International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: http://www.tandfonline.com/loi/tprs20

Scheduling/rescheduling in the manufacturing operating system environment

M. YAMAMOTO & S. Y. NOF

To cite this article: M. YAMAMOTO & S. Y. NOF (1985) Scheduling/rescheduling in the manufacturing operating system environment , International Journal of Production Research, 23:4, 705-722, DOI: 10.1080/00207548508904739

To link to this article: http://dx.doi.org/10.1080/00207548508904739

Published online: 03 Apr 2007.

INT. J. PROD. RES., 1985, VOL. 23, NO. 4, 705–722

Scheduling/rescheduling in the manufacturing operating system environment†

M. YAMAMOTO‡ and S. Y. NOF§

A scheduling/rescheduling procedure is proposed for real-time control of a computerized manufacturing facility managed by a central manufacturing operating system. The procedure implies schedule revisions upon significant operational changes such as machine breakdowns. Experiments to evaluate the total production time of a computerized manufacturing system with breakdowns under scheduling/rescheduling have yielded advantages of between 2·5% to 7·0% compared to fixed sequencing and priority despatching procedures, respectively. Computation times required for the scheduling procedures on a CDC 6500/6600 have also been studied. The scheduling/rescheduling procedure for an actual facility required less than two minutes, and the computation time can be regulated by the selection of parameters in an approximate method of scheduling.

Introduction

Job shop type machine scheduling problems have been well-known subjects in operations research for the last 25 years and extensive work about them has been presented (Baker 1974, Conway *et al.* 1967). Nevertheless, these theories and methods have scarcely been used to solve actual job shop problems in practice. Though the causes of this inability can be investigated from several aspects, it has been considered that one of the major causes is the high variability of schedule factors in an actual process. Significant differences can be shown between a schedule that results from traditional scheduling approaches and the real progress in a shop; for example, see Bestwick and Lockyer (1979). If we want to correct these differences and maintain control by the schedule we cannot avoid ceaseless rescheduling. Such a practice is usually undesirable from the standpoint of shop management and administrative cost. However, two major developments have taken place in recent years: computer aided scheduling and computerized manufacturing systems. This work develops and analyses the idea of rescheduling in the new environment of computerized manufacturing.

Computer aided scheduling

Computer aided scheduling includes the following.

(1) The development of interactive scheduling techniques that rely on computer systems, e.g., Godin (1978), Kerry (1980) and Hodgson and McDonald (1981).
(2) Scheduling with graphics capabilities, e.g., Hurrion (1978).

Revision received June 1984.

† A shorter version of this article has appeared in Japanese (Yamamoto and Nof 1982).

‡ College of Engineering, Hosei University, Tokyo, Japan. This research was conducted when Professor Yamamoto was a Research Scholar at Purdue University.

§ School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907, U.S.A.

(3) Decision support systems for manufacturing control, e.g., Nof and Gurecki (1980).

The main contribution of such approaches is that they allow non-experts to review and revise many schedule alternatives quickly. As a result, scheduling control of dynamic production systems becomes responsive and more effective.

Computerised manufacturing systems

The other major development is of computerized, or flexible, manufacturing systems (CMS). CMS are comprised of control computers, DNC machines, automatic transportation and material handling facilities, automatic tool changers, etc. CMS provide production flexibility and programmability, and therefore have been applied over the last decade in order to increase productivity in non-mass products, batch type production. CMS system design and the analysis of CMS operating for rules are investigated by many researchers (for example, see Buzzacot and Shantikumar 1981, Nof *et al.* 1978). A fundamental requirement for the effective use of CMS is a good central control.

The manufacturing operating system environment

The manufacturing operating system (MOS), was suggested by Nof, Whinston and Bullers (1980) for central control of automatic manufacturing. The MOS is a framework of multi-stage decision making in controlling the total system operation.

The MOS has three main components, namely, data management, logic management, and interfaces to human users and machine controllers. The data management component is responsible for the representation, storage, and retrieval of data used for operations management decisions. The logic management component involves the representation, storage, retrieval, and execution of algorithms (e.g., reasoning logic) at decision points in the system. Decision-making algorithms are applied for structured decision where automatic resolution is allowed. Decision support algorithms, on the other hand, manipulate and evaluate information that is useful as an aid to human decisions for non-automatic, unstructured decisions.

Such control is quite difficult to achieve in a regular job shop environment. A CMS, however, requires significant investment cost and intensive use of computers, thus it is possible to justify the more sophisticated and accurate shop control of the MOS type.

One of the major functions of the MOS is sequencing and timing of parts movement in a CMS. Since a CMS facility is an integrated complex equipment, proper coordination and synchronization are paramount. Examples of scheduling control issues are the control of the parts mix concurrently produced by the system; initial entry of parts to the system (i.e., entry to an empty system after each restart); general entry of parts; dynamic process selection when alternative processes are available for parts; dynamic selection of transporter and transfer route; rerouting of parts upon machine breakdown, and so on.

Nof *et al.* (1978) applied various heuristics for decision making in a CMS for priority despatching of parts in a particular CMS. An important factor in the ability to apply such decision making for control in a CMS is the fact that operations, process, and transfer time values in the automatic environment are much less variable than in systems involving human work. This relative stability of automatic performance time suggests, therefore, that the above-mentioned scheduling control has practical promise. On the other hand, dynamic changes in production requirements and frequent machine stoppage and breakdown complicate the CMS operation.

The purpose of this article is to present a concept of scheduling/rescheduling in the dynamic MOS environment, and examine its feasibility and use to increase the CMS operational efficiency.

The scheduling/rescheduling approach

A scheduling/rescheduling approach implies that a set schedule is revised at given points in time due to certain significant changes in operation requirements. A fundamental characteristic of the scheduling/rescheduling approach is that it is not planned in advance for a certain future time, but is invoked under certain circumstances. An approach of this type has mainly been applied for production and inventory control. For example, Mather (1977), describes rescheduling in application of MRP. He states that MRP rescheduling procedure is called upon for a variety of causes, such as vendor failure to supply, unexpected scrap or spoilage, lot-size changes, etc.

