

---

# A review of machine learning in dynamic scheduling of flexible manufacturing systems

---

PAOLO PRIORE, DAVID DE LA FUENTE, ALBERTO GOMEZ, AND JAVIER PUENTE

ETSII e II, Campus de Viesques, 33204 Gijón, Spain

(RECEIVED January 25, 1999; ACCEPTED February 1, 2001)

## Abstract

A common way of dynamically scheduling jobs in a flexible manufacturing system (FMS) is by means of dispatching rules. The problem of this method is that the performance of these rules depends on the state the system is in at each moment, and no single rule exists that is better than the rest in all the possible states that the system may be in. It would therefore be interesting to use the most appropriate dispatching rule at each moment. To achieve this goal, a scheduling approach which uses machine learning can be used. Analyzing the previous performance of the system (training examples) by means of this technique, knowledge is obtained that can be used to decide which is the most appropriate dispatching rule at each moment in time. In this paper, a review of the main machine learning-based scheduling approaches described in the literature is presented.

**Keywords:** Discrete Simulation; Dispatching Rules; Dynamic Scheduling; Flexible Manufacturing Systems; Machine Learning

## 1. INTRODUCTION

Scheduling, a part of any manufacturing system's control process, is necessary when a common set of resources needs to be shared to manufacture several different products during the same time period. The goal of scheduling is to assign machines and other resources to jobs, or operations within jobs, in an efficient manner, as well as to determine the moment when each of the jobs is processed (Shaw et al., 1992)

Processing times in flexible manufacturing systems (FMSs) are almost deterministic, as operations are computer-controlled and mainly processed by numerically controlled machines, and setups between consecutive operations are automated. As a result, providing the system is not disturbed in some way, results can be predicted and a fixed off-line scheduling system is sufficient.

However, the actual states of FMSs may not be predictable because of part arrivals, machine states (up or down), tool breakages, rushed jobs, and many other system disturbances. This dynamic, uncertain nature of the FMS suggests that an off-line scheduling system is not really the most adequate. Moreover, FMSs are more sensitive than

conventional manufacturing systems to disturbances, as their components are more synchronized, more integrated, and more interdependent. They therefore require immediate response to changes in system states, using a real-time scheduling method. If system states change dynamically, scheduling of parts should be done as a function of the current state of the system (Jeong & Kim, 1998).

The rest of this paper is organized as follows. The different techniques described in the literature to schedule FMS jobs are first described. Then, there is a description of two ways of modifying the dispatching rules dynamically so as to overcome the problem that dispatching rules cause when applied statically. One of these two ways is based on the use of a simulation model, whilst the other utilizes "scheduling knowledge" of the manufacturing system. A review of the work done on the two approaches is then provided, and their main characteristics are described. Finally, the paper ends with a consideration of a series of generalized shortcomings of knowledge-based systems that need to be dealt with in future research.

## 2. APPROACHES TO SCHEDULING IN FMSs

The different approaches available to solve the problem of FMS scheduling can be divided into the following categories:

---

Address correspondence and reprint requests to: Paolo Priore, ETSII e II, Campus de Viesques, 33203 Gijón, Spain. E-mail: priore@etsiig.uniovi.es

1. The analytical approach.
2. The heuristic approach.
3. The simulation-based approach.
4. The artificial intelligence-based approach.

The analytical approach interprets an FMS scheduling problem as an optimization model with certain constraints, in terms of an objective function and explicit constraints. An appropriate algorithm is then used to resolve the model (see, e.g., Stecke, 1983; Kimemia & Gershwin, 1985; Shanker & Tzen, 1985; Lashkari et al., 1987; Han et al., 1989; Hutchison et al., 1989; Shanker & Rajamarthandan, 1989; Wilson, 1989).

In general, these problems are of a NP-complete type (Garey & Johnson, 1979). Heuristic and off-line type algorithms are therefore usually proposed to resolve this kind of problem (Cho & Wysk, 1993; Chen & Yih, 1996). However, these analytical models contain simplifications that are not always valid in practice. Indeed, Basnet and Mize (1994) state that some models are so singular that one has the impression that the problems are invented to fit the model rather than *vice versa*. They are not efficient for reasonably large-scale problems either.

The above-mentioned difficulties of applying the analytical approach to scheduling problems led to research into many heuristic approaches. These are usually dispatching rules, although they may be more complicated than that, and they are generally used to schedule the jobs in a manufacturing system dynamically. These heuristics use different priority schemes to order the different jobs competing for the use of a given machine. Each job is assigned a priority index and the one with the lowest index is selected first.

Many researchers (see, e.g., Panwalkar & Iskander, 1977; Blackstone et al., 1982; Baker, 1984; Russel et al., 1987; Vepsalainen & Morton, 1987; Ramasesh, 1990; Kim, 1990) have evaluated the performance of these dispatching rules on manufacturing systems using simulation. The conclusion to be drawn from such studies is that their performance depends on many factors, such as the criteria that are selected, the system's configuration, the work load, and so on (Cho & Wysk, 1993). With the advent of FMSs came many studies analyzing the performance of dispatching rules in these systems (see, e.g., Stecke & Stolberg, 1981; Egbelu & Tanchoco, 1984; Denzler & Boe, 1987; Choi & Malstrom, 1988; Henneke & Choi, 1990; Montazeri & Van Wassenhove, 1990; Tang et al., 1993).

