The International Journal of Flexible Manufacturing Systems, 3 (1991): 189-211

Effectiveness of Flexible Routing Control

GRACE YUH-JIUN LIN
School of Industrial Engineering, Purdue University, West Lafayette, IN 47907

JAMES J. SOLBERG
School of Industrial Engineering, Purdue University, West Lafayette, IN 47907

Abstract. Flexibility in part process representation and in highly adaptive routing algorithms are two major sources for improvement in the control of flexible manufacturing systems (FMSs). This article reports the investigation of the impact of these two kinds of flexibilities on the performance of the system. We argue that, when feasible, the choices of operations and sequencing of the part process plans should be deferred until detailed knowledge about the real-time factory state is available.

To test our ideas, a flexible routing control simulation system (FRCS) was constructed and a programming language for modeling FMS part process plans, control strategies, and environments of the FMS was designed and implemented. In addition, a scheme for implementing flexible process routing called data flow dispatching rule (DFDR) was derived.

The simulation results indicate that flexible processing can reduce mean flow time while increasing system throughput and machine utilization. We observed that this form of flexibility makes automatic load balancing of the machines possible. On the other hand, it also makes the control and scheduling process more complicated and calls for new control algorithms.

Key Words: flexible manufacturing, routing, scheduling, simulation

1. Introduction

Since the beginnings of the formal study of flexible manufacturing systems (FMSs) some 15 years ago, much progress has been made in understanding the meaning and value of flexibility. Nevertheless, there is still some confusion about the relative worth of different kinds of flexibility. This confusion is no doubt attributable to the mixing of several concepts within the one general category of FMS. Several survey papers (Buzacott and Shanthikumar, 1980; Gerwin, 1982; Browne, Dubois, Rathmill, Sethi, and Stecke, 1984), have attempted to clarify the terminology by defining and categorizing several distinguishable types of flexibility. This article refines the notion of just one type of "process flexibility" and assesses its contribution to improved throughput and flow time.

It is in the very nature of FMSs that the machines are capable of handling a variety of jobs. Sometimes that potential is lost or restricted through tooling assignments or other operational decisions. Usually, however, there is at least some room for choice in the routing of parts through the system. For a given configuration of machines, transporters, tooling, etc. (so that the only controllable parameters relate to decisions about how the assigned work is to be done), it is known that flexible routing can improve throughput and flowtime by better balancing[1] of workloads, so that no one machine or machine type can bottleneck the flow.

It is possible, and generally desirable, to achieve balance or near balance among the average workloads by optimizing the mix of products that are resident in the system at any given time. Several studies have dealt with this problem and the associated tooling allocation problem (Kimemia and Gershwin, 1978; Stecke and Solberg, 1981; Stecke, 1983; Kimemia and Gershwin, 1985; Stecke and Kim, 1988). It has also been pointed out that, if there are alternative ways to produce a single part type, then these alternative process plans can be incorporated into the set of options available in determining the product mix and thereby can contribute to better solutions to the optimization problem. In other words, the different process plans can be treated as if they were for different part types.

The difference in system performance resulting from having such options can be either large or small, depending upon how well the workloads can be balanced without these options. For example, if the system is (temporarily) dedicated to producing just one type of part, which tends to distribute loads quite unevenly, alternative process plans may make a large difference. On the other hand, if the part mix was already varied enough to allow an optimization procedure to nearly balance the workloads, the remaining improvement that is possible from flexible routing may be small.

Until recently, it was necessary to assume that process plans were prepared in advance of production operations. Although it was possible to imagine a preestablished *set* of alternative process plans for a single part type (rather than just one process plan), the whole set had to be prepared well before releasing the job to the system. Furthermore, with the necessity of loading appropriate tools into machines well before they were needed, it was natural to select just one machine sequence for all parts of a given type entering the system during a period. This constraint could be relaxed just slightly if there were duplicate tools available. So, for example, if the next required operation in a plan called for drilling several half-inch holes, any available machine with a half-inch drill bit could conceivably perform the operation. Notice, however, that even in this case of alternative machines, the *sequence* of operations required to produce the part is assumed fixed. This generally is what has been meant by routing flexibility.

Recent research, however, has revealed the opportunity for *dynamic process planning*, in which the choice of the next operation is not made until just before it is to be carried out. Quite literally, one would have no idea what path through the system a part was going to follow at the time it started its processing. Depending upon the availability of tooling, machine queues, and a number of other factors that are dependent on real-time events, the computer would figure out on the fly the best thing to do next. In order to derive any benefit from such flexibility, it would be necessary to have rapid delivery of tooling, and of course some sophisticated control software.

The conditions postulated by this study have not, to our knowledge, ever been implemented in practice. In fact, we are evaluating a higher degree of real-time flexibility than may ever be reasonable to implement. It could also be quite expensive. The purpose of this study was to determine, before going to the effort of creating such options, whether the anticipated gain was worth the effort.

The answer to this question is a qulaified "yes." Under certain reasonable circumstances, the throughput can be improved around 10% over that which occurs with fixed process plans, while the average flow time is reduced by a similar factor. It should be noted, however, that there may be reasons apart from cost and performance to stay with the simpler fixed process plan approach.

This article is organized as follows. In section 2.1, four types of real-time routing flexibility will be described. Section 2.2 provides a concise and complete process plan representation called a part process network that allows flexible routing. General control and scheduling issues when using flexible routing are discussed in section 3. A flexible routing control simulation system is presented in section 4. Section 5 discusses the simulation experiment and results. Conclusions are given in section 6.

2. Definitions

Process planning is a general term referring to the task of establishing the technological requirements necessary to complete a specified part. A process plan usually contains the set of operations to be performed on the part, the order constraints among these operations, and resource requirements such as tooling. *Routing* is the assignment of operations to machines satisfying the technological order constraints, and *scheduling* involves timing specification. Traditionally, process planning, routing decisions, and sometimes even scheduling are done weeks in advance of processing (off-line) and thus produce a process plan with little routing flexibility. Obviously, no matter how these choices are made, performance may suffer when unexpected events occur.

There are two sources that may enhance the flexibility of the control process: flexibility in the part process plans and highly adaptive routing algorithms. In this article, we will study both options and show some promising results from using a flexible part process plan representation in the flexible manufacturing environment.

2.1. Routing flexibility

In this section, we will describe four types of routing flexibility as well as the corresponding process plan representations.

2.1.1. No routing flexibility. Each part must be processed according to a fixed linear sequence of operations that must be conducted at predetermined unique machines. No alternative operations or alternative machines are permitted in real time. A corresponding process plan example is shown in figure la. Traditionally, the sequencing and machine assignments are usually done considering only some technological constraints. Chang, Sullivan, and Bagchi (1984), Dar-El and Sarin (1984), and Wittrock (1988), among others, explore routing optimization by determining the best route for each part type through the system while generating schedules off-line. During operation in real time, only one route is used. Control theory has also been applied to this problem by Hildebrandt and Suri (1983).

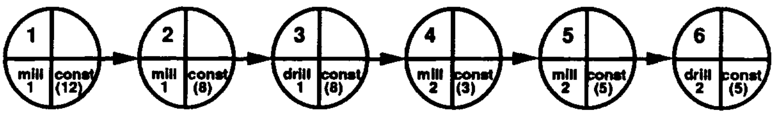

Figure la. Process plan with no routing flexibility.

2.1.2. Fixed sequencing. The operations must be performed in a predetermined linear sequence; however, there could be more than one machine capable of performing any given operation. The routing flexibility provided by this type of process plan is termed *routing flexibility* in Browne et al. (1984) (see figure 1b for the process plan representation). Stecke (1983) and Stecke and Solberg (1985) utilize the flexibility by pooling machines into machine groups and by duplicating operation assignments so that automatic rerouting of parts is attained. Wilhelm and Shin (1985) compared the results of using fixed sequencing process plan with the performance achieved by using no routing flexibility process plan for an FMS aggregate example through simulation and showed that the former can reduce flow time while increasing machine utilization.

