NISTIR 88-3837

On-Line Concurrent Simulation in Production Scheduling

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

August 1988

Issued February 1989

NISTIR 88-3837

On-Line Concurrent Simulation in Production Scheduling

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

August 1988

Issued February 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 A. Mosbacher, Secretary

Ernest Ambler, Acting Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

ON-LINE CONCURRENT SIMULATION IN PRODUCTION SCHEDULING

by

Wayne J. Davis
Professor
University of Illinois
Urbana, IL

Albert Jones
Automated Manufacturing Research Facility
National Institute of Standards and Technology
Gaithersburg, MD

ABSTRACT

Flexible manufacturing systems (FMS) have been installed in many factories around the world. Production scheduling is the function responsible for assigning FMS resources to various manufacturing tasks. On-line simulation is being used as an analysis tool to choose among several candidate scheduling rules. This paper defines on-line simulation, and describes the inputs to and outputs from the on-line simulation trials. It also addresses the statistical analysis of those outputs to determine the "best" compromise scheduling rule. Finally, it presents results from some preliminary scheduling experiments on the Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards.

1. INTRODUCTION

Flexible manufacturing systems (FMS) have been installed in many factories around the world. These systems typically contain several advanced machining centers tied together by an automated material handling system. Each piece of hardware is under some computer control, and the whole system is managed by an FMS supervisory computer. Considerable effort has been expended in integrating these FMSs with existing factory systems (scheduling, MRP, process planning, etc). Success varies considerably from one implementation to another.

In most cases, the tasks to be done in the FMS and the data required to perform those tasks are still generated by one or more factory existing systems external to the FMS. The tasks and data are transferred to the FMS supervisory computer over a computer network. The tasks are then decomposed into a set of commands for each piece of hardware in the system. The controlling computers monitor the execution of those commands and pass status feedback up to the FMS supervisor. This feedback is analyzed and sent up to the appropriate factory systems.

This implies that, today, most FMS systems have very little autonomy. They simply act as sophisticated executors of decisions made by someone else. Although they exercise "control" over internal systems to ensure that tasks are carried out correctly, they have little or no authority to change external decisions, even when unforeseen events occur.

This situation is slowly changing, and the FMS systems of tomorrow are expected to have increased autonomy. This means that they will make many of their own decisions and generate much, if not all, of the data needed to carry out those decisions. Consequently, the existing hierarchical structures [JON85] used to control FMS activities must be expanded to include decision-making.

Production scheduling is one such decision. Davis and Jones [DAV88] have recently proposed a algorithm for doing real-time production scheduling which can be integrated with most of the existing FMS hierarchical control architectures. That algorithm proposes to distribute scheduling across the levels of the hierarchy. This implies the ability to 1) quickly analyze alternatives at a given level, 2) perform contingency planning at each level, and 3) resolve conflicts between decisions at different levels. The primary analysis tools proposed for achieving these goals are on-line and real-time simulation.

On-line simulation requires each simulation to be initialized from a known state, which is tied to the actual state of the manufacturing system. That state remains constant throughout the scheduling analysis. Real-time simulation allows the initialization state to change from one trial to another, based on the actual evolution of the manufacturing system. This paper describes the approach being developed to performed on-line, simulation. A companion paper is being prepared to address the issues involved in doing real-time simulation.

2. PROBLEM STATEMENT

Before describing the techniques used in on-line simulation for production scheduling, we present some necessary background information on scheduling.

2.1 The Production Scheduling (PS) Problem

An FMS contains N distinct processes (see Figure 3) denoted by P_n (n=1,...,N). These processes can be one of three types. First, a process can perform operations that physically alter the state of a job such as machining or deburring. Second, a process can perform operations that ascertain the true attributes of the job such as inspection or performance testing. Finally, a process can perform operations that change the physical location of a job such as robots, conveyors, or automated guided vehicles (AGV).

We assume that jobs JOB_j (j=1,...,J) are available for scheduling. We also assume that each JOB_j has a specified due date D_j and requires the fabrication of a single preplanned product type ρ_m (m=1,...,M). The number of units of a given product type comprising the JOB_j will be denoted as $\#(JOB_j)$. We assume, without loss of generality, that $\#(JOB_j)$ is less than or equal to the maximum number of units of product ρ_m that can be transported in a single trip by the material handler, i.e. the maximum number of parts that will fit on a pallet or fixture. If more than one delivery is needed, we simply created new JOBs.

We note that this disaggregation is imperative to truly capture the material handling interactions which have been [GRA81, RAM85] neglected in most other formulations of the scheduling problem. We further note that the aggregation required to get any desired information about an original customer order from our definition of JOB is very simple.

