Multiple-objective Scheduling for the Hierarchical Control of Flexible Manufacturing Cells

(Revised December 1997)

Ling-Feng Tung, Li Lin, Rakesh Nagi
Department of Industrial Engineering, 342 Bell Hall,
State University of New York at Buffalo
Buffalo, NY 14260

ABSTRACT

This paper presents a hierarchical scheduling approach for flexible manufacturing systems that pursues multiple performance objectives and includes process flexibility by taking advantage of alternative process plans and alternative resources for operations. The scheduling problem is considered at two levels: the shop level, and the cell level. The shop level controller employs a combined priority rule to rank shop orders considering multiple scheduling objectives. To overcome dimensional complexity and keep a low level of work-in-process inventory, the shop controller selects up to three of the highest ranking orders as candidates and develops all possible release sequences for them, with or without multitasking. These sequences are conveyed to the cell controller, which performs detailed scheduling using a fixed priority heuristic for routing multiple part types while considering alternative process plans and alternative resources for operations. The cell controller reports the detailed schedule with projected order completion times back to the shop controller. Upon receiving these results from the cell controller, the shop controller evaluates each schedule using a multiple-objective function and selects the best schedule. By applying this hierarchical scheduling approach to the shop and cell levels of the control, multiple performance objectives of an FMS can be achieved in an integrated manner.

Key Words: flexible manufacturing systems, flexible manufacturing cells, hierarchical control, scheduling, multiple objectives.

1. Introduction

Facing increasing manufacturing competition in today's global market, many companies realize that flexible manufacturing systems (FMSs) provide significant advantages, including reduced work-in-process inventory, reduced throughput time, improved quality, and increased machine utilization. Many research topics in FMSs have been investigated: machine grouping, order scheduling, operations sequencing, system layout, resource allocation (automated guided vehicles or tools), economic consideration, and handling of unexpected events. This research mainly aims at accomplishing one of the most important goals: increasing FMS efficiency.

For many firms, improving FMS flexibility is equally desirable. It is well known that FMS flexibility depends not only on flexibility of hardware, e.g., machines and robots, but also on the flexibility of software, e.g., scheduling strategies. This is important because some systems lack the flexibility to be significantly modified after installation (Larin, 1989) and there is a need to allow a human scheduler to influence the final schedule and to choose scheduling rules according to particular applications (Montazeri and Van Wassenhove, 1990; Grabot and Geneste, 1994).

Much research treats FMS scheduling as a single-level problem for machines in a shop, even though an FMS is a hierarchical system with manufacturing cells that each consist of a group of equipment. This study, however, considers FMS scheduling as a multiple-level problem, and follows the hierarchical control architecture developed by the National Institute of Standards and Technology (NIST) for its prototype factory, the Automated Manufacturing Research Facility (AMRF) (Simpson et al., 1982; Jones and McLean, 1986).

Owing to the complexity of FMS scheduling and the need for real-time control, many heuristics were developed for reducing the complexity and computational burden. However, most of these are single-objective-driven (e.g., the earliest-due-date rule), even though FMS scheduling may be multiple-objective. For this reason, combining different heuristics seems to be a logical solution (Vepsalainen and Morton, 1987). Further, FMS scheduling should be able to: (1) cooperate with the sequencing mechanism in achieving the system’s global goal, (2) pursue multiple objectives of the system, (3) provide a quick response with short computation time, (4) provide flexibility for the human scheduler, and (5) provide a flexible software structure.

To accomplish these goals, in this study we develop a hierarchical scheduling model which can pursue multiple performance objectives by coordinating shop and cell controls. At the shop level, a rough-cut schedule is generated based on a combined priority index, which considers six factors of the existing orders: tardiness cost, inventory cost, profit, processing time, due date, and order quantity. The rough-cut schedule is provided to the cell controller that is responsible for machine loading within the cell. At the cell level, then, a fixed priority (FP) rule (Shmilovici and Maimon, 1992) is used for sequencing multiple part types within a single machine (or flexible manufacturing center). The cell controller is capable of routing parts with alternative operations. It provides detailed projected order processing times of the rough-cut schedule, back to the shop. Then the shop controller uses this information to evaluate the different possible schedules in making its final decision. A multiple-objective function is established to pursue the best performance of the entire FMS.

The control system is developed using object-oriented modeling and design methodology. It is chosen because of its strong linkages between real-world objects and software objects. Its inherent features of easy modification and reusability are also compatible with the FMS environment. The C++ language is used to evaluate system performance and is executed on an IBM compatible microcomputer. The following are the basic features and assumptions of this study:

(1) Continuous order arrival: Orders arrive continuously; thus the scheduling is dynamic;
(2) No order preemption: Once an order is processed, other orders cannot interrupt it;
(3) Order-dependent cost: Each order has its unique tardiness cost (an estimate of the loss-of-goodwill or lost sales), inventory cost (cost for tied-up capital, insurance and storage), and profit (difference between contract sale price and estimated cost with overhead), all of which are assumed specified by the management;
(4) Single part order: An order can only contain a single part type but with various quantities;
(5) Routing availability: Part process plans (routings) are available and alternative routing is allowed;
(6) Deterministic processing time: Part processing time on any specific machine is deterministic and alternative machines may require different processing times for the same operation;
(7) No setup time: Setup time is not considered by assuming rapid automatic tool changes;
(8) No system blocking: System blocking is assumed not to occur;
(9) Negligible part moving time: Robot moving time in the cell is short so that it can be neglected;
(10) No inter-cell moves: An order does not need to visit more than one cell in its processing;

(11) No unexpected events: Machine breakdowns and raw material shortage are not considered.

The rest of the paper is structured as follows. Section 2 reviews related work in hierarchical control and scheduling of FMSs. The proposed methodology for multi-objective hierarchical scheduling is described in Section 3. Section 4 presents illustrative examples and discusses certain issues related to system implementation. Finally, Section 5 concludes the paper.

2. Related work in FMS scheduling

2.1. Hierarchical control of FMSs

FMS are automated production systems capable of producing a wide variety of part types with the same equipment. In order to effectively control such complex production systems, NIST has developed a generic control hierarchy in the AMRF with four levels: facility, shop, cell, and equipment. A controller at each level is responsible for receiving commands from and reporting current status to higher-level controllers, making decisions, sending commands to and receiving status report from lower-level controllers, and retrieving data from a supporting database (Williams, 1988; Bedworth et al., 1991). In a leader-follower strategy, a hierarchical approach decomposes a large complex problem into manageable tasks, and uses results of the higher-level sub-problem (the leader) as input to the lower-level sub-problem (the follower) to reach a global solution. Due to the complexity of FMS control, a hierarchical approach is commonly implemented. Lin et al. (1994) reported advantages of using hierarchical control. It allows the control problem to be partitioned to limit the complexity of any module in the hierarchy, regardless of the complexity of the entire structure. Information is passed through the system in specified communication protocols. And, minimum reprogramming is needed when a controller is modified. Functionally, an FMS can be decomposed into several subsystems (MacCarthy and Liu, 1993; Changchien et al., 1995): a material processing subsystem (MPS), a material transport subsystem (MTS), a material storage subsystem (MSS), and the control subsystem. In addition, an FMS also contains a database management system, because the operational control of an FMS needs to access a large set of distributed process data (see Ranky, 1983; Bedworth et al., 1991; Lin and Fang, 1993).

Unlike the spatial decomposition in the AMRF hierarchical architecture, Gershwin (1989) developed a hierarchical flow control framework for scheduling and planning events in a flexible manufacturing system. Responding to potentially disruptive events that occur at certain distinct frequencies in a temporal hierarchy, the scheduling and planning policies seek a balance between the demand and resources of production.

Scheduling in the AMRF at any level of the hierarchy has four steps: selection of candidate scheduling rules, simulation of the scheduling performance, statistical analysis of the simulation results, and compromise analysis of the results for determining the schedule. Control is then applied by coordinating the operations based on a production cost analysis. The AMRF scheduler is thus a hierarchical, single-dispatching-rule-driven, multiple-objective, stochastic system. After investigating the AMRF and other research in FMS control, Boulet et al. (1991) concluded that since classical control theory cannot be directly applied due to the difficulty of defining transfer functions explicitly, simulation, heuristics, and expert systems are commonly used for real-time scheduling.