In view of the variable performance of dispatching rules, it would be interesting to modify these rules dynamically and at the right moment according to the system's conditions. That this approach will be better than the conventional system of using a dispatching rule constantly is an *a priori* assumption, for two reasons. Firstly, because it can identify the best rule for a given manufacturing scenario. Given such a selection capacity, the system should perform

at least as well as the best of the candidate dispatching rules being considered. Secondly, this approach can adapt its choices dynamically to changing scenarios. This adaptability should result in job scheduling of a higher quality than even the best dispatching rules (Shaw et al., 1992).

Basically, two approaches to modifying dispatching rules dynamically can be found in the literature. Firstly, the rule is selected at the appropriate moment by simulating a set of preestablished dispatching rules and choosing the one that provides the best performance. In the second approach, belonging to the field of artificial intelligence, a set of earlier system simulations (training examples) is used to determine which is the best rule for each possible system state. These training cases are used to train a machine learning module to acquire knowledge about the manufacturing system. Such knowledge is then used to make intelligent decisions in real time. These scheduling systems are normally said to be knowledge-based.

In contrast, there are other scheduling schemes within the artificial intelligence approach in which dynamic modification of the dispatching rule does not take place (see, e.g., Fox & Smith, 1984; Maimon, 1987; Maley et al., 1988; Shen & Chang, 1988; Shaw & Whinston, 1989; Chaturvedi et al., 1993; Dong & Kitaoka, 1994; De & Lee, 1998). Kanet and Adelsberger (1987), as well as Kusiak and Chen (1988), both present reviews of expert systems as applied to scheduling. Jain and Meeran (1998) review scheduling systems that use neural networks. Aytug et al. (1994) and Minton (1993) present a review of work in which machine learning is applied to solving scheduling and planning problems. Zweben and Fox (1994) give different scheduling systems that use artificial intelligence, including real systems used in different industrial fields (aerospace, defence, heavy industry, and semiconductor manufacturing).

Finally, in the literature on FMSs scheduling, there are many reviews of a general nature that apply any of the four above-mentioned approaches (see, e.g., Harmonosky & Robohn, 1991; Kouvelis, 1992; Gunasekaran et al., 1993; Basnet & Mize, 1994).

### 3. SIMULATION-BASED SYSTEMS

The general scheme of simulation-based scheduling systems is shown in Figure 1. When the simulator receives a request to select a rule, it carries out a series of simulations with each of the *a priori* selected rules. From amongst the results of the simulations carried out, the selector chooses the best dispatching rule to use to schedule the manufacturing system's jobs. Finally, if an anomaly occurs in the system, the control system sends a signal to the rule selector. This can then send the simulator a new request according to the type of anomaly observed. Some simulation-based systems that vary the dispatching rule applied at each particular moment dynamically will next be reviewed.

Wu and Wysk (1989) put forward a scheduling system and on-line control that selects the best rule *via* simulation

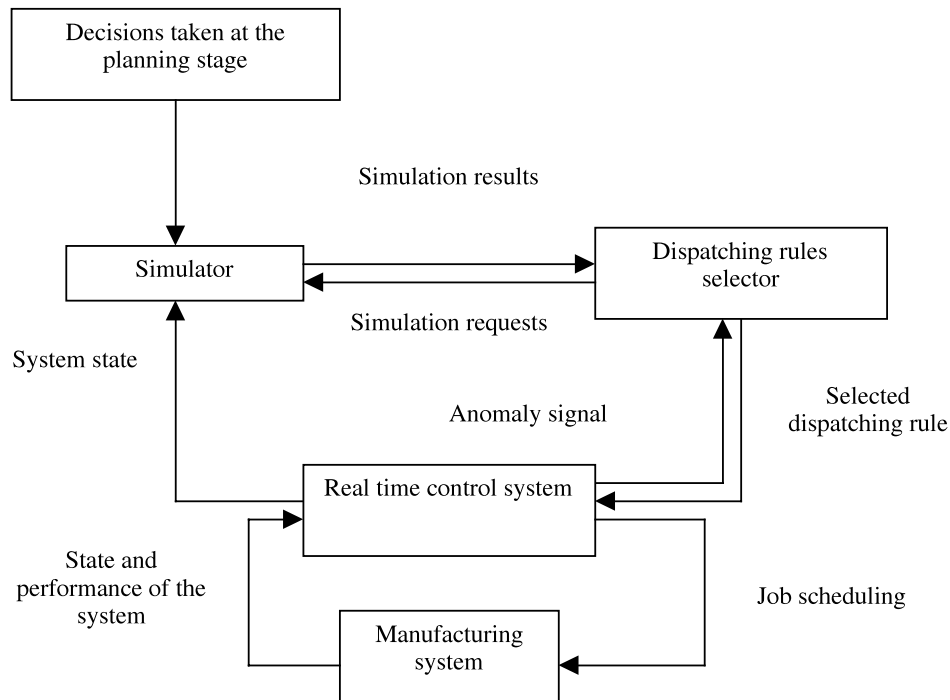


Fig. 1. General overview of a simulation-based scheduling system.

