Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

EXPLORATION OF A MINIMUM TARDINESS DISPATCHING PRIORITY FOR A FLEXIBLE MANUFACTURING SYSTEM — A COMBINED SIMULATION / OPTIMIZATION APPROACH

Susumu Morito
Keun Hyung Lee

Department of Industrial
Engineering and Management
School of Science and Engineering
Waseda University
Shinjuku, Tokyo 169

Kiyohisa Mizoguchi
Yamazaki Mazak Corporation
Oguchi-cho, Aichi, 480-01

Hiroshi Awane
Yamazaki Mazak Corporation
Ark-Mori Bldg.,1-12-32
Akasaka,Minato,Tokyo 107

This paper presents a simulation-optimization approach to find an appropriate dispatching priority. The study is based on a detailed simulator for a module-type commercial FMS. Specifically, after presenting the basic configuration and fundamental control logic of the system together with its main characteristics as a special type of a job shop, an algorithm is presented which combines simulated annealing and simulation to explore a dispatching priority of operations that minimizes the total tardiness. Computational performance of the algorithm is presented which shows that good solutions can be obtained mostly after 300 iterations or so.

The paper also compares the performance of the "optimal" or near optimal dispatching priority generated by the proposed algorithm with those generated by standard dispatching rules. The results show that standard dispatching rules such as EDD and SLACK which take due dates into considerations perform moderately, but not extremely well. The results also show that some of the workload-based dispatching rules such as SPT work consistently well.

1 INTRODUCTION

A determination of a proper dispatching priority of jobs is one of the most important decisions for efficient daily operations of a shop floor. There, meeting externally specified due dates is often regarded as the most important.

The authors have developed an extensive simulator for a commercial module type flexible manufacturing system (FMS) made and marketed by Yamazaki Mazak Corporation. The system is called the Mazatrol FMS (sometimes abbreviated as M-FMS). More details of the developed simulator together with analysis based on it can be found in Morito et al.(1991,1992). Since the simulator could be run at a shop floor based on the "current status" of the system, its repeated executions would allow us to find an appropriate control of the FMS so that some measure of performance is optimized.

2 SYSTEM DESCRIPTION

2.1 General Outline of the System

The Mazatrol FMS studied in this paper is a commercial module-type medium- to small-scale FMS, and more than 200 sets have been installed and currently in operation world-wide. The key ingredient of the system is horizontal machining centers, which we simply call machines, that are interconnected to load/unload stations (LUL) and pallet stocker by a stacker crane, which is the material handling device for this FMS. Other components of the system include pallets, fixtures, operators, and an FMS computer. Main roles of human operators are load/unload operations, changing tools, and also some software decisions to specify dispatching priority of parts and operations. For more details of the system, see, e.g., Morito et al.(1991).

2.2 Model Assumptions

We list basic assumptions of the model.

1) Jobs to be processed correspond to a finite number of distinct part types. For each part type, a predetermined production quantity is known which we call the "lot size". Processing of a specific part type consists of several "operations", where each operation can be performed without interuption on a single machine. A given series of operations for a particular part must be performed in the prescribed order.

2) A part (or parts) is held on a pallet with operation-dependent fixtures. The number of pallets for a particular operation is a fixed constant during a short time period, which is often a lower single digit number.

3) A pallet and the associated part(s) must be returned to LUL between any two operations. At LUL,

a part is removed from the pallet and the next part requiring the same operation, if available, is loaded on this pallet. Subsequently, the removed part will be loaded on the pallet for the "next" operation, if one exits.

4) What we call an "operation" generally consisits of many "sub-operations" and the number of tools required for an operation often reaches as high as 30 - 50. All these sub-operations will be performed automatically based on the prescribed NC programs, and the time required for tool changes is practically negligible with the help of automatic tool changer (ATC). In this study, however, we assume that tool assignments are already given and the machine(s) which can process a particular operation is known, and thus we pay no attention to sub-operations. Those machine(s) that can process a particular operation is sometime referred to as the "candidate machine(s)" for the operation.

5) Raw materials for the first operation of all parts are assumed to be available at time 0.

6) Any movement of pallets requires a stacker crane whose control logic is determined by the FMS and is not controllable by users. Essentially all control logics of the M-FMS are incorporated in the model as discussed later.

7) No preemption is allowed during processing.

8) No equipment (machines, LUL's, a stacker crane, tools) failure is considered.

