NISTIR 88-3891

Artificial Intelligence Techniques in Real-Time 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

November 1988

Issued February 1989

NISTIR 88-3891

Artificial Intelligence Techniques in Real-Time 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

November 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

ARTIFICIAL INTELLIGENCE TECHNIQUES IN REAL-TIME PRODUCTION SCHEDULING

Wayne J. Davis
Professor of General Engineering
University of Illinois
Urbana, IL 61801

Albert Jones
Deputy Project Manager, AMRF
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

This paper addresses the real-time production scheduling problem as a special case of a much larger class of real-time decision-making/control problems. The paper first reviews the definition of the scheduling problem, and then reviews a earlier algorithm proposed by the authors to address this problem. It then concentrates on the possible application of various AI techniques to many of the functions that make up that algorithm.

Keywords: artificial intelligence, automation, expert systems, information models, production scheduling, real-time decision-making, simulation

1.0 INTRODUCTION

Solution methodologies to classic decision-making and control problems in dynamic systems are undergoing a major metamorphosis. The significant thrust has been to computerize, automate, and integrate these methodologies, effectively removing the human being from the loop. Certainly one of the most publicized areas in which this is happening is discrete parts manufacturing. A particular problem that has received considerable research attention within the manufacturing arena is the production scheduling problem. This problem possesses several properties which allow it to serve as a representative example for the larger class of real-time decision-making/control problems.

First, the decision constraints and performance objectives must be modified frequently to reflect unexpected events in system's evolution. When these events occur, the problem must be reformulated and resolved as quickly as possible. Second, those events take place in a unpredictable, and usually stochastic manner. This implies that the robustness of any decision must be demonstrated against these uncertainties. Third, inputs and outputs for this problem must be coordinated with the inputs and outputs for other problems. For example the output from process planning is used by production scheduling. Fourth, the problem focuses upon a time interval which demands quick determination and implementation of a solution. Finally, the introduction of computers and advanced sensors furnish a great deal of feedback data on a real-time basis. That forces a control mechanism to be in place which can use that data in resolving deviations between planned and actual system response.

Davis and Jones [DAV88] have proposed an algorithm which has general applicability to such real-time decision-making problems. Their first application was to production scheduling within a hierarchical control framework. After defining the Production Scheduling problem, we will present an overview of that algorithm. We pay specific attention to the role that AI can play in carrying out the functions contained in the algorithm and managing their input/output data. We plan to use expert systems in a supporting role in generating real-time production schedules. In addition, we expect some of the information modeling methodologies used in AI to speed up many of the data handling and analysis functions in the algorithm.

2.0 THE PRODUCTION SCHEDULING PROBLEM (PSP)

We now give a formal statement of the production scheduling problem (PSP). We assume that JOB_j ($j=1,...,J$) has associated due date D_j and requires the production of a specific product ρ_m ($m=1,...,M$). We further assume that processes P_n ($n=1,...,N$) are available, and that $TASK_{ijn}$ ($i=1,...,I$) represent the individual processing tasks to be performed on JOB_j by P_n. If we define

E_{ijn} as the earliest start time for $TASK_{ijn}$
L_{ijn} as the latest finish time for $TASK_{ijn}$

the PSP is to optimize the utility function

$$W[f^1(E_{111},....,L_{IJN}),...,f^L(\ )]$$

where the $f^l(\ )$ are the criteria to be considered in the optimization. These criteria could include the minimization of tardiness with respect to assigned due dates, the maximization of production throughput or the maximization of process utilization. The optimization is carried out with several technological constraints. They include due dates, material handling, precedence relationships among the processes, precedence relationships among the tasks, and alternative routings.

An exact mathematical representation of the objectives and constraints for a given the production scheduling problem is quite complex. For a generic representation of this problem, and a detailed formulation, the reader should consult [MCP86,ROD86]. For a survey of mathematical programming approaches to solving the production scheduling problem, the reader should consult [RAM85, GRA82]. For a summary of some of the recent work in this area the reader is referred to [JAC86].