2.1.3. Flexible sequencing. Subject only to technologically prescribed precedence constraints, certain operations can be performed in arbitrary order. However, the operations are fixed; that is, there is only one way to achieve a given machined feature. Flexible sequencing is termed *operation flexibility* in Browne et al. (1984). A precedence digraph is usually used to represent flexible sequencing process plans (figure 1c). Buzacott (1982) conjectures that flexible sequencing can improve system performance. Hancock (1989) studied the effects of the flexible sequencing under variable lot-size conditions.

2.1.4. Flexible processing. Neither operations nor sequencing are fixed. All possible alternative operations and sequencing of performing a part are considered in real time to make routing decisions. For example, a slot might ideally be milled in one pass using a cutter

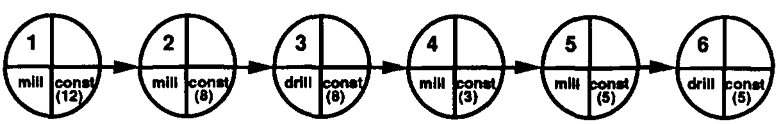

Figure 1b. Fixed sequencing process plan.

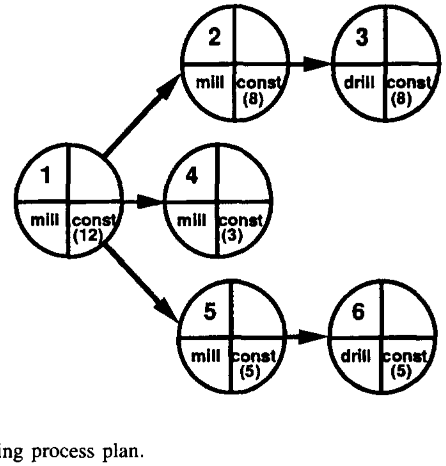

Figure 1c. Flexible sequencing process plan.

of a certain width; however, if that cutter is in use, the same slot could be produced in multiple passes using a smaller tool. Possibly even the raw material might depend upon what is available. The point is that not even the definition of the operation is fixed. Browne et al. (1984) called this type of flexibility *process flexibility*. Parunak et al. (1985) incorporated a hierarchy of process plans that use some flexible processing flexibility in a fractal factory model. A natural, complete, and concise representation for flexible processing process plans, called a part process network, will be presented in the next section.

2.2. *Part process network*

A part process network is a graph representation of the flexible processing process plan. It is an AND-OR digraph that represents the physical precedence constraints in producing a part without imposing unnecessary artificial constraints. All possible operations are represented as nodes. Duplication of nodes is allowed when needed. Information such as the resource requirements can be stored in the nodes also. Directed arcs of the digraph represent the physical precedence relations between the operations; that is, an arc from node A to node B means that operation B can be performed only after operation A is completed. An AND node in the graph means that all of its children operations must be performed. An OR node in the graph means that exactly one of its children operations needs to be performed. In addition to the AND nodes and OR nodes, a third kind of node is introduced to handle the cases when both AND arcs and OR arcs are needed from a node. This kind of node is called the *virtual AND-OR* node; it has zero processing time and is used to group all AND or OR nodes of its parent so that its parent can be classified as an OR or AND node. In figure 1d, we show an example of a simple part process network that is represented by the AND-OR digraph representation.

In the AND-OR graph representation, a node is called *solved* if one of the following conditions is satisfied:

1. The node is a terminal node, i.e., the node has no successors
2. The node is an AND node and all of its children are solved
3. The node is an OR node and exactly one of its children is solved

A solution to an AND-OR graph is given by a subgraph that is sufficient to show that the initial operation node is solved. A sequential part process plan is a topological ordering of a solution of the AND-OR graph of the part process network.

The following is only a small subset of all the sequential part process plans that are possible for the part shown in figure 1d: [1, 2, 4, 3, 8, 5, 6], [1, 2, 3, 4, 8, 5, 6], [1, 8, 7, 2, 3, 4], [1, 2, 8, 7, 4, 3], [1, 8, 5, 2, 3, 4, 6], [1, 4, 2, 3, 8, 7], [1, 2, 8, 7, 3, 4], etc. Actually, 42 different sequential part process plans can be generated for this simple part! This example shows how much flexibility can be lost due to the premature selections of alternatives and sequencing of the part process plans. The loss in flexibility can have a large impact on the performance of the control systems. For example, suppose after processing of operations 1 and 2, all machines that can perform milling are occupied by long jobs. A job process plan of order [1, 2, 4, 3, 8, 7] will have to wait for these machines, even though operations 3 and 7 could be processed immediately. Although this problem could be averted

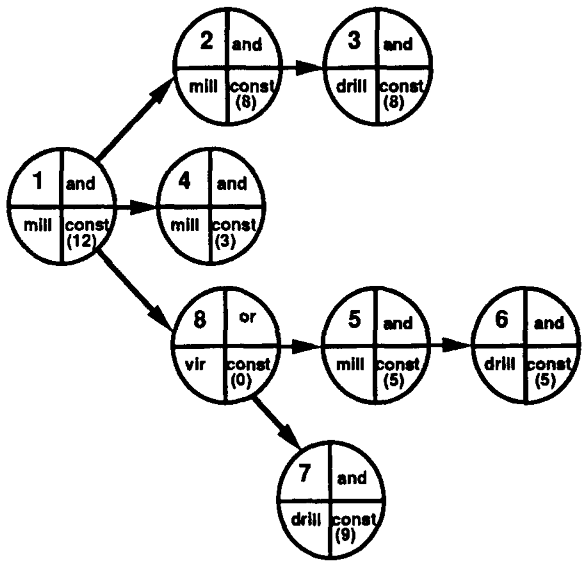

Figure ld. Flexible processing process plan.

by preempting unfinished operations of the milling machine (this is a costly and sometimes impossible option), the flexible part process network can solve this problem without performance degradation and produce better machine utilization and load balancing.

2.3. Flexible routing

Flexible routing in this article is referred to as the flexible processing routing that gives the most general routing flexibility. More specifically, we are interested in testing if the gain in flexibility by using the part process network justifies the added flexibility. General control/scheduling issues with flexible routing will be discussed briefly also.

3. Control and scheduling with flexible routing

When using flexible routing, there are three major control components in a scheduling system: part control, resource control (including controls for machines, tooling, fixtures, transportation, etc.), and the coordination between part and resource controls. Based on the control rules, the environment, and the global criteria of the system, the machine control selects an operation of a part to process. Similarly, a part control evaluates and selects a machine to perform ready operations in its process plan. Each part may have several operations ready to be processed, but the part can only be processed by one machine at a time. Therefore, a part control needs to "negotiate" with machines that are capable of performing the ready operations of the part and to select one ready operation to be processed by an available machine. Although the machine is the major resource to consider, other resources such as tools, fixtures, and transporters all can have great impact on the performance of the system. An operation can only be started when all needed resources are in place. Having dedicated controls on these resources to participate in the negotiation process will allow more thorough consideration and greater flexibility in the control algorithms, and will enhance the capability of the control to make better decisions. Coordination among controls is needed to resolve conflicts, provide communication for the involved controls, and more importantly, lay the groundwork for orchestration of different controls. Issues involved in this process include the communication bandwidth, evaluation of objectives, conflict resolutions, and intelligent decision making.