Several other applications of hierarchical scheduling have been reported. Shanker and Tzen (1985) use a hierarchical-heuristic approach for FMS scheduling. Wein and Chevalier (1992) use a two-step approach to job-shop scheduling with three dynamic decisions: assigning due-dates to arriving orders, releasing orders to the shop, and sequencing orders at the machines.

Since scheduling algorithms at the lower control levels of the FMS can only search for solutions under the guidance of algorithms of the higher level of FMS control, they have to perform within a reduced problem space. Thus, a hierarchical approach may potentially only at best find good lower-level solutions within the given poor higher-level results. For instance, machine loading at the cell level is limited to the orders scheduled to the cell. Therefore, the final solutions may not achieve the system’s global goals. This motivated Moreno and Ding (1993) to improve Shanker and Tzens' work by developing a constructive heuristic. With less restrained higher-level results, the lower level has more searching space, thus can reach a better global result.

2.2. Scheduling of FMSs: An overview

FMS performance strongly depends on the scheduling strategies used. FMS scheduling problems are known to be NP-hard and generally involve a large number of machines and part types. In addition, searching for an optimal schedule in a dynamic system, such as an FMS, may not be practical since it is too time-consuming to provide a quick response to real-time events.

Due to the complexity of FMSs, analytical approaches with closed form exact solutions can only be exploited under certain stringent assumptions (see, e.g., Ahluwalia and Ji, 1991; Wein 1990). Well-designed scheduling heuristics, due to their ease of computational burden and simplicity of implementation, are commonly used even though optimal solutions are not guaranteed (see, e.g., Chan and Bedworth, 1991; Liu and Lin, 1993). Among them, priority rules are the most popular.

Regardless of the rules used, FMS scheduling is either static or dynamic. Essentially, a static problem is a snapshot of a dynamic problem and scheduling decisions are made when orders enter the system. Whereas in dynamic scheduling, decisions are made at the completion of each operation step (Ahluwalia and Ji, 1991). Since no single scheduling rule has been found to produce superior results on all performance measures, the user must choose one or more rules according to the measures prevailing in particular applications (Montazeri and Van Wassenhove, 1990; Kim, 1990). Several issues in FMS scheduling are reviewed in detail next, including performance of single rules versus mixed rules, and scheduling with alternative operations and multiple tasks.

2.3. Single rule versus mixed rules (single objective versus multiple objectives)

Since no single rule consistently outperforms all other rules, Ishii and Talavage (1994) use a mixed dispatching rule (MDR) approach in FMS scheduling by mixing four rules: next-in next-out, shortest processing time (SPT), largest slack first, and first-in first-out. The criteria to evaluate system performance include mean flow time, mean tardiness, weighted mean flow time, weighted mean tardiness, and combinations thereof. They found that MDR improves system performance, compared to the best result using any single rule, due to a search strategy that focuses on bottleneck machines. However, the MDR approach does not work as efficiently for multi-objective scheduling as it does for the single objective case.

Several researchers use fuzzy logic to solve the multi-objective problem. Watanabe et al. (1992) combine a profit and a slack criterion through a defined function between profit and order completion time. Hatono et al. (1992) develop a more generic model for achieving multiple performance objectives. Grabot and Geneste (1994) introduce the importance rule which, often neglected in research, protects the late completion of particular orders. Combining SPT, slack time, and the importance rules, they demonstrate that aggregated rules can provide compromised performance between the elementary rules they are composed of. For example, the slack time rule is better than the SPT rule for maximum flow time and maximum lateness, but not so for average flow time, lateness, and tardiness. Note that the fuzzy logic approach assumes that a combination of rules provides a compromise between their advantages. However, this assumption is not necessarily true under certain circumstances. With the same assumption, mixed or combined priority rules work better than single rules with respect to multi-objective manufacturing systems.

2.4. Sequencing with alternative operations

Due to the interchangeability of machines, a part can have several alternative operations. To avoid long queues, an alternative operation may be used, often performed by machines of lesser capabilities. After testing three schemes of implementing alternative operations in FMSs, Wilhelm and Shin (1985) claim that alternative operations can reduce flow time (and consequently WIP), and increase machine utilization. Nasr and Elsayed (1990) also propose a heuristic to deal with alternative operations in a static job-shop environment. The heuristic is based on a mixed integer program.

Karsiti et al. (1992) present a two-level dynamic scheduling technique to take advantage of alternative operations. First, a minimum job in queue rule assigns priority to machines to achieve a fair distribution of the total load. Then at the second level, they use two tardiness-based performance measures to compare the results of four different heuristic rules: first-come-first-serve, SPT, earliest operation due time, and earliest final due date. They claim that earliest final due date is far superior to the other rules due to its emphasis on final due dates. A hierarchical approach is used in this work.

2.5. Scheduling with multiple tasks and alternative operations

One of the common objectives of industries is to reduce order cycle time or lead time, i.e., minimize the average flow time of products. This can be achieved not only by routing multiple part types concurrently but also by providing alternative operations for parts to avoid bottleneck congestion of certain resources. Shmilovici and Maimon (1992) propose three dynamic heuristics using this strategy:

(1) <u>Fixed priorities (FP) heuristic:</u> A fixed priority based on a cost function is assigned to [machine, process] pairs. The pair with the highest priority value will be processed first.

(2) Least reduction in entropy heuristic: By modeling an FMS as a Markov process, this rule uses an entropy function to calculate the system processing flexibility in choosing machines and processes (i.e., the FMSs' ability to route parts to alternative machines during breakdowns). First, the processing operation of a part with the minimum flexibility is chosen because it has the fewest choices on machines. Then the most available machine is chosen to satisfy the processing requirement so that the overall flexibility of the FMS would remain maximum (i.e., the remaining parts would enjoy maximum possible choices).

(3) Minimum flow resistance heuristic: This heuristic is derived from two simple rules on serial and parallel operations, similar to those in calculating the resistance of an electric circuit.

They conclude: (1) Operation flexibility increases the throughput for all heuristics while conserving lead time, and thus is desirable. However, it requires a large buffer space. (2) Increasing internal buffer space increases FMS throughput, but only up to a certain extent. (3) Minimum flow resistance outperforms the other heuristics in terms of throughput because of its ability to foresee the future, but at the expense of high buffer utilization and long lead times. It also imposes a high computational cost. (4) Priority should be given to simple heuristics that are easy to understand and implement, such as FP (for which SPT is a special case).

However, routing multiple orders concurrently within a cell is not always beneficial because it would: increase the frequency of changing fixtures and tools, cause conflict of sequence (and hence increase flow time), and require temporary storage space (inner buffer). Fortunately, today's advanced technology and machines allow fixtures and tools to be changed rapidly. Automated tool changers and tool magazines on machining centers significantly reduce setup times. Thus FP rule may work well for multiple-order sequencing problems.

2.6. Economic factors in FMS scheduling

Meeting due dates is one of the most important goals in scheduling. The consequence of tardy deliveries, such as delay penalties, customer discontent, and rush shipping costs can vary significantly over customers and orders. Therefore, a weighted tardiness cost should be reflected in order priority. Vepsalainen and Morton (1987) examine six priority rules with weighted tardiness costs within a job shop environment. They assume that a delay cost of v_i per unit time is charged if order i is completed after its due date d_i, and the objective is to minimize the weighted tardiness of orders:

$$WT = \sum_{i=1}^{n} v_i \, [C_i - d_i]^+$$

where $[C_i - d_i]^+ = \text{Max}\{0, C_i - d_i\}$ and C_i is the completion time for order i. The six heuristic rules used in their research are as follows:

(1) First-come-first-serve (FCFS): This rule does not consider due dates.

(2) Earliest due date (EDD): This rule assumes all orders have the same tardiness cost.

(3) Slack per Remaining Processing Time (S/RPT): This rule is derived from minimum slack (MSLACK) rule. An MSLACK index is the due date of order i minus its remaining processing time and minus the current time. It actually counteracts SPT: of all orders with equal due dates, the order with the longest processing time will receive the top priority. Thus, S/RPT is used to compensate this "anti-SPT" tendency. Its formula is: MSLACK index of order i divided by its remaining processing time. Same tardiness cost for all orders are also assumed.

(4) Weighted Shortest Processing Time (WSPT): As is well known, the SPT rule provides for single machine scheduling the minimum mean flow time, which is equivalent to minimizing mean lateness. Therefore, SPT could affect the objective of minimizing the number of tardy orders. WSPT proposed here is weighted by v_i, which means different orders could have different tardiness costs.