While rescheduling in production and inventory control is considered over periods of weeks or days, our focus is on applying the scheduling/rescheduling approach by the MOS more frequently if necessary, and in real-time control.

General scheme

The main feature of a scheduling/rescheduling approach is to establish a schedule of all operations in a system for a fixed time period in advance. The schedule is generated in consideration of the optimality or near optimality of the total system, instead of a series of local decisions by some priority rule applied at each machine and each time point. In order to realize this idea, a scheduling/rescheduling approach will have the following general scheme.

(1) Planning phase
  (*a*) Part-mix assignment
  (*b*) Initial scheduling
  (*c*) Machine loading table generation
(2) Control phase
  (*a*) Machine loading order instructions
  (*b*) Checking progress
  (*c*) Testing for abnormal status
(3) Rescheduling phase
  (*a*) Rescheduling
  (*b*) Revising loading table
  (*c*) Resorting to control phase

Each of the phases will now be described.

Planning phase

In the planning phase, an 'initial schedule' is generated just prior to the start of a new work period, based on all available production requirements data. The planning phase prepares all the information necessary for the operations during a given work period, say a work day or a shift. In the part-mix assignment the types and quantities of parts to be processed in the work period are decided according to the production requirement given for this CMS. The part-mix assignment has to satisfy requirements according to part types, quantities, and due dates in production during the same time period, in order to assume the proper loading for each machine. Next, a schedule is generated to process these part types and quantities. For all operations necessary for performing jobs in the work period, starting and finishing times on each machine are determined. A loading table can then be developed for each machine and facility directly from the resulting schedule. In some cases, we can assign only a processing sequence, instead of producing a complete timetable for each facility.

Control phase

Next, the control phase begins with the start of the work period. Operations are initiated one by one following instructions from the loading tables of the initial schedule, and progress is checked. If the loading table contains loading times, a signal is issued to begin the next operation when its starting time in the loading table is reached. However, if specified conditions that are required to start an operation are not met, e.g., a set temperature is not yet available, this signal will be held until the necessary conditions are satisfied. Similarly, the signal will be held if all preceding operations have not been completed. In the case of instructions given by a loading sequence, a signal to start an operation is issued as soon as all preceding operations in the sequence are completed and the specified conditions are met.

The actual progress data of operations are compared with a current schedule every time a new operation begins and finishes. If the difference exceeds a specified limit, the MOS will decide to enter a rescheduling phase. In addition, certain abnormal status such as machine troubles and operator's interruptions can cause entry into the rescheduling phase. In the latter situation, the rescheduling will be accompanied by other actions deemed necessary to respond to the abnormal status.

Rescheduling phase

In the rescheduling phase, the schedule times of operations already processed, in process, or expected immediately to follow into processing are fixed at first. The remaining operations are considered to be free operations. For them, computation of a revised schedule is carried out, considering the operational changes that have triggered the rescheduling. For instance, if a new part mix is required, then a revised schedule has to be generated. In the case of machine breakdown, the expected duration of the breakdown has to be considered. As a result, revised loading tables are generated. The scheduling procedure itself is essentially the same as in the planning phase, except that the schedule of the non-free operations is fixed.

Scheduling/rescheduling experiment

The scheduling/rescheduling approach has been tested on two case studies of typical CMSs with machine breakdowns, and compared to operations under two scheduling techniques: one following strictly the sequence of the original schedule; the other, applying priority despatching rules. The former corresponds to the extreme case that MOS doesn't enter the rescheduling phase at all in the scheduling/rescheduling approach. The latter is a traditional procedure in shop control. Figure 1 (*a*), (*b*) and (*c*) show the logic of the three scheduling procedures that have been compared, i.e., (*a*) scheduling/rescheduling; (*b*) fixed sequence; (*c*) priority despatching.

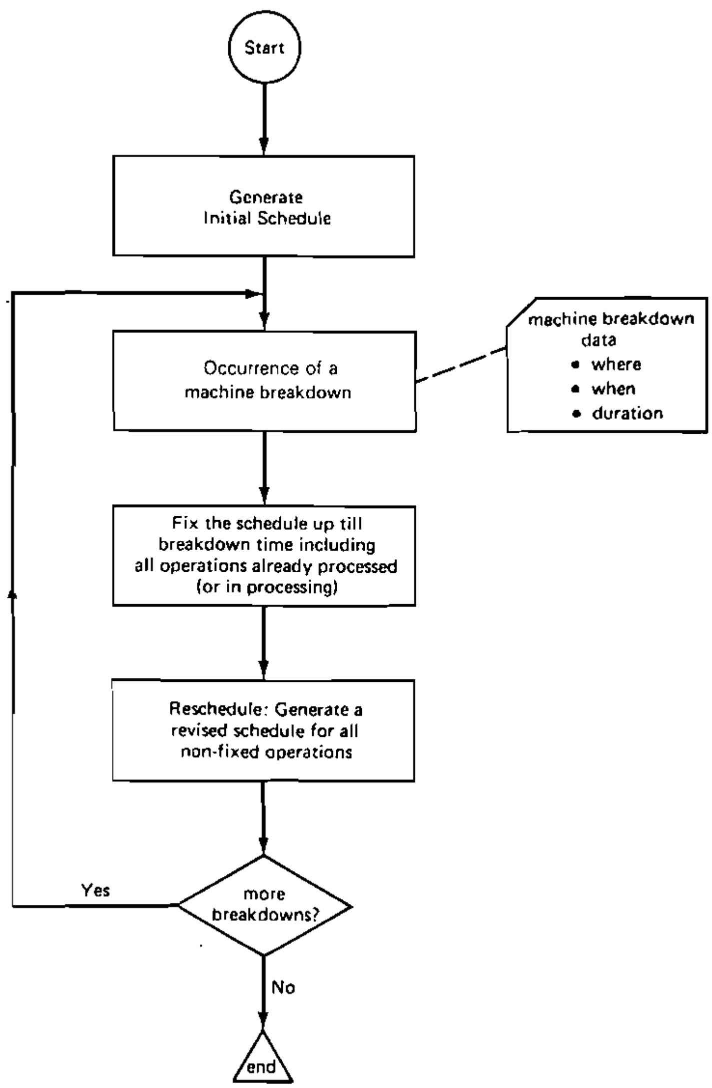


Figure 1 (*a*). Logic of scheduling/rescheduling procedure.

