NISTIR 89-4070

Real-Time Simulation and Production Scheduling Systems

Wayne J. Davis

Professor of General Engineering
University of Illinois
Urbana, Illinois 61801

Albert T. Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
Center for Manufacturing Engineering
Gaithersburg, MD 20899

April 1989

NISTIR 89-4070

Real-Time Simulation and Production Scheduling Systems

Wayne J. Davis

Professor of General Engineering
University of Illinois
Urbana, Illinois 61801

Albert T. Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
Center for Manufacturing Engineering
Gaithersburg, MD 20899

April 1989

National Bureau of Standards became the National Institute of Standards and Technology on August 23, 1988, when the Omnibus Trade and Competitiveness Act was signed. NIST retains all NBS functions. Its new programs will encourage improved use of technology by U.S. industry.

U.S. DEPARTMENT OF COMMERCE
Robert Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

REAL-TIME SIMULATION AND PRODUCTION SCHEDULING SYSTEMS

Wayne Davis
Department of General Engineering
University of Illinois
Urbana, Illinois 61801 USA

Albert Jones
Center for Manufacturing Engineering
National Institute of Standards & Technology
Gaithersburg, Maryland 20899 USA

ABSTRACT

The efficient scheduling of resources in a flexible manufacturing system (FMS) has a direct impact on a company's goal of increased profits. Many techniques, including mathematical programming, expert systems, and discrete event simulation have been used to solve these scheduling problems. However, they have all been ineffective in dealing with the unexpected delays that occur on the shop floor. This paper deals with a new approach to address production scheduling problems in an FMS - real-time, concurrent simulations. These simulations can be initialized to the current system state and run any time a new schedule is needed.

Keywords: automated manufacturing, optimization, production scheduling, real-time control, simulation, statistics

1. INTRODUCTION

Flexible Manufacturing Systems (FMS) have been installed in numerous factories around the world. Production scheduling is the function responsible for assigning FMS resources to various manufacturing tasks. Efficient use of these resources is critical to a company's goal of increased profits. In fact, poor scheduling decisions tend to reduce profits because they increase idle time on machines, cause bottlenecks on the shop floor, and push customer orders past their due date.

Mathematical programming approaches to solving the scheduling problem have received considerable attention in the literature. Graves [GRA81] and Raman [RAM85] have provided excellent surveys on these techniques. However, these techniques tend to have prohibitive computational requirements and restrictive assumptions. Another major drawback to these approaches is that they typically do not include material handling constraints. Discrete event simulation, expert systems, and other heuristics [MIL86, NOR86, JAC86] are other methods used to generate schedules. While simulation and expert systems packages allow the manufacturing system to be modelled to any level of detail, they still have unacceptable computational requirements. In addition, all of these methods generate feasible solutions with no measure of optimality. These undesirable properties tend to limit the applicability of all these approaches in a real FMS environment.

There are two other major drawbacks to all of the aforementioned approaches. First, they are all run "off-line", usually once or twice a day. Consequently, they are not able to respond quickly to unexpected events in the FMS. These events are usually handled on an ad hoc basis by the FMS supervisor with little understanding of the impact of his decisions on the overall schedule. Second, they do not take advantage of the vast amount of "real-time" shop floor data provided by FMS computer systems.

Davis and Jones [DAV88] have proposed an algorithm for real-time production scheduling (see Figure 1). The algorithm first selects R candidate scheduling alternatives and L performance indices to be used in evaluating those alternatives. Both selections based on actual shop floor data and are subject to continuous modification as the system evolves over time. On-line, concurrent, Monte Carlo simulations are run, in real-time, to evaluate the performance of these rules. This type of simulation analysis introduces four problems which typically are not addressed in the simulation literature.