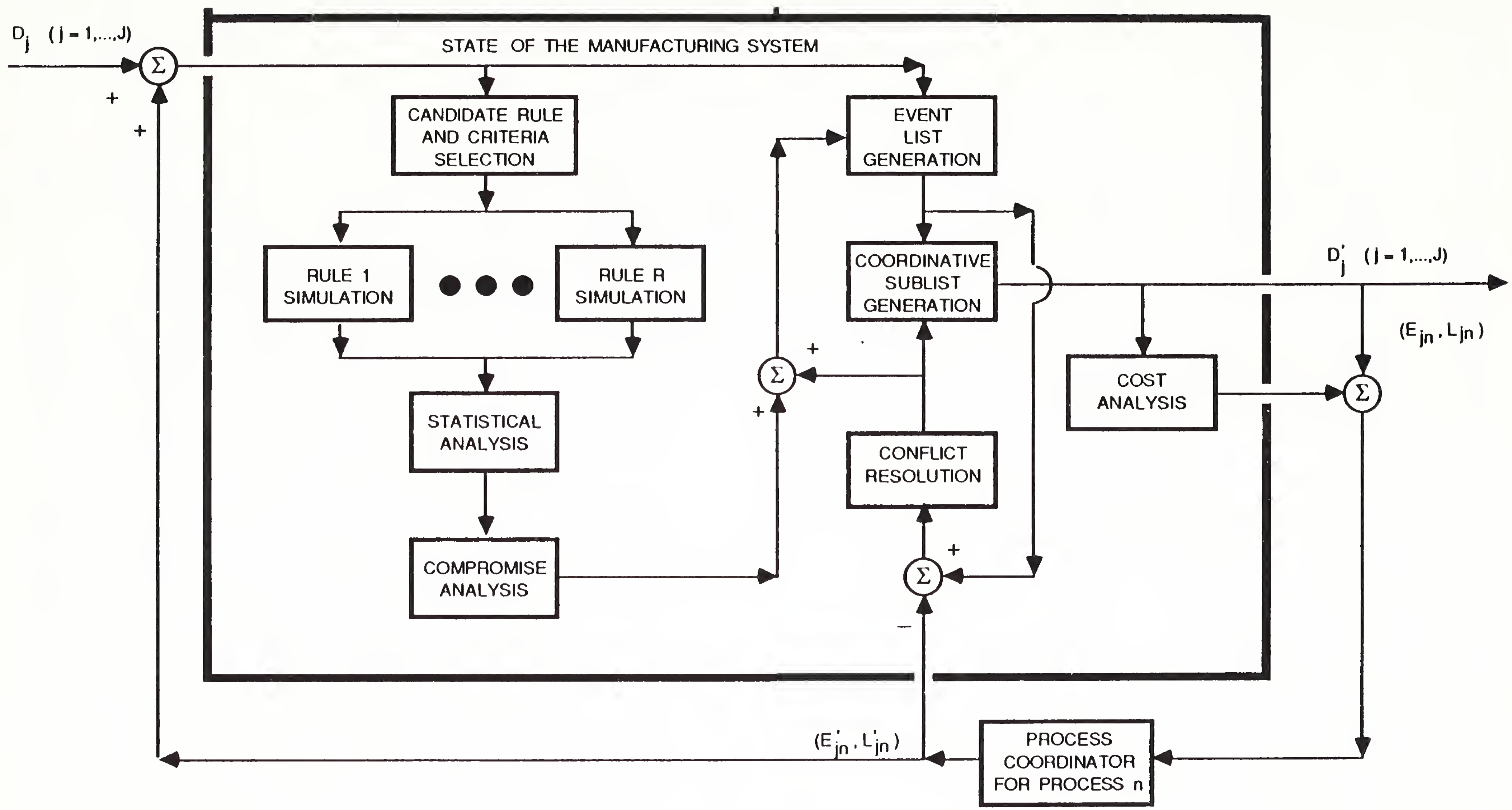

FIGURE 1. REAL-TIME SCHEDULING ALGORITHM

The production scheduler determines the following quantities

E_{jn} - the planned arrival time for JOB_j at process P_n,

L_{jn} - the planned pickup time for JOB_j at process P_n,

E'_{jn} - planned time for P_n to begin processing JOB_j, and

L'_{jn} - planned time for P_n to complete processing JOB_j

These quantities are chosen to optimize some multi-criteria, utility function subject to several types of constraints: due dates, precedence relations, processing capacity, resource availability, and material handling. The optimization criteria could include minimizing tardiness, maximizing production throughput, or maximizing process utilization.

2.2 Solution Methodologies

Mathematical programming approaches to solving the PS problem have received considerable attention in the literature. Graves [GRA81] and Raman [RAM85] have provided excellent surveys on these techniques. However, their computational requirements and restrictive assumptions, particularly about material handling constraints, tend to limit their applicability in a real FMS environment. Recently, off-line simulation studies [MIL86, NOR86] and AI heuristics [JAC86] have also become a popular means of generating schedules. Although these techniques do allow more realistic assumptions, they still have unacceptable computational inefficiencies. Furthermore, they only generate feasible solutions with no measure of optimality. In addition, these

methodologies have not been able to respond quickly to unexpected events in the FMS.

Davis and Jones [DAV88] proposed a algorithm for real-time production scheduling (see Figure 1). The foundations of this algorithm are two forms of simulation. The first consists of R on-line, concurrent evaluations of candidate scheduling rules. These are invoked whenever a new schedule needs to be generated, and are initialized to the current "state of the system". This occurs whenever a new job is ready for production or a shop floor problem invalidates the existing schedule. The second consists of continuously-running evaluations of potential scheduling rules to provide a contingency planning capability. This approach creates several problems in output analysis not covered in the existing simulation literature [LAW86].

This paper focuses on on-line simulation analysis. It first describes the data inputs to and outputs from the on-line simulation trials. It also addresses the statistical analysis of those outputs to determine the "best" scheduling rule. Finally, it presents results from some preliminary scheduling experiments on the Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards.

3. INPUT DATA FOR THE SIMULATIONS

The on-line, concurrent simulations described in [DAV88] require simulations to be initialized to the current "state" of the system. That state contains status information about the processes, buffers, and jobs currently on the shop floor. In addition, it includes the current schedule and information about the new jobs to be added to that schedule.

3.1 Processes

The state of each type 1 and type 2 process P_n contains the following information for each JOB_j at the process: job ID, the product type $_m$ corresponding to JOB_j, the batch size $\#(JOB_j)$, and E_{jn}, E'_{jn}, L_{jn}, and L'_{jn}. Although there are a variety of type 3 processes, material transportation devices, we limit our discussion to automatic guided vehicles (AGV). We note that expanding the definition to handle other devices is straight forward. In addition to a BUSY/IDLE indicator, the state of each AGV contains the following information for each JOB it is transporting: the JOB ID, destination and path being used, current location, expected completion time (E_{jn} for deliveries, L_{jn} for pickups). The topology of the transportation network has direct impact on the complexity of both location and path definitions. In small, simple networks the last node visited may suffice for location, and a list of nodes for the path. In more complicated systems, the network can be partitioned into sectors. These sectors IDs can then be used to define both pieces of data.

3.2 Buffers

Buffers are used as temporary storage repositories for work-in-process or raw material inventory. They can also be used to store other types of inventory such as tools, fixtures, and robot end effectors. Buffers typically have several distinct characteristics which impact the complexity of their state definition. Some buffers are located near and only store inventory for a unique process. Others can store inventory for more than one process regardless of their location. Some buffers have no natural ordering, such as bins. Others can have a two or three dimensional ordering, such as tables and shelves. Some buffers can hold one item per storage slot; others can hold several items per slot.

We limit our discussion to tables and shelves with one item per slot. Other implementations can be modeled as simple extensions of these two. For buffer tables, the state is a simple ordered list containing B entries, where B is the maximum number of items that the table can hold. Each entry contains the ITEM_ID and the time the item entered the buffer. For buffer shelves, the state is either an NxM or an NxMxL matrix. Each entry in these matrices has the same definitions as above.

3.3 Current Schedule

The current 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 each process. GANNT [BAK74] charts are the conventional method for representing all this information on one diagram (see Figure 2).

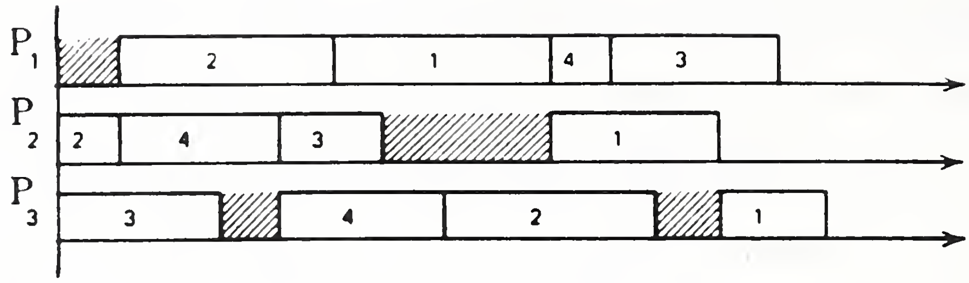

Figure 2. Sample GANNT Chart

3.4 Current Jobs

The "state of the system" also contains the progress of each job on the shop floor. The status of each job includes job ID, current location (buffer, transporter, or process), due date, expected completion time, shop floor release time, list of process to be used and any alternates, and expected/actual start and finish time at each process. The list of processes can be derived from the GANNT chart and is depicted in Figure 3.

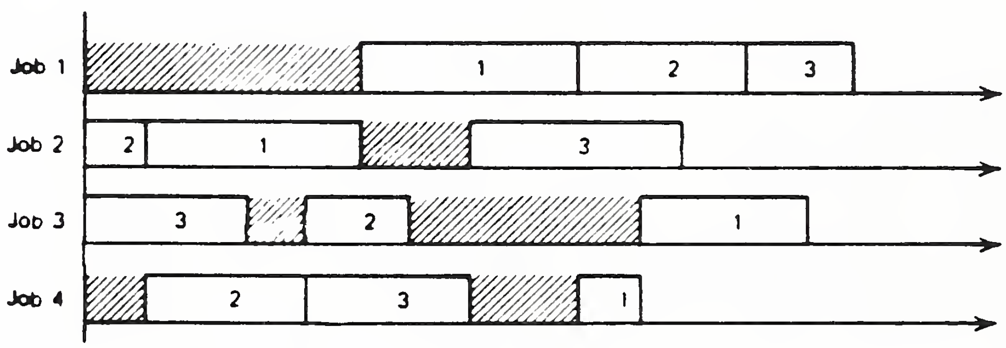

Figure 3. List of Processes to be visited by each JOB

3.5 New Jobs

Several pieces of information are required for each NEW_JOB to be schedule: a JOB_ID, due date, release time, expected completion time, and 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. ordered pairs (PROCESS_ID, DURATION). If we allow only one, completely-ordered, M step routing then a simple ordered list 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 (see Figure 4). Precedence relationships are enforced using the following convention: a given activity cannot begin until all activities ending at its start node have been completed.