Davis and Jones [DAV88] proposed a decomposition of the production scheduling problem into two levels (see Figure 1). The top level, the supremal, determines the start and finish times of each JOB_j at each process P_n, E_{jn} and L_{jn} respectively. The bottom level, the infimal, uses these bounds to determine the start and finish times for the $TASK_{ijn}$. The authors made two important and realistic assumptions in developing this decomposition. First, decision makers at each level will behave in a cooperative fashion in solving their own problems. Second, the decision maker at the Process Coordinator level possess more detailed information about the variables and constraints associated with his decisions than the supremal. These assumptions result in a downward flow of authority and an upward flow of aggregated information about the state of the process and duration of activities.

3.0 A DECOMPOSITION ALGORITHM

We now review the approach proposed in [DAV88] in more detail. The schematic is given in Figure 2. The discussion is divided into two parts: planning and control.

3.1 PLANNING

The planning elements include the selection of evaluation criteria and scheduling rules, simulations, statistical analysis, and compromise analysis. Their combined responsibility is to find, in real-time, the best compromise scheduling rule given the

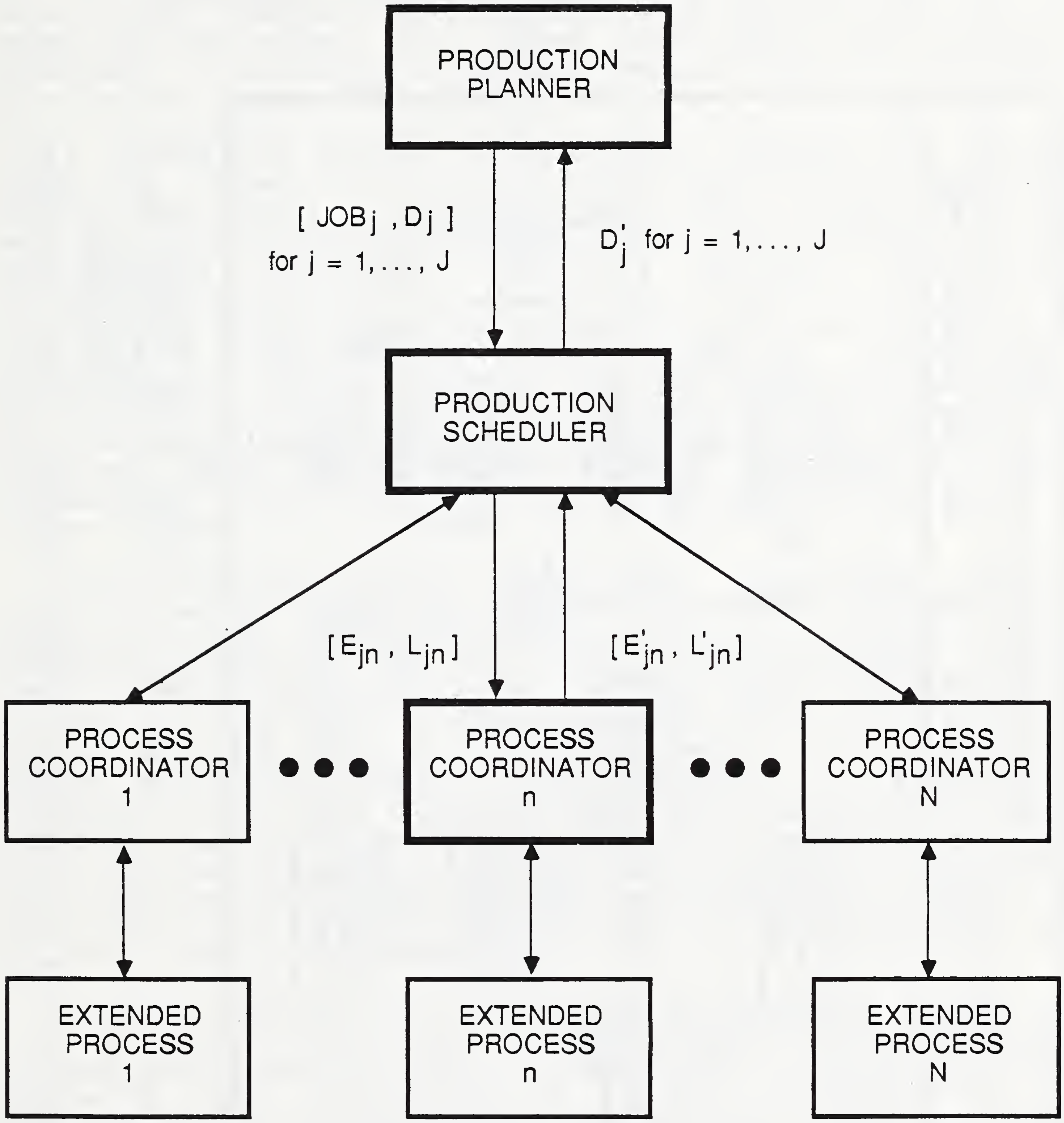

FIGURE 1 -- THE DECOMPOSITION STRATEGY FOR THE PRODUCTION SCHEDULING PROBLEM

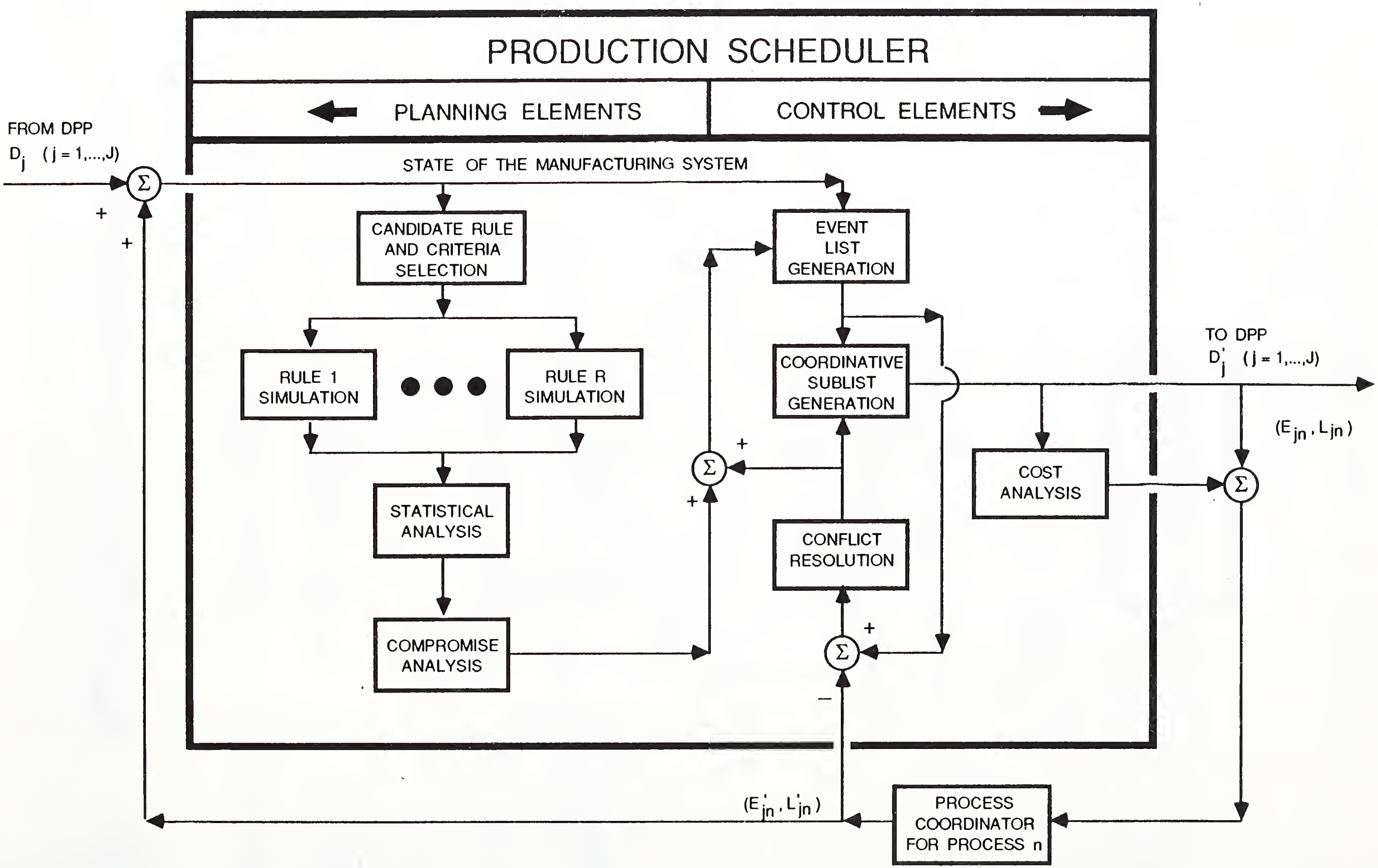

FIGURE 2 -- MODULAR SCHEMATIC FOR THE PRODUCTION SCHEDULER

current state of the system.