First, these simulations forecast the future response of the system from an known initial state **S**, which is changing over time. The state S_k represents the actual "state" of the manufacturing system at the time trial k of the simulation analysis is initialized. This means that different simulations

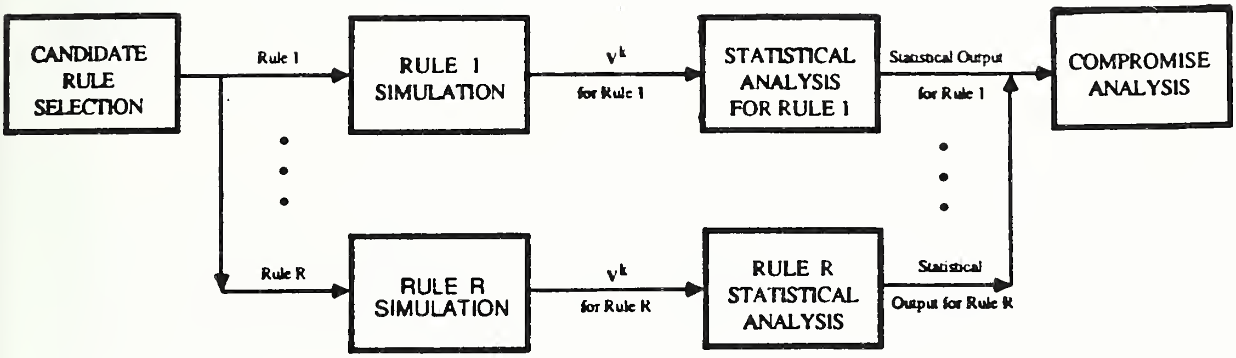

FIG. 1 SCHEMATIC FOR REAL-TIME PRODUCTION SCHEDULER

may have distinct initial conditions (see Figure 2). Section 2 defines what is meant by "state of the system" and addresses some of the problems that arise from initiating simulations in this manner.

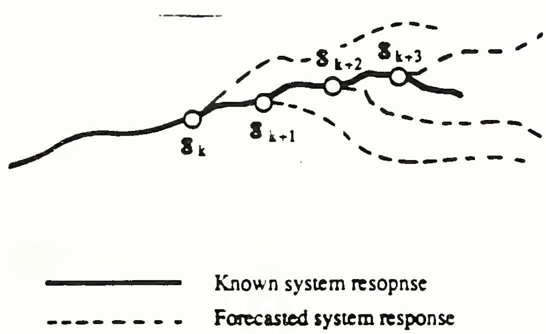

FIG 2. CHANGING THE INITIAL STATE

Second, an output data structure V^k must be defined which can support the performance calculations necessary to choose the best scheduling rule. In addition, this structure must be updated easily to incorporate shop floor events as they happen in real-time. These issues are discussed in Section 3.

The third problem arises in the calculation of statistics associated with a given performance criteria under a selected scheduling rule. These calculations are complicated because 1) each simulation trial is initialized to a possibly different state, and 2) previous outputs may have changed to reflect events that happened on the shop floor. Section 4 will explore these issues in detail.

The fourth problem area involves the consideration of multiple stochastic performance criteria to carry out a compromise analysis to determine the "best" rule. Section 5 addresses issues associated with that analysis.

2. INPUT DATA FOR THE SIMULATIONS

The input data for each simulation consists of two things: the current "state" of the system and process plans (see Figure 3). That state contains up-to-date information about all in-process jobs, processes, and the active schedule. Process plans contain all of the routing and timing data needed to create or update that schedule.

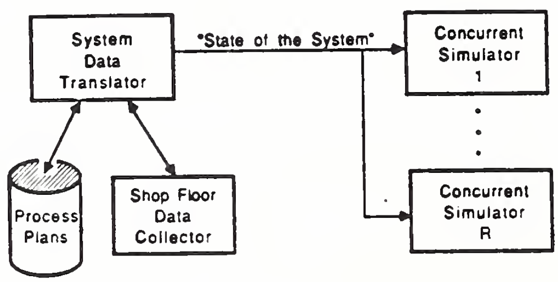

FIG 3. INPUT DATA FOR SIMULATIONS

2.1 Jobs

We assume that jobs JOB_j $(j=1,...,J)$ are active, with specified due dates D_j, and that JOB_j requires the fabrication of a single preplanned product type Φ_m $(m=1,...M)$. We also define $\#(JOB_j)$ to be the number of copies of Φ_m needed to fill the order. We can restrict $\#(JOB_j)$ to be less than or equal to the maximum number of product type Φ_m that can be carried in a single trip by the material transport system. If more than one delivery is needed, we simply create additional jobs. This, seemingly artificial, restriction is useful in scheduling the movement of material around the shop floor. These movements are typically ignored in most formulations of the production scheduling problem [GRA81, RAM85]. We note further that it is easy to obtain information about the original customer order from our definition of JOB.