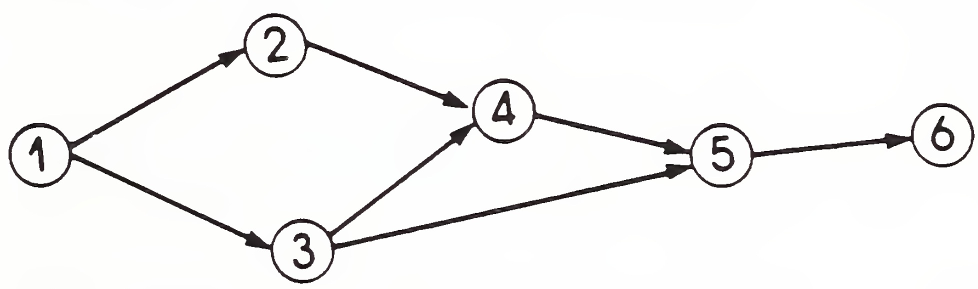

Figure 4. Sample PERT Diagram

If we allow the scheduler to consider more than one routing for each NEW_JOB, then the preceding definitions are inadequate. One possible representation for such a generalized routing uses an AND/OR graph (see Figure 5). This is an extension of the PERT graph used above. Each arc represents an activity, each activity has a start node and an end node, square nodes represent OR branches and circular nodes represents AND branches. Precedence relations are handled exactly as they described for the PERT diagram.

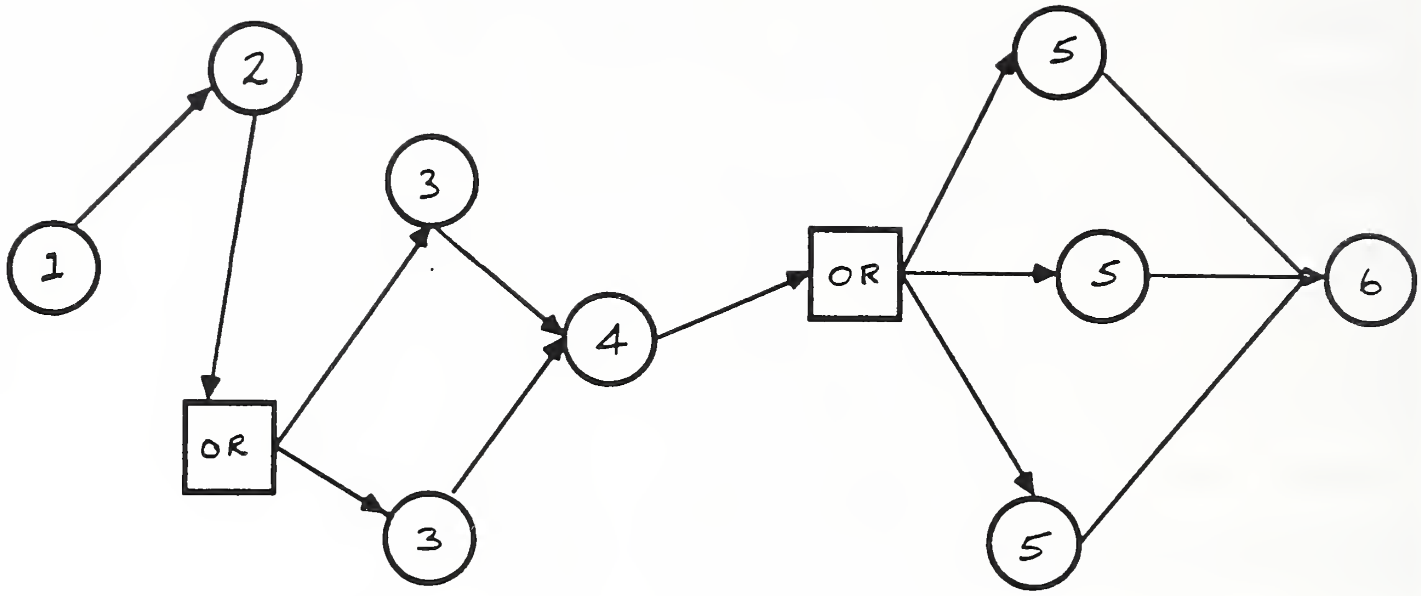

Figure 5. Sample AND/OR Graph

3.6 Remarks

We have described the information needed to define the "state" of the system which is used to initialize the real-time simulations. We are in the process of examining different data structures for storing and updating this information. Substantial testing is required to estimate the robustness and efficiency of various structures in both the laboratory and the real-world. There is an additional problem in a real-world FMS because the data required to generate those structures will come from the shop floor sensors and computers, the process planning data base, and the production scheduling data base. This "raw" data must be converted to the aforementioned structures before they can be used to initialize the R concurrent simulations.

Second, commercial simulation packages cannot easily be initialized to a predetermined state defined by an arbitrary collection of data structures. In SIMAN[1], this can be approximated by using the interactive debug facilities. The system state is saved in SIMAN format at prespecified time intervals. The generated system state file is then sent to each of the real-time concurrent simulators. Through their interactive debug facilities, each of the R concurrent simulations is initialized using the most recently transmitted file. However, the information contained within the SIMAN initialization file can be used to define the above data structures or the above data structures could also be employed to evaluate the SIMAN initialization file. The user must write all translators needed to carry out this conversion. Other simulation languages have similar features.

4. SIMULATION OUTPUT DATA

The output from each simulation trial k can be limited to $(E_{jn}, E'_{jn}, L_{jn}, L'_{jn})$. If we let $v_{jn} = (E_{jn}, E'_{jn}, L'_{jn}, L_{jn})$ we can use the following matrix notation to visualize the output from one trial run, for one potential scheduling rule, for J JOBs. Distinct values for the elements of this data structure will be derived on each simulation trial for a given rule. Each potential scheduling rule will be analyzed in precisely the same manner.

---

[1] Certain commercial equipment, instruments, or materials are identified in this paper. Such identification does not imply recommendation or indorsement by the National Bureau of Standards, nor does it imply that the material or equipment identified are necessarily the best available for the purpose.

$$V = \begin{vmatrix} v_{11} & v_{1N} \\ & \\ v_{J1} & v_{JN} \end{vmatrix}$$

The above definition for V obviously contains all the essential data to determine the times when each JOB_j arrived at given process P_n, when it was processed and when it departed from the given process. From this data, we can clearly determine the queue population at each process at any specific time. Also details pertaining to the specific material handling procedures used is given. Since the data were derived from a detailed simulation of the manufacturing system, we can be assured that a feasible material handling strategy does exist to effect the events as given. In summary, the defined data structure does impose some limitations upon the information that can be gathered from the simulation. However, as we will show, the compact structure does provide essential data to compute the performance criteria that are often considered in the simulation of manufacturing system.

Several performance criteria, can be defined using the output matrix V from a given simulation of a proposed scheduling rule. We give several examples:

$$\text{job tardiness} = \max\{\, 0, \max_n [\, L'_{jn} \,] \,\} = D^*_j \tag{1}$$

$$\text{average tardiness} = \sum_{j=1}^{J} D_j^* / J \qquad (2)$$

$$\text{process utilization} = \sum (L'_{jn} - E'_{jn}) / T \qquad (3)$$

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

$$\text{job flow time} = \max_n \{L_{jn}\} - \min_n \{E_{jn}\} \quad (\text{denote this } FT_j) \qquad (4)$$

$$\text{job productivity} = \sum_{n=1}^{N} (L'_{jn} - E'_{jn}) / FT_j \qquad (5)$$

We now discuss the statistical analysis required to analyze those performance measures.

4. STATISTICAL ANALYSIS OF SIMULATION OUTPUT

We first discuss methods for estimating performance measures $f_r^l(V^k)$ for l=1,...,L for a single scheduling rule r=1,...,R from the output V^k for k=1,...,K of K simulation trials. We then address the issue of comparing these performance measures across different scheduling rules to determine the "best" rule.

4.1 Estimating Performance Measures for a Single Rule

Assume, for now, that r is fixed. As noted above, on-line simulations are run whenever a new schedule must be generated. The inputs and the internal model must be updated to reflect the current state of system. All K simulation runs will use this as their initial state, will schedule the same J JOBs, and will run for the same amount of simulated time T. This is graphically depicted in Figure 6.

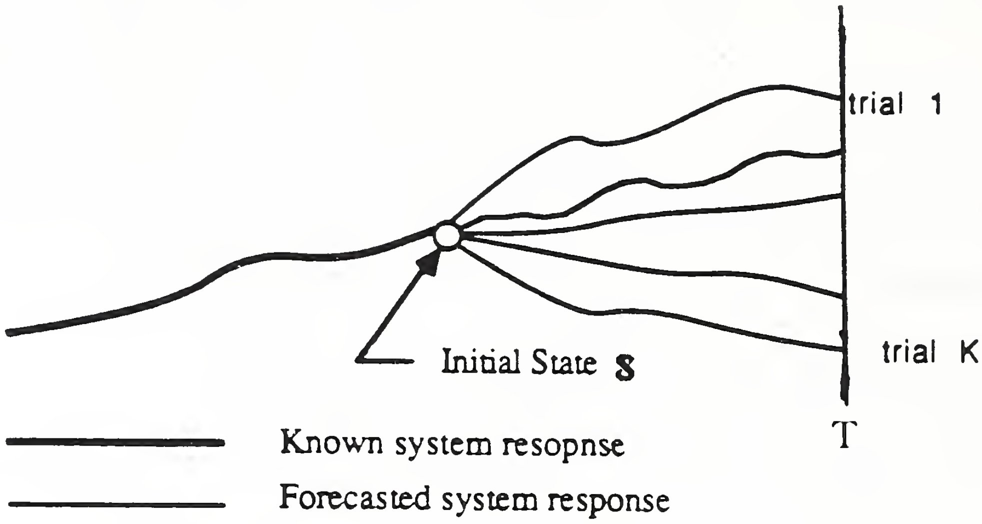

Figure 6. Graphic Depiction of On-Line Simulation

The output V_k from trial k can be used to get one estimate for each performance measure $f^l(\cdot)$ for scheduling rule r. Using the output $V_1,\ldots,V_K$ from the K runs, we get a collection of estimates $f^l(V_1),\ldots,f^l(V_K)$ from which we define an empirical cumulative density function for the prob{ $f^l(\cdot) \le z^l$ } as

$$F^l(z^l) = \# \{f^l(\cdot) \le z^l\} / K \quad (1)$$

From this empirical density, we compute the following statistics

$$\bar{f}^l = \text{Sample Mean or Ex } [f^l] \quad (2)$$

$$(s^l)^2 = \text{Sample Variance or Ex } [f^l - \bar{f}^l] \quad (3)$$

These statistics provide a summary for the performance of a given rule with respect to the various objective functions. We can also derive confidence intervals for the true mean Ex $[f^l]$ using the terminating simulation results given in [LAW86]. For example, if all simulation trials are run using different random number seeds, the 90% confidence interval is given by

$$\bar{f}^1 \pm t_{K-1,\,.05}\sqrt{\frac{(s^1)^2}{K}} \quad (4)$$

The preceding analysis can be carried for each of the R potential scheduling rules and each of the L objectives. The next step is to develop the best compromise scheduling rule across all objectives which we call r^*.

4.2 Comparing Scheduling Rules Across Performance Measures

We note that if R=2 and L=1, we can calculate a confidence interval for the difference between the two means to determine the best rule. Law and Kelton [LAW86] give two methods for computing such a confidence interval. One requires the estimates from the individual trials to be independent and one does not. We note further that whenever L=1, regardless of the value of R, we can use the "best of R systems" method described in [LAW86]. This method does not use confidence intervals. It attempts to select the "best" rule given a specified probability of making the correct selection. The method is straightforward, but does require independence across all trials.

For arbitrary R and L, we have developed the following approach to choose the best rule across all performance measures. First we determine the nondominated set of scheduling rules, denoted by R* using the approach in [DES86]. The set R^* will be defined here such that $r \in R^*$ if for every $r' \in R$ there exists an $l \in [1,...,L]$ such that

$$\bar{f}_r^l \geq \bar{f}_{r'}^l \qquad (5)$$

That is, scheduling rule r is in the nondominated set R^* if and only if it maximizes one of the L performance measures in the mean sense.

Since it is highly unlikely that a given rule will simultaneously maximize all L performance measures, we must calculate "compromise intervals" $[m^l, M^l]$ for each objective. Here, m^l is the minimum for each performance measure over R^* and M^l is the maximum for each performance measure over R^*.

$$m^l = \min_{r \in R^*} \{\bar{f}_r^l\} \qquad M^l = \max_{r \in R^*} \{\bar{f}_r^l\} \qquad (6)$$

We also define $r_m{}^l$ to be the rule which corresponds to m^l and $r_M{}^l$ to be the rule which corresponds to M^l. This gives the decision manker all the information needed to choose the "best" compromise strategy $r^* \in R^*$.

5. EARLY EXPERIMENTAL RESULTS

Preliminary testing of the on-line simulation approach is being done on the Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards (NBS). Before presenting some of our early results, we describe some of the relevant features of the AMRF.

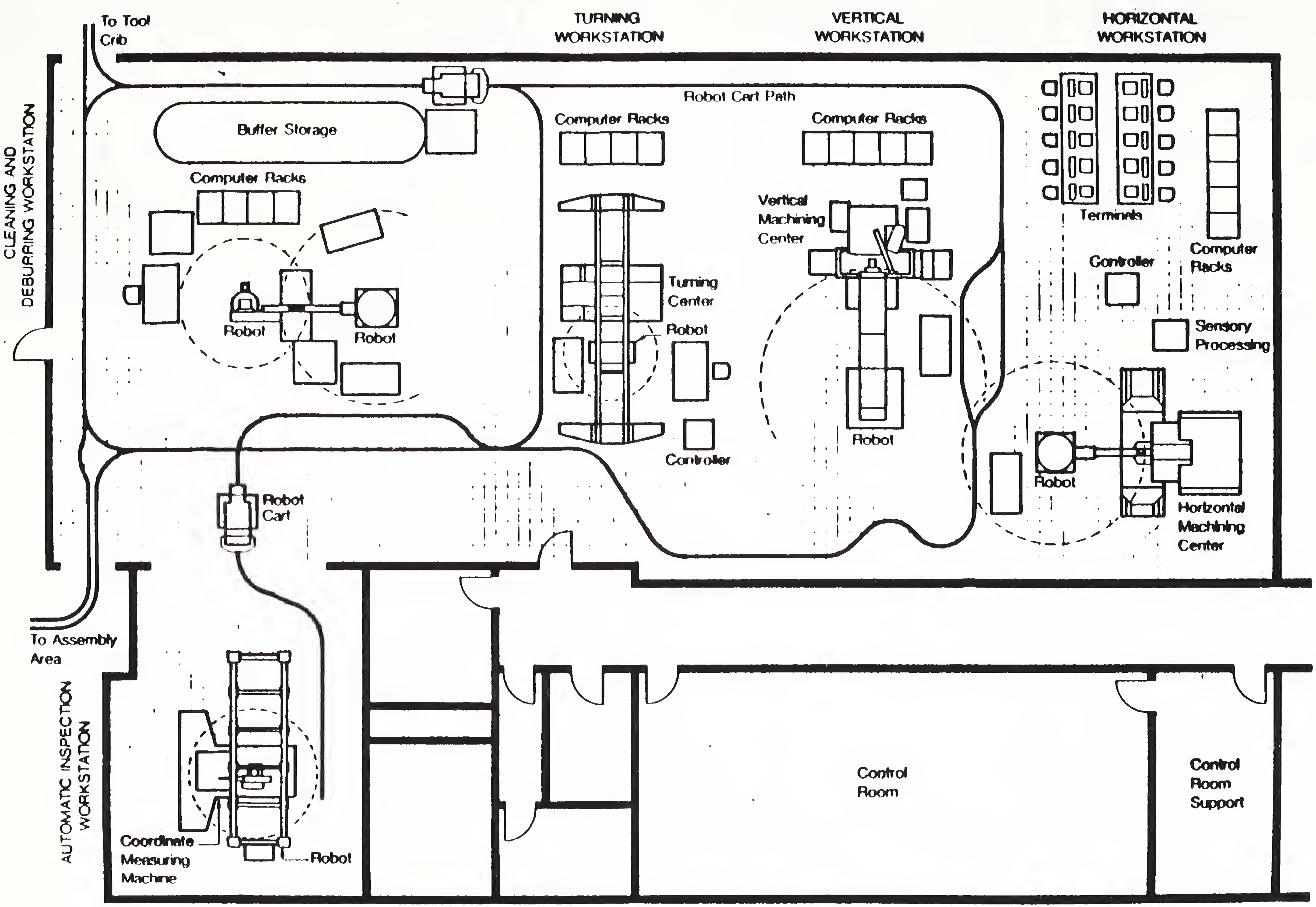

Figure 7. AUTOMATED MANUFACTURING RESEARCH FACILITY

5.1 The AMRF

The AMRF is a prototype FMS built at the National Bureau of Standards (NBS) where scientists conduct standards-related research for automated manufacturing systems [SIM82]. The AMRF contains six processes - three machining centers, a cleaning and deburring center, an inspection center, and a material transport center (see Figure 7). For our experiments, we assumed that each process had one input buffer and one output buffer, but could not queue any JOBs. We also assumed that each JOB had exactly one route that it could use to go through the processes.

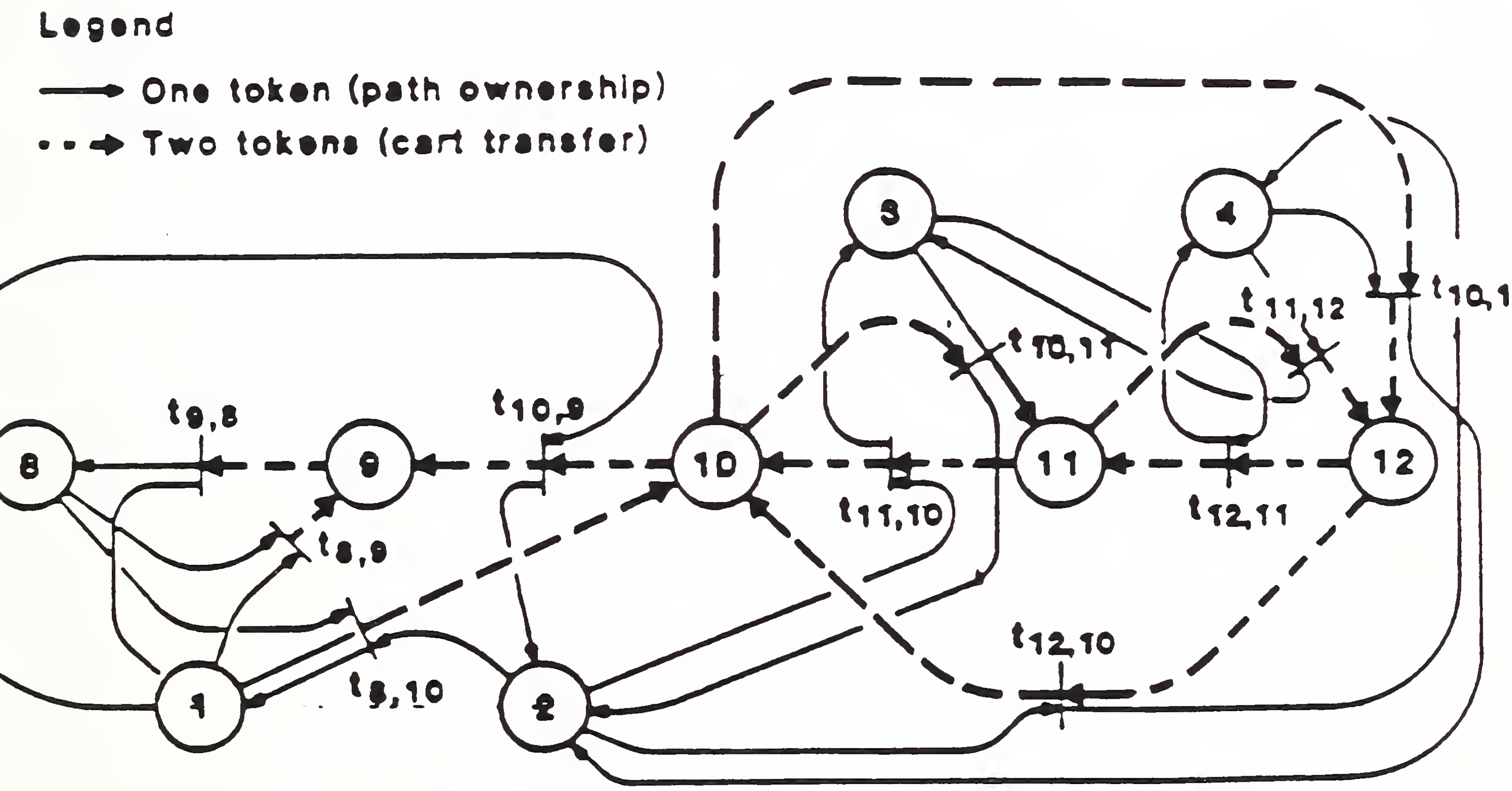

Figure 8. PETRI Net Representation of Cart Path

Material transportation posed some interesting problems in trying to generate schedules for the AMRF. Since there are two bi-directional AGVs operating on a single track which contains no loops, there is a very real potential for deadlocks. We used a PETRI net [AGE79] model (see Figure 8) to analyze potential deadlocks for material transfers. The nodes represent the various places that the cart can stop.

5.2 Initial Simulation Results

We are still in the model verification and time testing phase of the simulation analysis of the AMRF. To complete this phase, we are using a single scheduling rule and three different performance measures: time in system, productivity, and process utilization. Using a SIMAN simulation package on an INTEL 80286 based personal computer, we can simulate the future response of the system at approximately 1000 times the speed at which a system emulation takes place. These numbers are based on results obtained from making 100 runs with each run scheduling 50 JOBs. We are in the process of analyzing the performance measures obtained from the various trials to verify the current AMRF SIMAN model.

6. SUMMARY

We have discussed data requirements and statistical analysis techniques for using on-line simulation as a tool for real-time production scheduling. We have also presented some very early results from our testing on the AMRF. Our future work falls into several areas. First, we will complete the model verification and time testing of the AMRF. We expect to increase the speed by a factor of 10. Second, we will expand the current on-line analysis to include several scheduling rules. We intend to use a network of PCs to conduct the concurrent simulation trials. Third, we will relax the assumption that all trials must be initialized to the exact same state. This "real-time" simulation approach recognizes that the system continues to evolve during the scheduling analysis. We intend to examine the impact of this on both the input and output structures and the statistical analysis. This type of simulation has not been investigated. Finally, we will initiate a study on both conventional and AI approaches to developing data structures for representing the state of the system and the simulation output.

7. REFERENCES

[AGE79] Agerwala, T., "Putting Petri Nets to Work", **Computer**, 85-93, December, 1979.

[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, 101-112, 1988.

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

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

[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.

[JON85] Jones, A. and Whitt, N. (eds.), "Proceedings of Factory Standards Model Conference", National Bureau of Standards, November, 1985.

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

[MIL86] Miles, T., Erickson, C. and Batra, A., "Scheduling a Manufacturing Cell with Simulation," **Proc. of 1986 Winter Simulation Conference**, Washington, D.C., pp. 668-676.

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

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

[SIM82] Simpson, J., Hocken, R., and Albus, J., "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing Systems, Vol. 1, No. 1, pp. 17-31, 1982.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 88-3837	2. Performing Organ. Report No.	3. Publication Date FEBRUARY 1989

4. TITLE AND SUBTITLE

On-Line Concurrent Simulation in Production Scheduling

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)

National Bureau of Standards
Building 220 Room A-127
Gaithersburg, MD 20899

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)

Flexible manufacturing systems (FMS) have been installed in many factories around the world. Production scheduling is the function responsible for assigning FMS resources to various manufactring tasks. On-line simulation is being used as an analysis tool to choose among several candidate scheduling rules. This paper defines on-line simulation, and describes the inputs to and outputs from the on-line simulation trails. It also addresses the statistical analysis of those outputs to determine the "best" compromise scheduling rule. Finally, it presents results from some preliminary scheduling experiments on the Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards.

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

automation, flexible manufacturing, production control, simulation, scheduling statistics.

13. AVAILABILITY

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

14. NO. OF PRINTED PAGES
26

15. Price
$12.95