for each time period; they call this the scheduling interval. Although they do point out that the duration of this interval is an important factor insofar as it determines the system's performance, they do not provide a general procedure by which it might be defined. Simply by using multiples of the average total processing time, they make the claim and observation that three times this amount is the best possible scheduling interval. However, a constant interval cannot keep up with the changes of state in a system as dynamic as the FMS is. They likewise define a simulation window, to simulate a model and evaluate the performance of the candidate rules, which is the same as the scheduling interval. Because of the parts that remain in the system at the end of each simulation period, this window affects each of the rules differently (this problem is called "censored data"). Despite all this, the authors achieve an improvement of 7.7% and 21.11%, respectively, for the FMS they used, compared to employing one rule constantly, using mean tardiness and mean flow time as performance measurements.

Ishii and Talavage (1991) present an approach that attempts to solve the above problems which has three main components. First, it calculates the scheduling intervals as a function of an index that measures the system's state. The second element determines the dispatching rule to be used. The authors propose four strategies that define different simulation windows to reduce the problem of "censored data". The final element is the FMS simulator.

When the authors compared this approach with the one proposed by Wu and Wysk (1989), they observed that the latter displayed extremely variable behavior, owing to the

problem of "censored data" and the constant scheduling interval. Moreover, the average improvement according to the six performance criteria that are considered is 5.17% if the third strategy is used to lessen the problem of "censored data". Curiously, the authors claim that the algorithm proposed by Wu and Wysk (1989) is 4.12% inferior to the best of the rules used constantly.

Kim and Kim (1994) suggest a real-time simulation-based scheduling method whose dispatching rules vary dynamically. The main elements of the proposed method are the simulator and the real-time control system. The function of the former is to evaluate the rules and select the best one for a given performance criterion. The latter component supervises the manufacturing system and checks its performance periodically.

The selected dispatching rule is applied until the difference between actual performance and performance as calculated by the simulator exceeds a given limit, or until there is a major disturbance. When either of these occur, a new rule is selected by the simulator with the jobs that remain to be carried out. The authors study the methodology in relation to the monitoring period of the control system and the limits in performance differences. The improvement with respect to mean tardiness and mean flow time is 6.10% and 2.08%, respectively.

Jeong and Kim (1998) use a scheme based on the work of Kim and Kim (1994) and analyze two factors that can influence the scheduling system. First, the type of simulation model that is used; this can be dynamic or static, depending on whether it includes probability distributions of system

disturbances. Second, the authors study the right moment to select a new rule, for which there are four options:

1. Once only, at the beginning of the planning horizon.
2. When there is a major disturbance.
3. If there is a major disturbance or if the difference between the actual performance value and the one estimated by the simulator goes over a given limit.
4. When any kind of disturbance occurs.

Their results show that modifying the rules in response to changes in the manufacturing system reduces the mean flow time between 1.77% and 3.20%. Moreover, mean tardiness goes down between 8.34% and 12.05%. Furthermore, ANOVA analysis shows that the simulation model type is not relevant. The authors conclude by calculating the percentage use of each of the rules, and they come to the conclusion that no rule predominates over the others, and that it is therefore useful to substitute rules when the state of the system changes.

The main generalized drawbacks to these simulation-based systems are the following:

1. The time required to examine the performance of the set of candidate rules, which can make real-time scheduling difficult.
2. Very frequent changes in the system. As the evaluation of each of the rules is carried out until the end of the period considered, there may not be a match between the rule that is proposed and the one that is really required, as the one that is chosen is used for a period of time that is less than the time used during its evaluation.
3. Methods to avoid unnecessary modifications of the dispatching rules during transitory changes are not available.
4. No knowledge is acquired about the system.
5. There are no methods to determine a reasonable simulation window.
6. The scheduling interval must be defined at machine, not system, level (Chiu & Yih, 1995).

#### 4. KNOWLEDGE-BASED SYSTEMS

A real-time scheduling system that modifies dispatching rules dynamically should fulfil two contradictory characteristics to work adequately (Nakasuka & Yoshida, 1992):

1. Rule selection must contemplate a variety of information about the manufacturing system in real time.
2. Rule selection must be completed in such a short time that real operations are not delayed.

One way of achieving these characteristics is to utilize some class of knowledge about the relationship between the manufacturing system's state and the rule to be applied at that moment. It is therefore useful to use "scheduling knowledge" of the manufacturing system to save time and get a rapid response in a dynamically changing environment (as are FMS environments). However, one of the most difficult problems to solve in a knowledge-based system is precisely how this knowledge is to be acquired.