9) Processing time of operations, load/unload time at LUL, transportation time of pallets, the number of pallets for a specific operation, due dates given to each lot of a part type, are all known and deterministic. The FMS control logic does not include any randomness, and thus the simulation is completely deterministic given a set of production requirements.

10) An FMS user specifies priority ordering of parts/operations. This is the only user-controllable hardware independent decision, provided that production requirements are given.

2.3 Classification of User-Controllable Decisions

In this paper, we focus on short-term decisions FMS users normally make on a daily basis. These short-term decisions can be classified into two categories:

1. Hardware dependent decisions
2. Hardware independent decisions

Most of hardware dependent decisions relate to tools, pallets, and fixtures, as an operation depends very much on tools required for processing the operation (i.e., a set of sub-operations) and on how the part is fixed on a pallet. Typical hardware dependent decisions are listed below:

1) Assignment of tools to machines (within the capacity of a tool magazine)

2) Determination of machine(s), which we call candidate machine(s), that can process a particuar operation

3) Determination of the number of pallets with a particular fixture that can be used for a particular operation

One can safely assume that 1) and 2) above are effectively identical.

Hardware independent decisions, on the other hand, may be called software decisions, i.e., those decisions which can be made without taking hardware components of system such as tools and fixtures into considerations. Generally, software decisions are easier to change than hardware dependent decisions, as the latter decisions often require movement of hardware in addition to the corresponding software changes. For the system under study, the only software decision is a dispatching priority for parts/operations.

2.4 Basic Control Logic

Since pallets cannot be moved without the stacker crane, the control mechanism of the stacker crane plays the essential role in determining the behavior of the system. Two key elements of the control mechanism are 1)"from-to" priority of the stacker crane (Table 1), and 2)priority of operations (Table 2).

Table 1 : Dispatching Priority of Stacker Crane

Dispatching Priority
1. Move a loaded pallet to machine
2. Move a pallet to LUL for loading when a part is available for loading
3. Move a completed pallet to LUL for unloading when parts to be loaded on the pallet are unavailable
4. Move a pallet to be washed to a wash station
5. Move a loaded pallet from LUL to stocker
6. Move an empty pallet from LUL to stocker
7. Move an unloaded pallet to stocker
8. Move a washed pallet to stocker

Table 1 gives the "from-to" priority of the stacker crane, which specifies priority ordering among various movement types of the stacker crane. As achievement of high machine utilization is essential for these FMS's, the highest priority is given to moving a loaded pallet to a machine or to an input buffer of

Table 2 : Priority of Operations

Priority	Parts No./Opn.No.	No. of pallets	Candidate machines
1	B_2	2	0 1 0 0 0 1 0 0
2	B_1	1	1 0 0 1 0 0 1 1
3	A_1	3	0 1 1 0 0 1 0 0
4	A_2	1	0 0 0 0 1 0 0 0
5	A_3	2	1 1 0 1 0 0 0 1
6	C_1	1	0 0 0 1 0 0 1 1

a machine. There may exist several alternative parts that can be transported to some machine, and thus it is necessary to specify priority of parts and operations, based on which a specific part is chosen for movement.

Table 2 shows an example of a priority ordering of parts and the associated operations. Here, higher priority will be given to parts B, A, C, in this order. For each part type, users specify if earlier/later operations are given higher priority when the part requires multiple operations. In this example, We assume that the later operation gets higher priority for part B, which requires operations B_1 and B_2 to be processed in this order. On the other hand, for part A which consists of three operations A_1, A_2, A_3 to be processed in this order, the option of earlier-operations-first is assumed. In consequence, all associated operations are ordered by priority as follows : $B_2 \rightarrow B_1 \rightarrow A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow C_1$.

Vectors under the heading of candidate machines in Table 2 indicate machines capable of processing a particular operation. Table 2 assumes an 8-machine configuration, and a vector (0 1 1 0 0 1 0 0) for A_2 indicates that machines 2,3 and 6 are capable of processing operation A_2. That is, if the n-th entry is 1(0), machine n is (is not) capable of processing the operation. For a given "from-to" movement type, a priority ordering of operations is scanned to find the highest priority operation that waits for the movement type. For example, assume that a pallet for operation A_2 has been just loaded at one of the load/unload stations and any one of the candidate machines or their input buffer, say machine 3 is empty. If loaded pallets with higher priority such as operaions B_2, B_1, and A_1 are not currently available, the stacker crane is deployed for moving pallet of A_2 to machine 3.