(5) Weighted Cost OVER Time (COVERT): With a look-ahead dynamic feature, it considers the expected waiting time for a remaining operation. COVERT has been proven superior to other rules in mean tardiness performance. This rule is also weighted by v_i.

(6) Apparent Tardiness Cost (ATC): Also a dynamic rule, the ATC follows an exponential function of slack that involves a look-ahead parameter measured in units of average processing times. It is obtained by simulation runs or standard processing times. ATC is also weighted by v_i.

Vepsalainen and Morton (1987) ranked the performance of those six rules, in descending order: ATC - COVERT - WSPT - S/RPT - EDD- FCFS. Obviously, rules with a look-ahead feature outplay those without, and rules concerning order-dependent tardiness costs outperform those without.

From the above review, there is clearly a need for multi-objective FMS scheduling that can address the multi-level control for the shop and cells. We now present the development of such a system in the next section.

3. Development of multiple-objective scheduling for an FMS

We first describe the hierarchical configuration considered for the FMS. Then, we will present the hierarchical scheduling approach considering multiple objectives with alternate operations at the two levels of the shop and the cell.

3.1. Configuration of the FMS

The FMS in this study consists of a shop controller, an MPS controller (for the cell), a virtual MTS controller with several AGVs, four CNC machines, one robot, an input buffer, an output buffer, and an inner buffer (for storing parts). The scope of the entire system within the hierarchical organization of the FMS control is shown in figure 1.

Insert figure 1 about here

The shop controller receives orders from customers and schedules them according to shop objectives. Then, it requests the MSS controller to release raw materials and the MTS controller to send an AGV to load the material. MTS will report to the shop controller upon successfully delivering raw materials to MPS. Then, the shop controller will instructs the MPS controller to process the materials into finished parts. After finishing all operations, the MPS controller will inform the shop controller; then, again, the shop controller will request the MTS to deliver finished parts to MSS for storage. The temporal decomposition of processing an order is shown in figure 2.

Insert figure 2 about here

Our proposed methodology for FMC scheduling is divided into two portions: (1) development of heuristic priority rules to construct a flexible scheduling mechanism using the hierarchical approach, and (2) implementation of the scheduling system in a flexible software structure using object-oriented programming. We first discuss the scheduling mechanism.

3.2. Hierarchical approach for scheduling

Since an FMS has to achieve multiple performance objectives, a monolithic scheduling algorithm would be complex even if it is capable of addressing all objectives. Therefore, a hierarchical approach is considered for scheduling. It is achieved by the controllers at different levels of the hierarchical architecture, namely: the shop controller and the cell controller(s).

This work differs from the commonly used priority rules that usually target a single-objective. Since the shop controller is the "leader" of the shop, it must consider all factors that could influence its goal. It generates a good schedule that can maximize profits, and minimize tardiness cost, inventory cost and mean flow time. This is accomplished by ranking orders by a multiple-objective heuristic rule using a combined priority index. On the other hand, the shop controller would be over-burdened to attend every detail because of its governance over several subsystems. Thus, computation burden is a major consideration in scheduling. To provide a quick response, the shop controller can only concentrate on some carefully selected major factors. The combined priority index has all the aforementioned features and is easy to calculate. The result of applying this index is a rough-cut schedule, which provides the guidance to the cell controller for a detailed schedule.

The rough-cut schedule is refined by the cell controller. To keep the problem tractable, the shop controller provides to the cell controller, at one time, up to three candidate orders with the highest combined priority index values. By nature of design, the cell controller performs tasks in a more detailed manner, and hence is able to provide the shop controller a more precise schedule in terms of order completion times. Unlike the shop, a cell controller only concentrates on minimizing mean flow times of the orders, i.e., to process orders as fast as possible. According to the order completion estimates from the cell controller, the shop controller evaluates and modifies the corresponding schedules considering the overall objective function. Finally, the cell controller is directed to execute the selected schedule. This schedule has improved performance over the rough-cut one originally generated by the shop controller. The detailed procedure in a temporal decomposition format is shown in figure 3. Interactions between the shop and cell controllers in this iterative scheduling process are shown by their respective tasks.

Insert figure 3 about here

3.3. Overall scheduling objectives of the FMS

Making money is the goal of business (Goldratt, 1992). Derived from this concept, increasing profits and reducing costs are major business objectives. Manufacturing objectives for different shops might vary according to the nature of the shops. However, they should all aim at the ultimate goal of the business. Some commonly accepted objectives are listed as follows (Nahmias, 1993):

(1) meet due dates,
(2) minimize work-in-process (WIP) inventory,
(3) minimize the average flow time of orders through the system,
(4) achieve high machine and worker time utilization, etc.

To achieve these objectives, shop scheduling must consider profits and costs as major factors. In addition, it should also achieve a high system efficiency, which is the objective of cell controllers. Without considering it, system performance could seriously deteriorate. For example, it is not beneficial to use tardiness cost as the only factor to schedule orders.

There are three fundamental factors which the shop controller should take into account: (1) profits gained from orders, (2) costs associated with orders, and (3) system efficiency. It is noted that profit and cost estimates may not always be accurately available. Most companies might be able to download reasonable estimates for these from a cost accounting or engineering cost estimation software package.

The objective function of the example FMS is formulated as:

$$\text{Maximize:} \quad \sum_{\text{all } i} f_i - v_i \, [C_i - d_i]^+ - c_i \, (C_i - \text{Tnow}) - c'_i \, [d_i - C_i]^+ ,$$

where

f_i = profit gained from order i,

v_i = tardiness cost per unit time if order i is completed after its due date,

c_i = average WIP inventory cost per unit time of order i,

c'_i = finished-good inventory cost per unit time of order i,

C_i = completion date of order i,

d_i = due date of order i, and

Tnow = current time (the point of time when the schedule decision is made).

Note that the objective function concentrates on four objectives:

(1) Maximizing profit,

(2) Meeting due dates (minimize tardiness cost),

(3) Minimizing the WIP inventory cost, and

(4) Minimizing finished-good inventory cost.

To achieve them, a multi-objective priority rule for ranking orders needs to be developed.

3.4. Shop controller: Combined priority rule

As mentioned in Section 2, no single rule outperforms any others under a multi-objective environment. Thus, combining different rules by taking their advantages into account might create a good scheduling rule. Obviously, it is critical to choose the proper rules. From the objective function described in the preceding section, the SPT rule and Slack per Remaining Processing Time (S/RPT) rule are good candidates. Using SPT as a scheduling rule can reduce mean flow time; therefore, inventory costs of the system could be reduced. On the other hand, the S/RPT rule is due-date-driven. Aiming at different objectives, both of them have their own pros and cons. The index formula of S/RPT and SPT are as follows (Kim, 1990):

$$\text{S/RPT\_index} = \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{\sum_{q=j}^{m_i} p_{iq}}$$

$$\text{SPT\_index} = \frac{1}{p_{ij}}$$

where

d_i = due date of job i,

p_{ij} = processing time of the j th operation of job i,

t = current time,

m_i = numbers of operations job i will need, and

$\sum_{q=j}^{m_i} p_{iq}$ = the remaining processing time of job i, after j-1 th operation has been done.

For scheduling purposes, the remaining processing time is considered as the total processing time of an order, which is the sum of the processing time of each part within an order. Therefore, the S/RPT and the SPT indices become

$$\text{S/RPT\_index} = \frac{d_i - t - p_i}{p_i}$$

and

$$\text{SPT\_index} = \frac{1}{p_i},$$

where p_i represents the total processing time of order i. When there are alternative routings, it is the sum of process times in the preferred routing. This decision is normally made by manufacturing engineering at the process planning stage.

There is a difficulty combining these two rules because S/RPT ranks priorities in an ascending order, while SPT ranks priorities in a descending manner. To resolve this conflict, either rule may be modified. We choose to modify the S/RPT rule by reversing the sign of the index since later, the FP rule at the cell level ranks priorities in a descending manner. This function is plotted in figure 4.

$$\text{S/RPT\_index} = -\left(\frac{d_i - t - p_i}{p_i}\right).$$

Insert figure 4 about here

Then we multiply the S/RPT index by the SPT index and get the combined index formula.

$$\text{Combined\_index} = -\frac{1}{p_i}\left(\frac{d_i - t - p_i}{p_i}\right).$$

However, this formula has a drawback: when the portion of the modified S/RPT index is negative (t is far ahead from the due date), the order having heavy tardiness cost gets less priority. Thus, the S/RPT index should be modified again so that its value will always be positive. The new and improved index is shown below:

$$\text{Combined\_index} = \frac{1}{p_i}\left| 1 - \frac{d_i - t - p_i}{\text{Max}\{d_i - t\ ,\ p_i\}} \right| .$$

The S/RPT portion of the combined index is plotted in figure 5.

