Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

A SIMPLE SIMULATION FOR SCHEDULING IN A FLEXIBLE MANUFACTURING SYSTEM

Sherry D. Frese
The BDM Corporation
1801 Randolph Road, S.E.
Albuquerque, NM 87106, U.S.A

ABSTRACT

Scheduling a manufacturing system frequently involves a model or simulation of the system. The complexities of such systems often require sophisticated simulation techniques. These in turn require large sets of data, such as processing parameters and machine times, to accurately model the system.

This paper presents a much simpler approach to the modeling of a particular flexible manufacturing system (FMS). This approach has much smaller data requirements and uses very little computer time. The speed of the model makes it very useful in conjuction with a scheduling heuristic which requires repeated simulation of the system under different schedules.

1. INTRODUCTION

Simulations are often used a basis for scheduling in manufacturing systems. Normally these are based on modeling the paths of individual parts, using routing requirements and process times. In systems involving many parts with very short processing times, this can be prohibitive in both data requirements and modeling time. This expense also lessens the utility of the simulation as a tool for scheduling if iterative methods, requiring several simulation runs, are to be used.

Many of today's FMSs do not really require that level of detail especially if much of their operation is in a batch mode. In such a system, a simulation of batches rather than individual parts can be used.

The simulation discussed herein was developed as part of a supervisory control system for such an FMS. The simulation and scheduling are complicated by the fact that there synergy between different parts which can affect their processing times. Thus in some of the processes, knowing when a part begins the process does not insure that one can predict when it will finish. Instead, one must know what other parts are being run simultaneously to estimate the total processing time.

SLAM II simulations of this system were performed to assist in the sizing of equipment and to provide high-confidence predictions of potential throughput. However, both required too much data and used too much CPU time to be reasonably considered as a basis from which to schedule the system.

2. THE SYSTEM TO BE SCHEDULED

The system to be scheduled is an FMS which manufactures over 500 different part types which fall into 25 different classes. There is no significant set-up time involved in changing between parts within the same class. However, there is a 15 minute set-up associated with changing classes.

The system is scheduled on several levels. Outside the scope of our scheduling problem is the initial scheduling - comparing resources to orders and deciding which work orders are to be run on a given day. Instead, the scheduling problem of interest here is based on an agenda. A list of work orders, with associated priorities, is represented to the scheduler, and it must sequence them and assign them to the equipment. The objective is to complete the orders in as little time as possible.

2.1 System Operation

The system is illustrated in figure 1. It consists of two identical machining cells (labeled M1 and M2), a hardening furnace (labeled H1) and two tempering furnaces of different temperatures (labeled T1 and T2).

All parts follow the same basic flow. Raw material enters at a machining cell where it undergoes a series of six operations, including two inspections, to produce individual parts. The parts proceed through H1 and are then routed to either T1 or T2, depending on their size and the type of material from which they were produced.

Parts flow through the furnaces on conveyers consisting of two lanes. Thus parts from both M1 and M2 flow through H1, side-by-side. The conveyer travels at the same speed for both lanes. The tempering furnaces have similar conveyer arrangements.

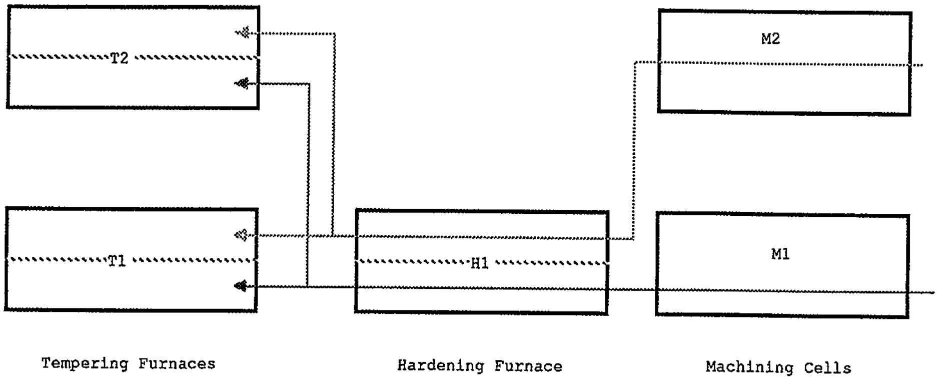


Figure 1: The FMS to be Scheduled

The raw material is loaded manually and varies with part type, hence thistoperator needs an agenda of parts to be run. At the end of the hardening furnace, H1, there is a quench operation. When parts types (actually part classes) are changed, a 15 minute manual set-up of the quench is required. The control system provides this window by holding new part types at the entrance to H1 for 15 minutes. However, the operator must be able to anticipate when the change will occur so that he can prepare for it.

