_i\}$$

go to Step 4.

Step 4. If set S_2 is empty go to Step 5, else, go to Step 4.1.

Step 4.1. Construct the set S_3 as all workstations i and all jobs j such that jobs at the blocked workstations have already completed their last operation

$$S_3 = \{(i, j) | i \in S_2, NR_j(t) = 0\}$$

If set S_3 is empty then go to Step 4.2. However, if S_3 is not empty and there is only one job in the set then set $X(t)=j$, $Y(t)=i$, and go to Step 4.5. Otherwise, set $S_2=S_3$ and go to Step 4.2.

Step 4.2. Reduce S_2 to the workstations with the highest demand

$$\sum_{k=1}^N D_{k,n}(t) \geq \sum_{k=1}^N D_{k,l}(t) \text{ where } n, l \in \{1, \dots, N\}, k \leq l$$

If there is only one workstation with the highest demand, then set $Y(t)=n$, select the related job and go to Step 4.5. If more than one job could be selected, go to Step 4.4. If there are more workstations with the highest demand, go to Step 4.3.

Step 4.3. Reduce S_2 to the workstations which are closest to the location of the AGV (i.e. apply the shortest travel distance (SDT) rule). If there is only one closest workstation n and one job j then set $Y(t)=n$, $X(t)=j$ and go to Step 4.5. Otherwise, go to Step 4.4.

Step 4.4. Search set S_2 to find the job j with either the least amount of work remaining (LWKR) or the earliest due-date. Set $X(t)=j$ (if there is a tie, select the jobs arbitrarily). If $Y(t)$ is not determined in the previous steps set the value of $Y(t)$ equal to the current location of job j .

Step 4.5. If the destination station of the selected job is full, then move the job to the closest central buffer. Update $C_p(t)$ and go to Step 8.

Step 5. Check the central buffers to determine if there are any jobs waiting to go to a workstation for their next operation. If $CT > 0$, then go to Step 5.1. Otherwise, go to Step 6.

Step 5.1. Construct set S_4 as all jobs that are in the central buffer and whose destination stations have available queue spaces

$$S_4 = \{(i, j) | LO_j = 1 \text{ and } IQ_i(t) + A_i(t) < q_i\}$$

If set S_4 is empty, go to Step 6. Otherwise, find the job j in the central buffer m which has the most destination queue space available. Set $X(t)=j$, $Y(t)=m$ and go to Step 8. If more than one job could be selected and they are in the same central buffer area then set $Y(t)=m$ and go to Step 5.3. Otherwise, go to Step 5.2.

Step 5.2. Reduce S_4 to the jobs to find a job j in the central buffer m which is closest to the location of the AGV (i.e. the SDT rule). If there is only one job then set $X(t)=j$, $Y(t)=m$, and go to Step 8. Otherwise, go to Step 5.3.

Step 5.3. Search set S_4 to find a job j in the central buffer m with the least amount of work remaining (LWKR) or earliest due-date. If there is a tie, select the job arbitrarily and set $X(t)=j$. If $Y(t)$ is not determined in the previous steps set the value of $Y(t)$ equal to the current central buffer location m of job j . Go to Step 8.

Step 6. Construct set S_5 as all workstations that are idle and have no jobs scheduled to be processed

$$S_5 = \{i | U_i(t) = 0 \text{ or } IQ_i(t) + A_i(t) = 0\}$$

If set S_5 is empty, go to Step 7. Otherwise, go to Step 6.1.

Step 6.1. Determine set S_6 from all workstations that can immediately send jobs to the idle workstations identified in the set S_5

$$S_6 = \{(k, l) | OQ_k(t) > 0 \text{ and } D_{k,l}(t) > 0 \text{ and } U_l(t) + A_l(t) = 0\}$$

where $k, l = 1, \dots, N$ and $k \neq l$. If set S_6 is empty, then go to Step 7, otherwise, go to Step 6.2.

Step 6.2. Find the source station m with the greatest number of jobs in it (or find the most congested station which can send a job to the idle station)

$$IQ_m(t) + OQ_m(t) > IQ_i(t) + OQ_i(t)$$

for all workstations where $(m, i) \in S_6$ and $m \neq i$. If there is a tie or more than one job in the selected source workstation, then go to Step 6.3. Ptherwise, set $X(t)=j$, $Y(t)=m$, and go to Step 8.

Step 6.3. Search set S_6 to find a job j in workstation i which is closest to the location of the AGV. If there is only one closest workstation and one outgoing job at this station then set $X(t)=j$, $Y(t)=i$, and go to Step 8. Otherwise, go to Step 6.4.

Step 6.4. Search set S_6 to find a job j at workstation i with the least amount of work remaining or the earliest due-date. If there is a tie, select the job and the central buffer arbitrarily. Set $X(t)=j$, $Y(t)=i$, and go to Step 8.

Step 7. Use set S_1 and calculate the expected waiting time of job j currently at the workcentre i waiting to be delivered to the workstation m .

$$EWAT_{i,j}(t) = \sum_{w \in W} OP_{m,w} \quad (2)$$

If the scheduling criteria is the mean flow-time, W is a set of jobs at workstation i with smaller operation times than job j . That is,

$$W = \{w | OP_{m,w} < OP_{m,j}\} \quad (3)$$

However, if the due-date based scheduling criteria is employed, then W is a set of jobs at the workstation i with smaller modified operation due-dates than job j . That is,

$$W = \{w | MOD_{m,w} < MOD_{m,j}\} \quad (4)$$

where $MOD_{m,j} = \max \{OD_{m,j}, t + OP_{m,j}\}$.

Find the job with the smallest waiting time on its next operation ($X(t)=j$ and $Y(t)=i$) and go to Step 8. If there is a tie, use SDT and LWKR or (EDD) rules to break the tie.

Step 8. Change the status of AGV from idle to busy and send to the workstation $Y(t)$ and the job $X(t)$ selected.

Step 9. Check if there are AGVs available to transfer the job to the next workstation. If there are AGVs then select the one closest to the current workstation. Change the status of AGV number $Z(t)$ from idle to busy and send to the workstation $Y(t)$ and the job $X(t)$ demanding the service. If there is no AGV available, update $OQ_i(t)$ and the set S_1 . Also search the set V_i if there is any job that can be processed by the current workstation i . If set V_i is empty, stop. Otherwise, go to Step 10.

Step 10. Determine the priority index of each job j at the current workstation i by considering the waiting time at the next workstation m . Priority index ($P_j(t)$) is calculated depending on the scheduling criteria. For the mean flow-time criteria, use the following equation:

$$P_j(t) = a_1 * OP_{i,j} + a_2 * EWAT_{m,j}(t) \quad (5)$$

where, $a_1 = IQ_i(t)/Q_i$, $a_2 = IQ_m(t)/Q_m$ and i and m are current and destination workstation indices respectively.

For the due-date based scheduling criteria, use the following equation:

$$P_j(t) = a_1 * MOD_{i,j} + a_2 * EWAT_{m,j}(t) \quad (6)$$

where, a_1 and a_2 are defined above.

Step 11. Select the job j which has the smallest priority index from the set V_i and stop.