3.1.1 Evaluation Criteria. The evaluation criteria can be a combination of goals related to the performance of the entire manufacturing system, some or all of the processes, and some or all of the jobs. These criteria are often fixed, and set by management. However, they can also be a function of the current state of the system and changed each time a new schedule is required. Currently, acceptable methods are not available for choosing the appropriate criteria.

3.1.2 Scheduling Rules. The current research in production scheduling appears to be focusing upon three methodologies [RAM85,GRA82,JAC86]. The first, and perhaps classic, methodology employs mixed integer programming, including branch-and-bound and combinatorial techniques. In this approach, the production scheduling problem is typically posed with a single objective function to be optimized over a set of mathematical constraints expressed as linear equalities and inequalities. Recent work with multi-criteria integer programs has developed methodologies to generate the nondominated vertices. An alternate approach toward the consideration of multiple objectives would define the Nadir solutions which optimize each singular objective. From the nondominated solutions, a class of candidate schedules would emerge [DES86].

Heuristic/artificial intelligence approaches attempt to define a set of scheduling rules which can be a combination of preselected job release strategies, queuing strategies, material handling strategies, and any number of well-known dispatching rules. As with the evaluation criteria, these rules can be fixed or vary with the state of all or part of the system. These rules can be selected using an algorithm [OGR85], or an expert system [WYS86, PAR86,FOX84,LAW86]. These approaches use a combination of production rules and constraint-directed reasoning to generate feasible schedules. We note two things. First, those approaches which use only an expert system to define the rules and generate the schedules have severe computational problems. Second, these approaches make no claim of optimality or provide no measure of how close they are to optimality.

Closely related to the heuristic approaches is the recent trend to employ simulation, or a model-based approach, to generate a production schedule [MIL86,NOR86]. The simulation approach begins by first constructing an accurate representation of the shop floor as a discrete event system. Next rules are specified which govern the behavior of the major components of that system including routing rules, queuing rules and dispatching rules. Given the list of the current jobs to be scheduled, a single-pass simulation is then executed. Critical events are recorded from which the production schedule is specified. For both the heuristic and simulation approach, the schedule is highly dependent on the rules. That is, using an alternative sets of rules will generate different schedules.

3.1.3 Simulations. Discrete event simulation is the primary analysis tool used in the Davis/Jones approach. Concurrent, independent, real-time simulations are performed for each of the candidate scheduling rules. These simulations are integrated with shop floor data collection systems so that each trial can be initialized to the current "state" of the manufacturing system. Assuming that K simulation trials are run for each scheduling rule, an output table of start and finish times can be generated. Table 1 shows a sample output for the supremal. Here E^k_{jn} is the k^{th} simulated response for the start time of JOB_j on process P_n and L^k_{jn} is the k^{th} simulated response for the finish time of JOB_j on process P_n.

k	Simulation Results											
1	E^1_{11}	L^1_{11}	...	E^1_{1N}	L^1_{1N}	...	E^1_{J1}	L^1_{J1}	...	E^1_{JN}	L^1_{JN}
⋮	⋮	⋮		⋮	⋮		⋮	⋮		⋮	⋮
K	E^K_{11}	L^K_{11}	...	E^K_{1N}	L^K_{1N}	...	E^K_{J1}	L^K_{J1}	...	E^K_{JN}	L^K_{JN}

Table 1--Simulation Results for each Scheduling Rule

From this table we can evaluate each of the L objectives

$$f^l_k = f^l(E^k_{11}, \ldots, L^k_{JN}) \text{ for } k=1,\ldots K \text{ and } l=1,\ldots L. \quad (2)$$

As an example, the tardiness of a given JOB_j could be computed as

$$T^k_j = \max\{0, \max_n[L^k_{jn}]-D_j\} \text{ for } j=1,\ldots,J \text{ and } k=1,\ldots K. \quad (3)$$

3.1.4 Statistical Analysis. A statistical analysis is performed on the resulting function data. Specifically, for each objective $f^l(\ )$ and for each scheduling rule r an empirical probability density function is developed giving

$$\Pr[f^l_r(\ ) \leq z] = F^l_r(z) \quad (4)$$

The following statistics can then be computed

$$\bar{f}^l_r = \text{Mean or Ex } [f^l_r] \quad (5)$$

$$(s^l_r)^2 = \text{Sample Variance or Ex } [\bar{f}^l_r - f^l_r]^2 \quad (6)$$

$$m^l_r = \text{Minimum or } \min_k [f^l_k|r] \quad (7)$$

$$M^l_r = \text{Maximum or } \max_k [f^l_k|r] \quad (8)$$

Developing confidence interval intervals using these statistical samples is complicated by the fact that each simulated trial will be initiated from a different initial condition or state. This is a topic of active research.