3.1. Control paradigms

The control in an FMS can be either centralized or distributed. There are actually two kinds of distribution: the distribution of the decision making and the distribution of the information about the system state. Therefore, the control systems can be classified into the following four categories: centralized information–centralized decision making (CICD); distributed information–centralized decision making (DICD); centralized information–distributed decision making (CIDD); and distributed information–distributed decision making (DIDD). Figure 2 depicts the organization of these four alternatives. In control systems that centralize decision making, there is a central control unit that makes the control decisions; in control systems with distributed decision making, the control decisions are made by a set of control units. Communication and cooperation are needed in order to make comprehensive decisions. In control systems with centralized information, all information is routed to a central spot and stored there when the information becomes available. In contrast, in distributed information systems, information is stored locally and is sent to other control entities only when requested. The size and the complexity of the control problem for large manufacturing systems suggests that the control processes need to be highly distributed for these systems (see Solberg, 1989).

3.2. Coordination among part and resource controls

Global coordination among the part and resource controls is quite complicated. The issues include coordination and competition among controls of different types and controls of the same type. The coordination method depends on the paradigm of the control and the selection of the algorithms and evaluation criteria.

One simple coordination scheme is to combine the dispatching rules with a data flow model application. As shall be seen in section 5, this approach provides simple but powerful results.

Another example of a global coordination scheme for distributed decision making is *bidding*. A basic bidding scheme is as follows: the part control takes bids from machine controls and awards an operation to the machine that submitted the best bid (see Parunak et al., 1985; Parunak, 1986; Maley, 1987a; Shaw, 1987, Shaw and Whinston, 1989). More

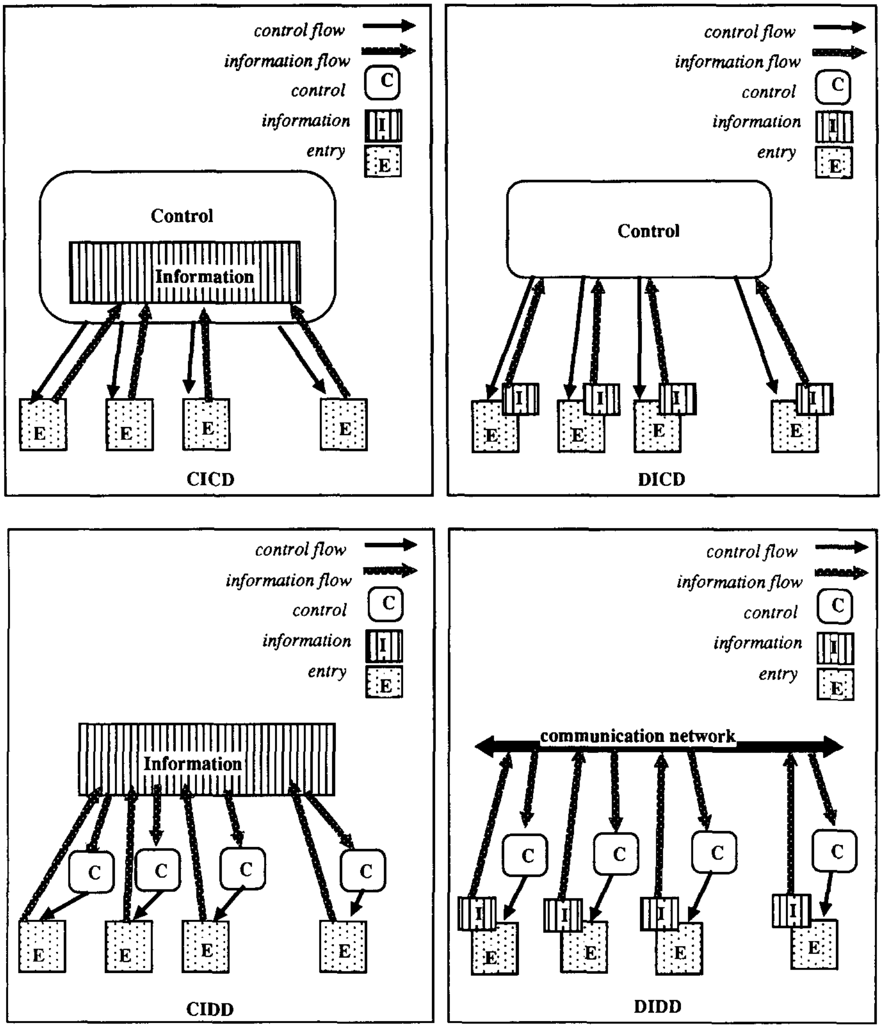

Figure 2. Different control models.

sophisticated bidding schemes include *biased bidding*, *two-way bidding*, and *look-ahead bidding*. In biased bidding, system objectives can be used to adjust or overwrite certain decisions. An expert system can be built to incorporate heuristic and biases in adjusting the weights of the bids. In two-way bidding, part controls send bids to machines that are capable of performing one of its ready operations, and the machine controls send bids to parts that they are "interested" in. A cooperating bid award scheme assigns the final match between the parts and the machines. Acquisition of other resources is similar. Look-ahead bidding is actually a form of planning where bids may be submitted for several operations ahead. In this form of bidding, controls of busy resources also submit bids that take their current and future loads into account.

4. The flexible routing simulation system

We built a simulation system that is called flexible routing control simulation system (FRCS) to study the gain in using the part process network representation and the behaviors of various control strategies in the FMS environment. The simulation system is coupled with a very high-level programming language, called the FRCS language. The FRCS language contains constructs to model FMS environments, part process networks, and the control structures and strategies. The modeling of the FMS environment includes the definition of machines, transportation, central and local buffers, machine breakdowns, factory layout geometry, system states, and other factors common in FMS environments. The part process network is represented by operation nodes and precedence arcs. The operation nodes contain the desired operations, constraints of the operations, and the types of precedence arcs originated from the nodes. The precedence arcs dtermine the execution orders of the operations nodes. The specification of the control includes the control algorithm and strategies, dispatching rules, evaluation functions, and predefined optimization options. A detailed definition of the FRCS language is presented in Lin and Solberg (1989). A sample FRCS program that models a small FMS is shown in appendix A. By employing the FRCS language as the front end to the FRCS simulation system, the simulation can be used to study the effects of different setups of FMSs, various control strategies, different part process representation schemes, etc.

The FRCS simulation system consists of four major components: the language processor, the simulator, the control, and the data collector/reporter. The basic configuration of the system is shown in figure 3. The language processor contains a scanner, a parser, and an

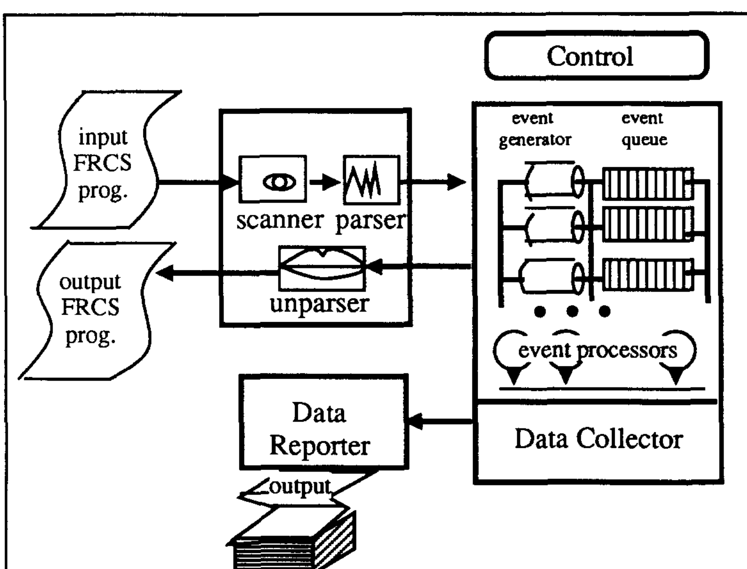

Figure 3. The FRCS simulation system.

unparser. The scanner and the parser are used to compile the FRCS programs and the unparser is used to generate FRCS programs based on the current system state. The simulator is the actual engine that runs the simulation. It is an event-based simulation system and has a list of primitives that act on the simulation events. Actions of the simulator are triggered by the active events under the direction of the control.