To acquire knowledge, machine learning techniques, such as inductive learning or neural networks, are used. These reduce the effort involved in determining the knowledge required to make scheduling decisions. However, the training examples and the learning algorithm must be right for this knowledge to be useful. Moreover, in order to get the training examples, the attributes that are selected are crucial to the performance of the scheduling system that is generated (Chen & Yih, 1996).

There are at least four reasons why a knowledge-based approach might perform worse than the best rules used individually:

1. The training set is a subset of the universe of all possible cases. However, situations in which the scheduling system does not work properly can always be observed and added as training examples.
2. The system's performance depends on the number and range of control attributes taken into account in the design of the training examples.
3. A rule may perform well in a simulation over a long time period for a set of given attributes, but will perform poorly when applied dynamically.
4. The system can be prone to inadequate generalizations in extremely imprecise situations.

An overview of a knowledge-based scheduling system is shown in Figure 2. The examples generator uses a simulation model to generate different manufacturing system states and search for the best dispatching rule for that state. The training examples that the machine learning module needs are generated by an information processor, based on the simulation results. The machine learning module acquires the knowledge that is necessary to make future scheduling decisions by using the training cases. The knowledge may need to be refined, depending on the manufacturing system's performance, by generating further training examples. The remaining elements of the system have similar functions to those described in Figure 1.

Several knowledge-based approaches that dynamically modify the dispatching rule being used at a specific instance are reviewed next. According to the type of machine learning algorithm used, these approaches can be divided into the following categories:

1. Approaches that do not use knowledge-acquisition algorithms.

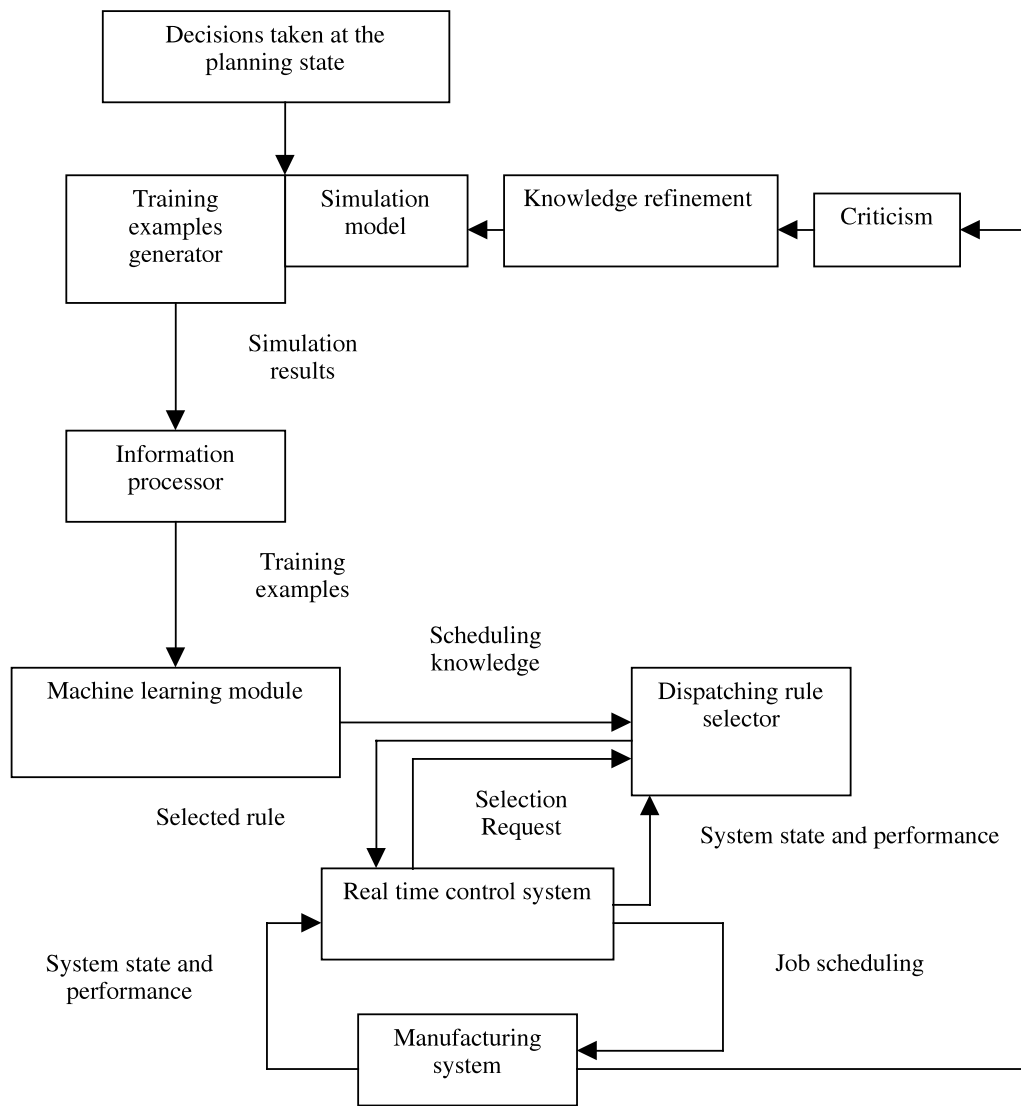


Fig. 2. General overview of a knowledge-based scheduling system.