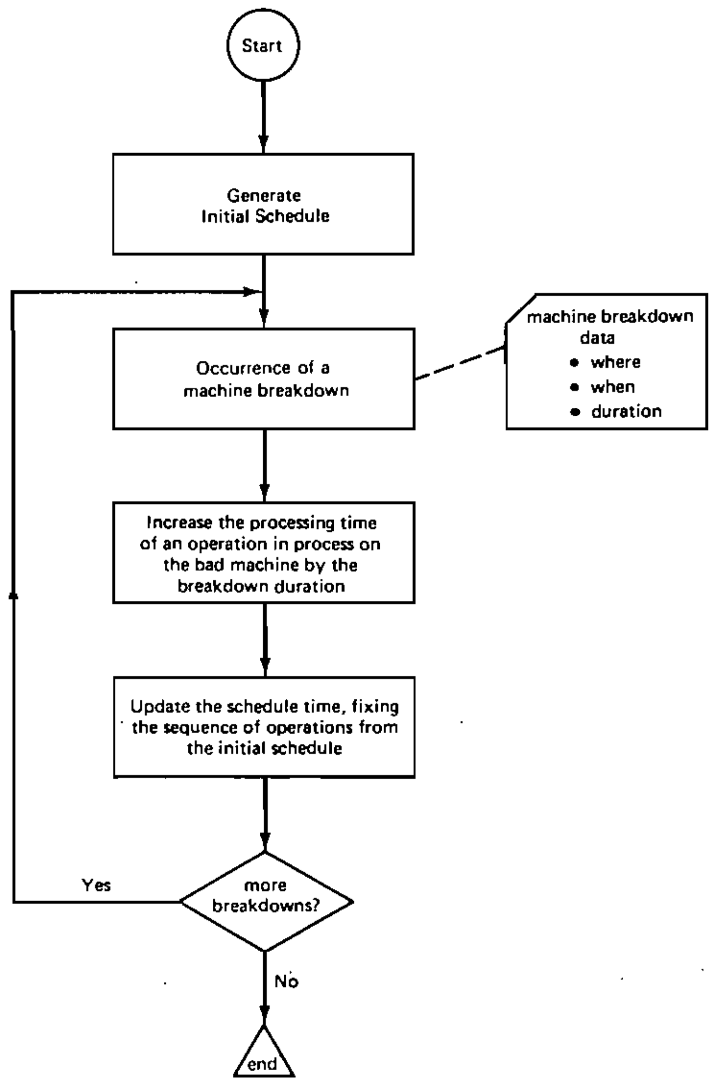


Figure 1 (*b*). Logic of fixed-sequence procedure. Maintain the sequence from the original schedule throughout all processing periods.

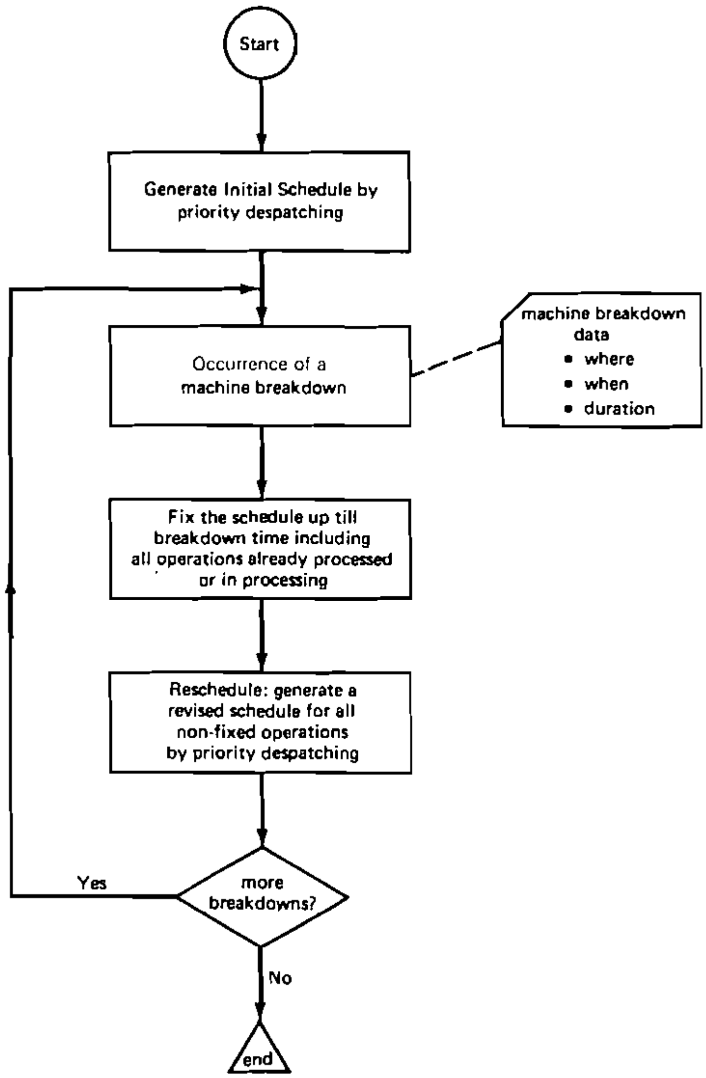


Figure 1 (*c*). Logic of priority despatching procedure. Same as (*a*), but the schedule is generated by priority despatching heuristics.