The control of the system consists of a set of cooperative control processes: part controls, machine controls, transporter control, and controls for other resources. The structure of the control is specified in the FRCS program and can be programmed to simulate centralized or distributed control processes. The simulation system is operated based on the priority of the entities. Each control has a private or shared priority queue that contains the entities that are related to the control. The priority of an entity is the function of the system states and the entity and is determained by a set of evaluation functions. The evaluation functions can be defined by either built-in functions or user-defined functions. Different evaluation functions can be combined to obtain new evaluation functions. The basic built-in functions are defined by some simple heuristics such as dispatching rules. Since the structure of the simulator is very modular, sophisticated control rules can be embedded easily. The user-defined function facility also allows the user to specify complicated or dedicated control strategies.

The FRCS currently supports nine predefined dispatching rules as shown in appendix B. These dispatching rules serve to define the basic evaluation functions of the system and can be used individually or be combinaed with a set of weights.

The data reporter includes a summary reporter, an event tracer, and a debugger. The summary report provides various statistics and the setting of the simulation. The latter includes the period for statistic collecting, seeds for random number generator, control rules, number of transporters, etc. The former includes statistics for parts (e.g., the numbers of parts released into the system and finished during the simulation period, mean flow, maximum and minimum flow time of each part type) and statistics for machines (e.g., machine utilization, mean and maximum ready queue lengths[2]), as well as mean and maximum lengths of various buffers. The event tracer records information of all events including the event name, event time, duration, part name, entity number, operation number, machine name, etc. The listing of the events forms a complete history of the trace. The debugger can be used to print out all invoked routines during the debugging period and the status of all related entities at the routine invocation time. It also prints the summarized information about transporters, parts, and machines at the end of the simulation. The amount of information produced can be controlled by the tunable debugging degree.

The information provided by the summary reporter can be used by a higher-level analyzer to adjust the setting of the control so that the best control strategy may be discovered by fine tuning the setting of the control automatically. Also, the control of the simulation system forms the core of a real-time system for FMS. The latter can be constructed by replacing the front end of the simulation system with sensors for an actual FMS.

The logic of the simulator is as follows: new part entities arrive at the system at a rate defined by the arrival functions specified in the FRCS program. At any instant, the state of a part entity inside the system is one of the following: being served by a machine, waiting to be served in the input buffer of a machine, waiting for transportation in the central buffer or output buffer of a machine, or being moved. A new part entity or a part entity whose operation is just finished by a machine announces its eligibility by sending its ready operations to the queues of machine controls. Part and machine controls then negotiate with each other to assign a ready operation of a part entity to a machine. Once a part entity is assigned to a machine, the part entity withdraws its requests for operations by pulling all its ready operations out of the ready queues of the machines.

Furthermore, if the machine has a space to hold the part entity, the part entity requests a transporter and waits to be moved to the input buffer of the designated machine. By default, part entities in the local input buffer of a machine get service based on the first-come first-served dispatching rule. When a part is served, an end-of-service event is scheduled to simulate the completion of the operation. Upon completion of the operation, the part entity requests a transporter to move it elsewhere unless the next operation is going to be processed by the same machine. If the machine has space in output buffer, the part entity whose operation was just finished will be moved to the local output buffer of the machine; otherwise, the machine is blocked by this part entity and cannot resume operation until the part entity is moved. When the transporter arrives, if the part entity has not been assigned a machine or the assigned machined has no place to hold it, it will be temporarily moved to the central buffer to free up the output buffer of the machine or avoid blocking the machine. This process continues until the part entity is finished, i.e., the AND-OR process network is solved. Breakdown events are scheduled according to the uptime functions of the machines specified in the FRCS program. When a machine is down, parts that were assigned to the machine (being processed by the machine, in the local input buffer of the machine, being moved to the machine by a transporter, or in the central buffer or local output buffer of other machines waiting for transporter assignment or arrival) are rescheduled and moved to a new destination machine or central buffer (if no machine is available). And a fixed event is scheduled according to the downtime function specification. When a machine is up again, it resumes its operation by declaring its status to be idle and invokes the machine control routine to look for parts.

5. Experiment with flexible routing control

The major objective of this experiment is to test the feasibility and advantages of using the part process network as input to the control of manufacturing systems. Some experiments have been conducted based on the current implementation of FRCS. In particular, we tested nine different dispatching rules on part process networks of different complexities in different manufacturing environments. We also compared the results with flexible sequencing and fixed sequencing part process plans for the same parts. Note that the routing for these three different types of process plans is all flexible in the sense that each operation can be performed in any machine capable of performing it; the differences are in the options and orders of performing it.

Our initial results are very encouraging. We found that even with simple control rules, the flexibility provided by the part process network reduces the mean flow time and increases the throughput and average machine utilization in most cases. In the following, we will present a typical example of our experiment including the experimental setup, control and routing strategies, FMS environment, the results, and the analysis of the results.

5.1. The routing strategy

In this experiment, we adopted a strategy that we call (DFDP). DFDP is a combination of dispatching rules and application of a data flow model (Lewis, Barash, and Solberg, 1987); it provides a simple means for the part-machine coordination. Under the DFDP model, the control of the part has a copy of the part process network to keep track of the current ready operations and their status. When an operation of a part is completed by a machine, the part control updates the ready operations and posts all possible ready operations to the queues of the machines that are suitable to process the operations. Identical machines share one priority queue that orders the ready operations based on the priority calculated by the evaluation functions based on the dispatching rules. When a machine becomes idle or there is a change in the queue, it checks the priority queue and picks the operation-entity with highest priority to perform. When an operation is selected by a machine, the entity that the operation belongs to withdraws all its ready operations from queues of other machines. Also, requests are made to acquire other needed resources such as transporters.

To simplify the experiment, tool changes and fixtures are not considered and transporter control uses the dispatching rules. In this model of control, the part controls and other resource controls have minimal impacts, and the flexibility of the part process plans becomes the major factor in the behavior of the system. In this way, we can test the system with simple dispatching rules and see how much performance gain we can obtain by using part process networks without sophisticated controls.

5.2. The environment

The FMS we model in this experiment consists of three milling machines with the same processing rates, two drilling machines with same processing rates, a load/unload station, a central buffer, some machine local input and output buffers, and a set of automated guided vehicles (AGVs). The machines' uptime and downtime are assumed to have negative exponential distributions with mean 1440 minutes and 60 minutes for milling machines and with mean 960 minutes and 60 minutes for drilling machines. The size of the central buffer and all local buffers, slack buffer sizes,[3] and number of transporters are the experimental parameters. The layout of the system is shown in figure 4. It was assumed that the AGVs traveled along shortest paths on the AGV pathway; no collision occurred on their paths of travel. It was also assumed that the AGVs travel at a constant speed of 0.75 feet per second and the loading and unloading time are 10 seconds each. A simple part type with AND-OR network process plan as shown in figure 1d was used. Two flexible sequencing process plans and two fixed sequencing process plans are also constructed from the network process plan. The part arrival rate is assume to be constant, and a new part will enter the system if the number of spaces available in the central buffer is greater than the slack buffer size.

Because of the regularity of the loading policy and the simplicity of the model, steady-state behavior of the system is readily achieved. The mean flow time, throughput, and machine utilizations are then collected for one week, two shift periods (4800 minutes) for each process plan and dispatching rule. The corresponding FRCS program for this FMS setup and the network part process plan is shown in appendix A.

