Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

A DISCRETE EVENT SCHEDULER IN A DYNAMIC PRODUCTION SYSTEM

Richard Q. Blackwell
Sr. Principal Engineer
Travenol Laboratories, Inc.
One Baxter Parkway
Deerfield, Illinois 60015

ABSTRACT

Many everyday production scheduling problems may not advantageously be represented by the classical optimum-seeking formulations of Operations Research. The assumptions underlying these approaches may be too limiting, the methods themselves not comprehensive enough or responsive to change, or the implementation too difficult for the intended user. On the other hand, heuristic approaches by themselves may not be capable of considering all relevant factors, and may not perform as well as the requirements dictate.

This article describes an approach which uses a deterministic event scheduling simulation as an adjusting procedure to develop a production schedule based on current equipment availability and batch status information. The problem is characterized by the sensitivity of the production materials to changes in processing time, and by extremely complex equipment constraints which make dispatching heuristics very difficult to define. The article details our experience in designing and implementing this approach, and discusses its performance.

1. INTRODUCTION

In practice the flow shop problem is perhaps the most commonly encountered of all the classical job scheduling problems. The objective is to schedule n jobs on m machines to optimize some measure of system effectiveness such as mean job flow time, subject to the special condition that all jobs follow the same sequence of operations (Conway, Maxwell, and Miller, 1967). An example is the ordinary assembly line. As important as the operation sequence restriction is in reducing the scope, it remains a very difficult problem to solve. The incentives, however, are substantial. Flow shop environments are characterized by very expensive equipment. Any additional capacity which may be developed by more effective scheduling is essentially free, and may forestall a significant investment. Moreover, the process of scheduling may be so difficult that its automation will reduce or eliminate a substantial expense. Both of these conditions were true in the situation described in this paper.

Many different optimization techniques have been proposed. Even conceptually, these are difficult, and in practice they generally fail because of their size and complexity, which prevent the preparation of anything like a realistic schedule without unreasonable programming and/or computation. After scanning the literature hopefully for a simple formulation which fits his needs, the practitioner is soon forced to consider approximate techniques such as heuristics.

Implementation of a heuristic is especially attractive in a situation where it is necessary to make dispatching type decisions at appropriate points in real time. An example would be the selection of the next piece of equipment as the current operation approaches completion, when several pieces might be used. It is possible however that no simple heuristic may be formulated which will adequately handle the situation. The particular conditions and constraints may be too complex, or heuristics may not adequately address some important general criteria.

An alternative is to attempt to duplicate the Gantt chart type of adjustment logic which human schedulers often employ. As tedious as manual line fitting can be, an experienced scheduler who knows the process can produce very effective although generally non-optimal schedules. Automating this sort of logic would eliminate the tedium, and allow the scheduler to simulate, and thereby develop even more effective schedules while allowing quicker response to changes. As anyone who has ever tried to write such a scheduler will confirm, however, it is extremely difficult to duplicate the intuitive human graphical ability.

The problem becomes apparent when one remembers that any operation currently scheduled on a processor depends upon the completion of many other preceeding operations. As new operations are considered, existing tentative assignments of operations to processors may have to be re-evaluated. Every re-evaluation requires the re-evaluation of all succeeding operations on that processor, as well as all preceeding operations for that job on other processors. Each of these re-evaluations generates others, and the problem snowballs. This logic is very complex to define, write, and test, and execution time can become horrendous.

The approach described in this paper attempts to mimic the Gantt chart approach without incurring the endless reconsideration. A deterministic simulation written in an event scheduling simulation language (SLAM) is used to generate a schedule which is guaranteed to be feasible. As the simulation unfolds, decisions are made in such a way that desirable choices are made within a framework which does not allow re-evaluation. Any incremental improvements which could be made by multi-pass adjustments are more than compensated for in this situation by the program's ability to consider options that the human scheduler could not, and by the certainty that the schedule will be acceptable.

2. DESCRIPTION OF THE PROBLEM

The program was written to automate production

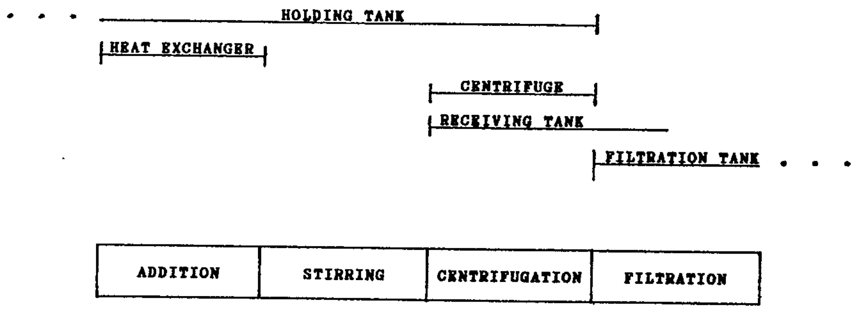

Figure 1: Batch Processing Equipment Utilization

scheduling in a plant which processes human blood plasma to produce various therapeutic agents for burn ictims and hemophiliacs. Through a complex set of operations, the plasma is reduced to various fractions which are then used by other departments in the plant. Although several different types of processing regimens are followed, and each has its own unique set of steps, they all may be reduced to series of stages which involve three operations - chemical addition, stirring, and centrifugation. Each of these stages has equipment requirements shown in Figure 1.

During the addition phase, alcohol and other chemicals are added to the plasma. Additions may have to be performed using a heat exchanger to cool the alcohol. The stirring phase causes the desired fractions to precipitate from the plasma. The centrifugation phase removes the precipitate from the supernate, and results in the batch moving from the holding tank into the receiving tank. In some cases, centrifugation may be followed by a filtration into another tank. Any batch may go through this same basic sequence up to four times. Thus, to plan each stage of each batch, it is necessary to schedule it through up to five pieces of equipment, a holding tank, a heat exchanger, a centrifuge, a receiving tank, and a filtration tank. As many as 20 batches of various types may be active at any time, all of them making conflicting demands on the same equipment.

To make the situation more difficult, the process yield depends critically on the length of the stirring operation. Variations either way from the optimum can drastically reduce the yield. It is possible to counteract this by slowing down the addition phase. Thus, if it is known in advance that there will be a delay in acquiring an acceptable centrifuge or receiving tank, the addition time may be extended to make sure that the stir time is optimal. This, of course, will also extend the time that the heat exchanger is in use. This set of conditions means that to optimize the process yield, it is necessary to plan the entire stage since heat exchanger requirements depend on centrifuge and receiving tank availability.

The complexity of manual scheduling was overwhelming. Over the years, the plant had developed template schedules which would process identical sets of batches acceptably at the cost of considerable equipment idle time. It was known that other schedules could reduce flow time and delays, and thereby increase capacity, but the incredible difficulty of developing and proving a new type of schedule prevented this type of improvement. The high marginal value of each batch and an increase in potential sales finally made it imperative that the department find a way to increase utilization or buy more equipment. It also became impossible to reduce demand to a simple set of template demands. More flexibility was necessary.

3. SELECTION OF APPROACH

Linear programming and other optimum seeking approaches were soon ruled out as feasible approaches to this problem. The number of distinct batch types, and the complexity of small but important production details which would have to be written into various kinds of structural constraints simply made any mathematical programming approach unworkable.

As critical as processing time is to yield, there is still a fair amount of random variation in the process. Thus every shift things happen which can affect the actual completion of any processing step, and hence the availability of the equipment needed for that step. The weekly schedule is merely the starting point and is not really essential for the day-to-day operation of the department. This suggested that a dynamic method which would respond periodically to actual experience and look ahead to plan equipment availability and setups would be most useful to department management.