2.5 Important Characteristics of the System

The FMS under study can be regarded as a special type of a job shop. However, there exist several factors which differenciate this system from the "standard" job shop as studied, e.g., by Conway,Maxwell,and Miller(1967).

1. Control by part/operation priority
2. Existence of lot sizes
3. Finite number of pallets
4. Different number of operations per part

There exist many studies on efficient operations of FMS's, but it is not necessarily clear which factors of the systems have been explicitly included in those analysis. The authors believe that those listed above constitute the key factors that differentiate many of existing FMS's from a traditional job shop.

2.5.1 Control by part/operation priority

Many of the existing FMS's use a control mechanism similar to the one discussed here, in which users specify priority ordering of parts and operations.

Therefore, once the short-term part mix and the associated lot size for each part type are determined, the determination of the part/operation priority is the only hardware-independent decision FMS users can make. In other words, the only user-controllable software decision is the determination of part/opereration priority.

2.5.2 Existence of lot sizes

Even under FMS's, there still exists a notion of lot sizes. As diversity of customer demand increases, lot sizes certainly decreased substantially, and lot sizes of typical FMS's are in the order of tens and hundreds. Lot sizes may be as low as single-digit numbers, but still it is rare that lot size is 1. Note that the classical job shop scheduling model does not consider the notion of lot sizes, even though a job lot of size k can be split into k "identical" jobs. The existence of lot sizes implies that the multiple part types are in production concurrently, and the same part repeats as many times as its lot size.

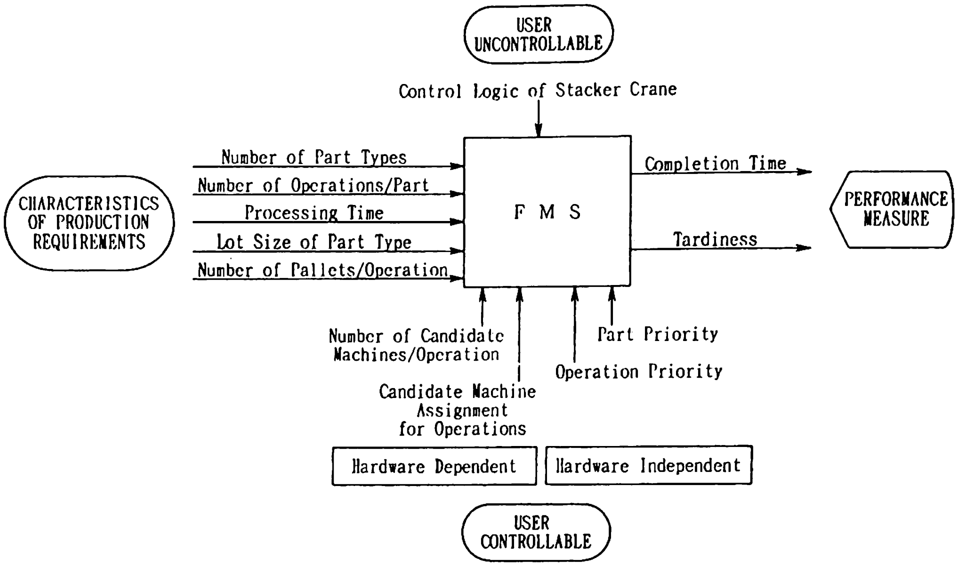


Figure 1 : Factors Surrounding the Problem

2.5.3 Finite number of pallets

Parts are introduced into the system after they are placed on pallets. Pallets are basically a square base on which fixtures are placed. A fixture, in turn holds parts. All pallets are identical in size and also in function, but fixtures are mostly operation-dependent. Due to limited storage space and also to high cost of pallets and fixtures, the number of pallets with identical fixtures is often small, say, a lower single digit number. From now on, a term "pallet" is used to mean a "pallet with a specific fixture for a specific operation". The fact that the number of pallets for a particular operation is limited means that one may not generally be able to process a particular operation consecutively. Therefore, a new part type is often started before required lot sizes of other part types have not yet been completed, thus leading to concurrent production of multiple part types.

2.5.4 Different number of operations per part

There are many papers on job shop scheduling. They deal with either static (and mostly deterministic) problems often using combinatorial optimization techniques, or dynamic problems using discrete-event simulation or queueing theory. Interestingly, most of these studies assume that the number of operations per part is identical for all parts, which is not generally true in real life.