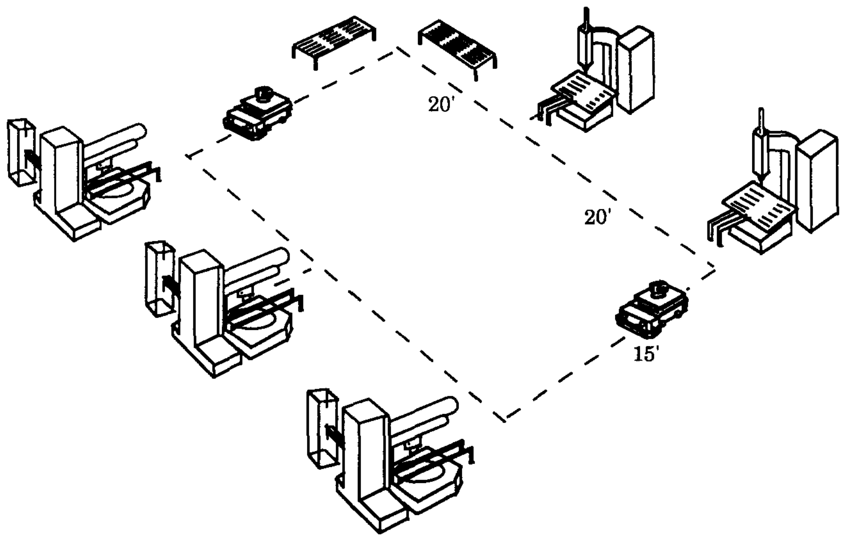

Figure 4. The FMS layout.

5.3. Results

Different parameter settings have been chosen to test the results. We found that the results are quite consistent—the network part process plan is better in all measures in most experiments. With realistic setting of the FMS parameters, the system shows a consistent improvement of around 10%. After the FMS reaches its capacity, increasing central buffer size does not increase its throughput, but the mean flow time increases rapidly. As the buffer size increases asymptotically to infinity, the part process network shows a much better mean flow time improvement over other part process representation schemes (250% as reported in Lin and Solberg, 1989). We found that with the no-look-ahead DFDP control strategy, the performance of the system is consistently better for different part process representations when using a lean loading policy and small input and output buffer sizes. We also noted that a small number of AGVs are sufficient for our FMS environment, since travel time is assumed small compared to machining time. Increasing the number of AGVs resulted in only slight performance improvement. Therefore, the data we presented was based on the following realistic FMS settings: part interarrival time is seven minutes; the sizes of central buffer, slack buffer, local input buffer, and local output buffer are ten, three, zero, zero, respectively, for the cases that ignore transportation, and six, four, one, one, respectively, for the cases with transportation taken into account. The number of AGVs is two, and they are stationed in the last dispatched location until the next call for service.

Figure 5-7 summarize the mean flow time, throughput, and average machine utilization among the flexible processing network, two flexible sequencing process plans, and two fixed sequencing process plans for the case in which both transportation delays and failures occur. Many more cases are covered by the data shown in appendix C. The performance

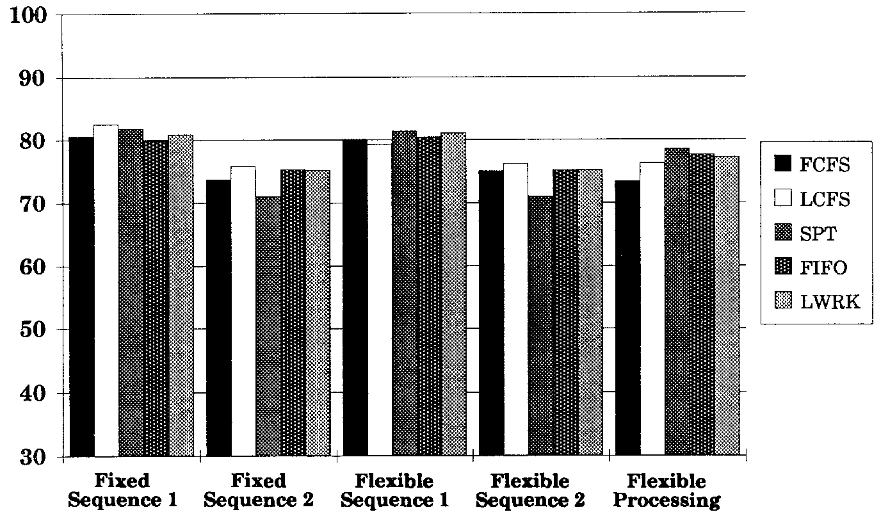

Figure 5. Average flow time comparison.

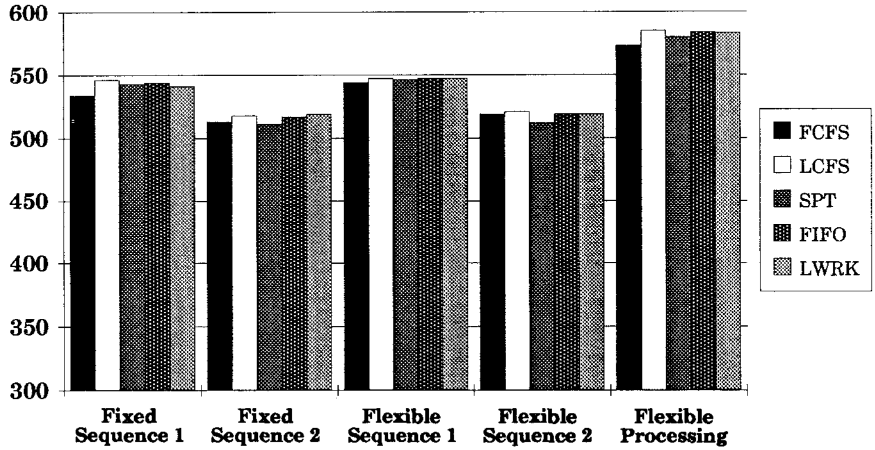

Figure 6. Throughput comparison.

measures with or without transportation and with or without machine failures, and for more dispatching rules, are provided there. Note that using the network plan, machines can achieve a very high utilization.

The flexible processing part process network is better for most of the cases and dispatching rules tested, because it has fewer constraints and more options (thus more flexibility) than the other part process plans. The throughput and average machine utilization is increased by around 10% compared to the flexible sequencing network and fixed sequencing process

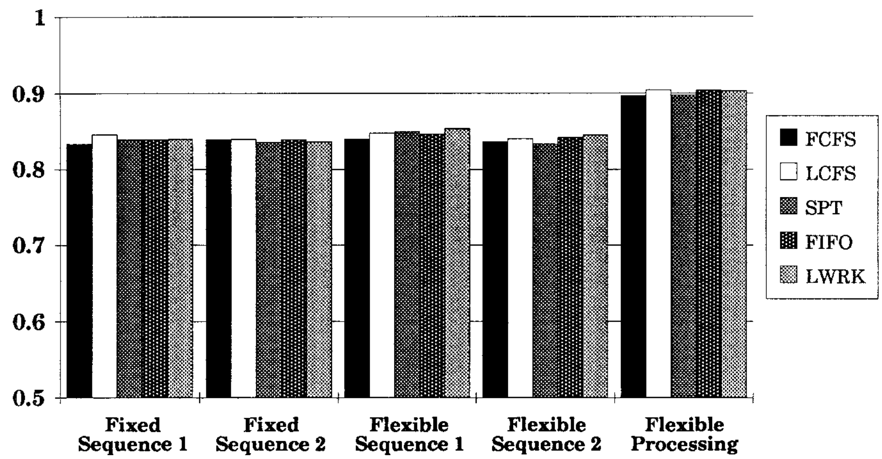

Figure 7. Utilization comparison.

plans. The average flow time comparison appears to indicate that some cases of fixed sequence of flexible sequence rules produce lower flow time than the flexible processing, but this is somewhat misleading because those cases also produced far fewer complete parts.