2. Inductive learning-based approaches.
3. Neural network-based approaches.
4. Mixed approaches. Here a combination of different types of learning algorithms is applied.
5. Other approaches based on machine learning algorithms.

**4.1. Approaches that do not use knowledge-acquisition algorithms**

Thesen and Lei (1986) propose an expert system for scheduling robots in a flexible electroplating system. The authors carry out a series of *a priori* simulations using different dispatching rules to study the performance of the manufacturing system in different situations; 38 training examples were acquired in the process. However, knowledge of the

system was not acquired by any machine learning procedure, but rather by inspection carried out directly on the simulation results. The authors observe that the manufacturing system increases the number of parts produced in percentages ranging between 7% and 30%.

Sarin and Salgame (1990) define an expert system to schedule dynamically. At the beginning of a given time period, the system has a known schedule of jobs that is followed during this period. This system reacts when a change occurs. Changes are classified into different groups: machine breakdown, rush jobs, new batch of jobs, material shortage, labor absenteeism, job completion at a machine, and change in shift. The system they propose has the following parts to it: a scheduling knowledge, a global database, a user interface, and a control block. The knowledge is divided into several groups, each of which has rules to solve different types of problems depending on the changes

that occur in the system. The rules represent the heuristics of a human expert.

The global database has information on the different jobs and shifts that exist at a specific point in time. The group of rules that needs information goes to the global database. Finally, the control block, which has the form of a tree (“meta-rules” or “knowledge about knowledge”), chooses the group of rules that fits the new problem that originated the change in the system.

The same authors likewise present an integrated system made up of two modules. The first, which is backed up by mathematical programming, determines a predictive schedule as a starting point. The second, the expert system, takes over control of executing dynamic or reactive scheduling as the new situation demands, whenever a change occurs. Finally, they point out that this integrated approach has still not been implemented in a real case.

Chandra and Talavage (1991) present a system called EXPERT, made up of a set of decision rules. The information that is used in the decision process is the congestion level of the manufacturing system, the preference of a part for a machine, how critical the part is (it indicates the part’s ability to meet its due date), and the objective of the manufacturing system at that specific moment. The authors state that in principle the aim of maximizing the work progress rate is an interesting one, even though there is the risk that some jobs are delayed (especially if the system is overloaded and there are a lot of critical jobs). Excessive preoccupation with critical jobs can make the system worse; for this reason the objective of maximizing the work progress rate is chosen as the first criterion.

Jobs are furthermore divided into groups (high, medium, and low preference) instead of them being classified individually. The system that is proposed selects the job assigned to a machine, beginning with the high preference ones, pursuing the primary objective whilst also searching for opportunities to improve the secondary one (minimizing the number of tardy jobs). In certain cases, jobs available in the near future are inspected. If there is a tie, or when a clear decision is not taken, then the shortest processing time (STP) rule is applied. At the experimental stage, it was shown that the EXPERT system is superior to conventional dispatching rules.

Sabuncuoglu and Hommertzheim (1992) suggest a dynamic algorithm to schedule jobs in machines and automatic guided vehicles (AGVs). The algorithm they propose is based on the idea that a job should not be assigned to a machine if it has to wait for an AGV in the following operation and *vice versa*. It uses several priority schemes (or rules) and information about the system (queue levels, the number of parts in the system, machine state, etc.) and about the jobs (processing times, number of operations, etc.). The algorithm has two fundamental parts: a set of procedures for scheduling jobs in the machines, and others to schedule the jobs in the AGVs. The latter check whether there are blocked or empty stations or parts in the central buffer.

The authors compare the algorithm that is proposed, using mean flow time and mean tardiness as performance criteria, with the two best dispatching rule combinations for machines and AGVs. To do this they proposed different scenarios, varying the load level, the queue capacity,  $F$  (flow allowance; Baker, 1984), the type of processing time distribution, and the performance criterion. The algorithm was found to perform better than the best rule combinations when machine load is high (if it is low there are hardly parts in the queue so the rule selected has no influence) and queue size is small. In such conditions an improvement of over 12% is achieved.

Pierreval and Mebarki (1997) introduce a heuristic methodology, called SFSR (shift from standard rules), to dynamically modify the dispatching rules according to two performance criteria (one primary and the other secondary). The SFSR heuristic checks the manufacturing system’s state when a resource becomes available or a new job arrives. By using rules that were defined beforehand that are functions of parameters to be optimized, the presence of certain symptoms in the manufacturing system can be detected (for example, when the system is congested or when there is a job which is waiting too long, etc.). The optimal values of the parameters are calculated by the Hooke-Jeeves method (Hooke & Jeeves, 1961). If there are no symptoms in the system, standard rules taken from the literature depending on the criterion to be optimized are used. The dispatching rule to be applied can thus be calculated.