By these experiments we can simulate the behaviour of a shop where machine breakdowns occur randomly. The completion time of parts in the middle of a process at the troubled machine is prolonged, and the schedule after the repair time is altered according to the respective scheduling logics.

We have tested these procedures on two case studies. The first case is a small CMS, which is a hypothetical model for ascertaining the idea of the proposed procedure. The second case is an actual CMS with six machine centres, the experimental data of which was taken from one actual case of the typical shop records.

Scheduling/rescheduling algorithms

Most job shop scheduling techniques could be used for the scheduling/rescheduling. The technique used in this study is the active schedule generation (Baker 1974). This technique involves a tree search procedure based on a branch-and-bound method. A schedule tree is produced by the active schedule generation technique, which creates an active schedule on a graph by introducing new arcs. The arcs represent order relation between operations due to machine constraints. An objective function defined for the search is to minimize total processing time.

As the lower bound for it, we use the combination of a job-based bound and a machine-based bound. The tree search procedure follows the depth-first search method. However, a pruning-off method (described in detail in Yamamoto 1977 a) is used as an approximation technique in order to regulate and limit the total computation time. The pruning-off method has been applied with $b^* = 1$ where b^* denotes the given order number of a branch, and the limitation of step count searched has been set at $l = 200$ nodes.

The same algorithm has been applied for both the initial planning phase and the rescheduling phase. The program package, which was originally used by Yamamoto (1977 a) and (1977 b), has been utilized with some necessary revisions on the CDC-6500/6600 computer at Purdue University computer centre, as described by Yamamoto (1979).

For the priority despatching procedure used in the study (Fig. 1 (c)), the priority rules that have been applied are as follows: for the entry of parts into the CMS a higher priority was given to part types that require longer total processing time; for the part-to-machine allocation, which assigns parts dynamically to particular machines inside the CMS, the first come first served rule was applied. These rules were chosen as typical of those practised at the actual CMS that was the object of this study.

Case 1. CMS with three machining centres

The system consists of three machining centres interconnected by a recirculating conveyor loop and a load/unload station. A part to be processed is set on an exclusive pallet at the loading station, and sent to the conveyor loop. The pallet with the part then enters a machining centre according to the programmed instructions (loading table in memory) if the assigned centre is available. Otherwise, the pallet and part recirculate until an appropriate machining centre becomes available. Additional assumptions about the operation are:

(1) A pallet can hold one item at a time.
(2) A pallet becomes free when a pallet with a part returns back to an unload station and the unloading work of the finished part is completed.

(3) When a pallet becomes free, if there remains an unprocessed part of the same type, the pallet is loaded with the new part.
(4) During the initial entry of parts to an empty CMS, the part that has the largest load on the first machine is given a highest priority. In the case of using an exclusive pallet system (i.e., different pallet types for different part types), the imbalance of load between different part types is stressed, because parts of the same type must be processed in series by the pallet restriction. Then this priority rule has an important role in the initial entry.

Let us consider an example of scheduling for one planning period. We have 12 parts to be processed, belonging to 5 part types respectively, and requiring a total of 84 operations as detailed in Fig. 2. (Each row in Fig. 2 represents one actual part.)

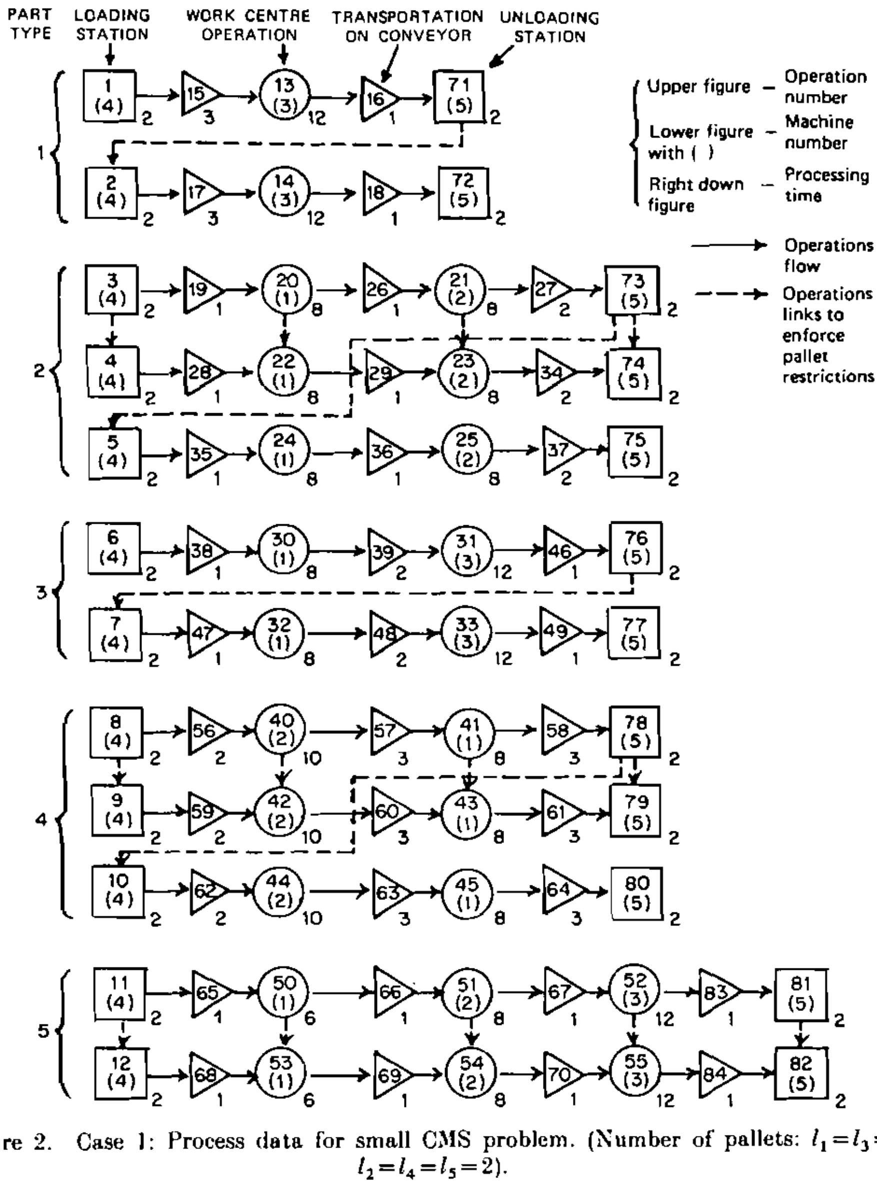


Figure 2. Case 1: Process data for small CMS problem. (Number of pallets: $l_1 = l_3 = 1$; $l_2 = l_4 = l_5 = 2$).

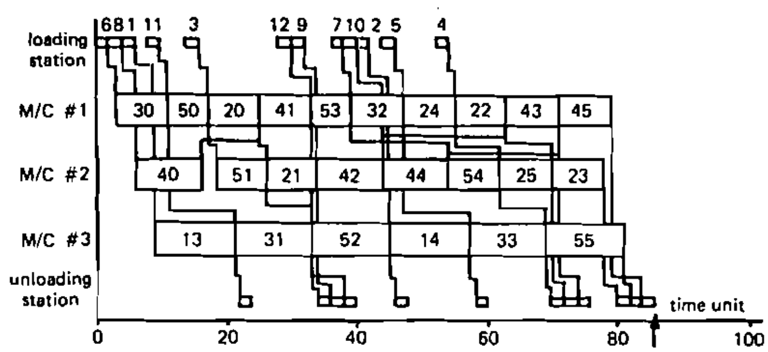
The figure represents the input network for scheduling the example case, where nodes are operations (loading, transporting, machining, and unloading) and arcs are operating sequences. Order relations dictated by pallet restrictions are shown hy dotted lines in Fig. 2. For parts of the same type, it is arbitrary which particular one precedes the other inside the system from the viewpoint of scheduling. For example, 3→4 and 20→22 for part type 2. As parts of the same part type are free in terms of the processing order and since there are two pallets availahle for part type 2, we can assign loading operation 3 to precede 4 at our option, if it is advantageous for saving the computation time of sequencing. In the scheduling algorithm, the dashed lines have the same meaning as regular arcs. For example, since part type 1 has only one availahle pallet, loading operation 2 of the second part type 1 can start only after 71 is completed .

The initial schedule for this case is shown in Fig. 3 (a). The completion of all operation requires 86 time units. After starting operations according to this initial schedule, suppose a machine breakdown occurs in machine 1 at time 18, the repair of which is expected to take 15 time units. Figure 3 (b) shows the change of schedule in the fixed sequencing case. In Fig. 3 (c) rescheduling is applied at time 18, and the execution after that time is carried out by the new schedule.

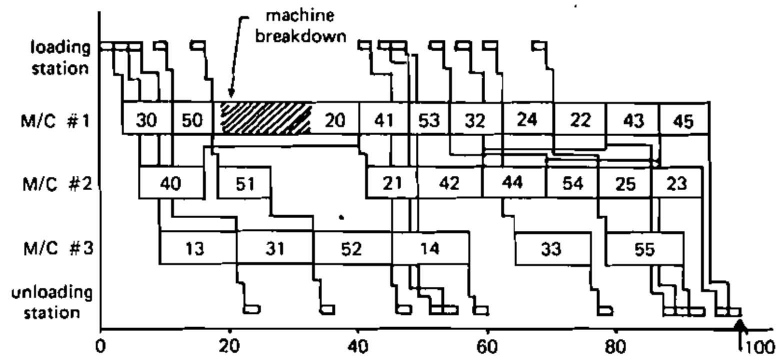
In order to ascertain the effect of the scheduling/rescheduling approach when machine hreakdowns take place, we performed a computational experiment by generating machine hreakdowns as shown in Table 1. Ten experiments were conducted, each with five random breakdowns. The three scheduling procedures were applied in each of the ten experiments; performance in terms of λ, total time required to complete all the scheduled parts, was computed.

Averages of total time in each of the ten experiments, $\bar{\lambda}$, are shown in Fig. 4. The $\bar{\lambda}$ values of initial schedules using three optional scheduling procedures increase with each additional machine breakdown that occurs as time passes by. However, the difference in the initial schedule between the rescheduling and the priority despatching is retained continuously during a series of machine breakdowns, though there is a tendency to decrease this dominance as the number of machine breakdowns increases. The results of the fixed sequencing procedure approach those of the priority despatching, but the dominance is kept during the whole range of this experiment.

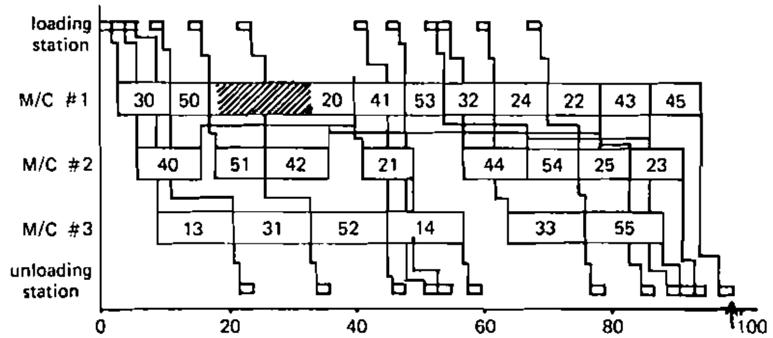
Comparison between the scheduling/rescheduling approach and the two other procedures are summarized in Table 2 (a) and (b). Out of the 100 pair comparisons in Table 2, the scheduling/rescheduling approach is worse than fixed sequencing in only



(a) Initial schedule (optimal schedule).



(*b*) Fixed sequencing in case of a machine breakdown in M/C 1 at time 18.



(*c*) Rescheduling in case of the same machine breakdown.

Figure 3. The alteration of a schedule at a machine breakdown.

5 cases, and worse than priority despatching in only 3 cases. Over all the experiments with breakdowns, the scheduling/rescheduling approach yields better performance than fixed scheduling by an average of about 2·5%, and better than priority despatching by an average of about 5·8%. The results were also compared statistically, since the ten experiments are random and can be considered independent. (Note that results of consecutive breakdowns of the same experiment are not independent.) In t-test comparisons, it was found that over the ten experiments λ_{RES} was significantly better than λ_{FIX} at a confidence level of 99% during the first 3 breakdowns, and 95% after the fourth breakdown, but not after the fifth breakdown. Compared to λ_{PRI}, λ_{RES} was significantly (at 99%) better during the first 4 breakdowns, but again not after the fifth.

Case 2. CMS with six machining centres

The second case that was studied involves an actual CMS with six machine centres interconnected by a conveyor loop (Nof *et al.* 1978). Fifteen different part types are scheduled for production, each with a different quantity. The operations required to produce each part and the quantity required from each part type are described in Table 3. The CMS is designed to handle up to 18 pallets simultaneously, either in-process at a machine, or recirculating in waiting. Since part type 3 requires the longest process, four pallets are assigned just for it, while all other part types are assigned only one pallet each.

Experiment number	Breakdown number	Occurrence of machine breakdown 1	2	3	4	5	Total breakdown time
1	M/C number	1	2	3	1	2	
	occurrence time	18	30	50	70	90	
	duration	15	10	20	10	15	70
2	M/C number	2	2	1	3	1	
	occurrence time	10	35	40	45	85	
	duration	15	10	20	15	10	70
3	M/C number	1	3	2	2	1	
	occurrence time	25	30	42	70	95	
	duration	10	15	5	15	20	65
4	M/C number	2	1	2	1	3	
	occurrence time	16	25	45	62	90	
	duration	20	10	15	10	15	70
5	M/C number	3	1	2	1	2	
	occurrence time	5	16	50	72	81	
	duration	15	5	20	20	10	70
6	M/C number	2	3	2	1	1	
	occurrence time	21	38	42	49	75	
	duration	15	10	20	15	10	70
7	M/C number	1	3	1	2	2	
	occurrence time	16	25	38	55	62	
	duration	15	10	10	5	20	60
8	M/C number	1	1	2	3	2	
	occurrence time	8	45	50	60	90	
	duration	10	10	15	20	15	70
9	M/C number	2	1	2	1	3	
	occurrence time	1	35	42	70	95	
	duration	15	15	10	10	10	60
10	M/C number	1	2	3	2	1	
	occurrence time	28	38	61	75	83	
	duration	20	15	10	15	10	70

Table 1. Input of machine breakdowns for small CMS problem.

For case 2, eight random machine breakdowns were considered, as shown in Table 4. Table 4 also shows the total time, λ, required by applying each scheduling procedure and considering each of the breakdowns; the CPU time required for computation by each procedure; and comparison ratios of the different values found for λ. As can be seen, in this experiment the scheduling/rescheduling procedure again provides better performance compared to the other procedures. In this experiment an advantage is found in all but one observation. On average, the scheduling/rescheduling procedure yields values for λ that were smaller by about 3·7% compared to the fixed sequence procedure, and by about 7·0% compared to the priority despatching procedure.

Computation time requirement

Two major objections to frequent schedule revisions during work have traditionally been that 'it is too complicated and disruptive', and that it requires too much

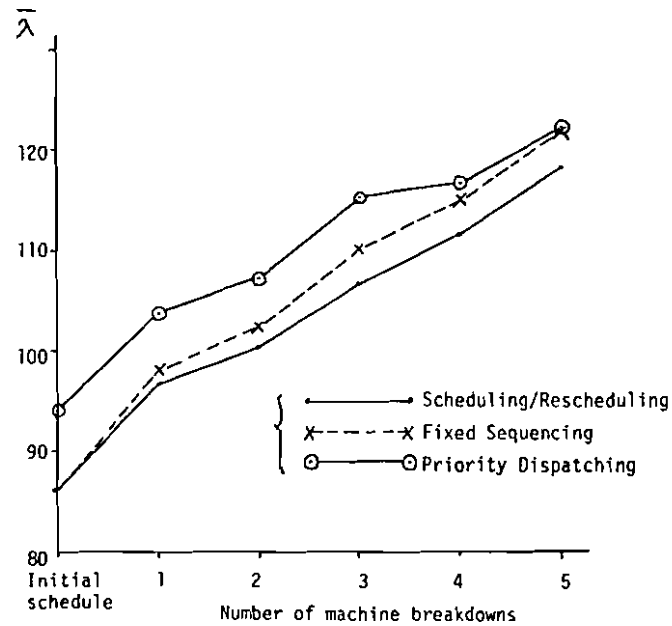


Figure 4. Increase of average total time by machine breakdowns.

(*a*) Comparison with fixed sequencing.

$\lambda_{RES}/\lambda_{FIX}$	After machine breakdown number					
Experiment number	0	1	2	3	4	5
1	†	98·0	98·0	98·1	100·0	96·7
2		96·0	96·3	96·5	98·2	95·1
3		100·0	98·0	98·0	90·5	90·5
4		96·2	96·2	96·6	96·6	104·9
5		97·9	97·9	100·9	100·0	100·8
6		96·0	98·0	96·7	96·7	95·9
7		98·0	98·0	98·2	94·0	89·7
8		97·9	100·0	94·6	100·9	99·2
9		95·7	96·1	98·1	98·2	101·8
10		100·0	100·0	100·0	95·0	96·6
Average‡		97·6	97·9	97·8	97·0	97·1
Number of the same or worse cases		2	2	2	3	3

† Both procedures start from a common initial schedule.
‡ Total average is 97·5%; that is 2·5% improvement.