The superiority of the flexible processing network can be attributed to its ability to direct parts away from machines that are heavily loaded. Although the chosen path might appear to take longer processing time (in static terms), it achieves the effect of automatic load balancing. This example shows that even parts with very simple part process networks and controls with unsophisticated control rules can benefit from the flexibility in the part process networks. Network plans were expected to perform better when machine breakdown occurred. However, because of the consistency of the loading policy and the existence of identical machines, the percentage of the gain is about the same as those without breakdown.

It is interesting to note that only slight or no improvement has been observed by using flexible sequencing network in the experiments. This is in part due to the fact that we are using only the simple DFDP control rules in the experiment. When the operations of the part are waiting for services from the machines, the first machine that picks a ready operation of the part actually decides the routing of the part. There is no guarantee that the best active operation is chosen by the machine it is assigned to, because the machine that selects the operation has no knowldge of the other operations of the part. Therefore, a busy machine may be avoided in the short terms, but it must be revisited later, and it is possible that the situation of the machine at that time may be even worse. The sequencing flexibility defers the problem but doesn't solve it, and the net long-term effect can be worse. This supports our suggestion that coordination between controls of the parts and the machines needs to be studied in more detail.

We observed that among different dispatching rules of the machines, FCFS and SPT work better than other rules for this particular example. There are some cases of surprisingly good performance for LCFS, LIFO, LPT rules, but, after we examined the parts history, we found that there were deadlocks using these rules, that is, some parts were stuck in the system for a long time. Therefore, unless there are good deadlock resolution strategies, these rules are not recommended.

We also did some experiments by changing the part arrival rate so that the system utilization is decreased to about 70%. We found that the benefit of the flexibility in the part process network becomes negligible. This is because only dispatching control rules are used and also because most parts get service very promptly even with a fixed sequencing process plan, which makes the flexibility in part process plans less important. We also noted that only the flexibility of the part process network is exploited in the above experiment.

In general, there is usually more information about the merits of the various routes available at the part dispatching time. Parts can discriminate operations (routes in the graph) that are not very promising and avoid these paths by excluding them from the subset of machines to wait for service. Adding this part control to the decision-making process will enhance the intelligence of the control and may be expected to produce better results.

We are currently investigating control algorithms for part dispatching and the coordination of part and machine controls. Note that the part dispatching controls for flexible processing networks are much more general (and complicated) than the controls for sequential part process plans. This is because the latter has no choice in operation selections and very few choices in machine selections. We do expect that with better part control strategies, the flexible processing part process network will show even more improvements over the sequential part process plans in FMSs.

6. Conclusions

In this article, we have shown that the flexible processing part process network offers an opportunity for performance improvement. We argue that, when feasible, the choices of operations and sequencing of the part process plan should be deferred until detailed knowledge about the real-time factory state is available. This form of flexibility makes automatic load balancing of the machines possible. On the other hand, it also makes the control and scheduling process more complicated and calls for new control algorithms.

As a conclusion, we note that the flexible processing process plan has very high potential in improving the control and scheduling of part shops and the FMS; and the flexible routing scheduling will have significant impact on the control and scheduling of the FMS.

Acknowledgments

We would like to thank Dr. Ted C. Chang and Dr. Ben Montreuil for their valuable suggestions. We would also like to acknowledge the effort put forth by the editor, the associate editor, and reviewers of this article. Their comments and suggestions were of great help in revising the article.

This work was supported by the Engineering Research Center for Intelligent Manufacturing Systems under a grant from the National Science Foundation, number CDR 8803017, with additional funding from the Defense Advanced Research Projects Agency (DARPA).

Appendix A

A sample FRCS program that models the FMS of three milling machines, two drilling machines, two AGVs, and one part as described in section 5.2:

```
    /* Header Section */
TITLE: test_program                  /* Title of the Program */
RUN: 0.0, 4800;                      /* Simulation Period */
SEED: 12345;                         /* Random Number Seed */
RULE: FCFS;                          /* Dispatching Rules */
BUFFER_SIZE: 6;                      /* Central Buffer Size */
SLACK_BUFFER: 3;                     /* Slack Buffer Size */
IN_BUFFER: 1;                        /* Local Input Buffer Size */
OUT_BUFFER: 1;                       /* Local Output Buffer Size */

TRACE: 200.0, 600.0;                 /* Trace Period */

    /* Resource Section */
RESOURCE                             /* Resource Block Begins */

  MACHINE                            /* Machine Block Begins */
    mill: 3;                         /* Three Identical Milling Machines */
    drill: 2;                        /* Two Identical Milling Machines */
  END MACHINE;                       /* Machine Block Ends */

                /* May have Tool or Fixture Blocks Here */

  TRANSPORTATION                     /* Transportation Block Begins */
    AGV: 2;                          /* Two AGVs in the system */
    DISTANCE    /* AGV Travel Time Computed from the layout Figure 4 */
      MACHINE 1, MACHINE 2: 0.41; /* The Order of Machines is the Order
                                     Specified in Machine Block */
      MACHINE 1, MACHINE 3: 0.66;
      MACHINE 1, MACHINE 4: 0.85;
      MACHINE 1, MACHINE 5: 0.60;
      MACHINE 2, MACHINE 3: 0.41;
      MACHINE 2, MACHINE 4: 0.60;
      MACHINE 2, MACHINE 5: 0.85;
      MACHINE 3, MACHINE 4: 0.35;
      MACHINE 3, MACHINE 5: 0.60;
      MACHINE 4, MACHINE 5: 0.41;
      CENTRAL_BUFFER, MACHINE 1: 0.35;
      CENTRAL_BUFFER, MACHINE 2: 0.60;
      CENTRAL_BUFFER, MACHINE 3: 1.02;
      CENTRAL_BUFFER, MACHINE 4: 0.66;
      CENTRAL_BUFFER, MACHINE 5: 0.41;
  END DISTANCE;
END TRANSPORTATION;                  /* Transportation Block Ends */
```

```
  BREAKDOWN: MACHINE mill;          /* Breakdown Block Begins */
                              /* Breakdown Specification for Milling Machine /
    UPTIME: EXPON(480.0,0.0,0.0);   /* The Uptime for each milling machine
                                    has mean 1440 minutes */
    DOWNTIME: EXPON(60.0,0.0,0.0);  /* The Downtime has mean 60 minutes */
  END BREAKDOWN;                    /* Breakdown Block for mill Ends */

  BREAKDOWN: MACHINE drill;         /* Breakdown Block for Drilling Machines */
    UPTIME: EXPON(480.0,0.0,0.0);
    DOWNTIME: EXPON(60.0,0.0,0.0);
  END BREAKDOWN;
END RESOURCE;                       /* Resource Block Ends */

    /* Part Definition Section */
PART__LIST                          /* Part Block Begins */
  PART part__A, P1;                 /* Part Name and Part Number */
    ARRIVE: CONST(0.0,0.0,0.0);     /* First Arrival Time */
    INTERVAL: CONST(7.0,0.0,0.0);   /* Interarrival Time */
    PART__GRAPH                     /* Part Graph Block Begins */
      OPERATIONS                    /* Operation Block Begins */
        1: mill, CONST(12.0,0.0,0.0), AND;
                /* Operation 1 Can Be Processed on Milling Machines,
                 * Processing Time is Constant 12,
                 * It is an AND Node */
        2: mill, CONST(4.0,0.0,0.0), AND;
        3: drill, CONST(8.0,0.0,0.0), AND;
        4: mill, CONST(3.0,0.0,0.0), AND;
        5: mill, CONST(5.0,0.0,0.0), AND;
        6: drill, CONST(5.0,0.0,0.0), AND;
        7: drill, CONST(9.0,0.0,0.0), AND;
        8: virtual, CONST(0.0,0.0,0.0), OR;
      END OPERATIONS; a             /* Operation Block for Part A Ends */
      RELATIONS                     /* Relation Block for Part A Begins */
        1 -> 2;   /* Operation 1 Has To Be Done Before Operation 2 */
        1 -> 4;
        1 -> 8;
        2 -> 3;
        8 -> 5;
        5 -> 6;
        8 -> 7;
      END RELATIONS;                /* Relation Block Ends */
    END PART__GRAPH;                /* Part Graph Block Ends */
  END PART;
                                    /* Can Have More Part Blocks */
END PART__LIST;                     /* Part List Block Ends */
```

Appendix B

Below are the dispatching rules implemented in FRCS.