If the opposite is the case, rules defined by the authors are used; these depend on the criteria to be optimized, on the symptom detected, and on the state of the system. In general, the methodology that is proposed improves upon the alternative of using a rule constantly according to the primary criterion; if this is not the case, it compensates with the secondary criterion. The improvements vary between 12.3% and 33.8%. The greatest defect of the methodology is that the standard rules are defined according to research results already presented in the literature. An alternative approach would be for them to be generated by inductive learning, so as to take account of the peculiarities of the system under study. Inductive learning could also be used to generate other types of rules used in SFSR.

#### 4.2. Inductive learning-based approaches

Pierreval and Ralambondrainy (1990) suggest an inductive learning algorithm called GENREG, to obtain heuristic rules to know system performance with different dispatching rules and states of the manufacturing system. In the methodology that they propose, a rule is obtained with each training example. As the number of rules is extremely large, GENREG is then used to generalize them, and thereby reduce this number. The approach they propose is applied in a simplified flow-shop configuration with two machines, using 198 training examples. However, the rules obtained using GENREG are not used dynamically.

Shaw et al. (1992) present a system called PDS (Pattern-Directed Scheduling) to schedule jobs in an FMS that uses inductive learning. Here, the learning algorithm that is used is ID3. During the knowledge-acquisition stage 130 training examples are used, applying mean tardiness as performance criterion. This system provides a mean tardiness reduction of 11.5%.

The authors observe that the maximum effectiveness of the approach is obtained when the number of changes in the manufacturing system's states (patterns) is between medium and reasonably high. Moreover, the number of alternative machines to process a given operation does not need to be very high. The authors confirm that all these characteristics occur in most real FMSs.

Nakasuka and Yoshida (1992) propose a scheduling scheme called LADS (Learning-Aided Dynamic Scheduler) that incorporates an inductive learning algorithm with two characteristics that differentiate it from conventional algorithms. Firstly, there is a new criterion to decide how to separate data groups; owing to data noise in scheduling problems, dividing them so that they belong to a single class is of no interest, as the number of data in each group would be very small. The second characteristic of the algorithm that is proposed is the generation of linear combinations of attributes fed into the system. The scheduling scheme being proposed is used in a simplified flow-shop system with three machines so as to minimize the makespan and keep mean tardiness below a set level. The authors point out that the system is superior (by both criteria) to using one rule constantly.

Piramuthu et al. (1993) define an approach to scheduling jobs dynamically. The approach they propose, along with the examples they apply it to, is similar to the one the authors present in others works (see for example, Shaw et al., 1992; Piramuthu et al., 1994). What this paper does contribute, compared with others that the same authors have published, is a more elaborate theoretical framework for each of the parts that make up the job scheduling system.

Piramuthu et al. (1994) define a methodology to schedule jobs using inductive learning in a flexible flow-shop manufacturing system with mean flow time as the performance criterion. By using C4.5 as the learning algorithm, two decision trees are generated *via* 66 training examples. The first is to schedule jobs in the machines themselves, and the second is for part-release decision. They also present a refinement procedure for the decision trees, which consists of including cases that the system misclassifies in the training set.

The authors observe that incorporating a decision tree to select the dispatching rule does not improve results significantly with respect to the alternative of using the decision tree only for part-release decision using a dispatching rule constantly. Moreover, this methodology is particularly useful when input buffer size is limited and small, and there is a great variation in processing times for parts in the bottleneck machines.

### 4.3. Neural network-based approaches

Chen and Yih (1996) define an approach to determine the most important attributes as a first step to constructing knowledge-based scheduling systems. The approach for identifying attributes has three steps to it:

1. Data collection *via* manufacturing system simulation.
2. Building of attributes/performance mapping functions of the manufacturing system using back-propagation neural networks.
3. Selection of essential attributes.

To do this, an attribute is omitted and the difference between the original output and the output obtained with the attribute omitted is measured. More important attributes have a greater difference than those with a lower significance level. Twenty candidate attributes taken from the literature (Nakasuka & Yoshida, 1992; Cho & Wysk, 1993; Chiu, 1994) are used in the experimental study (using attribute variability defined as the variance of the attribute divided by its mean), along with six performance measurements and ten dispatching rules. One thousand three hundred training examples are generated for each type of rule and the corresponding neural network, and the ten most important attributes are selected.

Finally, the authors compare three neural networks with input nodes formed by alternative groups of attributes (with the ten selected attributes, with the first twenty, and with the other ten) and output nodes corresponding to the rules that are selected. There is verification of the fact that the network's capacity for generalization obtained with the ten significant attributes is 9% superior with respect to the network formed by the first twenty attributes. The authors claim that using as many attributes as possible to build up a knowledge base does not improve generalization or prediction capacity. The more attributes that are included, the greater the effort required to develop the knowledge base, and the more complex its structure becomes. The main defects of the approach being proposed are that it does not identify important attributes if they are not considered initially and the process must be repeated if performance measurements change.