Insert figure 5 about here

Indeed, this index combines the S/RPT and SPT rules. At the left hand side of the delay point, (d_i - 1.0 p_i) in figure 5, SPT plays the dominant role and S/RPT has little influence. On the other side of the delay point, when the current time approaches or exceeds the due date, S/RPT dominates the index. This index can be further generalized to account for processing delays, transport times and waiting times. A factor w between 1.1 and 1.5, depending on the system efficiency can be employed as a multiplier to the processing time p_i. This is based on common agreements that proper machine utilization is between 50 and 80% and the machine utilization of an FMS could be up to 90%. Users can estimate w according to historical data of their systems. The S/RPT portion of the combined index is plotted in figure 6.

$$\text{Combined\_index} = \frac{1}{p_i}\left| 1 - \frac{d_i - t - w\ p_i}{\text{Max}\{d_i - t\ ,\ w\ p_i\}} \right| .$$

Insert figure 6 about here

This combined index includes only two factors, processing times and due dates of orders. Tardiness cost, profit and inventory cost of orders should also be considered, for which there are three basic ways of assigning priorities to orders:

(1) The more profit (f_i) an order can make, the higher the priority it will receive.
(2) The higher tardiness cost (v_i) an order is charged, the higher the priority it will receive.
(3) The higher inventory cost (c_i) an order is charged, the lower the priority it will receive.

The priority index formula, therefore, is developed accordingly as follows:

$$\text{Priority\_index} = \frac{v_i}{p_i}\left| 1 - \frac{d_i - t - w\ p_i}{\text{Max}\{d_i - t\ ,\ w\ p_i\}}\ \frac{f_i}{c_i} \right| .$$

Finally, a quantity factor ($1/q_i$) can be used to give preferential treatment to orders with fewer items when the above priority index is the same for two orders.

3.5. Cell controller: Multiple-task sequencing with fixed priority rule

As mentioned in Section 2, the capability of simultaneously routing multiple types of parts could significantly increase machine utilization. However, this capability does not always benefit the FMS, because routing different part types at the same time could increase the frequency of changing fixtures and tools, and cause conflict of sequence. Therefore, selecting the proper combination and number of orders to be processed is very important. For example, when the machines needed by order A are identical to those needed by order B, it might be advantageous to process A and B separately instead of processing them at the same time. This is because even processing orders A and B together will not increase machine utilization; it will only increase WIP inventory. Also, it is not appropriate to process concurrently too many orders, say, greater than the number of machines within the cell. Routing too many part types into the system at the same time could cause serious sequence conflict and overflow of the inner buffer, given its limited capacity. Therefore, the shop controller has to select the proper number and combination of orders to be scheduled.

Let us assume that the maximum number of orders that can be processed at the same time is the number of machines within the cell minus one. For example, for a four-machine cell, the maximum number of orders allowed to be concurrently processed is three. As a result, if the number of existing orders is less or equal to three, the shop controller will directly make different schedules for them, determine the schedule with the maximum objective function value, and then process those orders according to that schedule. If there are more than three orders on hand, the shop controller will select the three orders with the highest priority, then make a schedule.

The FP rule employed here is based on the presentation in Shmilovici and Maimon (1992). For the sake of completeness, we have included the notation and algorithm in the Appendix.

Let us assume that the SPT rule, a special case of the FP rule, is used. The priority index formula, therefore, is 1 divided by operation time, p_{mij}, the time needed to operate process j of part type i on machine m. The SPT index formula is shown below.

$$\text{SPT\_index} = \frac{1}{p_{mij}} .$$

The priority is given in a maximum fashion, which means the operation m_{ij} having the highest SPT-index value will be processed first. An example priority matrix $(q)_{m,(i,j)}$ is shown below (where Ø represents nonexistent operations):

$$(q)_{m,(i,j)} = \begin{pmatrix} \frac{1}{p_{111}} & \frac{1}{p_{112}} & \emptyset & \emptyset & \emptyset & \frac{1}{p_{122}} & \emptyset \\ \frac{1}{p_{211}} & \frac{1}{p_{212}} & \emptyset & \emptyset & \frac{1}{p_{221}} & \frac{1}{p_{222}} & \frac{1}{p_{223}} \\ \emptyset & \emptyset & \frac{1}{p_{313}} & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \frac{1}{p_{414}} & \emptyset & \emptyset & \frac{1}{p_{423}} \end{pmatrix}.$$

In this example, there are two orders waiting to be processed. Order 1 needs four operations and order 2 needs three. The first (j=1) and second (j=2) operations of order 1 (i=1) can be done by either machine 1 (m=1) or machine 2 (m=2), the third operation of order 1 can only be done by machine 3, and the fourth operation can only be done by machine 4. A numerical example illustrating the seven steps that the FP algorithm makes for a routing decision is now presented.

<u>A numerical example:</u>

<u>Cell configuration and part types:</u>

M = {1, 2, 3}: Three machines　　Queue size = 2: The capacity of the inner buffer

U = {a, b}: Two part types

P = {(a,1), (a,2), (a,3), (b,1), (b,2)}: Five operations (a requires three and b requires two)

<u>The priority matrix showing alternative operations is:</u>

$$(q)_{m,(i,j)} = \begin{bmatrix} \frac{1}{4} & \frac{1}{6} & \emptyset & \emptyset & \frac{1}{5} \\ \frac{1}{4.5} & \frac{1}{5} & \emptyset & \frac{1}{3.5} & \frac{1}{7} \\ \emptyset & \frac{1}{5.5} & \frac{1}{3} & \emptyset & \emptyset \end{bmatrix} = \begin{bmatrix} 0.250 & 0.167 & \emptyset & \emptyset & 0.200 \\ 0.222 & 0.200 & \emptyset & 0.286 & 0.143 \\ \emptyset & 0.182 & 0.333 & \emptyset & \emptyset \end{bmatrix}.$$

RS(a) = [(a,1), (a,2), (a,3)]: Process plan for part type a

RS(b) = [(b,1), (b,2)] or [(b,2), (b,1)]: Alternative process plans for part type b

<u>Current conditions:</u>

M_I = {1, 3}: Machines 1 and 3 are idle.

M_W = {2}: Machine 2 is busy.

U_W = {b1}: Machine 2 is performing process (b,2) on part b1.

U_I = {a1}: Part a1 is waiting to be processed within the cell

H(a1) = [(a,1)]: Part a1 has been processed for (a,1) operation.

H(b1) = [(b,1)]: Part b1 has been processed for (b,1) operation.

WU = {a2, a3, a4, b2, b3, b4, b5}: Seven parts are waiting in the input buffer

F(WU) = {(a,1), (b,1), (b,2)}: Three operations are requested by the parts in the input buffer.

Now, the parts are scheduled by selecting appropriate process plans and sequencing their operations. For the notations see the Appendix.

Step 1: If NOL(u_i) is the next operation list for waiting unit u_i, then NOL(a1) = {(a,2)}, and the set of immediately desired processes of parts P_D = {(a,2)}.

Step 2: The set of available processes for idle machines P_I = {(a,1), (a,2), (a,3), (b,2)}.

Step 3: The set of relevant processes for routing $P_R = P_I \cap P_D = \{(\alpha,2)\}$? {Ø}; therefore, go to step 4.

Step 4: Reduced matrix: (eliminate row 2 and column 1, 3, 4, 5).

$$(q_r)_{m,(i,j)} = \begin{array}{cc} m \setminus p & (\alpha,2) \\ \begin{array}{c}1\\3\end{array} & \begin{bmatrix}0.167\\0.182\end{bmatrix} \end{array}.$$

Step 5: Max $(q_r)_{m,(i,j)}$ = 0.182, m* = 3, p* = (a,2).

Step 6: Part a1 that requested process (a,2) is chosen.

Step 7: Part a1 is assigned to machine 3 for process (a,2).