2.2 The Scheduling Problem

At first glance, the scheduling of this system which is basically a transfer line, appears to be quite straightforward. Since the hardening furance can handle the output of both cells, as can either of the tempering furnaces, it would seem reasonable to merely divide the orders between the two cells based on simple estimates of their processing times. Of course the orders should be sorted in such a way as to minimize setup time, i.e. grouped by class.

The difficulty occurs because different classes of parts have different furnace speeds, sometime differing by a factor of two. While parts may be run at a slower than otpimum speed (within bounds), they cannot be run at a faster than optimum speed. Thus each furnace cannot be run any faster than the minimum of the speeds associated with the parts currently in it.

The parts spend a significant amount of time (1.5-2 hours) in the furnaces, so following a "fast" part with a "slow" part can significantly slow down the processing of the first order. Similarly, introducing "slow" parts along side "fast" parts can decrease production rates.

This interaction of orders also makes it impossible to predict the processing time of an order in isolation, since the processing time for an order is affected by the orders surrounding it in the schedule. Thus, to estimate how long it will take to complete a given schedule, a simulation, or predictor, which can account for the synergisms is required. However, if the predictor is to be used to determine an optimum schedule, it must be reasonably quick, so that alternative schedules can be examined.

3. SIMULATION APPROACH

The simulation approach used is very simple, and runs very quickly - on the order of 1 minute to produce a complete picture of the system's operation with 300 orders under a particular schedule. Of course, this predictor is used several times to develop an "optimal" schedule. (The word optimal is used advisedly, as the schedule developed is not necessarily optimal, but shuold be quite close to optimal within several other constraints which have been placed on the scheduling.)

3.1 The Events Used

The simulation was written in FORTRAN 77, on a MicroVAX II, in order to interface well with several other pieces of the system. It is an event stepped model, but the events are associated with entire orders, not individual parts. The events used for each order are listed in Table 1.

Table 1: Events Used in Simulation

EVENT	MNEMONIC
First part sawed	(SAW)
Last part sawed	(LSW)
First part enters H furnace	(FEH)
Last part enters H furnace	(LEH)
First part exits H furnace	(FXH)
Last part extis H furnace	(LXH)
First part enters T furnace	(FET)
Last part enters T furnace	(LET)
First part exits T furnace	(FXT)
Last part exits T furnace	(LXF)

The significant events are the SAW, LSW, FEH, FXH, LXH, FET, FXT, and LXT events. The others have been included to give a more complete picture of the system. They are also used to estimate times for other events with more accuracy than would be available otherwise.

Each event has a time associated with it. When a new event is created and placed in the event list, its time is also placed there. Thus, at time 0, there are 2 saw events, one for each cell placed on the list. As each event is "executed", new events are placed on the list with their times computed from the executed event. Figure 2 illustrates which events are produced as other events are executed.

After an event is executed, the next event in the list (that is, the one with the next time) is found and executed.

3.2 Estimating Event Times

The time estimates for the events are derived from the following principle:

> If it takes T seconds to process 1 part from point A to point B, and parts are produced at a rate dt (seconds/part), then it takes T+(N-1)*dt seconds to process N parts from point A to point B.

Of course T and dt will vary with part type, and, in this system, they can vary within a batch because of the furnace conveyers speeding up or slowing down. The events have been chosen in such a way that in generating other events only a T term or an (N-1)*dt term is required. This makes it simpler to adjust event times.

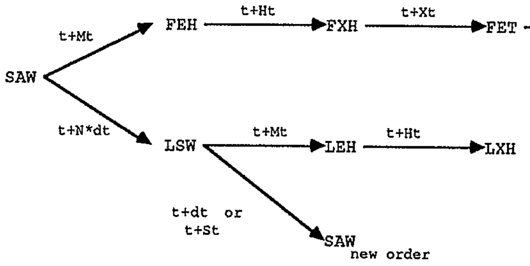
Thus, when the first part of an order is sawed (a SAW event) at time t_0 , one can estimate that the last part will be sawed a time t_0 + (N-1)dt. Similarly, when the first part enters the H1 furnace (an FEH event), we can estimate the time it will exit as (FEH time + (length of H1)/H conveyer speed).