Symbol	Rule: From among the jobs in queue, choose the one that:
FCFS	— entered first this queue
FIFO	— entered first into the system
LCFS	— entered last this queue
LIFO	— entered last into the system
LPT	— has the longest processing (imminent operation) time
LWRK	— has the smallest remaining processing time
MWRK	— has the largest remaining processing time
SPT	— has the shortest processing (imminent operation) time
FLEX	— has the largest number of children operations

Appendix C

The following data relates to the system shown in figure 4. Fixed Sequence 1, Flexible Sequence 1, and Flexible Process are shown in figures lb, lc, and ld, respectively. Fixed Sequence 2 consists of operations [1, 2, 3, 4, 7] in that fixed order. Flexible Sequence 2 replaces operations 5 and 6 by 7 in figure lc.

For each combination of dispatching rule and process plan, four data results are given in a two-by-two cell. The upper left number represents the total number of parts completed. The upper right number represents the average flow time for those parts. Note that some parts remaining in the system at end of the simulated period are not included in this calculation; for some dispatching rules (especially LCFS and LIFO), which may retain parts for long residency times, this mean flow time may underestimate a true long-term mean value. The lower left and lower right numbers represent the mean utilizations of the drilling machines and milling machines, respectively.

Table C1. Results without transportation and without failures

Rule	Fixed Sequence 1		Fixed Sequence 2		Flexible Sequence 1		Flexible Sequence 2		Flexible Process	
FCFS	593	85.41	560	78.84	593	85.58	559	80.04	650	76.08
	0.828	0.999	0.996	0.808	0.829	0.999	0.996	0.924	0.996	0.999
LCFS	593	75.67	560	71.73	593	59.98	561	54.50	640	76.78
	0.816	0.999	0.996	0.780	0.832	0.999	0.996	0.900	0.969	0.998
SPT	594	84.85	559	81.94	595	85.63	558	72.9	640	80.15
	0.750	0.999	0.996	0.848	0.871	1.000	0.996	0.903	0.960	0.999
FIFO	594	82.25	561	82.21	596	87.14	561	81.07	638	80.27
	0.826	0.999	0.996	0.826	0.810	0.999	0.996	0.925	0.964	0.999
LWRK	594	82.25	561	82.21	596	87.14	561	81.07	641	79.6
	0.826	0.999	0.996	0.826	0.810	0.999	0.996	0.925	0.969	0.999
FLEX	593	85.41	560	78.84	592	84.7	559	80.04	649	76.48
	0.828	0.999	0.996	0.808	0.838	0.999	0.996	0.924	0.996	0.999
LIFO	591	54.9	560	74.72	593	70.28	559	57.72	644	65.84
	0.827	0.999	0.996	0.773	0.819	0.999	0.996	0.925	0.996	0.988
LPT	592	83.32	561	82.21	594	73.53	559	74.55	618	59.83
	0.816	0.999	0.996	0.826	0.857	0.995	0.996	0.925	0.917	0.992
MWRK	591	87.08	560	76.59	592	81.89	559	74.55	642	73.39
	0.827	0.999	0.996	0.773	0.857	0.999	0.996	0.925	0.996	0.978

Table C2. Results without transportation and with failures

Rule	Fixed Sequence 1		Fixed Sequence 2		Flexible Sequence 1		Flexible Sequence 2		Flexible Process	
FCFS	544	89.31	517	86.74	551	89.2	519	85.23	596	79.14
	0.760	0.943	0.919	0.757	0.777	0.944	0.921	0.852	0.923	0.927
LCFS	544	86.85	515	82.35	552	87.34	523	63.01	593	81.34
	0.766	0.944	0.915	0.752	0.779	0.946	0.923	0.851	0.891	0.945
SPT	553	87.97	514	84.37	551	87.41	517	80.87	590	80.95
	0.762	0.945	0.914	0.770	0.775	0.944	0.920	0.835	0.889	0.941
FIFO	552	88.28	521	88.07	556	90.3	523	87.38	594	84.56
	0.773	0.946	0.921	0.780	0.791	0.952	0.923	0.856	0.888	0.949
LWRK	552	88.28	521	87.88	556	90.32	523	87.37	593	84.31
	0.773	0.946	0.920	0.780	0.768	0.950	0.923	0.856	0.885	0.951
FLEX	544	89.31	517	86.74	547	88.74	521	85.27	597	79.26
	0.760	0.943	0.919	0.757	0.771	0.937	0.922	0.852	0.922	0.932
LIFO	535	88.09	509	82.96	547	87.41	516	72.34	587	77.05
	0.750	0.932	0.911	0.744	0.762	0.938	0.923	0.847	0.913	0.920
LPT	541	89.12	514	89.03	545	76.83	520	79.69	568	70.32
	0.749	0.939	0.915	0.765	0.768	0.931	0.923	0.838	0.861	0.929
MWRK	538	87.9	509	87.09	535	84.46	516	79.82	582	73.2
	0.753	0.934	0.909	0.740	0.752	0.925	0.919	0.849	0.913	0.919

Table C3. Results with transportation and without failures

Rule	Fixed Sequence 1		Fixed Sequence 2		Flexible Sequence 1		Flexible Sequence 2		Flexible Process	
FCFS	592	78.94	560	69.03	593	78.56	559	68.95	635	72.74
	0.808	0.997	0.995	0.754	0.806	0.998	0.994	0.759	0.978	0.971
LCFS	594	79.23	560	72.15	594	77.47	561	69.22	630	74.24
	0.808	0.998	0.995	0.745	0.808	0.998	0.995	0.749	0.933	0.997
SPT	595	79.49	553	64.81	595	80.63	557	63.99	635	77.26
	0.809	0.998	0.986	0.742	0.867	0.999	0.991	0.750	0.946	0.998
FIFO	592	78.35	560	71.44	594	79.69	561	69.83	637	76.96
	0.806	0.998	0.995	0.765	0.807	0.998	0.995	0.750	0.952	0.997
LWRK	594	81.47	560	71.44	594	79.03	561	69.83	638	76.43
	0.808	0.998	0.995	0.765	0.807	0.998	0.995	0.750	0.953	0.997
FLEX	592	78.94	560	69.03	594	74.86	559	68.95	640	70.15
	0.808	0.997	0.995	0.754	0.808	0.998	0.994	0.759	0.978	0.983
LIFO	594	77.2	558	65.45	592	73.64	555	63.09	643	72.28
	0.808	0.997	0.989	0.745	0.805	0.995	0.985	0.742	0.980	0.988
LPT	593	77.96	560	71.44	591	74.11	553	64.2	620	71.33
	0.808	0.998	0.995	0.765	0.805	0.992	0.981	0.739	0.911	0.991
MWRK	592	74.42	553	64.81	591	73.31	553	64.2	644	73.14
	0.805	0.994	0.986	0.742	0.806	0.997	0.981	0.739	0.982	0.992