Now, the system's variables are updated:

M_I = {1}, M_W = {2, 3},

U_I = {Ø}, U_W = {a1, b1},

H(a1) = [(a,1)], H(b1) = [(b,1)],

WU = {a2, a3, b2, b3, b4} and F(WU) = {(a,1), (b,1), (b,2)}.

Then the algorithm is repeated. Go back to Step 1. In the next iteration, it can be shown that part a2 is assigned to machine 1 for process (a,1). Then, the system's variables are updated again:

M_I = {Ø}, M_W = {1, 2, 3},

$U_I = \{Ø\}$, $U_W = \{a1, a2, b1\}$,

H(a1) = [(a,1)], H(a2) = [ ], H(b1) = [(b,1)],

WU = {a3, b2, b3, b4} and F(WU) = {(a,1), (b,1), (b,2)}.

Then the algorithm is repeated. Go back to Step 1.

Step 1: NOL() = {Ø}, and PD = {Ø}.

Step 2: PI = {Ø}.

Step 3: PR = {Ø}. Since $M_I = \{Ø\}$, STOP and wait for the next event to occur.

The Gantt chart up to this point of time is shown in figure 7.

Insert figure 7 about here

Although the FP rule is able to sequence multiple part types which could have alternative operations and alternative process plans, some limitations are: (1) it does not consider transport and setup times, (2) it is sensitive to buffer size, and (3) it introduces the possibility of machine blocking (although rarely). Fortunately, these problems can be overcome with a more sophisticated algorithm (not presented here for sake of brevity).

3.6. The hierarchical and multiple objective scheduling algorithm

Now we present the proposed hierarchical scheduling algorithm for an FMS using the developed multi-objective priority index. It consists of nine steps:

Step 1: At the shop level, calculate priority for each order according to the combined priority index formula described in Section 3.4.

$$\text{Priority\_index} = \frac{v_i}{p_i} \left| 1 - \frac{d_i - t - w\,p_i}{\text{Max}\{d_i - t\,,\, w\,p_i\}}\,\frac{f_i}{c_i\,q_i} \right| .$$

If there is no order waiting to be processed or the cell is busy, wait at Step 1 until an order arrives or the cell becomes idle.

Step 2: Rank orders according to their priority index values.

Step 3: If the number of orders on hand is less or equal to three, directly go to Step 4; otherwise, choose three existing orders with the highest priority, then go to Step 4.

Step 4: Make all possible schedules of chosen orders.

If there is only one order, go to Step 5.

If there are two orders, there are three possible schedules.

(1st -> 2nd): The order with the highest combined index gets processed first, followed by the order with the 2nd highest index; (2nd -> 1st): The order with the 2nd highest index goes first, followed by the order with the highest priority; and (1st and 2nd): Process both of them together.

If there are three orders, there are thirteen possible schedules.

Six permutation schedules: (1st -> 2nd -> 3rd), (1st -> 3rd -> 2nd), etc., six schedules with two orders routed together: (1st and 2nd -> 3rd), (3rd -> 1st and 2nd), etc., and a schedule with all orders processed together (1st and 2nd and 3rd).

Step 5: Request the cell controller to estimate the completion date of each schedule using the FP heuristic rule of Section 3.5 (and the Appendix). The SPT rule is used as the priority index of the FP rule.

Step 6: Feedback the result to the shop controller.

Step 7: The shop controller calculates the objective value of each schedule and selects the schedule with the maximum objective value. The objective function is:

$$\text{Max}\left\{ \sum_{\text{all i}} f_i - v_i\,[C_i - d_i]^+ - c_i\,(C_i - \text{Tnow}) - c'_i\,[d_i - C_i]^+ \right\},$$

where C_i is the estimated completion date of order i.

Step 8: Issue orders, according to the best schedule, to the cell.

Step 9: Set the cell as busy and go back to Step 1.

4. System Implementation and Results

Based on the efficient hierarchical control architecture, and powerful object-oriented modeling methodology, the multi-objective scheduling mechanism was developed for a prototype FMS. Furthermore, to study the performance of the hierarchical scheduling algorithm for an FMS, a simulation framework was also developed.

4.1. System implementation

The simulation program was built for an FMS with a shop controller, a cell controller, four machines, a robot, and three buffers (i.e., the input, output, and inner buffer). Orders arrive continuously with interarrival times taken from an exponential distribution with mean of 140 minutes. Each order only contains a single part type, and the system can produce 15 different part types. The quantity of parts in an order is assigned in the range of 5 to 30 units with equal probability. The parts have one to four operations with a randomly assigned routing through the machines, both with equal probability. An operation can be performed on alternative machines which are randomly chosen. The number of alternative process plans for a part is generated randomly between 1 and 3 with equal probability. Processing times of standard operations are uniformly distributed between 1 and 15 minutes. Processing times of alternative operations equal the processing time of standard operation weighted by a factor that is uniformly distributed between 1.0 and 2.0.

The profit that a part can gain is uniformly distributed between 50 and 100 dollars. Therefore, the profit of an order is equal to quantity of parts times the profit of a unit part. Given the profit, tardiness, and inventory costs per minute are generated by multiplying profit by a random number from U(5e-5, 8e-4) and U(5e-5, 4e-4) respectively. Finally, the due date of an order is a uniform distribution between 1.5 to 3.0 times the total processing time of the order.

4.2. Illustrative examples

Detailed studies of the system performance under a multi-objective and multi-task environment were conducted employing the simulation framework proposed in the previous section. Using the combined priority index formula described earlier, priority index values of orders existing in the system can be determined. The shop controller can rank those orders according to their priority; therefore, the shop controller determines which order(s) should be considered first. This procedure is illustrated by some numerical results shown in figure 8.

Insert figure 8 about here

By applying the Fixed Priority (FP) heuristic, multiple-task sequencing can be achieved. A simplified numerical result is shown in figure 9. According to the simulation results, each dispatching decision only takes a few milliseconds of computation time.

Insert figures 8 and 9 about here

After tasks described in the previous two sections have been done, the shop controller will have the following information: the priority of the existing orders, a ranked list of orders, and estimated processing times of different schedules which are fedback from the cell controller. Based on this information, the shop controller then can choose the best schedule by comparing the objective value of each schedule.

An example is shown in figure 10. In this example, the best schedule is to process the second-priority order (order ID 17662) first then process the first- and third-priority orders (orders ID 9829 and 15463, respectively) together right after the second-priority order is completed. From our simulations, the shop controller only takes less than 0.3 CPU seconds on a 486/66 PC to determine a final schedule, so the scheduling mechanism can provide real-time response.

Insert figure 10 about here

4.3. Results and discussion

Heuristics with or without applying the hierarchical approach

To evaluate the performance of the scheduling system, six cases were tested. The combined index, shortest processing time (SPT) and earliest due date (EDD) rules were tested under the environment with or without applying the hierarchical approach for making schedules. Consequently, there are six simulation runs involved in this test. Regardless of the heuristic rule implemented, each simulation run will generate the same part types, order arrival pattern and order data. Therefore the same orders were tested in all six cases. The simulation program runs until over 100 orders have been completed. The results are shown in Table 1.

Insert Table 1 about here

As it can been seen, for all the heuristics, using the hierarchical approach consistently provided higher objective function values and higher objective function values per part than those without it. Therefore it indicates that applying a hierarchical approach in the scheduling mechanism provides great benefits for the system. This is because that the shop controller can obtain more information from the cell controller, and those information will help the shop controller to make a better decision.

Comparison of heuristics

In Table 1, the performance of three heuristics is also shown. The combined index rule with consideration of multiple parameters of the system outperforms SPT and EDD rule in terms of objective function values and objective function values per order. The reason is that SPT and EDD rules do not consider all the parameters that are in the objective function.

From these results, we can see that using the SPT as a priority index does reduce mean flow time of the system. However, since the FMS is not a single-objective manufacturing system, reducing mean flow time does not necessarily benefit the entire system.

The EDD rule, on the contrary, does not consider the efficiency of the system at all. This leads to the result that the EDD rule performs the worst among all three heuristics.

In summary, using the combined index under the hierarchical environment results in the highest objective value and the highest objective value per part among the six cases tested here. It also has the lowest value in completion time per part, which, calculated by dividing the total completion time of all parts (makespan) by the number of parts, can be used as an indicator for system throughput. When a large number of parts are made, it approaches system cycle-time. Therefore, the lower the completion time per part, the higher the system throughput. However, we do not claim that combined index rule fits all situations beyond experimental conditions presented here. Just as what was pointed out by Grabot and Geneste (1994): "...the performance of the scheduling rules depends strongly on the criterion chosen and on the characteristics of the production systems." It is still important to indicate that for the performance measures considered in pursuit for the global benefits of the system, the proposed index has superior performance to some other commonly used approaches.