Sun and Yih (1996) apply an approach that uses a back-propagation neural network on each machine to select the most adequate dispatching rule in a multiple criterion environment. At each point of decision, when a machine has to select a new job, an adjustment module determines the relative importance of each performance criterion as a function of the desired and current values. Taking the value provided by the adjustment module and the current state of the machine as the input value, the neural network provides the most adequate dispatching rule.

The authors used around 1000 training examples for each neural network, and show that the proposed approach has an average performance 4.2% better than the best rules used

constantly. Moreover, it is very adaptable to changes in the choice of performance criteria that are given priority. The greatest defect of this approach is that the manufacturing system being studied does not have flexible routes and the number of parts is limited.

Min et al. (1998) propose a methodology that uses competitive neural networks. Here, differences between the values of performance criteria and system state variables in different time intervals are used as attributes. By simulating the system for a long time period and modifying the dispatching rules randomly, the training examples are obtained with these differences. Three thousand five hundred examples are used in the training stage and 40 network nodes or classes are defined. Neural networks are then used to obtain classes from the training examples.

Their scheduling system works in real time in the following way. First, the user sets differences as an objective and the class is identified using the neural network. Then, from amongst all the training examples, those that have the same class and the same decision variables (dispatching rules) from the previous interval are sought. If this example is not found, which is the likely case, the most used rule within this class is chosen for each decision variable.

An interesting characteristic of this approach is that it uses earlier dispatching rules to find the new ones. The system proposed is compared to another one that chooses rules randomly (ten replicas of the random system are in fact made and the best from amongst them is chosen) and it is shown to have superior performance. The drawbacks of the approach are the lack of a method to systematically search for an optimum number of output nodes to the neural network, and that it is compared to a random system rather than the best possible combination of the proposed dispatching rules.

#### 4.4. Mixed approaches

Wu and Wysk (1988) propose a control and scheduling scheme called MPECS (Multipass Expert Control System) that combines expert systems, simulation, and inductive learning. The system they propose has three modules: an intelligent scheduling module, a manufacturing system simulator, and a cell control module. The first element is in turn made up of a knowledge base, an inference engine, and a learning module.

The base has declarative knowledge (information about the system state, scheduling heuristics, and rules), and procedural information (general criteria, in rule form, to select dispatching rules). The inference engine is a search mechanism to select the right rules of procedural knowledge. Finally, the learning module generates a set of rules from the training examples that associate dispatching rules, performance measurements, and the system's characteristics. The rules created by the learning module are sent to procedural knowledge.

The intelligent scheduling module is activated when a new job arrives, or if there is an anomaly in the system. The task of the simulator is to examine the performance of the dispatching rules suggested by the intelligent scheduling module and to select the best. Finally, the control module allows scheduling in the physical cell. Moreover, most information about the cell, which is essential to control it, is obtained and manipulated through this module. The authors point out that the proposed scheme produces a performance improvement in the manufacturing system of between 2.3% and 29.3%, when compared to the alternative of using a dispatching rule constantly.

Rabelo and Alptekin (1989) define an approach called ISS/FMS (Intelligent Scheduling System for FMS) to schedule jobs, made up of three basic modules. The first of them is an expert system which decides the heuristic rule to use based on certain information (data on work to be done, constraints imposed by the workshop, cell state, etc.). It takes into account data provided by a neural network and a statistical analysis model that study past cases obtained *via* a simulation study. The second module carries out a heuristic process (Kiran & Alptekin, 1989) that depends on two coefficients determined by a neural network as a function of the characteristics of the scheduling problem being solved. The third component chooses the best of the solutions calculated by the previous two modules.

Cho and Wysk (1993) present a system called IWC (Intelligent Workstation Controller) that uses neural networks and a simulator, based on the work of Wu and Wysk (1989). The neural network has seven input nodes, corresponding to the state of the system, and nine output nodes, one for each of the dispatching rules considered. The network is trained with 90 examples taken from the literature, taking into account different configurations of the hidden layers and different learning rates.

Using the two best rules provided by the network, the simulator selects the better of them as a function of the state of the manufacturing system. Moreover, the authors experimentally calculate the most adequate simulation window for the performance criterion chosen using a set of simulations. It is observed that IWC is superior to the use of one rule constantly, although percentage improvement never goes beyond 3%.

Li and She (1994) use an approach that utilizes cluster analysis (Evert, 1980) and inductive learning. Six hundred examples are uniformly generated from the decision spectrum. By using cluster analysis, seven classes with similar performance values are established. Then an algorithm similar to C4.5 establishes two sets of rules that determine the class as a function of the decision and performance attributes.

One way that the authors suggest of using this scheduling knowledge is to set performance conditions and determine the class they correspond to. Once the class is known, the decision variables that will be taken are determined using the other set of rules. However, this methodology is not compared with any other to check how it works.