The state information for each job includes job ID, current location (buffer, transporter, or process), due date, expected completion time, shop floor release time, current active routing, and expected/actual start and finish time at each process in that routing.

2.2 Processes

We assume that the FMS contains N distinct processes denoted by P_n ($n=1,\ldots,N$). These processes can be one of three types. A type 1 process can perform operations that physically alter the state of a job such as machining or deburring. A type 2 process can perform operations that ascertain the true attributes of a job such as inspection or non-destructive performance testing. Finally, a type 3 process can perform operations that change the physical location of a job such as conveyors, robots, or automated guided vehicles.

The state of type 1 and type 2 processes will contain the following information for each JOB_j at process P_n: job number for JOB_j, product number for JOB_j, batch size $\#(JOB_j)$, the planned delivery time (E'_{jn}), the planned start time (E_{jn}), the planned completion time (L_{jn}), and the planned pickup time (L'_{jn}). We note that for type 3 processes, transporters, these variables, with slightly different interpretations, are still valid. However, we also need information about each transporter, location and path. The topology of the transportation network has direct impact on the complexity of location and path. In small, simple networks the last node visited may suffice for location, and a list of nodes for path. In more complicated systems, the network may have to be partitioned into sectors each having a distinct ID. These sectors can then be used to determine both location and path.

2.3 Current Schedule

The production scheduler determines the values for the variables E_{jn}, L_{jn}, E'_{jn}, and L'_{jn}. These quantities are chosen to optimize some multi-criteria, utility function subject to several types of constraints: due dates, precedence relations, capacity, resource availability, and material handling. The optimization criteria could include minimizing tardiness, maximizing production throughout, or maximizing process utilization.

The resulting schedule contains timing data on all jobs and processes on the shop floor for some period T into the future. (Typically, T is one day or one shift.) For each process, that data includes the expected start and finish time for each job to be executed during T. For each job, that data includes the sequence of processes to be visited, and the start and finish times at that process. GANTT charts [BAK74] are the conventional method for representing all this data in one diagram.

2.4 Process Plans

Several pieces of information are required to schedule a NEW_JOB to be schedule. The JOB_ID j and due date D_j are provided by the production planner. All other information is provided by the process planner in a routing. A routing is either a completely-ordered or partially ordered listing of the processes needed to produce, transport, and inspect this NEW_JOB and the expected time spent at each process. If we allow only one, completely-ordered, M step routing then a simple ordered list of processes and durations is sufficient. If we allow the routing to be a partially-ordered list of M activities, then we must include the precedence relations among processes. This can be visualized using the concept of a PERT [BAK74] diagram. If we allow the scheduler to consider more than one routing for each NEW_JOB, then the preceding definitions are inadequate. One representation for such a generalized routing uses an AND/OR graph [DAV89] which is an extension of the PERT graph.

2.5 Remarks

We have described the inputs to the concurrent, real-time simulations. It is imperative that robust and efficient data structures be found to store and update these inputs as needed. We note three things. First, substantial testing will be required to determine the robustness and efficiency of candidate structures both in the laboratory and in the real-world. Second, shop floor data must be converted to the candidate structures before they can be used to initialize the simulations. Third, commercial simulation packages cannot easily be initialized to be predetermined state defined by an arbitrary collection of data structures.

3. SIMULATION OUTPUT DATA

The output from each simulation trial can be limited to $\mathbf{v}_{jn} = \{(E_{jn}, E'_{jn}, L_{jn}, L'_{jn})\}$. We can use the following matrix notation $\mathbf{V}=[\mathbf{v}_{jn}]$ to visualize the output from one trial run, for one potential scheduling rule, for J JOBs.

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_{11} & \cdots & \mathbf{v}_{1N} \\ \vdots & & \vdots \\ \mathbf{v}_{J1} & \cdots & \mathbf{v}_{JN} \end{bmatrix}$$

We note that whenever a routing requires a JOB_j to visit a given process P_n more than once, multiple entries for the corresponding $\mathbf{v}_{jn}$ may exist.

Distinct values for the elements of this data structure will be derived on each simulation trial for a given rule. These values allow one to determine the times when a given JOB is predicted to arrive at, be processed, and depart from each process. We can also determine all required information about the buffers and material transporters. Since this data is derived from detailed simulations of the manufacturing system, we can be sure that 1) no jobs are assigned to "down" processes, and 2) a feasible material handling strategy exists for effecting the predicted events.

Several performance criteria can be defined using the output matrix **V** from a given simulation trial of a proposed scheduling rule. These criteria typically fall into three classes. First, we can fixed a point in time and estimate the probability that a given event has occurred. For example, if we desire to compute the probability that a given JOB_j will be completed by time t_1, then the state of the system at t_1 must be analyzed (see Figure 4). In the second case, we may desire to develop an empirical distribution for the time at which a given event occurs. For example, one might desire to evaluate the expected tardiness of a given JOB_j which requires the distribution of completion time for JOB_j. Here the state of system at the completion time for JOB_j on the simulation trial k, t'_k, would be recorded as illustrated in Figure 4.

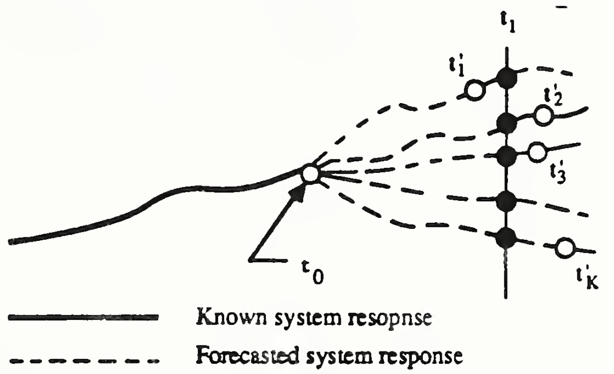

FIG. 4 PERFORMANCE MEASURES AND SIMULATION OUTPUT

Finally, the performance index might be defined upon the entire projected trajectory beyond t_0. For example one might desire to investigate the project utilization of a given process. We give several examples:

$$\text{Job Tardiness} = \max\left\{ 0, \max_n [\, L'_{jn} \,] \right\} = D_j^* \tag{1}$$

$$\text{Average Tardiness} = \sum_{j=1}^{J} \frac{D_j^*}{J} \tag{2}$$

$$\text{Process Utilization} = \sum_{j=1}^{J} \frac{(L'_{jn} - E'_{jn})}{\Gamma} \tag{3}$$

where the summation is over all JOBs that have process P_n in their active routing and Γ is the length of the simulation run on a given trial

$$\text{Job Flow Time} = \max_n \{L_{jn}\} - \min_n \{E_{jn}\} = FT_j \tag{4}$$

$$\text{Job Productivity} = \sum_{n=1}^{N} \frac{(L'_{jn} - E'_{jn})}{FT_j} \tag{5}$$

We note that these functions are valid regardless the initial state of a particular trial. The collection of performance measures from these trials provides the samples used in the statistical analysis discussed in the following section.

4. STATISTICAL ANALYSIS OF SIMULATION OUTPUT

We now discuss issues surrounding the estimation of statistics for all L performance measures $f_r^1(V^k)$ for each scheduling rule r = 1,...,R from the output from each simulation trial V^k.

4.1 Traditional Approach

When all K simulation trials are independent, start at the same initial state, schedule the same J JOBs, and run for the same amount of simulated time, we proceed as follows. First, we use the output V_r^k from trial k to get one estimate for each performance measure $f^1(V_r^k)$ for each fixed scheduling rule r. Using the output tables from the K runs $V_r^1,\ldots,V_r^K$ we can get a collection of sample estimates $f^1(V_r^1),\ldots,f^1(V_r^K)$. We then get an empirical cumulative density function for the $\text{prob}\{f^1(\bullet) \le z^1\}$ by dividing $\#\{f^1(\bullet) \le z^1\}$ by the number of observations.

$$\Pr[\, f_r^l(V_r^k) \le z \,] = F_r^l(z) \tag{6}$$

From this density function we compute estimates for the true means and variances:

$$\bar{f}_r = \text{mean or } \mathrm{Ex}[\, f_r \,] \tag{7}$$

$$\sigma_r^2 = \text{Sample Variance or } \mathrm{Ex}[\, (f_r - \bar{f}_r)^2 \,] \tag{8}$$

These statistics provide a summary for the performance of a given rule with respect to a single objective function.

Since all simulations schedule the same J JOBs on the same N processes over the same time horizon, we can use the terminating simulation approach described in [LAW82] to generate $(100 - \alpha_1)\%$ confidence intervals for the mean of each of the L objectives. Then, as also shown in [LAW82], the probability that these L intervals simultaneously contains their respective means is $> 1-\alpha$ where α is given by

$$\alpha = \sum_{l=1}^{L} \alpha_l$$

We note two things. First, this calculation does allow performance measures to be correlated. Second, the magnitude of α_l can be chosen to reflect the relative importance of performance measure l.

4.2 Issues in Real-time Simulations

The preceding statistical analysis ignores the fact that the systems continues to evolve during the actual scheduling analysis. We now discuss several issues which arise because of this phenomenon and their potential impact the preceding statistical analysis.

4.2.1 Updating Output Tables. The validity of an output table from a given simulation trial is limited by two facts. First, since each trial simulates a finite period of time, its life cannot extend beyond the time associated with last temporal event contained in the table. Second, the state of the system will change during the scheduling analysis. That is, as the system evolves, the output table for a completed trial may contain predictions for which have actual measurements are available. This was illustrated in Figure 2, where the trajectory initialized from S_k, (the initial state for trial k) does not pass through S_{k+1} (the initial state for trial k+1).

One might conclude that the forecasted trajectory is no longer useful. However, the trajectory does contain considerable forecasted information which is both valid and valuable to the scheduling analysis. Furthermore, if one discarded a simulation trial as soon as this happened, it would be very difficult to generate a sufficient number of trials for the meaningful computation of statistics and confidence intervals.

An alternative approach would be to update the simulation trial as the state of the system changes. In Figure 5, the evolution of a single simulated trial is depicted. Here "state" changes are recorded at intervals of length and the simulation output table is updated to pass the simulated trajectory through the measured state. The updating procedure must preserve the sampled duration of future processing tasks while ensuring precedence relationships and material handling constraints are satisfied. This, of course, may not always be possible. As soon as V^k is updated, all performance measures must also be updated. As noted above, the calculation of these performance is not impacted by these changes. But the statistic estimates and confidence interval calculations are.

4.2.2 Initializing Simulations. Let Δ represent the average time to compute a given simulation trial's output table V^k. Figure 6 Represents a more realistic depiction of the proposed real-time simulation environment. Here simulation trial k is initialized to the state S_k and the simulation trajectory for trial k is generated. During this simulation time Δ, the system evolves to state S_{k+1}. Ideally trial k+1 should be initialized from S_{k+1}.

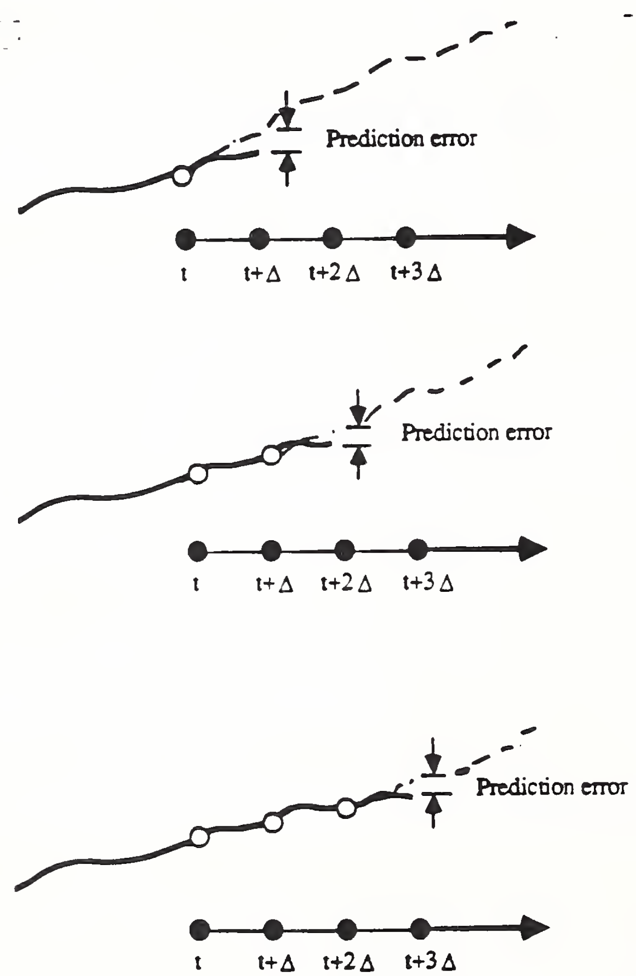

FIG. 5 UPDATING OUTPUT TRAJECTORIES

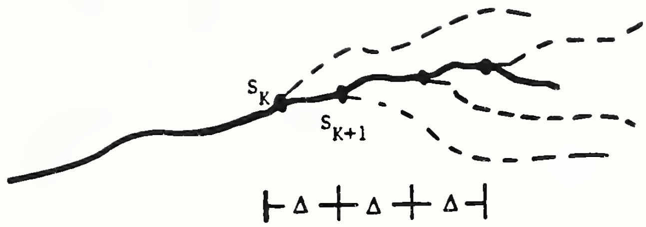

FIG. 6 SYSTEM EVOLUTION AT Δ INTERVALS

If we ignore this and use S_k to initialize all simulation trials, as shown in Figure 7, then we may deliberately introduce an error into the analysis. That error arises because, as discussed above, the probability that the simulated system response will pass through the known state S_{k+1} may be zero. On the other hand, if we use S_{k+1}, each simulation trial will start from a different initial state. The impact of this decision is discussed below.

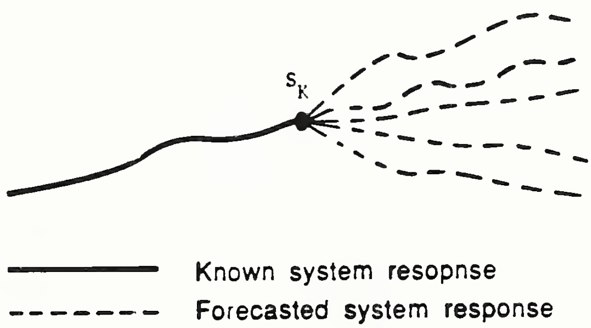

FIG. 7 ALL TRIALS INITIALIZED TO S_K

4.2.3 Calculating Statistics. State changes fall into three classes: number of jobs, number of processes, status of scheduled events. Whenever a state change is detected during a simulation analysis, two things happen. We must try to update output tables from completed trials and we must initiate the next trial at a different state than the current one. This can impact the statistical computations in two ways.

First, the calculation of means and variances for individual performance measures may be biased. The amount of bias depends on the type of change, the action taken, and the performance measure itself. Some changes, such as a new job to be scheduled or a process breakdown, will terminate the current analysis and cause a new one to be initiated. Others will require table updates and the recalculation of individual performance measures. To illustrate these points, consider (separately) the impact of a finished JOB and a delayed event during trial K. In the first case, output tables for trials 1,...,K will contain data about this JOB. However, output tables for the remaining trials in the analysis will not. We can easily remove all of the data related to that JOB from the first K output tables. The impact of this is most obvious on those performance measures that are directly related to the number of jobs in the system such as average tardiness. Instead of the first K computations being based on J jobs, and the remainder on J-1 jobs, all will be based on the same J-1 jobs, thereby removing this bias. The impact of this data removal on the statistical properties of performance measures not directly related to the number of jobs, such as process utilization, is minimal.

The difficulty involved in eliminating the bias introduced whenever a scheduled event will be delayed depends on the "severity" of the delay. Severity is measured in terms of its impact on the current schedule and completed trials 1,...,K. A delays which results 1) in any new material movements (a new tool delivery) or 2) from a process breakdown will automatically force a termination of the current analysis and a new schedule to be generated. Any delay which makes it impossible to update the output tables for trails 1,...,K, forces them to be discarded, and the analysis to be initiated again. This means that the delay invalidated so many of the events in the tables that it was necessary to run all of the simulations over again. All other delays require all performance measures to be recomputed, but they introduce little bias into the statistical calculations.

The second, and more significant, problem involves the calculation and accuracy of the individual confidence intervals. At this point in time, there are four major unresolved issues. The first involves the "type" of simulation analysis being performed. Since each trial is started from a different state, the simulations are not "terminating" according to [LAW82]. Moreover, since they also do not run long enough for the system to reach statistical equilibrium. Hence, they are also not "steady state" simulations. The second issue involves the selection of a stopping rule for each trial. If we simulate out to the end of the current day, then each trial will have a variable length. This introduces a bias into the calculation of all "time-weighted statistics". This can be removed by running each trial **T** simulated time units into the future. However, the impact of doing this on the statistical properties of other performance measures in not known.

The third issue arises from the fact that all of the statistical samples from a given trial are computed from the same output tables. This introduces the very likely possibility that there will be correlation (both positive and negative) among the performance measures. While we can arrange for independence between trial outputs by careful selection of random number streams, we cannot remove this type of correlation. The fourth issue concerns the accuracy of any confidence intervals computed in this way. The accuracy depends on the underlying distributions of the various performance measures. If measure l has a highly skewed or non-normal distribution, then the actual coverage may be significantly less then $(100 - \alpha)\%$. For those measures which are expressed as averages, such tardiness or process utilization, this should not be a problem. Measure such as makespan may require additional analysis before accurate confidence intervals can be derived.

At this point in time, there are no theoretical results which account for all of these issues.

4.3 Remarks

Although many of the above issues merit considerable further research, their criticality is directly related to the time required to perform table updates and complete the scheduling analysis. If that time is long, relative to frequency of events which cause changes in the initial state, then updating output trials and generating statistics to account for different initial conditions in the simulations are major concerns. As that time decreases, the probability that such events will occur is significantly reduced. We hope to achieve such a reduction by networking a collection of computer workstations and running simulation trials in parallel.

5. CHOOSING THE BEST SCHEDULING RULE

Whenever there is only one performance objective f, we can determine the best scheduling rule using one of two methods. If R=2, we can construct a "paired t-confidence interval" [LAW82] for the difference between the means $EX(f_1) - EX(f_2)$. This approach does not require independence across trials. In fact, a positive correlation can be beneficial, since it reduces the variance which reduces the size of the confidence interval. If R>2, we can use the "best of R systems" approach described in [LAW82]. This method does not use confidence intervals but does require independence across trials. The scheduler specifies two numbers: a probability P^*, and an "indifference amount" d^*. The method selects a rule r^* such that 1) the probability that r^* is the "best" rule is > P^*, and 2) the performance of r^* differs from the performance of the "best" rule by no more than d^*.

For arbitrary R and L, the selection of the best rule is less straightforward. The work done by [ZEL74] suggest that following approach. We first use the mean values defined in (7) to define the nondominated set of scheduling rules, denoted by R^*. R^* will be defined such that $r \in R^*$ if for every $r' \in R$ there exists an $l \in [1,\ldots,L]$ such that

$$\bar{f}_r^l > \bar{f}_{r'}^l$$

That is, scheduling rule r is contained in the nondominated set R^* if and only if it maximizes one of the L considered objectives in the mean sense. In actuality we have defined the Nadir set or the set of scheduling rules which contain the maximal mean values for the considered performance indices. Since it is unlikely that a single rule will simultaneously optimize all objectives, we define an "interval of compromise" for each objective l, $[m^l, M^l]$ where

$$m^l = \min_{r \in R^*} \{m_r^l\} \qquad M^l = \max_{r \in R^*} \{M_r^l\}$$

where m_r^l is the smallest and M_r^l is the largest of all the samples, f_r^l, from the simulation trials. That is m^l is the minimum value assumed by each performance measure and M^l is the maximum value. We can also define r_m^l and r_M^l to be those rules corresponding to the values m^l and M^l respectively.

This information provides the basis for choosing the "best" compromise rule, but methods for actually selecting that rule have not been developed for stochastic systems [DES86, FIS78, WHI79]. The current research in this area attempts to define a finite set of states that the decision-making environment can assume and then estimate the probability that each state will occur. This approach is similar to the definition of "state" used in the classic decision theoretic analysis [LAP81]. For a simulated system, however, there is a continuum of states that can be assumed by the system, and this set of states is obviously conditioned by the initial state employed in the simulation. Therefore, direct application of existing methodologies will be extremely difficult.