The effect of multitask sequencing, alternative process plans, and alternative operations

Two more simulations are performed in this study, which: (1) explore the effect of multitask sequencing, and (2) examine the influence of alternative process plans and operations. The combined index rule is used in both of them as a priority index. The results are shown in Table 2.

Insert Table 2 about here

From these result, we can find that a cell with the capability of routing multiple tasks clearly outperforms the ones without this capability by large margins in all categories. This result is intuitively appealing. As described in Section 2, multitask sequencing will increase machine utilization. Consequently, the system's efficiency will also be improved.

The result associated with alternative operations and process plans is consistent with the studies conducted by Shmilovici and Maimon (1992), and Wilhelm and Shin (1985): a system will be more efficient if alternative operations and alternative process plans are available. Since this point has been well recognized in the literature, additional simulations applying different heuristics were not conducted.

Flexibility of the scheduling mechanism

Since the combined index is parameterized and the index formula of the FP heuristics can be easily modified to suit user needs, the scheduling mechanism can be implemented under different manufacturing objectives by changing factors within the combined index or the FP rule's priority matrix. A human scheduler, therefore, can influence the final result according to the needs of the particular application. Alternatively, multiple scheduling rules such as the MDR approach of Ishii and Talavage (1994) can be applied at the shop level and can be used in the proposed hierarchical framework.

5 Conclusions

FMS scheduling is a difficult problem to solve optimally due to the complex nature of the system. By using well-designed heuristics, it can be rendered tractable. In facing the scheduling problem of FMS, we should notice that FMS is a multi-objective, hierarchical manufacturing system with a number of manufacturing cells. Thus, it is important to consider these two important features when designing an FMS scheduling algorithm.

This paper develops an FMS scheduling mechanism with the capability of pursuing global benefit of the system under a multi-objective and multitask environment. Consistent with the physical hierarchy of the system, the FMS scheduling problem is divided into two levels: the shop controller and the cell controller. Different objectives are set to these two hierarchical levels. In general, orders with potential high tardiness cost (if not meeting the due dates), short processing times, high profit and low inventory holding cost are chosen to be processed first by the shop level. This is accomplished with the help of a combined index formula that involves several major parameters of the system and assigns priority to different orders at the shop level. The fixed priority heuristic (Shmilovici and Maimon, 1992) that is able to route multiple part types within the cell is adopted by the cell controller. This procedure is able choose among multiple process plans and alternative machines for operations. In our case, the SPT fixed priority rule targets the objective of minimizing flow time at the cell level.

By adopting this hierarchical approach, we do not over-constrain the scheduling mechanism to be a strict leader-follower or top-down procedure. The composite index that possesses the advantages of the individual rules it is composed of, helps the shop controller pick out the highest priority orders. All permutations and combinations of this small set of orders are completely characterized using tentative detailed scheduling by the cell controller. After obtaining this additional information from the cell controller, the shop controller makes the final selection. Due to the small number of solutions being computed by the cell controller the computational time is not excessive, and at the same time the mechanism is not limited by a “blind” top-down decision. Empirical studies demonstrate the benefit of this procedure.

Numerical studies also attest that the composite index is able to balance the multiple objectives unlike the chosen representative rules such as SPT and EDD that exhibit one sided behavior (or have preference towards one objective). In despite this, it is widely recognized that no single rule is universal. And, no matter how good a scheduling algorithm is, it should allow a human scheduler to influence the final result, perhaps using a parameterized algorithm. Thus, our framework along with its object-oriented implementation enables such flexibility, and alternative indexes and multiple rules can be employed in place of the combined index or the FP heuristic.

Our studies also confirm the previously reported benefits of using multi-tasking, alternative operations and alternative process plans as means of improving system utilization and throughput. We have been able to replicate these findings in a hierarchical setting and with the combined index that targets multiple FMS objectives.

Future work should aim at integrating the scheduling mechanism with the material transport subsystem (MTS), and the material storage/retrieval subsystem (MSS) in an FMS. Further, the fixed priority (FP) rule could be extended to consider sequencing robot moves within the MPS, and include a conflict resolution scheme to avert potential system blocking. We believe that a well-designed robot-sequencing algorithm will overcome this problem, and offer further improvement of the schedules for an entire FMS.

References

Ahluwalia, R.S. and Ji, P., "A Distributed Approach to Job Scheduling in a Flexible Manufacturing System," *Computers and Industrial Engineering*, Vol. 20, No. 1, pp. 95-103 (1991).

Bedworth, D.D., Henderson, M.R., and Wolfe, P.M., *Computer-Integrated Design and Manufacturing*, McGraw-Hill, NY (1991).

Boulet, B., Chabbra, B., Harhalakis, G., Minis, I., and Proth, J.M., "Cell Controllers: Analysis and Comparison of Three Major Projects," *Computers in Industry*, Vol. 16, pp. 239-254 (1991).

Chan, D.-Y. and Bedworth, D.D., "Design of a Scheduling System for Flexible Manufacturing Cells," *International Journal of Production Research*, Vol. 29, No. 11, pp. 2037-2049 (1990).

Changchien, S.-W., Lin, L., and Sun, D., "A Dynamic Control Model of Flexible Manufacturing Cells Using the Information Processing Object Hierarchy," *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 3, Nos. 3 & 4, pp. 193-223 (1995).

Gershwin, S.B., "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems," *Proceedings of the Institute for Electrical and Electronic Engineers (IEEE)*, Vol. 77, No. 1, pp. 195-299 (1989).

Goldratt, E.M., *The Goal*, North River Press, Croton-on-Hudson, NY (1992).

Grabot, B. and Geneste, L., "Dispatching Rules in Scheduling: a Fuzzy Approach," *International Journal of Production Research*, Vol. 32, No. 4, pp. 903-915 (1994).

Hatono, I., Suzuka, T., and Umano, M., "Towards Intelligent Scheduling for Flexible Manufacturing: Application of Fuzzy Inference to Realizing High Variety of Objectives," *JAPAN/USA Symposium on Flexible Automation*, Vol. 1, pp. 433-440 (1992).

Ishii, N. and Talavage, J.J., "A Mixed Dispatching Rule Approach in FMS Scheduling," *The International Journal of Flexible Manufacturing Systems*, Vol. 6, pp. 69-87 (1994).

Jones, A. and McLean, C., "A Proposed Hierarchical Control Model for Automated Manufacturing Systems," *Journal of Manufacturing Systems*, Vol. 5, No. 1, pp. 15-25 (1986).

Karsiti, M.N., Cruz, Jr. J.B., and Mulligan, Jr. J.H., "Simulation Studies of Multilevel Dynamic Job Shop Scheduling Using Heuristic Dispatching Rules," *Journal of Manufacturing Systems*, Vol. 11, No. 5, pp. 346-358 (1992).

Kim, Y.-D., "A Comparison of Dispatching Rules for Job Shops with Multiple Identical Jobs and Alternative Routing," *International Journal of Production Research*, Vol. 28, No. 5, pp. 953-962 (1990).

Larin, D.J., "Cell Control: What We Have, What We'll Need," *Manufacturing Engineering*, January 1989, pp. 41-54 (1989).

Lin, L. and Fang, Y.-J., "Database Model for Hierarchical Control of Manufacturing Cell Systems in a Manufacturing Shop," *Computer Integrated Manufacturing Systems*, Vol. 6, No. 3, pp. 185-194 (1993).

Lin, L., Wakabayashi, M., and Adiga, S., "Object-oriented Modeling and Implementation of Control Software for a Robotic Flexible Manufacturing Cell," *Robotics and Computer-Integrated Manufacturing*, Vol. 11, No. 1, pp. 1-12 (1994).

Liu, S.-C. and Lin, L., "Dynamic Sequencing of Robot Moves in a Manufacturing Cell," *European Journal of Operational Research*, Vol. 69, No. 3, pp. 482-497 (1993).

MacCarthy, B.L. and Liu, J., "A New Classification Scheme for Flexible Manufacturing Systems," *International Journal of Production Research*, Vol. 31, No. 2, pp. 299-309 (1993).