Chiu and Yih (1995) put forward a system that uses inductive learning and genetic algorithms. The latter are used to search for a set of good-quality training examples. To do this, at each point of decision the best dispatching rule is chosen and, this rule forms a training case along with the state of the system. Furthermore, the learning algorithm can modify the decision tree when new examples are presented only if the change is significant. The authors show that the proposed approach is superior to using a dispatching rule constantly. The greatest defect of the approach lies in the need to change the induced scheduling knowledge when there are small modifications in the manufacturing system.

Lee et al. (1997) propose a scheme that also uses inductive learning and genetic algorithms. The first technique serves to generate a decision tree using C4.5 to select the best rule to control the input flow of jobs to the system. The genetic algorithms are used to select the most appropriate dispatching rules for each of the system's machines. The authors verify the approach they propose with two job shop systems (one of them with a bottleneck machine), using mean tardiness as performance criterion, and show that it beats the best combination of rules used constantly, at a rate of between 20.34% and 25.28%. However, times required (26 and 168 min for the first and second cases, respectively) are rather high for this system to work in real time.

Kim et al. (1998) suggest a scheme that broadens the scope of earlier work (Min et al., 1998) and uses competitive neural networks and inductive learning. Once the classes are obtained from the neural networks, inductive learning is applied to express knowledge in tree form and production rules. The authors use 99,999 training cases and establish a network with 100 groups or classes.

The scheduling system works in real time in the same way as the previously described approach (Min et al., 1998). The only difference is that the class is identified by the production rules obtained from the C4.5 inductive learning algorithm. The authors compare this system with another that only uses a competitive neural network, and demonstrate its superiority, due to the C4.5's tree pruning algorithm, which deals with noise in the data more efficiently. This scheme displays the same defects as the approach suggested in Min et al. (1998).

#### 4.5. Other approaches based on machine learning algorithms

Quiroga and Rabelo (1995) solve the problem of scheduling jobs for a machine by inductive learning (ID3), back-propagation neural networks, and fuzzy logic. They use 358 training cases and 198 test cases, and the test error is lower than 10% in the three methodologies. Inductive learning and fuzzy logic have the advantage of generating rules that are intelligible to humans, which is not the case for neural networks. However, the latter are less sensitive to noise or

incomplete data, and have the lowest level of test error (1.2%).

Bowden and Bullington (1996) suggest a scheme called GARDS (Genetic Algorithm Rule Discovery System) to determine control strategies using genetic algorithms. GARDS has three fundamental modules:

1. A simulation model to analyze the performance of the different strategies that are generated.
2. An algorithm that determines the most adequate rule within a strategy or plan for the current state of the manufacturing system.
3. A genetic algorithm that uses traditional crossover and mutation operators to improve the initial plans by choosing the best for the control system.

The system that is proposed is tested on two configurations of different complexity with the aim of minimizing the number of tardy jobs. It was observed that GARDS improves the performance of manufacturing systems with respect to several classical heuristic methods (e.g., sending jobs to the queue of the machine with fewest jobs).

Table 1 recapitulates and summarizes the different approaches to be found in the literature, classified by the methodology used. Table 2 shows a collection of different approaches that dynamically modify dispatching rules, classified by the type of machine learning algorithm used.

### 5. LIMITATIONS OF THE SCHEDULING APPROACHES AND FUTURE RESEARCH DIRECTIONS

A number of limitations that would be desirable characteristics can be detected across the board in the knowledge-based approaches using machine learning algorithms that have been considered above. These limitations point to future directions of research in the field of dynamic scheduling of manufacturing systems, by modifying the dispatching rule that is employed. Future research directions would include the following:

1. Comparison of the different machine learning methodologies. The approaches described in the literature employ a methodology or, in certain cases, a combination of methodologies. However, there is no comparative study that determines which of them is the best. Furthermore, because of the wide range and disparity of the FMSs used in the literature that has been reviewed, it is not possible to have even an inkling of which of the methodologies described is the most adequate for resolving this type of scheduling problem.
2. The use of CBR (Case-Based Reasoning) as a machine learning methodology in scheduling systems. These algorithms are very efficient at classification, despite their simplicity (Rachlin et al., 1994). Yet none

**Table 1.** Classification of references according to methodology applied

Methodology	References
Analytical approach	Han et al. (1989); Hutchison et al. (1989); Kimemia & Gershwin (1985); Lashkari et al. (1987); Shanker & Rajamarthandan (1989); Shanker & Tzen (1985); Stecke (1983); Wilson (1989).
Heuristic approach	Choi & Malstrom (1988); Denzler & Boe (1987); Egbelu & Tanchoco (1984); Henneke & Choi (1990); Montazeri & Van Wassenhove (1990); Stecke & Solberg (1981); Tang et al. (1993).
Simulation-based approach	Ishii & Talavage (1991); Jeong & Kim (1998); Kim & Kim (1994); Wu & Wysk (1989).
Artificial intelligence-based approach	Bowden & Bullington (1996); Chandra & Talavage (1991); Chaturvedi et al. (1993); Chen & Yih (1996); Chiu & Yih (1995); Cho & Wysk (1993); De & Lee (1998); Dong & Kitaoka (1994); