A Review of Machine Learning in Scheduling

Haldun Aytug, Siddhartha Bhattacharyya, Gary J. Koehler, and Jane L. Snowdon

Abstract—**This paper has two primary purposes: to motivate the need for machine learning in scheduling systems and to survey work on machine learning in scheduling. In order to motivate the need for machine learning in scheduling, we briefly motivate the need for systems employing artificial intelligence methods for scheduling. This leads to a need for incorporating adaptive methods—learning.**

I. Introduction

IN THIS PAPER we summarize work done on machine learning in scheduling. The backdrop for this activity is artificial intelligence (AI) approaches to dynamic scheduling. This has been a very active area of research since the early 1980's.

According to Mayer, Phillips, and Young [32] over 98% of the time work-in-process is involved in non-value-added activities. Proper scheduling of activities is an important tool for improving this situation.

However, scheduling is a very difficult task. Manufacturing environments have high levels of uncertainty, processes have detailed and specific requirements, and management objectives are varied, dynamic and often conflicting [54].

Over time there have been a large number of approaches studied to solve scheduling problems. Had these approaches been successful, the current interest in artificially intelligence-based methods for scheduling would probably not be as intense.

In Section II we discuss the scheduling problem and the context in which scheduling is typically encountered.

In Section III we review various traditional approaches to scheduling and offer an assessment of these methods.

One increasingly popular approach to scheduling uses artificial intelligence methodology. AI approaches to scheduling are discussed and assessed in Section IV.

An exciting new trend in AI approaches to scheduling is to add a learning capability. The purpose of this paper is to summarize the work in this area. In Section V we motivate the need for a learning capability in scheduling systems. In Section VI we present an overview of various machine learning approaches used in scheduling.

Finally, in Section VII we suggest areas for future research.

Manuscript received November 25, 1992. Review of this manuscript was processed by Editor S. S. Erenguc. This work was supported by Grant N-UN-224 from IBM Corp.

H. Aytug, S. Bhattacharyya, and G. J. Koehler are with Decision and Information Sciences, University of Florida, Gainesville, FL 32611 USA.

J. L. Snowden is with IBM Corporation, Boca Raton, FL 33429 USA.

IEEE Log Number 9400877.

II. The Environment and Scheduling

Before summarizing traditional approaches to scheduling, it is worthwhile to describe our application environment and to discuss the various roles of scheduling.

A. The Environment

The complexity of the processing environment can be as simple as requiring one processing step on one machine or facility. A slight generalization would have multiple parallel processing facilities. A flow shop environment is a more complex, multi-stage configuration. However, all tasks are processed on the same facilities with the same precedence requirements. Finally, a job shop environment permits alternative routings.

The tasks performed at a facility include machining, fabrication, assembly, and inspection. The tasks may be carried out by humans, fixed machines, or Numerical Control (NC) machines.

B. Scheduling

There are three views of planning and scheduling, depending on the time horizon. The highest level is production planning which usually covers details at a multi-week level. The output of production planning is used for release scheduling. Finally, actual item movement planning occurs in real-time. (See [3], [24] for related discussions).

The first two planning levels "predict" the production plan and schedule. The third level is more involved with "reacting" to the current local situation and is often called reactive scheduling.

It is the area of reactive scheduling that is most difficult to analyze and provide meaningful automated help. A large number of factors may cause disruptions in a production environment necessitating a new schedule. Machines break down, management submits "hot-listed" items requiring immediate attention, workers call in sick, critical materials fail quality screens, materials are out of stock, due dates change, etc.

Schedules are obsoleted by such disruptions—requiring a new schedule. Foremen and dispatchers are accustomed to dealing with such disruptions. However, their decisions are often crisis-oriented with little attention given to the bigger picture and impact therein.

If a computer-aided method is used for scheduling, it must be run often and fast. Both requirements often dictate a simplistic model that trades off solution time for realism and solution quality. (See [58] for a good discussion of this issue).

In the next section we look more closely at the traditional methods of supporting scheduling decisions.

III. Approaches to Scheduling

Most traditional approaches to solving scheduling problems use either simulation models, analytical models, heuristic approaches or combinations of these methods.

Simulation is primarily used to assess schedules and scheduling heuristics and is most useful for production planning and release scheduling. It is also a good tool to explore "what-if" scenarios at an experimental level. Seldom is simulation used for real-time scheduling. (See [43] for a good discussion of simulation in scheduling).

Analytical models include mathematical programming models, stochastic models, and control theory approaches. There are many algorithmic models developed over the last thirty years, mainly by the Operations Research (OR) community. Several review papers are available, e.g., [18].

Analytical approaches are plagued by the complexity of the scheduling problem. These problems involve stochastic elements in a dynamic environment with complex, multiple objectives and discrete variables.

Even the most simplified models for scheduling are NP hard problems. Fox [14] makes the NP-hard point more tangible: "85 orders moving through eight operations without alternatives, with a single machine substitution for each and no machine idle time has over 10^{880} possible schedules".

The implication is obvious—realistically sized problems cannot be solved optimally [59]. Furthermore, the static nature of most OR models ignores the dynamic nature of most real-time scheduling environments [56]. The OR scheduling problem is a classic. Yet, although textbooks promote these analytical models, few real-life applications are known [8].

Largely due to the failure of classic OR approaches to scheduling, a wide body of heuristic approaches have been investigated. Most work with heuristics has focused on dispatching rules. For example, the EDD dispatching rule says to pick the next job to be processed based on earliest due date. Blackstone, Phillips, and Hogg [6] provide a good survey of dispatching rules.

Some dispatching rules are known to be optimal in certain situations. For example, EDD minimizes maximum lateness. Others have been shown to perform well in general, but not always. Often, mixtures of dispatching rules out-perform single rules.

The main problem with using such heuristics is that one seldom knows whether a given decision will be good or not [4]. Thesen and Lei [59] showed that human interaction with automated heuristic methods often offers improved performance.

So it seems that systems that use human insights with heuristics have some possible advantages. AI offers a means to combine various types of knowledge in a manner that can be used in the scheduling environment. This is explored in the next section.

IV. Artificial Intelligence Methods for Scheduling

Over the last ten to fifteen years, considerable effort has been devoted to applying AI methods to scheduling. The first such application can be traced back to Gere [17].

AI methods could be considered as sophisticated heuristics. However, the emphasis of AI methods is to solve a problem using methods that "appear" intelligent.

Knowledge about a problem domain is encoded in some structure (rules, logic, frames, scripts, semantic nets, etc.) and manipulated to solve a problem. Manipulation is performed by an inference mechanism. The inference mechanism employs a search according to one of many control strategies.

The domain knowledge of interest to a practitioner is knowledge about scheduling in his specific production environment [39]. AI methods are flexible and rich enough to capture idiosyncratic knowledge while providing an automated way to use this knowledge. AI theory and methodology also provides a way to study how people do scheduling [32].

When an AI scheduling system is built, knowledge about a domain becomes explicit. This enables one to study that knowledge, to critique the knowledge, to use it in training, and to preserve it over time 12].

AI methods are often called knowledge-based systems. Expert system (ES) used for scheduling represent a special case of knowledge-based scheduling systems. Expert systems encompass the knowledge of a human expert. ES's are more simplified versions of more complex AI methods. As a result, ES's were among the early AI approaches used in scheduling. The next section investigates ES's used in scheduling.

A. Expert Systems in Scheduling

"The primary goal of the ES applications is to consistently duplicate the results of a human expert" [32].

Expert Systems are constructed by first acquiring knowledge from a human expert and then codifying this knowledge in some knowledge representation scheme (such as rules, frames, etc.) This knowledge acquisition usually involves time consuming interviews with the human expert.

Once the knowledge has been verified and validated, it can be used to make decisions similar to those that the human expert would make.

Reinschmidt, Slater, and Finn [42] provided a list of benefits that an ES might provide in scheduling. Basically, the Scheduling ES could make quick decisions on actual or "what-if" cases and do so in a way that captures the idiosyncratic nature of the specific production environment.

In the next section we provide an assessment of ES's used for scheduling.

B. An Assessment of Expert Systems

Miller, Lufg, and Walker [34] reported on an ES used for scheduling at a Westinghouse plant which achieved a $10 million annual payback.

Savell, Perez, and Koh [44] explain the success of some scheduling ES's as resulting from the experience gained by the scheduling experts over time. However, as the production environment complexity increased, human experts seemed to be less effective in evaluating alternatives.

Many researchers are more skeptical that true human experts exist in scheduling. Fox [14] believes that expert system approaches are not suitable for scheduling because the complexity of most real-life environments is beyond the cognitive capabilities of most schedulers and that most environments are so dynamic that knowledge becomes obsolete too fast anyway.

Fox [14] saw that over 80% of a scheduler's time was spent in trying to determine the actual constraints governing the situation at a point in time rather than in deciding on a schedule. This supports an observation by Steffen and Greene [57] that, while schedulers are often very experienced and have insights into production problems, they seldom know how to produce near-optimal schedules.

Even if human experts exist, Kanet and Adelsberger [23] argue that an ES merely automates their decisions—good or bad. At the least, an ES could never perform better than the human expert from whom the knowledge was acquired.

Much of the case against expert systems argues against the existence of bona fide experts. Do they exist, or not? Steffen [55] gives many reasons against their existence or use. Human schedulers employ greedy strategies because of the complexity of the production environment. They are prone to react to management dictates and other disturbances in non-optimal, myopic ways. Today's production environment is so dynamic that scheduling knowledge is a "moving target".

Not everyone is convinced that human expert schedulers don't exist. Mckay, Buzacott, and Safayeni [33] insists that real human expert schedulers exist and that they should be used in the development of knowledge-based systems for scheduling.

In any case, more general AI methods have also been applied to scheduling problems. We discuss these approaches in the next subsection.

C. Other AI Methods

Drawing from Steffen [55], Kusiak and Villa [27], and Fox [14], we note a large number of AI approaches in scheduling. AI search can be random, blind or heuristic:

Random Search: A space of schedules is randomly searched to find a satisfactory schedule. Genetic algorithms fall into this approach when no information of the environment is used to shape the algorithm.

Blind Search: Methods such as depth-first search, breadth-first search, or combinations thereof, are used to exhaustively explore the space of schedules.

Heuristic Search: Here the large space of schedules is searched with the aid of heuristics that reflect knowledge about the problem structure.

A common heuristic search uses knowledge about the constraints of a scheduling problem to guide and limit the search. These methods are called constraint-directed, constraint-guided, or constraint-based methods.

The problem space can be divided in a way to reduce the overall search. Common methods include:

Hierarchical search: Here the scheduling problem is structured as a sequence of progressively more detailed and less aggregated problems. This hierarchical approach provides a further limitation to the search.

Distributed search: In distributed search, several different computing units are involved in solving the scheduling problem. Coordination between different units is often handled by a bidding or auction-based system. See [11], [53], [36], [45], [46].

We will not discuss the details of these various approaches but do offer a short assessment in the next section.

D. An Assessment of Other AI Approaches

Overall, knowledge-based approaches seem to do a better job of capturing the idiosyncrasies and nuances of an environment than, say, OR models. However, AI approaches are not without problems.

Steffen and Greene [56] found the use of constraint-directed search useful in capturing expert knowledge, reducing the search and in handling the trade-offs between conflicting objectives. On the other hand, Fox [14] found that the constraints did not yield consistently better search paths.

There have been criticisms against the use of AI in general. Rightly so, the AI community has a historical record of overhyping their abilities. Parnas [37] likens the use of AI to that of having a machine wash clothes by having two robotic arms using a washboard.

A serious limitation of AI methods used in scheduling was their inability to adapt. This observation led to including learning components in AI systems for scheduling.

V. The Need for Machine Learning in Scheduling

Ebner and Wollmann [13] predicted several trends in computer-based scheduling. A key observation is that knowledge-based systems will have to be adaptive (that is, they will have to learn).

Steffen [55] forecasts that learning will become the third fundamental AI method applied to scheduling problems (search and knowledge representation being the other two).

Machine learning is the name of the class of automated methods of knowledge acquisition. Imbuing a system with learning gives it the ability to adapt.

One compelling reason for automated knowledge-acquisition stems from the difficulty of obtaining knowledge from experts. This has been realized by many AI researchers, including by Yih [61], [62] in the scheduling area.

The cases where learning systems have been built as part of an AI system for scheduling are relatively few in number. Fox and Smith, [15] hinted at the need to learn in their early work on AI systems for scheduling. Some authors take the capability of learning as an integral part of an expert system approach. For example, Bitran and Papageorge [5] argue that the expert system should be able to learn about the particular environment in which they operate.

Several researchers have recognized the need for learning in scheduling systems (e.g., [28]). Fox and Smith [16] place learning at an important level. A system should be able to correct its misconceptions and improve its performance based on experience. This is learning.

May, *et al.* [31] insist that we don't know how to incorporate learning into our systems. However, there have been several attempts at incorporating learning into knowledge-based systems. In the next section we provide a survey of the more prominent efforts.

The methods used to learn include a large number of different approaches. Several researchers learn from experiments on a simulation model. See, for example, [1], [2], who use statistical inference; [38], [47]–[52], [60], who use inductive learning methods; and [9], who learn with neural nets.

One type of learning, called cased-based or exemplar-based reasoning, uses past situations to infer actions for new situations [30]. Koton [25], [26] notes that these approaches have merit when similar problems recur, when the domain is stable under perturbations and when interactions are limited.

VI. Survey of Machine Learning in Scheduling

In this section we present and summarize a number of papers where machine learning of scheduling is studied.

A. Rote Learning

Blessing and Watford [7] describe a FMS scheduling system, INFMSS, where the knowledge is represented by frames. They structure the knowledge base in three dimensions namely:

- FMS description on one axis,
- part mix on another, and
- schedule proposed on the third axis.

If any point in that three dimensional space is already defined, the expert system uses that point as the best schedule. Otherwise a search for a few scheduling strategies for the given part mix and FMS description is carried out, and evaluated by a simulation system. The schedule that gives the best result is chosen as the answer. Those points are saved for further reference.

This mechanism is referred to as rote learning. Even though the system saves old decisions that gave good results, it has no means of generalizing them and we suspect that in such a complex environment the chances of observing the same state is very low.

Thesen and Lei [59] use a simulation based approach to construct a rule base for scheduling the operations of a single track material handling robot in an automated electroplating line. The performance of several dispatching rules under varied system conditions is obtained through simulation, and manual analysis of results indicates the best rule to use in different situations. The system state is continually monitored, and the knowledge base is used to dynamically change the decision rules depending on the current conditions.

Though the expert scheduling system is noted to have surpassed the performance of any single dispatching heuristic, this approach is limited by the initially chosen set of dispatching rules and system states considered in the simulation experiments. The authors note the need for further research to develop an automated approach for knowledge acquisition, which can yield better scheduling methods.

B. Induction

In view of the difficulty that human experts have in verbalizing their own decision processes, Yih [61], [62] develops a method for knowledge extraction from a trace of an expert's decision sequences. This method, called Trace Driven Knowledge Acquisition, has been applied to a circuit board production line consisting of a series of chemical process tanks. A material handling robot moves jobs through the tanks, and adherence to accurate timing constraints between different chemical treatments is crucial for maintaining defect free jobs.

The system operates in three steps: (1) data collection, where a trace of decisions and corresponding system states is obtained from the simulated system; (2) data analysis, where the traces are examined to determine the scheduling rules used; and (3) rule evaluation, where the performance of these obtained rules is compared against the expert's performance. If found inadequate, the rules are fed back to the second step.

A set of variables defines the state space, which is considered divided into a number of classes. For each class, a simple decision rule determines the next course of action. The complete set of decision alternatives is provided in terms of decision rules that specify the possible actions given any state.

A trace is a sequence of $\{$existing state S_i, action taken $A_j\}$ tuples. During data analysis, the class that state S_i belongs to is obtained. In the rule evaluation stage, all decision rules in each class are scored according to the number of matches with the decision maker's actions. If the highest score is above some predefined threshold, the corresponding decision rule is assigned for that class. When the scores do nor meet the minimum requirements, that class is further sub-divided, and the process repeated.

Experiments were conducted with student subjects as human schedulers, and the best amongst them taken as experts. The system-obtained rules were noted to out-perform the human experts.

In a later paper Yih, [63], recognizing that true experts seldom exist in real-life production environments, utilizes a semi-Markov modeling of the decision traces, together with an optimization procedure, to enrich the knowledge quality. Here the data analysis phase is replaced by an improvement procedure, where transition probabilities are estimated from the traces, and then an optimal decision policy algorithmically derived. A third rule formation stage then determines decision rules from the optimal policy, and the decision rules for each class of the state space are obtained as in the rule evaluation phase.

Rule formation is, however, noted to be a bottleneck in this process, requiring substantial human intervention.

Piramuthu *et al.* [38] and Shaw, Raman, and Park [49] present an inductive learning approach from training examples generated through simulation. The system learns the appropriate dispatching rule to use under different factory floor conditions. They characterize an FMS environment by a set of attributes such as the number of machines, buffer size, machine contention factor, etc. The system is simulated under varying values for these attributes. This attribute valued "pattern" together with the dispatching rule resulting in the best performance, is taken as an example to be input to the learner.

Quinlan's [40] ID3 inductive learning algorithm is used and the resulting knowledge is represented by a decision tree. A critic mechanism monitors the performance of the learned rules, and if found undesirable, moves to refine the rules by suggesting further simulation to generate more training examples. A set of meta-rules are utilized to recommend attribute values under which to run additional simulations. Experiments with the learned rules indicate improved performance over that obtained by applying any single dispatching rule.

Lamatsch *et al.*, [28] describe an expert system named SCHEDULE that uses reduction digraphs to match the given problem as an instance of a scheduling problem solvable by the stored heuristics. All the heuristics are polynomial time algorithms that can produce optimal or near optimal results published in various journals. If no match is found, the system first tries to find a problem that is close to the original problem with respect to a metric they define. If that fails, a relaxation of the problem is solved.

The system has a learning capability by adaptively adjusting the distances between two nodes of the graph. As the user prefers a problem to solve as a substitute for the original problem, the arc weight is reduced. By this way the system learns which problem is a better approximation for a certain type of problem. Note that each node is a type of problem that can be reduced to each other if there is a path between them.

Case Based Reasoning (CBR) is another machine learning paradigm being considered for application to scheduling problems. CBR seeks to exploit experience gained from past similar problem-solving cases. Mark [30] outlines a case based system for autoclave management.

The problem here consists of loading parts of varying shapes and sizes into a large convection oven, where the process is sensitive to the part layout, and of scheduling the flow of parts through the autoclave so as to meet global requirements. Jobs arrive with different priority rankings. The manual procedure is based on the past experience of an expert operator. The proposed Clavier system seeks to obtain good autoclave loading patterns through matching the current set of jobs and priorities against a case library of previously successful layouts.

The larger scheduling problem requires the identification of salient features of past schedules, generalizing these, and a mechanism for determining which stored case best matches and is useful in the current problem context. Given the large number of interacting constraints inherent in scheduling, existing case indexing schemes are noted to be inadequate for building the case base and subsequent retrieval, and research avenues are suggested.

Another case-based reasoner for scheduling large scale airlift operations was proposed by Koton [25], [26].

A neural network approach to learning appropriate criteria weights for multiple criteria decision making for job shop scheduling is presented in Chryssolouris *et al.* [9]. A three-layer network using back-propagation algorithm is applied to learn weights to be assigned to local work-center scheduling criteria so as to meet a set of global performance objectives.

A system simulation first provides the performance measure values for different operational policies (criteria weights) and varying workload levels at the work-centers, and this serves as the training inputs to the neural net. The output criteria-weights reflect the relative importance of the different criteria for attaining the specified performance levels.

Three performance objectives and workloads at four work-centers considered require seven input neurons. There are three output nodes each corresponding to a local criteria. Ten nodes are used in the hidden layer. After training, the network is used by specifying the anticipated workload levels and the desired performance measures, and necessary criteria weights are obtained at the output nodes.

Davis [10] outlines one of the earliest Genetic Algorithm (GA) approaches for the scheduling problem. A job shop setting is considered, and a time-dependent preference list representation used. A preference list consists of a starting time at which the list goes into effect, and a sequence in which a machine prefers to process different job types. Idle and Wait operations may be included in the sequence to indicate the machine's preference to remain idle, or, if none of the specified job types are present, to wait for the next job. This representation of schedules ensures that manipulation of the lists by GA search operators yields legal schedules. Non-conventional GA operators are modeled after manual scheduling procedures that have been found to work well in the considered domain.

Hilliard *et al.* [19]–[21] present a series of experiments with a classifier system approach (using GA search) for learning job sequencing heuristics. The first two papers consider a single machine scheduling task with an objective of minimizing total lateness.

The optimal strategy for this problem is to order the jobs by increasing processing time. In initial experiments, environmental detector message lists are comprised of binary coded job run-times and run-time rankings, and effectors moved jobs to specific queue locations. Different conflict resolution and credit assignment strategies are examined in [20].

The classifier system with this binary representation was found deficient in learning optimal rules. In [19], environmental detectors are modeled using high-level predicates that compare attribute values of jobs. Job processing time values, due dates, and current queue positions were considered.

Each predicate represents a heuristic strategy. For example, Processing-time (job i) < Processing-time (job j) corresponds to a shorter run-time first strategy. A bubble sort routine is used to compare all job pairs, and tests on the defined predicates provide the condition part of a classifier. An action either exchanges job positions or leaves the queue unchanged. This system was able to obtain the optimal rules.

The third set of experiments considered the objective of minimizing weighted tardiness, known to be an NP-complete problem. The classifier system learned rules which were found to perform better than the application of any of the single heuristics alone.

Husbands, Mill, and Warrington [22] propose a GA based scheme where separate populations simultaneously evolve production plans for different components to be manufactured. A population member is represented as

$$op_1\, m_1\, s_1\, op_2\, m_2\, s_2\, G\, op_3\, m_3\, s_3\, op_4\, m_4\, s_4\, op_5\, m_5\, s_5\, G\, \ldots$$

where each $\langle op_i\, m_i\, s_i \rangle$ set represents an operation op_i to be performed on machine m_i with setup s_i. Interdependent operations are separated into groups demarcated by G. Crossover is permitted only at the group boundaries, and mutation is also designed to ensure valid plans.

In the first phase of learning, different populations evolve independently, based on the machining costs involved in the plans embodied by the population members. Different populations are ranked according to the summed costs of their members. In the second phase, the plans of equally ranked populations are simulated simultaneously. Interactions amongst the plans for different components are resolved by an arbitrator that decides which population's member to give precedence to. The arbitrator is also modeled as another population that evolves through genetic learning.

As can be seen, there have been many types of learning studied in the scheduling domain. In the next section we look at possible areas of future research.

VII. Future Research Issues

Many types of learning have been explored for scheduling, including rote learning, inductive learning (ID3), neural network learning, case-based learning, classifier systems, and others. While each particular methodology offers positive and negative features, what is most important is that these learning systems generally seem to improve a scheduling system's performance.

Future research needs to focus on several important dimensions of the scheduling environment. One dimension is to provide means to learn how to trade off management goals, which are often inconsistent and time-varying. Another is to learn about the uncertainties that exists in a given production environment.

An important issue for future research is validation. With few exceptions, very little work has been undertaken to validate the results that machine learning has produced.

Most learning systems in scheduling have used straightforward, simplistic forms of knowledge representation. A richer representation is needed to adequately handle the complexities and nuances of production environments.

This, in turn, dictates more involved learning paradigms and inference strategies.

If the typical production environment is as dynamic as many believe them to be, the learning methods need to be dynamic themselves. Thus, fixed experiments for inductive learning may not be appropriate. Inherently dynamic learning methods are needed.

Ideally, a scheduling system should be able to learn and deploy that knowledge in real-time. Learning methods that require extensive experimentation are inappropriate under such situations. Future research is needed to understand the limitations of and methods for real-time learning.

References

[1] T. Adachi, J. J. Talavage, and C. L. Moodie, "A rule based control method for a multi-loop production system," *Art. Int. in Eng.*, vol. 4, no. 3, pp. 115–125, 1989.

[2] T. Adachi, C. L. Moodie, and J. J. Talavage, "A pattern-recognition-based method for controlling a multi-loop production system," *Int. J. Prod. Res.*, vol. 26, no. 12, pp. 1943–1957, 1988.

[3] J. C. Ammons, T. Govindaraj, and C. M. Mitchell, "A supervisory control paradigm for real-time control of flexible manufacturing systems," *Annals of Operations Res.*, vol. 15, pp. 313–335, 1988.

[4] G. Bel, E. Bensana, D. Dubois, J. Erschler, and P. Esquirol, "A knowledge-based approach to industrial job-shop scheduling," in *Knowledge-Based Systems in Manufacturing*, A. Kusiak, ed. Taylor & Francis, 1989, pp. 207–246.

[5] G. Bitran and T. Papageorge, "Integration of manufacturing policy and corporate strategy with the aid of expert systems," in *Intell. Manufact., Proc. 1st Int. Conf. Expert Systems and the Leading Edge in Production Planning and Control*, 1988, pp. 13–43.

[6] J. H. Blackstone, D. T. Phillips, and G. L. Hogg, "The state-of-the-art survey of dispatching rules for manufacturing job shop operations," *Int. J. Prod. Res.*, vol. 20, no. 1, pp. 27–45, 1982.

[7] J. A. Blessing and B. A. Watford, "INFMSS, an intelligent fms scheduling system," in *World Productivity Forum and 1987 Ann. Int. Industrial Engineering Conf. Proc.*, pp. 82–88.

[8] G. Buxey, "Production scheduling: Practice and theory," *Eur. J. Operational Res.*, vol. 39, pp. 17–31, 1989.

[9] R. Davis and R. Smith, "Negotiation as a metaphor for distributed problem solving," *Artificial Intelligence*, vol. 20, pp. 63–109, 1983.

[10] S. De, "A knowledge-based approach to scheduling in an FMS," *Annals of Operations Res.*, vol. 12, pp. 109–134, 1988.

[11] M. L. Ebner and T. E. Vollmann, "Manufacturing systems for the 1990's," in *Intelligent Manufacturing*, M. Oliff, ed. Benjamin/Cummings, 1988, pp. 317–335.

[12] M. S. Fox, "Constraint-guided scheduling—A short history of research at CMU," *Comp. in Industry*, vol. 14, pp. 79–88, 1990.

[13] M. S. Fox and K. P. Sycara, "Overview of CORTES: A constraint based approach to production planning, scheduling and control," in *Proc. 4th Int. Conf. Expert Systems in Production and Operations Manage.*, May 1990.

[14] M. S. Fox and S. F. Smith, "The role of intelligent reactive processing in production management," *Proc. CAM-I's 13th Ann. Meeting and Technical Conf.*, Clearwater Beach, FL, Nov. 13–15, 1984.

[15] M. S. Fox and S. F. Smith, "ISIS—A knowledge-based system for factory scheduling," *Expert Systems*, vol. 1, no. 1, pp. 25–44, 1984.

[16] W. S. Gere Jr., "A Heuristic Approach to Job Shop Scheduling," Ph.D. dissertaion, Graduate School of Industrial Administration, Carnegie Institute of Technology, Sept. 1962.

[17] S. C. Graves, "A review of production scheduling," *Operations Res.*, no. 2914, pp. 646–675, Apr. 1981.

[18] J. J. Kanet and H. H. Adelsberger, "Expert systems in production scheduling," *Eur. J. Operational Res.*, vol. 29, pp. 51–59, 1987.

[19] F. K. Kemp, "Artificially intelligent tools for manufacturing process planners," in Intelligent Manufacturing, *Proc. 1st Int. Conf. Expert Systems and the Leading Edge in Production Planning and Control*, 1988, pp. 131–163.

[20] P. Koton, "SMARTplan: A case-based resource allocation and scheduling system," in *Proc. DARPA Workshop on Case-Based Reasoning*, Pensacola Beach, FL, 1989, pp. 285–289.

[21] P. Koton, "Evaluating case-based problem solving," in *Proc. DARPA Workshop on Case-Based Reasoning*, Pensacola Beach, FL, 1989, pp. 173–175.

[22] A. Kusiak and A. Villa, "Architectures of expert systems for scheduling flexible manufacturing systems," in *Proc. IEEE 1987 Int. Conf. Robotics and Autom.*, pp. 113–117.

[23] A. Lamatsch, M. Morlock, K. Neumann, and T. Rubach, "SCHEDULE—An expert-like system for machine scheduling," *Annals of Operations Res.*, vol. 16, pp. 425–438, 1988.

[24] W. S. Mark, "Case-based reasoning for autoclave management," in *Proc. DARPA Workshop on Case-Based Reasoning*, Pensacola Beach, FL, 1989, pp. 176–180.

[25] J. H. May, L. G. Vargas, R. De, S. A. Slotnick, T. E. Morton, D. W. Pentico, and G. J. Tatar, "Multex: An integrated AI/OR system for factory scheduling," in *Proc. Northeast DSI Conf.*, Pittsburgh, PA, Apr. 1991.

[26] J. R. Mayer, D. T. Phillips,and R. E. Young, "Artificial intelligence—Applications in manufacturing," *Smart Manufacturing With Artificial Intelligence*, pp. 10–24, 1987.

[27] R. K. Miller, E. Lufg ,and T. C. Walker, *Artificial Intelligence Applications in Manufacturing*. 1988, pp. 166–177.

[28] P. S. Ow, S. F. Smith, and R. Howie, "A cooperative scheduling system," in *Proc. 1988 Expert Systems and Intelligent Manufact.*, pp. 70–89.

[29] D. L. Parnas, "Why engineers should not use artificial intelligence," *Info.* vol. 26, no. 4, pp. 234–245, 1988.

[30] S. Piramuthu, S.-C. Park, N. Raman, and M. J. Shaw, "Integration of simulation modeling and inductive learning in an adaptive decision

support system," in *Model Management Systems,*\Bonczelc and A. B. Whinston, ed. IEEE Society Press, 1991.

[31] E. T. Powner and D. H. Walburn, "A knowledge based scheduler," in *1st Int. Conf. Expert Planning Systems*, Brighton, U. K., June 27–29, 1990, pp. 82–87.

[32] K. F. Reinschmidt, J. H. Slate, and G. A. Finn, "Expert systems for plant scheduling using linear programming," in *Proc. 4th Int. Conf. Expert Systems in Production and Operations Manage.,* Head Island,SC , May 1990, pp. 198–211.

[33] F. A. Rodammer and K. P. White, "A recent survey of production scheduling," *IEEE Trans. Syst., Man,and Cybernetics*, vol SMC-18, pp. 841–851, 1988.

[34] D. V. Savell, R. A. Perez, and S. W. Koh, "Scheduling semiconductor wafer production: An expert system implementation," *IEEE Expert,* pp. 9–15, Fall 1989.

[35] M. J. Shaw, "FMS scheduling as cooperative problem solving," *Operations Res.,* vol. 17, pp. 323–346, 1988.

[36] M. J. Shaw, "Dynamic scheduling in cellular manufacturing systems: A framework for networked decision making," *J. Manufact. Syst.*, vol. 7, no. 2, pp. 83– 94, 1988.

[37] M. J. Shaw, "Knowledge-based scheduling in flexible manufacturing system: An integration of pattern-directed inference and heuristic search," *Int. J. Production Res.*, vol. 26, no. 5, pp. 821–844, 1988.

[38] M. J. Shaw, N. Raman, and S.-C. Park, "Intelligent Scheduling with Machine Learning Capabilities: The Induction of Scheduling Knowledge," Tech. Rep. AI-DSS-91-01, Beckman Institute, Univ. of Illinois, Urbana-Champaign, 1991.

[39] M. J. Shaw, S.-C. Park, and U. Menon, "Incorporating Machine Learning in Knowledge-Based Process Planning Systems: An Explanation-Based Approach," Tech. Rep., Beckman Institute, Univ. of Illinois, Urbana-Champaign, October, 1988.

[40] M. J. Shaw, U. Menon, and S.-C. Park, "Machine learning methods for computer-aided process planning," in *Expert Systems and Manufacturing Designs,* A. Kusiak, ed. Dearborn, MI: Soc. of Manufact. Eng. Press, 1988.

[41] M. J. Shaw, P. L. Tu, and P. De, "Applying machine learning to model management in decision support systems," *Decision Support Syst.,* vol. 4, pp. 285–305, 1989.

[42] M. J. Shaw and A. B. Whinston, "Task bidding and distributed planning in flexible manufacturing," in *2nd Conf. on Artificial Intelligence Applications, IEEE Computer Society,* Miami Beach, FL, Dec. 1985, pp. 184–189.

[43] S. F. Smith, "Knowledge-based scheduling systems," in *Proc. 1st Int. Conf. Expert Systems and the Leading Edge in Production Planning and Control,* 1988, pp. 381–383.

[44] M. S. Steffen, "A survey of artificial intelligence-based scheduling systems," in *Proc. Fall Industrial Engineering Conf.,* Dec. 7–10, 1986, pp. 395–405.

[45] M. S. Steffen and T. J. Greene, "Hierarchies of sub-periods in constraint-directed scheduling," in *Symp. Real Time Optimization in Automated Manufact. Facilities,* National Bureau of Standards, Gaithersburg, MD, Jan. 1986, pp. 167–183.

[46] M. S. Steffen and T. J. Greene, "A prototype system for scheduling parallel processors using artificial intelligence methods," in *1986 Ann. Int. Industrial Eng. Conf. Proc.,* pp. 156–164.

[47] J. A. Svestka, "a real time rescheduler—Supplying the missing link," in *Proc. IXth ICPR,* Cincinnati, OH, Aug. 1987, pp. 1689–1693.

[48] A. Thesen and L. Lei, "An 'expert' system for scheduling robot in a flexible electroplating system with dynamically changing work loads," in *Proc. 2nd ORSA/TIMS Conf. Flexible Manufact. Syst.: Operations Res. Models and Appl.,* 1986, pp. 555–566.

[49] R. A. Wysk, S. Y. D. Wu, and N. S. Yang, "A multi-pass expert control system (MPECS) for flexible manufacturing systems," in *Symp. Real Time Optimization in Automated Manufact. Facilities,* National Bureau of Standards, Gaithersburg, MD, Jan. 1986pp. 9–25.

[50] Y. Yih, "Learning decision rules for FMS from the optimal policy of user-based semi-Markov decision processes," in *Proc. 4th Int. Conf. Expert Systems in Production and Operations Manage.,* Head Island, SC, 1990, pp. 175–183.

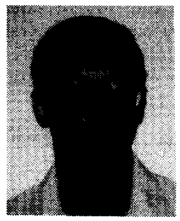
[51] Y. Yih, "Trace driven knowledge acquisition (TDKA) for rule based real time scheduling systems," working paper, Purdue Univ..

[52] Y. Yih and A. Thesen, "Semi-Markov decision models for real time scheduling," *Int. Journal of Production Res.,* vol. 29, no. 11, pp. 2331–2346, 1991.

Haldun Aytug received the B. S. in industrial engineering in 1988 from Bogazici University, Turkey and his Ph. D. from the University of Florida in 1993.

He is currently an Assistant Professor of MIS/DSS at the School of Business and Engineering Administration at Michigan Technological University. His research interests include machine learning applications and application of genetic algorithms i n production problems, object oriented simulation modeling, and simulation generators.

Dr. Aytug is a member of TIMS, Alpha Iota Delta, and Beta Gamma Sigma honorary societies.



Siddhartha Bhattacharyya received the Ph. D. degree from the University of Florida in 1993.

His work experience includes an assistant professorship in the Department of Management in Southern Illinois University at Carbondale. His current research interests are in the areas of machine learning, genetic algorithms, and manufacturing. Dr. Bhattacharyya is a member of TIMS and DSI.

Gary J. Koehler received the Ph. D. from Purdue University in management science in 1973.

He has held academic positions at the Univsity of Florida, Northwestern University, and Purdue University. He is also a founder and past president of Micro Data Base Systems, Inc. At present, he is a Professor and area head for Management Iinfor mation Systems in the Krannert Graduate School of Business at Purdue University. His current research includes machine learning in manufacturing environments, information systems, and decision support systems.

Dr. Koehler is a member of TIMS.

Jane L. Snowden received the B. S. degree in industrial and management systems engineering from Pennsylvania State University and the M. S. degree in industrial and operations engineering from the University o f Michigan. She is completing the Ph. D. degree at the Georgia Institute of Technology.

She is a consultant with the Buisness transformation Practice in the IBM Consulting Group. She has over 11 years of engineerin experience in both project management and technical roles. She specializes in business analysis and simplification, qu ality function deployment, and expert systems. She is currently Project Manager for an Integrated quality management system, which combines quality function deployment (QFD) techniques, customer satisfaction surveys, statistical process control (SPC), an d a quality monitoring system for suppliers. She is also Project Manager for joint research between IBM and the University of Florida on dynamic scheduling of manufacturing systems using machine learning. She is a frequent speaker on the subject of busin ess process simplification and has participated in panel discussions and forums at conferences and universities.

Ms. Snowden is a Senior Member of the Institute of Industrial Engineers, and serves as the Vice-President of the West palm Beach, Fl chapter. She is a member of the American Society of Quality Control and the Operations Research Society of America.