Montazeri, M. and Van Wassenhove, L.N., "Analysis of Scheduling Rules for an FMS," *International Journal of Production Research*, Vol. 28, No. 4, pp. 785-802 (1990).

Moreno, A.A. and Ding, F.-Y., "A Constructive Heuristic Algorithm for Concurrently Selecting and Sequencing Jobs in an FMS Environment," *International Journal of Production Research*, Vol. 31, No. 5, pp. 1157-1169 (1993).

Nahmias, S., *Production and Operations Analysis*, Irwin, Homewood, IL (1993).

Nasr, N. and Elsayed, E.A., "Job Shop Scheduling with Alternative Machines," *International Journal of Production Research*, Vol. 28, No. 9, pp. 1595-1609 (1990).

Ranky, P., *The Design and Operation of FMS (Flexible Manufacturing Systems)*, IFS (Publications) Ltd., Kempston, Bedford, UK (1983).

Shanker, K. and Tzen, Y.-J., "A Loading and Dispatching Problem in a Random Flexible Manufacturing System," *International Journal of Production Research*, Vol. 23, No. 3, pp. 579-595 (1985).

Shmilovici, A. and Maimon, O.Z., "Heuristics for Dynamic Selection and Routing of Parts in an FMS," *Journal of Manufacturing Systems*, Vol. 11, No. 4, pp. 285-296 (1992).

Simpson, J.A., Hocken, R.J., and Albus, J.S., "The Automated Manufacturing Research Facility of the National Bureau of Standards," *Journal of Manufacturing Systems*, Vol. 1, No. 1, pp. 17-31 (1982).

Vepsalainen, A.P.J. and Morton, T.E., "Priority Rules for Job Shops With Weighted Tardiness Costs," *Management Science*, Vol. 33, No. 8, pp. 1035-1047 (1987).

Watanabe, T., Tokumaru, H., and Nakajima, Y., "Job-shop Scheduling Using Fuzzy Inference to Make Profit into Account," *proceedings of the 1992 JAPAN/USA Symposium on Flexible Automation, ASME*, Vol. 1, pp. 423-427 (1992).

Wein, L.M., "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-station Network with Controllable Inputs," *Operations Research*, Vol. 38, No. 6, pp. 1065-1078 (1990).

Wein, L.M. and Chevalier, P.B., "A Broader View of the Job-shop Scheduling Problem," *Management Science*, Vol. 38, No. 7, pp. 1018-1033 (1992).

Wilhelm, W.E. and Shin, H.-M., "Effectiveness of Alternate Operations in a Flexible Manufacturing System," *International Journal of Production Research*, Vol. 23, No. 1, pp. 65-79 (1985).

Williams, D.D., *Manufacturing Systems*, Open University Press, Milton, U.K. (1988).

Appendix

In this appendix, we introduce the notation and algorithm of implementing the fixed priority rule for real-time routing of parts with flexible process plans. This is based on the presentation in Shmilovici and Maimon (1992).

Notation

M_I	Set of idle machines
M_W	Set of working (busy) machines
$M = M_I \cup M_W$	Set of all machines on shop floor
U_I	Set of idle parts, waiting in the storage area (physical or virtual) to be processed
U_W	Set of working parts, currently processed by working machines
$U = U_I \cup U_W$	Set of all parts ready to be routed in the cell
P_I	Set of available processes for idle machines (generated from M_I)
P_D	Set of immediately desired processes of parts (generated from U_I)
$P_R = P_I \cap P_D$	Set of relevant processes for routing
P_M	Set of all processes that the cell can do (generated from M)
P_U	Set of all processes the parts can desire (generated from U)
$P = P_M \cup P_U$	Set of all considered processes
$H(u_i)$	A function that generates the processing history of a unit of part type u_i
$NOL(u_i)$	A function that generates the next operation list for part u_i
WU; $wu \in WU$	Parts waiting for production outside the cell (WU is generated by a shop control function)
F(wu)	First operation of part wu waiting outside the cell
F(WU)	set of first operations from parts outside the cell
FU	set of finished parts at output buffer
$(q)_{m,p}$	priority matrix

A new part is transferred from WU to U_I (part leaves the input buffer and enters the cell) when there is some idle production capacity in the cell which can be used to process that part:

(1) M_I ? {Ø}; There are idle machines

(2) $P_I \cap P_D$ = {Ø}; Parts in the cell do not require the idle processes

(3) $P_I \cap F(wu) \neq \{\emptyset\}$; There is a waiting part that needs the processes

(4) $|U_I| <$ queue size; There is space in the storage area.

Routing decisions are made when there is a request for idle production capacity in the cell:

(1) $M_I \neq \{\emptyset\}$; There are idle machines

(2) $U_I \neq \{\emptyset\}$; There are idle parts in the cell

(3) $P_I \cap P_D \neq \{\emptyset\}$; The idle parts request idle processes.

Algorithm

There are seven steps to make a routing decision:

Step 1: Search all parts in U_I. For each part in U_I, find its next process(es) on the process plan (PP). P_D is the set of those processes.

Step 2: Search all machines in M_I. Get the available processes from the idle machines and put them in P_I.

Step 3: Find $P_R = P_I \cap P_D$. If this set is not empty, then go to the heuristics for the routing decision. Otherwise, test the aforementioned conditions for transferring a new part from W_U to U_I, or return to Step 1.

Step 4: Find the reduced priority matrix $(q_r)_{m,(i,j)}$ from the original priority matrix, $(q)_{m,(i,j)}$, by deleting rows that belong to busy machine (machines in M_W) and columns that belong to processes not in P_R.

Step 5: Find $\mathrm{Max}\{(q_r)_{m,(i,j)}\}$, i.e., the machine m* and the process (i,j)* that have the highest value in $(q_r)_{m,(i,j)}$. If there are two or more entries with the same value, choose one of them arbitrarily.

Step 6: Search U_I for a part that requests process (i,j)*. If there are several such parts, then apply the first-come-first-served (FCFS) rule.

Step 7: Route the selected part to machine m* for performing process (i,j)*. Go to Step 1.

List of Tables

Table 1 Performance of heuristics with and without hierarchical approach.

Heuristics	With Hierarchical Approach	Number of Completed Orders	Number of Completed Parts	Completion Time	Completion Time/part	Profit/part	Inventory Cost/part	Tardiness Cost/ part	Objective Value/ part
Combined Index	Yes	101	1712	14557.81	8.5	73.62	4.28	15.48	53.85
Combined Index	No	100	1688	14587.43	8.64	73.60	5.70	17.24	50.65
SPT	Yes	100	1641	14511.45	8.84	73.61	4.61	16.04	52.95
SPT	No	100	1667	14323.79	8.59	74.59	5.62	17.83	51.14
EDD	Yes	101	1707	15340.65	8.98	74.24	4.22	23.59	46.42
EDD	No	100	1695	14993.50	8.84	73.83	5.45	23.63	44.75

Table 2 Influence of multi-task scheduling, and alternative operations and process plans.

Multitask	Alternate operation & process plan	Number of Completed Orders	Number of Completed Parts	Completion Time	Completion Time/part	Profit/part	Inventory Cost/part	Tardiness Cost/ part	Objective Value/ part
Yes	Yes	101	1712	14557.81	8.503	73.62	4.28	15.48	53.85
Yes	No	100	1624	15650.98	9.637	74.95	4.13	21.93	48.88
No	Yes	100	1562	15981.36	10.23	74.23	2.90	27.76	43.56
No	No	100	1513	17005.94	11.24	74.56	3.03	38.83	32.69

Note: The combined index is used in all four cases.