Computation of dt (as is computation of T) in this simulation is complicated by the varying of the furnace speed. For example, if the mahcining times are such that the time between parts off the machining cell is 2 minutes, but the furnace can only take one part every 130 seconds, a backlog will quickly develop at the entry to the furnace. In this system, there are no queuing areas, hence dt is effectively 130 seconds. Of course if the furance speeds up, it will be able to admit parts more quickly, and dt may return to 120 seconds, the limit imposed by machining. Similarly the time required to get from point A to point B will vary with changes in furnace speed.

These changes are handled in the simulation by adjusting upcoming event times. Of course, the furnace speeds can only change when a new order enters a furnace or when the last of an order exits a furnace, i.e. at FEH, LXH, FET or LXT events. By having all the events generated by addition of T or (N-1)dt to the time of an event that has already occurred the adjustments of event times are simple.

For example, when the last of an order exits the H1 furnace, only orders with SAW, FEH, FXH, LEH, LXH, and LSW events on the list are affected, because they are the only ones with parts either in the H1 furnace, or scheduled to enter the H1 furnace. Any order which has an FXH event pending must have already had an FEH event occur. Since the FXH time was based soley on when the FEH occurred and the furnace speed, the new FXH time can be computed by using the ratio of the furnace speeds to adjust the time remaining until the FXH. Thus,

new FXH time =
T + (old FXH time - T) *
old H speed/new H speed,



Mt = time required for machining
Ht = time in H1 (Hlength/speed)
Tt = time in T (Tlength/speed)
Xt = time required for H to T transfer
St = Set-up time (if needed)
N = Number of parts -1
dt = delta t between parts
t = time of originating event

Figure 2: How Events Generate Other Events in the Simulation

where T is the time the H1 speed changed. Similarly, the LSW event is derived based on the dt and the time the SAW event occurs; so its adjustment is based upon the ratio of its previous dt and its new dt.

Similar changes can be made in the T furnace events when T furnace speeds change. Since T furnace changes can affect the dt computation, an adjustment of the H furnace events may also be required when a T speed changes.

4. USE OF THE SIMULATION IN SCHEDULING

The scheduling of this system is centered on a list of orders supplied from outside the system. The objective of the scheduling is to complete the orders in as little time as possible. Since both machining cells can produce any kind of part, the scheduling becomes a question of the sequence in which the orders will be done and upon which cell.

As was mentioned earlier, the first step is to sort the orders by class, and run all orders within the same class together. Normally, since splitting the class between the cells causes an extra set-up, one would schedule all of a class for the same cell. However, a large set of orders for a very slow class will slow down all of the orders run beside it, as well as those before and after it.

The preliminary sort (by furnace speed) provides a basic list from which to work. Of course, minimizing completion time means minimizing the number of setups performed. Thus, if the first several orders on the list are in the same class, they should be assigned to the same cell. The predictor is used to estimate when orders will complete sawing and then assign the next order (with appropriate provisions for set up time). This process of assigning orders from the sequenced list is illustrated in Figure 3.

Use of the predictor for scheduling does not stop here. Although one normally would require that a minimal number of die changes be made to minimize completion time, there are circumstances where this is not the case. In the event that there are very long strings of slow orders, followed by faster orders, it is conceivable that splitting the slow order(s) between 2 cells would decrease completion time by lessening the influence of the slow order(s) on the faster ones.

For example, we consider two part classes with the parameters shown in Table 2. If they are run side by side through the two machining cells, it would take 5.3 hours to complete Class 1 and, since its furnace speed is limited by the speed at which Class 2 can be run, Class 2 would be completed in 3.7 hours. Thus these two require 9 hours of system time.