6. SUMMARY

We have discussed data requirements and statistical analysis techniques for using real-time, concurrent simulation as a tool for production scheduling. Our future work falls into several areas. First, we are developing real-time simulations for a small manufacturing system. Second, we are attempting to determine the frequency at which the simulation analysis should be performed. Should the analysis be run continuously, or should they be triggered by events? If the latter is the case, what events should be considered? Given the frequency of analysis, the next concern is the development of an interface to the data generated in advanced real-time simulation environments.

A third research concern is the development of simulation techniques that will support contingency analysis of rare events. That is, how does one compare expected performance against a potential rare event? This leads to the fundamental issue of how one performs comprise analysis among considered performance indices across a collection of potential scheduling rules.

In conclusion, it is becoming apparent that although the computational capacities are emerging to implement real-time simulation, several theoretical and practical concerns remain. The authors believe that these concerns will evolve into major research efforts within the near future.

7. REFERENCES

[BAK74] Baker, K. R., **Introduction to Sequencing and Scheduling**, John Wiley & Sons, New York, 1974.

[DAV88] Davis W. J. and Jones A. T., "A Real-Time Production Scheduler for a Stochastic Manufacturing Environment", **International Journal of Computer Integrated Manufacturing,** Vol. 1, No. 2, 1988, p. 101-112.

[DAV89] Davis W. J. and Jones A.T., "On-line Concurrent Simulation in Real-time Production Scheduling", **Proceedings of ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems**, Boston, MA, Aug., 1989.

[DES86] Dessouky, M. I., Ghiassi, M., and Davis, W. J., "Estimates of the Minimum Nondominated Criterion Values in Multi-Criteria Decision-Making, **"Engineering Costs and Production Economics"**, Vol. 10, 1986, p. 95-104.

[FIS78] Fishburn, P. C., "Stochastic Dominance without Transitive Preferences," Management Science, Vol. 24, No.12, pp. 1268-1277, 1978.

[GRA81] Graves, S. C., "A Review of Production Scheduling", **Operations Research**, Vol. 29, 1981, p. 646-675.

[JAC86] Jackson, R., and Jones, A., (eds), **Proceedings of the Symposium on Real-time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards Special Publication 724, Gaithersburg, Maryland, 1986.

[LAW82] Law, A. M. and Kelton, W. D., **Simulation Modeling and Analysis**, McGraw Hill, New York, 1982.

[LAP81] Lapin, L. L., **Quantitative Methods for Business Decisions**, (2nd ed.) Harcourt, Brace and Jovanovich, Inc., New York, 1981.

[MIL86] Miles, T., Erickson, C., and Batra, A., "Scheduling a Manufacturing Cell with Simulation", **Proceedings of 1986 Winter Simulation Conference,** Washington, D.C., 1986, p. 668-767.

[NOR86] Norman, T. A. and Norman, V. B., "Interactive Factory Scheduling using Discrete Simulation", **Proceedings of 1986 Winter Simulation Conference,** Washington, D. C., 1986, p. 665-667.

[RAM85] Raman, N., "A Survey of the Literature on Production Scheduling as it Pertains to Flexible Manufacturing Systems", **National Bureau of Standards Technical Report NBS-CGR-85-499,** Gaithersburg, MD, 1985.

[WHI79] Whitt, W., "Computing Probability Measures on a Set with an Intransitive Preference Relation," **Management Sciences**, Vol. 25, No. 6, pp. 505-511, 1979.

[ZEL74] Zeleny, M., **Linear Multiobjective Programming**, Springer-Verlag, New York, 1974.

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 89-4070	2. Performing Organ. Report No.	3. Publication Date APRIL 1989

4. TITLE AND SUBTITLE

Real-Time Simulation and Proauction Scheduling Systems

5. AUTHOR(S)

Wayne Davis, Albert Jones

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

The efficient scheduling of resources in a flexible manufacturing system (FMS) has a direct impact on the company's goal of increased profits. Many techniques, including mathematical programming, expert systems, and discrete event simulation have been used to solve these scheduling problems. However, they have all been ineffective in dealing with the unexpected delays that occur on the shop floor. This paper deals with a new approach to address production scheduling problems in an FMS - real-time, concurrent simulations. These simulations can be initialized to the current system state and run any time a new schedule is needed.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

automated manufacturing; optimization; production scheduling; real-time control; simulation, statistics

13. AVAILABILITY

[X] Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

19

15. Price

$9.95

USCOMM-DC 6043-P80