Thus, very early on some sort of dynamic dispatching approach was attractive. The real problem was the requirement that stirring time be fixed, and the implied requirement that each stage be entirely planned. Since other batches might require the heat exchanger being selected for any given batch, the dispatching logic had to be smart enough to look ahead and accurately predict the heat exchanger completion times. Based on the availability of tanks and centrifuges, the only approach that seemed capable of this sort of precision was the Gantt chart approach. The probable complexity of accurate adjusting logic in this setting, however, was quite intimidating.

At the same time, we were planning on writing a simulation to investigate the effect of different dispatching rules on system capacity. Our intent was to develop a simple scheduler on the PC using dBase III, and to employ dispatching heuristics developed using the simulation. As the difficulty of the problem unfolded, we shifted from the dBase approach to the simulation itself. It gradually became clear that the resource logic within SLAM could be used as a schedule generator to resolve the equipment availability problems automatically, and thus to avoid writing the logic to carry out constant adjustments and re-evaluation.

The simulation makes one pass through time and generates one schedule which can process the load according to processing constraints embedded in the SLAM network language or FORTRAN subroutines. The advantage over manual methods was that the program could be more wide-ranging in its examination of candidate equipment than the human schedulers ever could afford to be. The benefit of this flexibility has far outweighed the disadvantages of not being able to adjust over multiple passes through a candidate schedule.

4. THE PROGRAM

SLAM was chosen as the framework within which to develop the scheduler for the following reasons.

- SLAM contains a network based language which can easily and effectively represent the basic logic underlying plasma batch processing.

- Where necessary, the network language can be expanded using FORTRAN subroutines. It was clear that the network language itself could not completely represent the scheduling logic.

- The SLAM network language includes a set of nodes which effectively model the impact of constrained resources. This capability can be used to make actual equipment selections.

- Eventual translation of the model from the mainframe to the PC was eased by the very close correspondence between mainframe and PC versions of SLAM.

The use of network based logical frameworks to represent complex systems is familiar to most practitioners. A brief description of SLAM resource modeling logic will make it easier to visualize our approach. SLAM is basically a list processing program which moves associated groups of data called entities from one list to another as simulated time advances. Each entity may at any point be associated with a resource by placing it in a list of entities currently using that resource. In this way, nodes in the network language or calls to FORTRAN subroutines may be used to control access to the resource. Our simulation uses this resource logic to guarantee feasible scheduling since any entity which has seized a particular resource must have done so when it was idle.

The scheduling model uses a very basic SLAM network representation of the four process steps in each stage (addition, stirring, centrifugation, and filtration), to move each plasma batch through time. At the beginning of each stage in the process, SLAM calls a subroutine which makes a tentative selection of all the equipment needed in that stage - tanks, heat exchangers and centrifuges. All need to be chosen at one time to insure that the stirring time is fixed. This tentative selection is made based on the best current information on anticipated completions. Thus when the entity returns to SLAM it knows of a set of pieces of equipment of each type (a zone) which is likely to contain an idle member at the necessary time. SLAM then moves the entity through each operation in the stage. At each point in simulated time that a piece of equipment is actually needed, SLAM first uses its own resource logic to seize a piece of equipment from the zone chosen previously, and then calls FORTRAN to figure out which piece of equipment has been seized. Heat exchangers are held until it becomes clear, via the SLAM resource logic, that a feasible receiving tank and centrifuge have become available. Only in this way can actual heat exchanger utilization be determined.

This somewhat convoluted approach was developed as a result of experimentation. The original design had the initial call to FORTRAN make selections of individual pieces of equipment, each of which was represented in SLAM as a separate resource. The idea was that FORTRAN could make effective decisions using current availability and heuristic decision rules. In practice, this didn't work well. The makespan for each stage is sufficiently long that good decisions based on current data become bad decisions as delays occur. The idea of equipment zones was intended to allow SLAM some flexibility in responding to inevitable changes and unforeseen circumstances. When equipment is actually needed in simulated time, the SLAM model has a range of good choices, and is then more likely to find a piece actually available.

The initial selection phase has been retained to act as a sort of valve. Even though the actual piece chosen will not necessarily be free when planned, as the logic looks through available zones it tends to spread its demands and increase the likelihood that a piece will be available in the chosen zone.