For some unknown reasons, most studies consider jobs with identical number of operations.

2.6 A Minimum Tardiness Dispatching Priority Problem

For the system and the associated model described earlier, we limit our attention to the hardware independent decision problems, and consider the problem of finding the best dispatching priority of operations that minimizes the total tardiness, when due dates are given externally to individual lots of parts, rather than to induvidual parts. That is, tardiness of a lot of a given part type is determined by comparing the completion time of the required lot size of the part type and the associated due date, and we try to minimize the sum of tardiness for all part types.

This problem is the outgrowth of the real issues faced by operators of the particular module-type commercial FMS's. The authors believe that the problem is very important one, which is faced by many people who use similar FMS's.

Figure 1 summarizes major factors and elements surrounding the problem considered in this paper. In this paper, user (i.e., FMS user) uncontrollable factors are assumed to be fixed. Among user controllable factors, only hardware independent decisions are considered. Characteristics of production requirements would correspond to those factors determined by what has to be produced with the FMS. We seek some relationship between factors of production requirements and user controllable software decision, namely, the dispatching priority.

3 COMBINED SIMULATED ANNEALING AND SIMULATION ALGORITHM FOR FINDING THE BEST DISPATCHING PRIORITY

3.1 Combined Simulation / Optimization Approach

The control logic of FMS is generally very complicated, and thus it is difficult to foresee future. Yet, the detailed simulator developed for M-FMS allows us to evaluate its performance and also to predict its immediate future.

We try to "solve" the minimum tardiness dispatching priority problem by using the simulator's capability for a short-term look ahead in conjunction with a variant of local search techiniques known as the simulated annealing method.

A "solution" in this paper means a specific dispatching priority ordering of all operations, and the terms "solution" and "dispatching priority" are used interchangeably. In our algorithm, a neighbourhood of a given solution is defined as follows: Given a specific dispatching priority ordering x, a set of other priority orderings which can be obtained by interchanging priorities of any two parts *or* switching operation priority of any one part constitutes the neighborhood of x, which is denoted as $N(x)$.

3.2 Prototype Simulated Annealing Algorithm

The outline of the prototype algorithm is shown below:

```
Step 1 Initialize $I$ and $T$,
           and let $x$ be a random priority order.
           a simulation based on $x$ to obtain $D(x)$.
Step 2 $I \leftarrow I + 1$
           If $I = 0(modW)$, then $T \leftarrow T \times 0.95$
Step 3 $N(x)$ be neighbor of $x$.
           Select randomly $x' \in N(x)$,
           and run a corrsponding simulation to
           obtain $D(x')$.
Step 4 $\Delta E \leftarrow D(x') - D(x)$
           If $\Delta E < 0$, then $x \leftarrow x'$,
           $D(x) \leftarrow D(x')$, and go to Step 6
Step 5 If $R < exp(-\Delta E/T)$, then
           $x \leftarrow x'$, $D(x) \leftarrow D(x')$.
Step 6 If $T < \varepsilon$, then stop.
           Otherwise, go to Step 2.
```

I:repetition counter T:temperature
x:dispatching priority under consideration
$D(x)$: total tardiness under dispatching priority x
$N(x)$: neighborhood of x
W: total number of operations
R: a random number in [0,1]

3.3 Computational Refinements

Several computational refinements are being tested some of which are listed below:

1) Changing parts priority generally affects due date performance more drastically than changing operation priority. As a result, during the earlier phase of the algorithm, one may focus on changing parts priority, and later increase the probability of changing operation priority.

2) The prototype algorithm selects any two parts randomly when an interchange of parts priority is considered. However, it appears profitable that at least one of the switched parts is tardy. One could modify the definition of neighborhood so that interchanges are limited to such pairs, or alternatively may distort selection probabilities in such a way that such pairs are more likely to be chosen.

3) As in branch-and-bound algorithms, simulation could be terminated as soon as it is known that the given priority cannot produce a better due date performance. That is, when the lower bound of the total tardiness is found to exceed the current best tardiness, terminate the simulation and try another dispatching priority.

3.4 Standard Dispatching Rules

A huge number of dispatching rules have been proposed for dynamic job shop scheduling. It is generally understood that the effectiveness of these dispatching rules is often limited to a particular situation. In this study, we focus on those dispatching rules claimed to be effective for the control of FMS's. Those dispatching rules considered in this study are listed below. For details, refer to Montazer and Van Wassenhove (1990) and Blackstone Jr. et al.(1982).