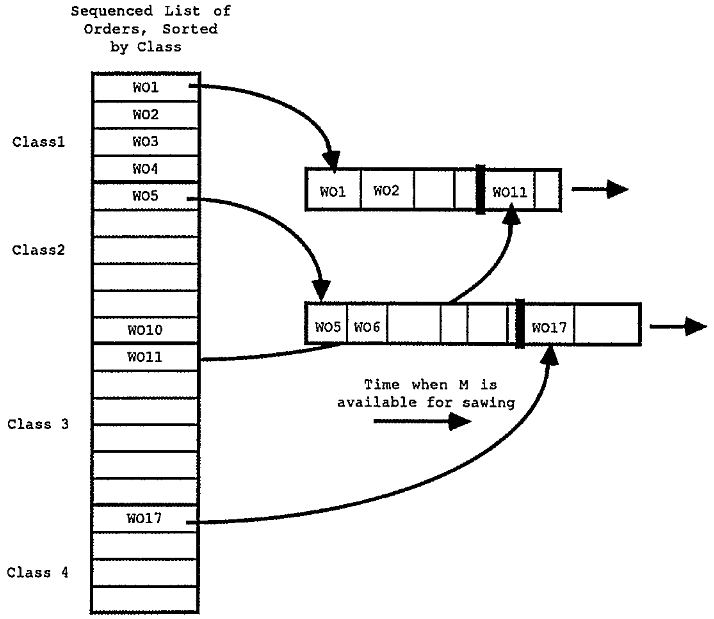


Figure 3: Use of the Simulation for Distributing Orders between the Two Machining Cells

Table 2: Sample Data for Two Classes of Parts

	Class 1	Class 2
Mach. dt (sec)	60	30
Total Machining time (sec)	90	70
Length (in)	30	20
H Speed (in/sec	.250	.500
T Speed	.250	.500
Quantity	100	100

H furnace - 50 ft., T furnace - 100 ft.

If Class 1 and Class 2 are each split into two pieces so that each class can be run along with itself, the influence of Class 2 is lessened, but an extra 2 die changes are introduced. It takes 8.6 hours of system time to complete the two classes in the schedule. If we are willing to live with uneven end times, or if there are more orders to schedule, the completion time could be cut to 8.5 hours by running Class 1 split and Class 2 whole, following the completion of half of Class 1.

The system we are scheduling has both very large lots and very small ones. Even with the predictor running at 1 minute per 300 orders, it would be prohibitive to consider all possible combinations of split and unsplit classes. Instead a "greedy" heuristic is used. The sequenced list is first run through the predictor with no classes split. Then each class is split into two pieces, one at a time, and the predictor is re-run. At each split, the shorter of the two system times is chosen in order to determine whether to leave the class whole or split.

This heuristic is greedy since it chooses the path along the local optima. This does not guarantee that the global optimum is achieved. However, in the system we are running, disparities in class size resemble those shown in the example, and hence significant savings can be produced following this method.

5. ANALYSIS OF RESULTS

A realistic collection of 300 orders in 14 classes with parameter ranges illustrated in table 3 was used to determine operating speed of the algorithm and to analyze the quality of the results.

The initial set of orders, arranged from slow to fast, required 100 hours of FMS system to complete. The final schedule required that 3 classes be split and resulted in a savings of 1.2 hours in system time. It took approximately 20 minutes of cpu time to produce this schedule.

It should be noted that much simpler cases were also run for comparison with the more complicated SLAM II model which could run very few part types, and required over 10 minutes of cpu time to model the processing of approximately 4000 parts in 3 classes. These comparisons showed production rates of 62 parts/hour per cell in the CLAM II model, as opposed to 63 parts/hour per cell in the simple predictor.

Table 3: Range of Parameters Used in Large Scheduling Problem

	Maximum	Minimum
WO size	100	2
H Speed (in/sec)	.6	.3
T Speed (in/sec)	.6	.3
WO's/Class	50	1

6. CONCLUSION

The nature of this particular application, with its competing furnace speeds, made this batch approach attractive, in that part-oriented modeling becomes very difficult under these circumstances. The SLAM II model required extensive Fortran code to modify internal event tables when furnace speeds changed.

This simple method of modeling a system is not adequate for developing confidence in the system's ability to operate without significant bottlenecks. It does not provide details of system operation such as machine utilization. However, its speed and simplicity make it useful for evaluating alternative scheduling strategies.

AUTHOR'S BIOGRAPHY

SHERRY FRESE is currently a manager and technical leader in The BDM Corporation's Advanced Manufacturing Group. She received a B.S. in mathematics from New Mexico State University in 1970 and a Ph.D. in mathematics from Northeastern University in 1975. She has taught mathematics at both Dartmouth College and New Mexico State University. At BDM she has been involved in numerous large software projects in both the scientific and manufacturing arenas.

Sherry D. Frese
The BDM Corporation
1801 Randolph Road, S.E.
Albuquerque, NM 87106, U.S.A.
(505) 848-5453