(*b*) Comparison with priority despatching.

$\lambda_{RES}/\lambda_{PRI}$	After machine breakdown number					
Experiment number	0	1	2	3	4	5
1	91·5	92·5	92·5	89·7	94·0	100·0
2	91·5	90·5	98·1	92·4	94·1	97·5
3	91·5	92·2	91·7	91·7	89·7	82·0
4	91·5	98·0	90·9	92·0	98·3	98·5
5	91·5	100·0	96·0	101·8	100·0	101·7
6	91·5	92·4	94·3	93·6	100·0	98·3
7	91·5	92·5	92·5	93·2	93·2	93·1
8	91·5	94·0	94·5	95·5	96·6	105·9
9	91·5	98·2	92·5	91·1	100·0	98·3
10	91·5	92·9	92·9	92·9	91·1	92·7
Average†	91·5	94·3	93·6	93·4	95·7	96·8
Number of the same or worse cases	0	1	0	1	3	3

† Total average is 94·2%; that is 5·8% improvement.

Table 2. Comparisons of scheduling/rescheduling with alternative procedures—small CMS problem.

computation. Since the first argument may be much less significant in the automatic CMS, attention should be paid to the second.

CPU time was tallied for computation of each schedule in this study. In the first case study, the average CPU time for the scheduling/rescheduling procedure ranged from 12·6 seconds for the initial schedule (as for the fixed sequence procedure) down to 0·8 seconds with the rapid decrease in the size of the scheduling problem. The CPU time required to obtain the initial schedule for the fixed sequencing procedure is the same as for the scheduling/rescheduling procedure. In this experiment all schedules but the initial one required 0·8 seconds. In a practical situation, however, the computation time for the schedules would not be necessary because the system always keeps the given initial sequences.

For the priority despatching procedure about 0·7 seconds was required for each schedule in the experiments. In the practical use of the procedure a decision by priority rule is made at each machine locally every time a machine completes an operation. The number of operation completions does not change by the machine breakdowns, and therefore the total computation time for scheduling by priority despatching remains fixed.

In the second case study, the size of which may be of the order of practical situations, the CPU time required for the scheduling/rescheduling procedure increases. But, as shown in Table 4, a revised schedule could be generated within less than two minutes. In practical CMS operations such responses are certainly reasonable. It must be pointed out that these CPU times are obtained under certain given conditions of an approximate method, as mentioned before. We stress that the

Part type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of processes	2	2	4	1	2	1	2	2	1	2	2	2	1	2	2
Total processing time	36·4	73·0	127·6	49·6	42·1	44·6	93·7	36·9	42·0	68·9	80·9	44·3	34·5	59·5	47·3
Quantity required	9	4	10	7	6	8	2	9	3	4	5	5	6	1	12
Number of pallets	1	1	4	1	1	1	1	1	1	1	1	1	1	1	1

Total number of parts	91
Total number of operations	629
Total processing time	4690·8 minutes
Total number of pallets	18

Table 3. The problem data of Case 2.

Machine breakdown data				Scheduling procedure: Rescheduling		Fixed sequence		Priority despatching		Ratio (%)	
Number	M/C	Time	Duration	λ_{RES}	t	λ_{FIX}	t	λ_{PRI}	t	$\lambda_{RES}/\lambda_{FIX}$	$\lambda_{RES}/\lambda_{PRI}$
0 (Initial schedule)	—	—	—	967·2	125·0	967·2	125·0	1058·7	15·3	—	91·4
1	2	120	45	987·3	101·4	977·8	16·5	1060·4	14·9	101·0	93·1
2	1	250	90	957·8	103·0	1037·1	16·5	1062·5	14·5	92·3	90·1
3	3	280	50	964·5	95·7	1037·1	16·4	1062·5	14·4	93·0	90·8
4	5	390	60	1014·9	90·4	1044·7	16·6	1079·2	12·2	97·1	94·0
5	4	550	55	1038·9	93·3	1076·7	16·6	1093·3	11·6	96·5	95·0
6	6	630	35	1042·2	84·0	1076·7	16·6	1106·5	11·6	96·8	94·2
7	5	810	70	1106·4	22·0	1144·1	16·5	1172·7	11·3	96·7	94·3
8	1	970	30	1106·4	13·8	1144·1	16·4	1172·7	11·3	96·7	94·3
				Average:	91·0		32·1		14·6	96·3	93·0

Table 4. Comparison of scheduling/rescheduling with alternative procedures—large CMS problem. (λ—Total production time required, in minutes; t=CPU time in seconds.)

computation time can be regulated by the selection of parameters in the approximate method, according to the speed of the specific computer and the size of problems in the application of this procedure.

Analysis and conclusions

The scheduling/rescheduling approach is defined in this article as a procedure to revise an initial schedule each time certain significant changes occur in operation requirements. The feasibility of this approach in the manufacturing operating system environment of a CMS is also discussed, and two case studies of CMS operations are described. The operational changes that trigger rescheduling in our experiment are machine breakdowns. It has been shown in this experiment that on average the scheduling/rescheduling approach improves the performance of a small CMS by 2·5% and 5·8%, and of a practical CMS by 3·7% and 7·0% compared to the fixed sequencing and priority despatching procedures, respectively.

The case studies were designed to identify the effect of rescheduling when machine breakdowns occur. Therefore, breakdown duration and frequency of its occurrence are relatively exaggerated. If the breakdown ratio is decreasing, the necessity of entry into a rescheduling phase goes down. The extreme state, that no rescheduling take place, is the fixed sequence case itself. The results of the case studied here are better with fixed sequencing based on active schedule than those of the priority despatching used, even though machine breakdowns do occur.

Several other specific observations can be pointed out based on the experiment.

(1) The initial schedules generated by the active scheduling algorithm require less total production time relative to the schedules that result from priority despatching used here; about 9% less for both the small and large problems.