5. PERFORMANCE AND BENEFITS

The current version of the models run on an IBM mainframe. In this setting execution is extremely fast, on the order of 10 to 15 seconds for a six week long schedule. Since there are no random elements, and no replication, this is not surprising. There are two other measures of performance which are more important than speed, however.

- The model quickly generates acceptable schedules under new sets of demands.

- Average makespans of batches scheduled using the model are usually substantially less than current experience using the manual scheduling rules. The implied increase in capacity this represents results from relaxing the old rules,- and has allowed the department to schedule an increased load through the same equipment.

Table 1: Simulated Batch Transit Times (Hrs.)				
Prod. Type	Min.	Curr. Act.	Sim. Mean	Sim. Variance
A	155	180	165.6	74.0
B	134	159	144.7	79.7
C	92	-	97.1	31.0
D	37	45	46.0	88.7
E	109	120	121.7	228.0

While an actual statistical comparison of the performance of the model relative to current methods has not been developed, Table 1 shows typical results for several important batch types. In general, improvement increases as batch processing becomes more complex and lengthy since the model then has more opportunities to identify opportunities for improvement.

In addition to mechanizing the preparation of schedules, the model has been extensively used to examine potential changes to methods and equipment capacities. While yielding the benefits one would expect from this sort of simulation analysis of design alternatives, employing the model in this way has also been effective in developing credibility for the model at the plant level. Other less tangible benefits result from an enhanced responsiveness to unplanned circumstances, and increased control over batch completion times and yield.

All of the work described here has been performed on the mainframe. Current plans are to downsize and streamline the current model for installation on a fast PC. This installation will be embedded in a user interface to collect batch status data at the beginning of each shift. After data entry, the model would be re-run to determine equipment schedules for the following shift, and equipment setup schedules for the next two shifts. Eventually, there is no reason why the model could not communicate directly with existing process control equipment to automatically monitor batch status and automatically perform periodic schedule adjustments.

6. SUMMARY DISCUSSION

While this approach has been quite effective in this application, it is important to recognize that its success is to some extent a reflection of the difficulty of this particular situation. Manual scheduling was so difficult that almost any mechanized approach which could yield a feasible schedule would have represented a major improvement. There is no question that a multi-pass approach could reduce batch transit time further but this one-pass method has achieved a substantial reduction with much less effort than would be necessary to mount a more comprehensive effort. I believe that this method would be useful wherever a Gantt chart methodology seems to be the only way to model a particular schedule.

In the course of developing this model, our experience has highlighted several more general issues concerning the development of OR based models in an operational setting.

- There was no deadline set on this project. Management was concerned that it succeed, but from the start, it was perceived as a risky endeavor. The lack of deadline pressure allowed much experimentation and development to occur which would not have been possible under more usual project management conditions.

- The choice of the modeling world view, in this case, the event scheduling logic in SLAM, was critical in guiding us to a solution. The effect that the selection of approach has in focusing the attention and progress of the modeler is often overlooked.

- The sort of deterministic simulation employed in this application is rarely taught in the universities where much more attention is paid to the Monte Carlo methods. This is unfortunate since a deterministic approach can be quite powerful in tackling very difficult problems which will not yield to any of the more usual methods.

REFERENCES

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

Pritsker, A. A. B. (1984). Introduction to Simulation and Slam II. John Wiley & Sons, New York.

AUTHOR'S BIOGRAPHY

RICHARD Q. BLACKWELL is a Senior Industrial Engineer with Travenol Laboratories, Inc. in Deerfield, Illinois. He received a B.A. (1969) and an M.S.I.E. in Operations Research (1978) from the University of Illinois at Champaign-Urbana. Areas of particular interest include production planning, design of decision support systems, strategic planning, and simulation. He is a member of IIE and SCS.

Richard Q. Blackwell
Travenol Laboratories, Inc.
One Baxter Parkway
Deerfield, Illinois 60015
(312) 948-4341