1. SDT(Smallest ratio SIO / SPT)
2. LMT(Largest value SIO * SPT)
3. LIO(Longest Imminent Operation time)
4. LPT(Longest Processing Time)
5. SLACK(minimum SLACK time)
6. FRO(Fewest number of Remaining Operations)
7. RAN(RANdom)
8. LRPT(Longest Remaining Processing Time)
9. LDT(Largest ratio SIO / SPT)
10. EDD(Earliest Due Date)
11. SIO(Shortest Imminent Operation time)
12. MRO(Largest number of Remaining Operations)
13. SRPT(Shortest Remaining Processing Time)
14. SMT(Smallest value SIO * SPT)
15. SPT(Shortest Processing Time)
16. SNOP(Smallest Number of OPerations)
17. SSNOP(SNOP+SPT)

4 EVALUATION OF THE COMBINED SIMULATION - SIMULATED ANNEALING APPROACH AND ITS COMPARISON WITH STANDARD DISPATCHING RULES

4.1 Experimental Conditions

In this study, we assume the FMS configuration of 8 machines, 4 LUL's and a single stacker crane. The number of part types, processing times of operations, the number of pallets for a particular operation, and the number of "candidate" machines that can process a particular operation, are fixed to 18, uniform distribution between 15 and 75 minutes, 1 pallet for each operation, and 1 machine for each operation, respectively. A constant lot size of 10 is assumed for each part type. Candidate machines are randomly assigned to each operation.

The numbers of operations for part types may not be identical, and our experiments are performed on the case where 6 part types consist of a single operation, 6 part types of two operations, and the remaining 6 part types of three operations. The maximum of 3 operations for a part type are considered throughout this study.

Simulation experiments have been performed on a basic data set reflecting typical daily operation. For the basic data set, six distinct sets of due dates have been tried.

In Figures 3 and 4, the horizontal axis corresponds to various dispatching priorities. In particular, RSA and SSA, which are the two right-most entries of the horizontal axis, indicate the Simulated Annealing (SA) methods where the initial dispatching priority is based on the random (RAN) rule and the SPT rule, respectively. On the other hand, the vertical axis shows relative due date (i.e., tardiness) performance of various dispatching rules, where the total tardiness obtained by the SSA method is taken as the base performance. For example, relative performance of 1.15 for some dispatching rule means that the total tardiness of the dispatching rule is 15% more than that obtained by the SSA method.

4.1.1 Determination of due dates

Performance of dispatching priorities certainly depends on the due dates assigned externally to individual part types. Even though an issue of due date assignment is outside the scope of this study, there are many studies on proper determination of due dates. Noting some results of these studies (See, e.g., Ragatz and Mabert (1984)) on how due dates could be assigned, due dates are generated randomly or deterministically by the following two methods for the experiments described below.

- Consider two parameters regarding the tightness of due dates and the range over which due dates are distributed. Assume that the total processing time of all part types is M. Using the tightness parameter $t(0 \leq t \leq 1)$ and the range parameter $d(0 \leq d \leq 1)$, determine the range over which due dates are distributed by the following formula, and determine due dates by generating random numbers uniformly from the range.

$$\text{due date range} = \begin{array}{l}\{M(1-t)+Md/2,\\ M(1-t)-Md/2\}\end{array}$$

- Considering lot sizes of part types, numbers of operations/part types and the minimum required time in system for each part type, due dates are determined by the following formula in a deterministic fashion. There is only one parameter k which controls the tightness of due dates.

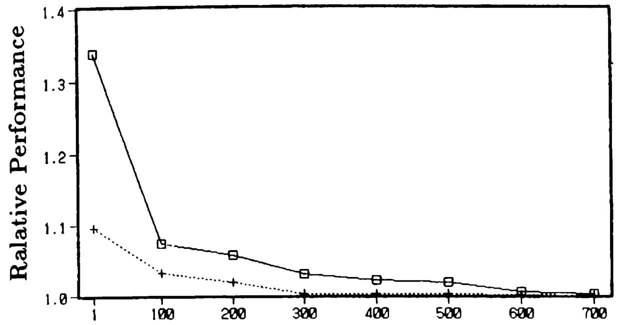
$$D_i = kQ_i(P_i + N_iL)$$

D_i : Due date of part type i
k : Due date factor($k > 1$)
Q_i : Lot size of part type i
P_i : Total processing time of part type i
N_i : No. of operations for part type i
L : Transportation and load/unload time

4.2 Optimization with the SA Method

Figure 2 shows typical processes of improvement as the simulated annealing algorithm progresses. Experiments have been performed on SUN SPARC Station 2, and a problem data set with 18 part types and 36 operations, lot sizes of 10, took roughly 12 minutes of its CPU time. Each point corresponds to an average performance for 6 sets of due dates.

The vertical axis corresponds to the relative performance where the best solution found after 700 iterations of SSA is taken as the base of 1.0. The best solution found by RSA is 1.0036, just slightly worse than that of SSA. A combined simulation/optimization approach requires extensive computations, and thus one cannot expect a huge number of iterations. Keeping this in mind, it appears profitable to start from good solutions. As shown in Figure 2, the SSA algorithm gets "stable" after approximately 300 iterations, after which one cannot expect dramatic improvement. As a result, all the subsequent results are based on the termination of RSA or SSA after 300 iterations.



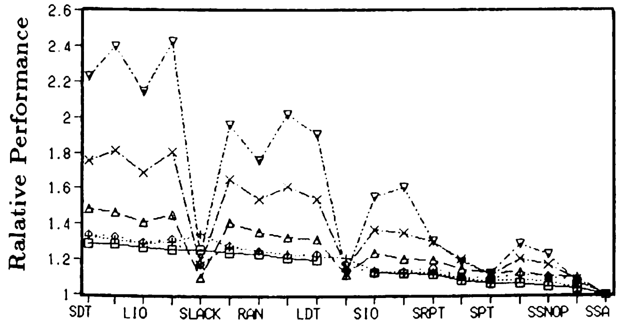
Iteration Number

Figure 2 : Optimization by a Simulated Annealing Method

4.3 Comparison of SA Methods and Standard Dispatching Rules

Figure 3 compares, for six distinct sets of due dates, performances of various dispatching rules against the best dispatching priority obtained by the SA method with a random priority. The vertical axis of 1 corresponds to the total tardiness of the best dispatching priority obtained by the SSA. Figure 3 shows results for 6 distinct sets of due dates.

Those dispatching rules which take due dates into considerations such as SLACK and EDD perform reasonably well as compared with other standard dispatching rules. It should be noted that the variability of due date performance is larger for those poorer dispatching rules.

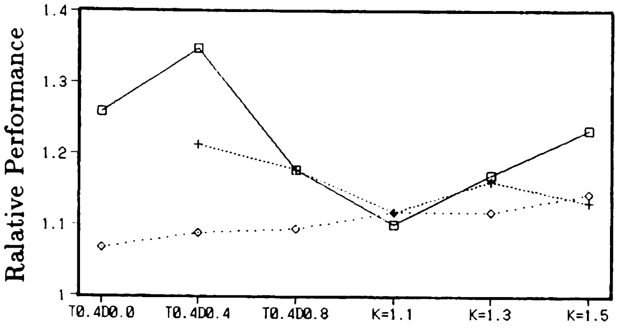


Dispatching Rules
□$t = 0.4, d = 0$, $+t = 0.4, d = 0.4$, $\diamond t = 0.4, d = 0.8$,
$\triangle k = 1.1$, $\times k = 1.3$, $\nabla k = 1.5$

Figure 3 : Comparison of SA Methods and Standard Dispatching Rules

4.4 Performance of Due Date Based and Workload Based Dispatching Rules

Even though dispatching rules such as SLACK and EDD which take due dates into considerations perform better than other standard dispatching rules, their performances are not necessarily satisfactory when compared with the best dispatching priority. More specifically, performances of SLACK and EDD fluctuate depending on given due dates. These dispatching rules work well for some sets of due dates, but not so well for others.



Due Date Factor
□$SLACK$, $+EDD$, $\diamond SPT$

Figure 4 : Performance of Due Date Based Rules and SPT Rule

It is also interesting to note that some of the workload-based dispatching rules such as SPT and SMT work well and more importantly consistently without regard to given due dates.

The best solution obtained by the annealing algorithm is found to be, roughly speaking, 10% better than the one found by SPT. This implies the merit of using this type of optimization as opposed to the good standard dispatching rules.