(2) When machine breakdowns occur, the above advantage (i.e., of the active schedule compared to priority despatching) is retained during the rescheduling phase when active scheduling is applied. However, since scheduling is performed over a finite period, there is a tendency to decrease this dominance as the number of machine breakdowns increases, especially for the breakdowns that occur at later times in the scheduled period, the advantage of rescheduling decreases.

(3) The schedule resulted by fixing the operations' sequences of the inital schedule is better than the results of priority despatching. This shows a certain degree of robustness of the initial active schedule. It may imply that it is not worthwhile to switch to rescheduling phase as long as the difference between a schedule and the actual progress is not over a specified limit. The setting of this limit is an interesting problem in itself. Another conclusion may be that, even if we cannot reschedule in some situation, we are guaranteed at least the performance of the fixed-sequence schedule.

(4) For the case study of the actual CMS, the computation time required to obtain the optimal scheduling solution becomes huge. In the computation shown here a tree search procedure was applied with a search limit of 200 nodes. In this experiment it means that in most cases the solution of the approximate schedule algorithm is the first solution of the tree search. However, λ of the first solution, 967·2 minutes, is close to the theoretical lower bound of 951·8 minutes. Thus there is only a remaining potential improvement of 1·6%.

Despite the indication gained to advantages that can accrue by the implementation of the scheduling/rescheduling approach, further research is still required on two main issues. First, it is necessary to establish the limits that will prevent too frequent and not necessarily beneficial schedule revisions, that may cause waste (e.g. of energy) in the CMS; second, careful economic justification is required for investment in the necessary computer capability *vis-à-vis* the potential performancé improvement.

Une méthode de programmation/reprogrammation est proposée pour le contrôle en temps réel d'un dispositif de fabrication informatisé géré par un système central d'exploitation de la fabrication. Avec cette méthode, il faut revoir les programmes lors de changements opérationnels importants tels que des pannes de machine. Des expérimentations faites dans le but d'évaluer le temps total de production d'un système de fabrication informatisé avec pannes et programmation/reprogrammation ont donné des avantages allant de 2,5% à 7% par comparaisons respectives avec des méthodes de séquentialisation fixe et d'envoi par priorité. Les temps de calcul requis pour les méthodes de programmation sur un CDC 6500/6600 ont aussi été examinés. La méthode de programmation/reprogrammation pour un dispositif réel demandait moins de deux minutes, et le temps de calcul peut être réglé par la sélection de paramètres dans une méthode approximative de programmation.

Ein Planungs-/Umplanungsverfahren wird für die Echtzeitüberwachung einer auf Rechner umgestellten Herstellungseinrichtung, die durch ein zentrales Herstellungsbetriebssystem geleitet wird, vorgeschlagen. Der Vorgang umfaßt Planungsüberprüfungen nach wichtigen betrieblichen Veränderungen wie z. B. Maschinenausfall. Versuche zur Feststellung der Gesamtproduktionszeit eines auf Rechner umgestellten Herstellungssystems mit Störungen bei der Planung/Umplanung führten zu Vorteilen von 2,5 bis 7,0% im Vergleich zu festgelegten sequenzierten bzw. Prioritätsversandverfahren. Die für die Planungsverfahren benötigte Rechnungszeit bei einem CDC 6500/6600 wurde ebenfalls untersucht. Das Planungs-/Umplanungsverfahren für eine effektive Einrichtung betrug weniger als 2 Minuten, und die Rechnungszeit kann durch die Wahl der Parameter bei einer approximierten Planungsmethode reguliert werden.

References

Baker, K. R., 1974, *Introduction to Sequencing and Scheduling* (New York: John Wiley and Sons, Inc.).

Bestwick, P. F., and Lockyer, K. G., 1979, A practical approach to production scheduling, *International Journal of Production Research*, **17**, 95.

Buzacott, J. A., and Shantikumar, J. G., 1980, Models for understanding flexible manufacturing systems, *AIIE Transactions*, **12**, 339.

Conway, R. W., Maxwell, W. L., and Miller, L. W., 1967, *Theory of Scheduling* (Reading: Addison-Wesley).

Godin, V. B., 1978, Interactive scheduling: historical survey and state of the art, *AIIE Transactions*, **10**, 331.

Hodgson, T. J., and McDonald, G. W., 1981, Interactive scheduling of a general flowshop, *Interfaces*, **11**, 83.

Hurrion, R. D., 1978, Investigation of visual interactive simulation methods using the job-shop scheduling problem, *Journal of the Operations Research Society*, **29**, 1085.

Kerry, T. M., 1980, Computer based scheduling for job-shops, *Proceedings of Autofact-West*, Society of Manufacturing Engineers, Anaheim, California, November, 179.

Mather, H., 1977, Reschedule the reschedules you just rescheduled—way of life for MRP?, *Production and Inventory Management*, 1st Qtr., p. 60.

Nof, S. Y., Barash, M. M., and Herald, M. J., 1978, Analysis of operating rules in a computerized manufacturing system, *ASME Publication*, No. 78-WA/Prod-38, December.

Nof, S. Y., and Gurecki, R., 1980, MDSS: manufacturing decision support system, *Proceedings of AIIE Spring Conference*, Atlanta, Georgia, May, 274.

Nof, S. Y., Whinston, A. B., and Bullers, W. I., 1980, Control and decision support in automatic manufacturing systems. *AIIE Transactions*, **12**, 156.

Yamamoto, M., 1977a, An approximate method of machine scheduling problems, *Sixth Management Science Colloquium*, Osaka.

Yamamoto, M., 1977b, An approximate solution of machine scheduling problems by decomposition method, *International Journal of Production Research*, **15**, 599.

Yamamoto, M., 1979, Scheduling software manual. Purdue University, School of Industrial Engineering, West Lafayette, Indiana 47907.

Yamamoto, M., and Nof, S. Y., 1982, A scheduling method in automatic manufacturing systems (in Japanese), *Journal of Japan Industrial Management Association*, **33**, 189.