Figure Captions

Figure 1: The scope of the proposed FMS

Figure 2: The temporal decomposition of processing an order

Figure 3: The temporal decomposition of determining a schedule

Figure 4: Modified S/RPT index

Figure 5: S/RPT index within the combined index

Figure 6: S/RPT index with weighted processing time

Figure 7: Gantt chart of the schedule using FP rule

Figure 8: Numerical example of calculating combined index value and ranking orders

Figure 9: Numerical example of routing parts in the cell using FP rule (with SPT)

Figure 10: Numerical example of the hierarchical scheduling algorithm

```
AN ORDER ARRIVES AT 388.46
-----------------------------------------------------------
 Order ID:   15463
 Part  ID:      12           Arrival  time:  388.46   min.
 Prc time:  101.50  min.      Av. proc time:   15.69   min.
 Quantity:    7.00  units     Due date     :  566.08   min.
 Profit  :  479.50  $/order   Tad cost     :    0.12 $/min.
 Inv cost:    0.03  $/min.

AN ORDER ARRIVES AT 414.83
-----------------------------------------------------------
 Order ID:   16772
 Part  ID:       8           Arrival  time:  414.83   min.
 Prc time:   63.49  min.      Av. proc time:    6.09   min.
 Quantity:    7.00  units     Due date     :  530.39   min.
 Profit  :  667.66  $/order   Tad cost     :    0.48 $/min.
 Inv cost:    0.26  $/min.

AN ORDER ARRIVES AT 506.21
-----------------------------------------------------------
 Order ID:   10751
 Part  ID:       9           Arrival  time:  506.21   min.
 Prc time:  242.06  min.      Av. proc time:   12.73   min.
 Quantity:   19.00  units     Due date     : 1012.12   min.
 Profit  : 2329.97  $/order   Tad cost     :    1.19 $/min.
 Inv cost:    0.47  $/min.

AN ORDER ARRIVES AT 705.52
-----------------------------------------------------------
 Order ID:    3943
 Part  ID:       5           Arrival  time:  705.52   min.
 Prc time:  144.43  min.      Av. proc time:    9.72   min.
 Quantity:   11.00  units     Due date     : 1138.81   min.
 Profit  : 1321.76  $/order   Tad cost     :    0.19 $/min.
 Inv cost:    0.11  $/min.


Tnow = 705.52

0: Calculated Index_value =  8.60  for order  15463
1: Calculated Index_value = 11.43  for order  16772
2: Calculated Index_value =  1.32  for order  10751
3: Calculated Index_value =  0.63  for order   3943

Rank 0:  order 16772   index value = 11.43
Rank 1:  order 15463   index value =  8.60
Rank 2:  order 10751   index value =  1.32
Rank 3:  order  3943   index value =  0.63
```

Figure 8 Numerical example of calculating combined index value and ranking orders.

```
CELL IS IDLE      MAKE DECISIONS  because new order arrives at 126.44
case 1:     Part_id = 7
ROUTE WU part 0  in order 0  to machine 3  for process 0
ROUTE WU part 1  in order 0  to machine 0  for process 0
ROUTE WU part 2  in order 0  to machine 1  for process 0
ROUTE WU part 3  in order 0  to machine 2  for process 0
FINISHED UNIT=0  0  0   ROUTE part 0  in order 0  to machine 3 for process 1
FINISHED UNIT=0  0  0   ROUTE part 1  in order 0  to machine 0 for process 1
FINISHED UNIT=0  0  0   ROUTE part 2  in order 0  to machine 1 for process 1
FINISHED UNIT=0  0  0   ROUTE part 3  in order 0  to machine 2 for process 1
FINISHED UNIT=0  0  0   ROUTE part 1  in order 0  to machine 0 for process 2
FINISHED UNIT=0  0  0   ROUTE part 3  in order 0  to machine 2 for process 2
FINISHED UNIT=0  0  0   ROUTE WU part 4  in order 0  to machine 3 for process 0
FINISHED UNIT=0  0  0   ROUTE part 4  in order 0  to machine 3 for process 1
FINISHED UNIT=0  0  0   ROUTE WU part 5  in order 0  to machine 1 for process 0
FINISHED UNIT=0  0  0   ROUTE part 5  in order 0  to machine 1 for process 1
FINISHED UNIT=0  0  0   ROUTE part 0  in order 0  to machine 2 for process 2
FINISHED UNIT=0  0  0   ROUTE part 2  in order 0  to machine 0  for process 2
FINISHED UNIT=0  0  0
FINISHED UNIT=0  0  0   ROUTE part 4  in order 0  to machine 2  for process 2
                        ROUTE part 0  in order 0  to machine 3  for process 3
FINISHED UNIT=0  0  0   ROUTE part 1  in order 0  to machine 1  for process 3
FINISHED UNIT=0  0  0   ROUTE part 5  in order 0  to machine 0  for process 2
FINISHED UNIT=0  0  0
FINISHED UNIT=1  0  0   ROUTE part 2  in order 0  to machine 3  for process 3
                        ROUTE WU part 6  in order 0  to machine 2 for process 0
FINISHED UNIT=1  0  0   ROUTE part 6  in order 0  to machine 2  for process 1
FINISHED UNIT=2  0  0   ROUTE part 3  in order 0  to machine 1  for process 3

•••

FINISHED UNIT=12  0  0  ROUTE part 12  in order 0  to machine 1  for process 3
FINISHED UNIT=13  0  0
Estimated completion time of Order 0 = 132.19
```

Figure 9 Numerical example of routing parts in the cell using FP rule (with SPT).

```
RANK ORDERS
Rank 0:  order =  9829  index value = 30.33
Rank 1:  order = 16772  index value = 19.90
Rank 2:  order = 15463  index value = 19.77
Rank 3:  order = 10751  index value =  3.62
MAKING DECISION  (idle cell); FIRST THREE ORDERS ARE CONSIDERED
ESTIMATE THE 1ST ORDER'S PROCESSING TIME
  Part_id = 8  Order_id = 9829
  Estimated processing time = 117.42
ESTIMATE THE 2ND ORDER'S PROCESSING TIME
  Part_id = 8  Order_id = 16772
  Estimated processing time = 49.86
ESTIMATE THE 3RD ORDER'S PROCESSING TIME
  Part_id = 12 Order_id = 15463
  Estimated processing time = 115.70
ESTIMATE THE PROCESSING TIME OF PROCESSING THE 1ST AND 2ND ORDER TOGETHER
  Part_id = 8  Order_id =  9829
  Part_id = 8  Order_id = 16772
  Estimated processing time of Order  9829 = 126.88
  Estimated processing time of Order 16772 = 158.49
ESTIMATE THE PROCESSING TIME OF PROCESSING THE 1ST AND 3RD ORDER TOGETHER
  Part_id = 8  Order_id =  9829
  Part_id = 12 Order_id = 15463
  Estimated processing time of Order  9829 = 127.55
  Estimated processing time of Order 15463 = 227.02
ESTIMATE THE PROCESSING TIME OF PROCESSING THE 2ND AND 3RD ORDER TOGETHER
  Part_id = 8  order_id = 16772
  Part_id = 12 order_id = 15463
  Estimated processing time of Order 16772 = 55.57
  Estimated processing time of Order 15463 = 153.56
ESTIMATE THE COMPLETION TIME OF PROCESSING ALL ORDERS TOGETHER
  Part_id = 8  Order_id =  9829
  Part_id = 8  Order_id = 16772
  Part_id = 12 Order_id = 15463
  Estimated processing time of Order  9829 = 126.88
  Estimated processing time of Order 16772 = 159.22
  Estimated processing time of Order 15463 = 257.96
Objective function value for case 0  (1st -> 2nd -> 3rd) = 1827.17
Objective function value for case 1  (1st -> 3rd -> 2nd) = 1779.88
Objective function value for case 2  (2nd -> 3rd -> 1st) = 1806.77
Objective function value for case 3  (2nd -> 1st -> 3rd) = 1881.07
Objective function value for case 4  (3rd -> 2nd -> 1st) = 1759.48
Objective function value for case 5  (3rd -> 1st -> 2nd) = 1747.34
Objective function value for case 6  (1st &  2nd -> 3rd) = 1832.23
Objective function value for case 7  (3rd -> 1st &  2nd) = 1743.45
Objective function value for case 8  (1st &  3rd -> 2nd) = 1783.39
Objective function value for case 9  (2nd -> 1st &  3rd) = 1881.78
Objective function value for case 10 (2nd &  3rd -> 1st) = 1815.60
Objective function value for case 11 (1st -> 2nd &  3rd) = 1825.95
Objective function value for case 12 (1st &  2nd &  3rd) = 1833.79
THE BEST COMBINATION IS 9;          CUMULATIVE OBJECTIVE VALUE = 1758.37
case(2,1&3):  Tnow = 631.17;        objective value for this case = 360.60
CUMULATIVE OBJECTIVE VALUE = 2118.97
CELL IS IDLE
CELL BEGIN OF SERVICE at 631.17;    ORDER 16772 is working
CELL END OF SERVICE at 681.03
```

Figure 10 Numerical example of the hierarchical scheduling algorithm.