4.5 More Flexible Priority Specification

One can easily observe that the control by parts/operation priority as in the existing M-FMS can be made further more flexible if any permutations of all operations are permissible. Note that, under the current rule, all operations of a particular part must be adjacent to each other in the priority ordering, and only earlier-operation-first or later-operation-first are the two possible ways to sequence priority of operations. As a result, a priority ordering such as $A_3 \rightarrow B_2 \rightarrow A_2 \rightarrow C_1 \rightarrow B_1 \rightarrow A_1$ is not allowed in the current M-FMS.

The modification of the control logic might be justified if a) the added flexibility leads to more efficiency, and b) there exists knowledge or mechanism for specifying the priority. The latter seems to be justified because the simulation-optimization approach as discussed here seems to provide a mechanism for the priority determination, even though no a priori knowledge exist for its proper determination.

The annealing algorithm can easily be revised to see the effect of the above modification. Preliminary computational results show that added flexibility achieved by allowing any permutations of operations in the priority specification contributes little and only leads to 0.2% better due date performance than the current method of priority specification.

5 CONCLUSIONS AND FUTURE RESEARCH

A simulation-optimization approach to find an appropriate dispatching priority was presented. The algorithm combined simulated annealing and simulation to explore a dispatching priority of operations that minimizes the total tardiness. Computational performance of the algorithm showed that good solutions can be obtained mostly after 300 iterations or so.

The best solution found yields, in most cases, tardiness 10% smaller than those obtained by well-performed standard dispatching rules such as SPT and EDD.

The paper also compared the results of the optimization with standard dispatching rules, and showed that standard rules such as EDD and SLACK which take due dates into considerations perform moderately, but not extremely well.

Our results are thus far limited to a particular FMS's, namely, the Mazatrol FMS's. The authors do believe, however, that the findings are general in the sense that a certain type of job shops have similar characteristics. Further research is being performed to study generality of the results. Further analyses are being performed also to find characteristics of those "best" diapatching priorities obtained by the simulation/optimization approach.

ACKNOWLEDGMENTS

This research is supported by Waseda University Grant for Special Research Projects (91C-7, 92A-83, 93A-79), a research grant from Progressive Manufacturing Foundation and Ministry of Education, Grant-in-Aid for General Scientific Research (C 132-268).

The authors would like to express appreciations to Mr. Isamu Kawashima of Yamazaki Mazak and Mr. Hiroshi Takayama for discussions and supply of data.

REFERENCES

Morito, S., K. H. Lee, K. Mizoguchi, and H. Awane. 1992. Development of a simulation-based planning system for a flexible manufacturing system. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, 908-915. Association for Computing Machinery, Arlington, Virginia.

Morito, S., T. Takano, H. Mizukawa, and K. Mizoguchi. 1991. Design and analysis of a flexible manufacturing system with simulation : effects of flexibility on FMS performance. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, and G. M. Clark, 294–301. Association for Computing Machinery, Baltimore, Maryland.

Montazer, M., and L. N. Van Wassenhove. 1990. Analysis of scheduling rules for an FMS. *International Journal of Production Research* 28:785-802.

Ragatz, G. L., and V. A. Mabert, 1984. A simulation analysis of due date assignment rules. *Journal of Operations Management* 5:27-39.

Blackstone Jr, J. H., D. T. Phillips, and G. L. Hogg. 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research* 20:27-45.

Conway, R. W., W. L. Maxwell, and L. W. Miller. 1967. *Theory of scheduling.* Addison-Wesley, Reading, Mass.

AUTHOR BIOGRAPHIES

SUSUMU MORITO is Professor of Operations Research at Department of Industrial Engineering and Management, School of Science and Engineering, Waseda University. He received his B.Engr. and M.Engr. from Waseda, and also M.S. and Ph.D. from Department of Operations Research, Case Western Reserve University, Cleveland, Ohio. Professor Morito taught at Case Western Reserve and also at the University of Tsukuba prior to joining Waseda. His main area of research is operations research, with particular emphasis on applications of discrete-event simulation in manufacturing. He serves as the president of the special interest group on simulation at the Operations Research Society of Japan.

KEUN HYUNG LEE is a doctoral student at Department of Industrial Engineering and Management, Waseda University. He holds Master of Management from Yokohama National University.

KIYOHISA MIZOGUCHI is Manager of Engineering Department at Yamazaki Mazak Corporation.

HIROSHI AWANE is Tokyo Liaison Officer at Tokyo Liaison Office of Yamazaki Mazak Corporation. He received B.Engr. in Industrial Engineering from Waseda University. Prior to joining Yamazaki Mazak, Mr.Awane worked for Hitachi.