3.1.5 Compromise Analysis. The next step is to determine the best compromise scheduling rule. First we determine the nondominated set of scheduling rules, denoted by R^*. The set R^* is defined such that $r \in R^*$ if for every $r' \in R$ there exists an $l \in \{1,\ldots,L\}$ such that

$$\bar{f}_r^l \geq \bar{f}_{r'}^l, \tag{9}$$

Given the nondominated rule set R^*, the next step is to determine the minimum and maximum for each objective function over R^* as

$$n^l = \min_{r \in R^*} (m_r^l) \tag{10}$$

$$M^l = \max_{r \in R^*} (M_r^l) \tag{11}$$

In this manner, the range of compromise for each objective f^l over R^* is defined as the interval $[n^l, M^l]$. Using the statistics for the nondominated strategies R^* and the associated range of compromise, the "best" compromise strategy $r^* \in R^*$ is then chosen. Methods for making this choice in a stochastic, multi-criteria, decision-making framework are being developed.

3.2 Control

Implementing the best compromise scheduling rule r^* is the primary function of the control elements: list generation, coordination, and conflict resolution.

3.2.1 List Generation. The first major control function generates an event list which the supremal will attempt to implement. Using the current state of the processes P_n $(n=1,\ldots,N)$ with the selected best compromise rule r^*, an additional single pass of the simulation is made to generate the following event list

$$\underline{E} = [E_{11}\ L_{11}, \ldots, E_{JN}\ L_{JN}] \tag{12}$$

from which we easily obtain the anticipated duration of process P_n on JOB_j, called $d_{jn} = L_{jn} - E_{jn}$. The event list $\underline{E}$ is then sorted into three sublists: 1) chronologically into a master schedule $\underline{T}$, 2) by JOB_j $(j=1,\ldots,J)$ into a scheduling list $\underline{J}$, and 3) by process into a process control list $\underline{C}$. The job scheduling list $\underline{J}$ provides the information necessary to track each JOB_j at any given time. The process scheduling control list $\underline{C}$ will permit the prediction of the status of a given process P_n at any given time.

3.2.2 Coordination. The event lists $\underline{T}$, $\underline{J}$ and $\underline{C}$ provide the data needed by the supremal to 1) coordinate the activities of the subordinate PC_n $(n=1,\ldots,N)$ and 2) provide feedback status on job completion. The coordination comes from all required precedence completion. The coordination comes from all required precedence relationships and the material handling considerations which are monitored continuously to ensure feasibility. Under the assumption of a cooperative hierarchy, we assume that each PC_n will actively attempt to fit the actual process duration, t_{jn}, within the time interval $[E_{jn}, L_{jn}]$. If changes are to be made to the list $\underline{T}$, then the feasibility of the entire list must be restored.

3.2.3 Conflict Resolution. The feedback information from the PC_n, (E'_{jn}, L'_{jn}) gives E'_{jn} as the actual initiation time and L'_{jn} as the expected completion time for JOB_j. This implies that the actual processing time is given by

$$t_{jn} = L'_{jn} - E'_{jn} \tag{13}$$

Whenever t_{jn} does not equal d_{jn}, the event list $\underline{T}$ is no longer valid since JOB_j will not be completed at the scheduled time. This can happen for two reasons. First, the process coordinator selects the initial t_{jn} based on the status of the process. Second, as the process evolves in time, the process coordinator may change the original t_{jn}. In either case, the actual duration, t_{jn}, can differ from the supremal's estimated duration, d_{jn}. When this happens, the Cost Analysis module is invoked to negotiate the determination of an acceptable process duration. This usually means that $\underline{T}$ must be updated to resolve this discrepancy. This requires the supremal to update its solution, in real-time, to restore feasibility.

First, we determine if the event list $\underline{T}$ can be updated without redoing the entire analysis. We are investigating the Perturbation Analysis technique described in [HO83,SUR84] and the Match-up approaches discussed in [BEA86,SAL88]. If restoration is possible, then we simply update the estimates for the expected durations, d_{jn}, and generate new lists $\underline{T}$, $\underline{J}$, and $\underline{C}$. If restoration is not possible, then a complete regeneration of $\underline{E}$ is required. This will happen if 1) the deviations between the planned and actual durations are large, 2) the compromise rule r^* is changed, and 3) a new Job_j is added to the list for scheduling.

3.2.4 Remarks. Normally we would anticipate that the continuous rescheduling would introduce nervousness or instability in the interaction of the supremal with the subordinate PC's. However, the (E_{jn}, L_{jn}) pair will be released to the PC_n on a need to know basis only. Once the Job_j has been released to PC_n, the PS can no longer enforce (E_{jn}, L_{jn}), but rather reacts to (E'_{jn}, L'_{jn}). The updating of the schedule is in anticipation of future system performance. Since the PS can not enforce a modification upon (E'_{jn}, L'_{jn}), the PC's are insulated from this potential problem.

4.0 AI FOR INFORMATION MODELING

We are beginning to implement the various modules described above. We envision using two major AI methodologies: information modeling and expert systems. Information modeling is necessary to handle the simulation input/output data and the various control lists. Although expert systems are not used as the primary analysis tool, they will be used them to aid in selecting performance criteria and candidate scheduling rules, and in analyzing some statistical output from the simulations. We discuss these issues in more detail in the following sections.

4.1 Input Data

The algorithm described in [DAV88] require simulations to be initialized to the current "state" of the system. At the supremal level the 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.

4.1.1 Processes. We assume that the shop floor contains N distinct processes 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).

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.

4.1.2 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.

The state should include some information on the type of buffer, the ID and location of each item in the buffer, and a linking mechanism to link items together that are logically connected.

4.1.3 Current Schedule. The current schedule at the supremal level 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

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

GANNT [BAK74] charts are the conventional method for representing all this information on one diagram.

4.1.4 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 being used to fabricate a given part can be derived easily from the GANNT chart.

4.1.5 New Jobs. Several pieces of information must be generated by the process planning department before a NEW_JOB can 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 3). Precedence relationships are enforced using the following convention: a given activity cannot begin until all activities ending at its start node have been completed.

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 4). 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.

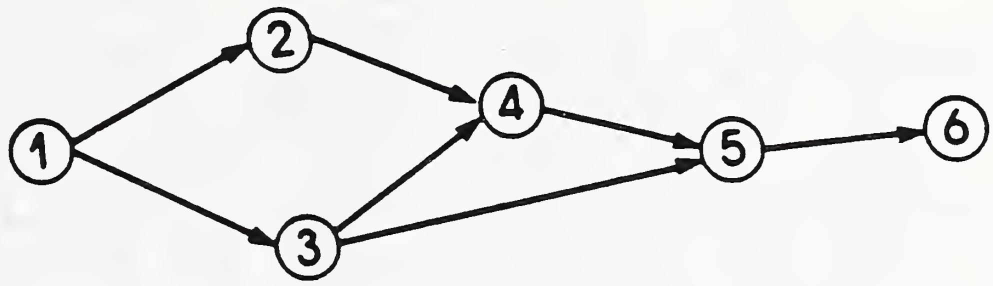

FIGURE 3. SAMPLE PERT CHART

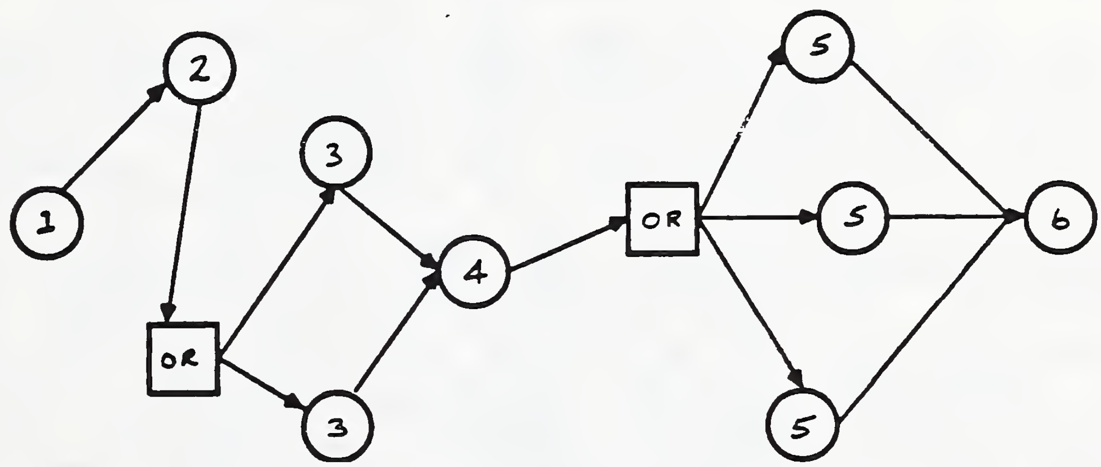

FIGURE 4. SAMPLE AND/OR GRAPH

4.2 Output Data

As we saw earlier, the output from each simulation trial at the supremal level is limited to E^k_{jn} and L^k_{jn}. Distinct values for the start and finish times are derived on each simulation trial for a given rule. Since each simulation trial requires a finite time to implement, a generated output will contain both events for which actual system response will have already been measured and events for which system response has not yet been measured. As the simulated output record in Table 1 ages, the number of events representing predicted system response must decrease as more events are realized by the actual system. One immediate consequence is the essential requirement that the simulated output records must be updated as each newly measured event is recorded. In addition, all of the performance measures and their associated statistics must also be updated.

4.3 Control Lists

As noted above, the control lists T,J, and C form the major outputs from the scheduler. They are, in fact, the master, job, and process schedules. They also form the principal information used in the negotiation and conflict resolution that takes place between the various levels in the control hierarchy.

4.4 Using AI Methodologies

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 representations, including semantic networks and object-oriented programming, 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 selected structures before they can be used to initialize the R concurrent simulations.

We have also described the output and related statistical calculations generated by those simulations. Here again, we foresee the need to develop efficient data structures for storing and updating this information. In addition, computer based procedures must be designed to filter the statistical results and bring the salient features of the data to the decision-maker. The automated procedure must then act to determine the best compromise rule. In this function, it is essential that the decision-maker be able to query the algorithm for the basis of its current selection. Finally, the algorithm must be capable of recognizing instances which are beyond its logical or programmed capacity and request human intervention. In these instances, the algorithm should employ the human's response to improve its knowledge base, increasing its capacity to handle similar future instances.

Finally, we have described the information that is contained in the control lists. It is extremely important to develop structures that capture the relationships between the entities in these lists and which allow one to quickly search through each list to determine the impact of any delay or other anticipated changes.

5. RULE-BASED METHODOLOGIES

Although we do not plan to use a large expert system to generate schedules, we do plan to use that paradigm in executing several functions within our algorithm.

5.1 Planning Functions

We expect to use rule-based systems to aid in determining both the evaluation criteria and candidate scheduling rules. In addition, we will develop some rule-based procedures for analyzing the statistical data generated from the simulations.

This analysis will help the decision maker understand the complex relationships and correlations that may exist among the various performance measures. It will also provide a means to filter that data and aid in the selection of the best compromise rule. Finally, it will provide the opportunity to learn from past decisions, thereby increasing the effectiveness of the scheduler.

5.2 Control Functions

As described above, the scheduler at the supremal level will attempt to implement the selected compromise scheduling rule r^*. The event list generator will first generate a tentative list of future events given the current system status while using preselected processing durations, d_{jn}. The process coordinator will then determine his "best" schedule based on his own determination of the actual process duration, t_{jn}. The cost analysis function is invoked whenever there is a difference between the estimated and actual process durations. We expect to use some type of rule-based system to carry out this negotiation. We note that early all the current research in real-time production scheduling has ignored this issue by assuming that processing durations are fixed. We believe that if time-varying processes are to be considered and hierarchical approaches used for production scheduling, that this issue cannot be ignored.

We also plan to use some type of rule-based system to determine the appropriate course of action to take each time a conflict occurs. That means that a decision must be made about the current schedule. Today's ad hoc approaches are simply inadequate. We need a fast method to determine the impact of any problem on the current schedule, determine if a quick fix is possible and the impact of that "fix", or determine that a completely new schedule must be generated.

5.0 CONCLUSIONS

This paper has focused on the potential for applying AI techniques to the functions described in [DAV88] algorithm for doing real-time production scheduling. Due to the embryonic nature of the algorithm itself, specific AI algorithms can not yet be prescribed. Nevertheless, the nature of the problem which is being addressed provides attributes which point to the appropriateness of considering AI methodologies.

Before concluding this paper, it should be noted that the problems cited herein are not limited to real-time production scheduling. Several other problem domains could have been used. We can expect a more generalized algorithm for real-time decision-making to emerge and that additional problem areas will be defined for which AI will provide a beneficial solution methodology.

REFERENCES

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

[BEA86] Bean, J. and Birge, J., "Match-up Scheduling", **Proceedings of the Symposium on Real-Time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards, Gaithersburg, Maryland, January 21-22, 1986, pp. 197-212.

[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.

[FOX84] Fox, M. S. and Smith, S. F., "ISIS -- A Knowledge-Based System for Factory Scheduling," **Expert Systems**, Vol. 1, No. 1, 1984, pp. 25-49.

[GRA82] Graves, S, C., "A Review of Production Scheduling," **Operations Research**, Vol. 29, 1982, pp. 646-675.

[HO 83] Ho, Y. C., **SPEEDS: A New Technique for the Analysis and Optimization of Queuing Networks**, Technical Report No. 675, Division of Applied Sciences, Harvard University, Cambridge, Massachusetts, 1983.

[JAC86] Jackson, R. H. and Jones, A. T., (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.

[LAW86] Lawrence, S. R. and Morton, T. E., "PATRIARCH: Hierarchical Production Scheduling", **Proceedings of the Symposium on Real-time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards Special Publication 724, Gaithersburg, Maryland, 87-98, 1986.

[MCP86] McPherson, R. F. and White, K. P., "A Management Control Approach to the Manufacturing Planning and Scheduling Problem", **Proceedings of the Symposium on Real-time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards Special

Publication 724, Gaithersburg, Maryland, 145-166, 1986.

[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.

[OGR85] O'Grady, P. and Harrison, C., "A General Search Sequencing Rule for Job-shop Sequencing," **International Journal of Production Research**, Vol. 23, No. 5, 1985.

[PAR86] Parrello, B. D. and Kabat, W. C., "Job-Shop Scheduling Using Automated Reasoning: A Case Study of the Car-Sequencing Problem," **Journal of Automated Reasoning**, Vol. 2, 1986, pp. 1-42.

[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, U.S. Department of Commerce, National Bureau of Standards, Gaithersburg, Maryland, 1985.

[ROD86] Rodammer, F. A. and White, K. P., "Design Requirements for a Real-time Production Scheduling Decision Aid", **Proceedings of the Symposium on Real-time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards Special Publication 724, Gaithersburg, Maryland, 401-422, 1986.

[SAL88] Saleh, A., "Real-time Control of a Flexible Manufacturing Cell", unpublished Ph.D dissertation, Lehigh University, 1988.

[SUR84] Suri, R. and Dille, J. W., "On-Line Optimization of the FMS Using Perturbation Analysis," **Proc. of Flexible Manufacturing Systems Conference**, First ORSA/TIMS Special Interest Conference, Ann Arbor, Michigan, Aug. 1984.

[WYS86] Wysk, R., Wu, S. and Yang, N., "A Multi-pass Expert System (MPECS) for Job-shop Scheduling", **Proceedings of the Symposium on Real-Time Optimization in Automated Manufacturing Facilities**, National Bureau of Standards, Gaithersburg, Maryland, January 21-22, 1986, pp. 251-278.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 88-3891	2. Performing Organ. Report No.	3. Publication Date FEBRUARY 1989

4. TITLE AND SUBTITLE

Artificial Intelligence Techniques in Real-Time 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 Institute of Standards & Technology
Bldg. 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)*

This paper addresses the real-time production scheduling problem as a special case of a much larger class of real-time decision-making/control problems. The paper first reviews the definition of the scheduling problem, and then reviews a earlier algorithm proposed by the authors to address this problem. It then concentrates on the possible application of various AI techniques to many of the functions that make up that algorithm.

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

artificial intelligence, automation, expert systems, information models, production scheduling, real-time decision-making, simulation

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

19

15. Price

$$9.95

USCOMM-DC 6043-P80