Table C4. Results with transportation and failures

Rule	Fixed Sequence 1		Fixed Sequence 2		Flexible Sequence 1		Flexible Sequence 2		Flexible Process	
FCFS	534	80.57	513	73.70	544	80.03	519	75.03	573	73.35
	0.741	0.926	0.915	0.763	0.749	0.931	0.920	0.752	0.896	0.898
LCFS	546	82.50	518	75.79	547	79.24	521	76.15	585	76.22
	0.758	0.934	0.915	0.765	0.757	0.938	0.921	0.760	0.871	0.938
SPT	543	81.81	511	71.05	546	81.31	512	70.95	580	78.47
	0.748	0.930	0.912	0.759	0.762	0.936	0.911	0.755	0.873	0.922
FIFO	544	79.79	517	75.29	547	80.43	519	75.12	584	77.61
	0.748	0.930	0.915	0.763	0.756	0.937	0.917	0.767	0.877	0.932
LWRK	541	80.80	519	75.15	547	81.05	519	75.15	583	77.19
	0.755	0.925	0.916	0.757	0.770	0.937	0.917	0.774	0.879	0.927
FLEX	534	80.57	513	73.70	538	78.85	519	74.69	582	75.23
	0.741	0.926	0.915	0.763	0.748	0.928	0.921	0.766	0.902	0.916
LIFO	533	78.90	505	70.65	539	75.95	511	69.34	580	74.26
	0.741	0.922	0.899	0.749	0.748	0.929	0.909	0.770	0.891	0.915
LPT	539	79.75	511	75.23	539	80.63	506	68.73	572	76.14
	0.747	0.926	0.907	0.755	0.754	0.928	0.900	0.744	0.850	0.925
MWRK	535	77.50	496	69.19	542	78.23	508	68.26	579	73.16
	0.741	0.920	0.881	0.748	0.751	0.934	0.903	0.742	0.888	0.915

Notes

1. Strictly speaking, exact equality of workloads is not an optimal condition (see Stecke and Solberg, 1985). However, this is a fine point that does not materially affect this discussion; we will use the term *balance* to mean assigning the work to achieve the correct relative average proportions among the stations.
2. The ready queue length is defined as the number of operations ready to be served in this machine, and it could be in the queue of other machines; also, a part may have several ready operations because of the network process structure.
3. The slack buffer is the space in the central buffer reserved for parts already in the system; that is, new parts cannot occupy the slack buffer. This concept is important in avoiding deadlocking.

References

Browne, Jim, Dubois, Didier, Rathmill, Keith, Sethi, Suresh P., and Stecke, Kathryn E., "Classification of Flexible Manufacturing Systems," *The FMS Magazine*, pp. 114–117 (April 1984).

Buzacott, J.A., "Optimal Operating Rules for Automated Manufacturing Systems," *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 1 (February 1982).

Buzacott, J.A. and Shanthikumar, J.G., "Models for Understanding Flexible Manufacturing Systems," *AIIE Transactions*, Vol. 12, pp. 339–350 (December 1980).

Chang, Y.L., Sullivan, R.S., and Bagchi, U.S., "Experimental Investigation of Real-Time Scheduling in Flexible Manufacturing Systems," *Proceedings of the First ORSA/TIMS Conference on Flexible Manufacturing Systems*, K.E. Stecke and R. Suri (eds.), The University of Michigan, Ann Arbor, MI, pp. 307–312 (August 1984).

Dar-El, E.M. and Sarin, S.C., "Scheduling Parts in FMS to Achieve Maximum Machine Utilization," *Proceedings of the First ORSA/TIMS Conference on Flexible Manufacturing Systems*, The University of Michigan, Ann Arbor, MI, pp. 300–306 (August 1984).

Gerwin, D., "Do's and Don'ts of Computerized Manufacturing," *Harvard Business Review*, Vol. 60, No. 2, pp. 107–116 (1982).

Hancock, Terence M., "Effects of Alternate Routings under Variable Lot-Size Conditions," *International Journal of Production Research*, Vol. 27, No. 2, pp. 247–259 (1989).

Hildebrandt, R.R. and Suri, R., "Methodology and Multilevel Algorithm Structure for Scheduling and Real Time Control of Flexible Manufacturing Systems," *Proceedings, 3rd International Symposium on Large Engineering Systems*, Memorial University of Newfoundland, Canada (July 1983).

Kimemia, Joseph and Gershwin, Stanley B., "Network Flow Optimization in Flexible Manufacturing Systems," *Proceedings of the IEEE Conference on Decision and Control*, pp. 633–639 (December 1978).

Kimemia, Joseph and Gershwin, Stanley B., "Flow Optimization in Flexible Manufacturing Systems," *International Journal of Production Research*, Vol. 23, No. 1, pp. 81–96 (1985).

Lewis, W.C., Barash, M.M., and Solberg, J.J., "Computer Integrated Manufacturing System Control: A Data-Flow Approach," *Journal of Manufacturing Systems*, Vol. 6, No. 3, pp. 177–191 (1987).

Lin, Yuh-Jiun and Solberg, James J., "A Flexible Control Simulation System for FMS," Technical Report, Engineering Research Center for Intelligent Manufacturing Systems, Purdue University, West Lafayette, IN (1989).

Lin, Yuh-Jiun and Solberg, James J., "Flexible Routing Control and Scheduling," in *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems*, K.E. Stecke and R. Suri (Eds.), Elsevier Science Publishers B.V., Amsterdam, pp. 155–160 (August 1989).

Maley, James G. and Solberg, James J., "Part Flow Orchestration in CIM," *Proceedings of the International Conference on Production Research*, Cincinnati, OH, pp. 17–20 (August 1987).

Maley, James G., "Managing the Flow of Intelligent Parts," *International Conference on the Manufacturing Science, Technology of the Future*, MIT, Cambridge, MA (June 1987).

Parunak, H.V.D., Irish, Bruce V., Kindrick, James, and Lozo, Peter W., "Fractal Actors for Distributed Manufacturing Control," *The Second Conference on Artificial Intelligence Applications*, pp. 653–660 (December 1985).

Parunak, H.V.D., "Manufacturing Experience with the Contract Net," Technical Report ITI TR-86-36, Communications and Distributed Systems Laboratory, Industrial Technology Institute, Ann Arbor, MI (1986).

Shaw, Michael J.P., "Distributed Planning in Cellular Flexible Manufacturing Systems," *Infor*, Vol. 25, No. 1, pp. 13–25 (1987).

Shaw, Michael J.P. and Whinston, Andrew B., "A Distributed Knowledge-based Approach to Flexible Automation: The Contract Net Framework," *International Journal of Flexible Manufacturing Systems*, Vol. 1, No. 1 (September 1988).

Solberg, James J., "Production Planning and Scheduling in CIM," *Proceedings of the 11th World Computer Congress*, IFIP, San Francisco, CA (August 1989).

Stecke, Kathryn E., "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems," *Management Science*, Vol. 29, No. 3, pp. 273–288 (March 1983).

Stecke, Kathryn E. and Kim, Ilyong, "A Study of FMS Part Type Selection Approaches for Short-Term Production Planning," *International Journal of Flexible Manufacturing Systems*, Vol. 1, No. 1, pp. 7–29 (September 1988).

Stecke, Kathryn E. and Solberg, James J., "Loading and Control Policies for a Flexible Manufacturing System," *International Journal of Production Research*, Vol. 19, No. 5, pp. 481–490 (1981).

Stecke, Kathryn E. and Solberg, James J., "The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multiserver Queues," *Operations Research*, Vol. 33, No. 4, pp. 882–910 (July–August 1985).

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

Wittrock, R.J., "An Adaptive Scheduling Algorithm for Flexible Flow Lines," *Operations Research*, Vol. 36, No. 3, pp. 445–453 (1988).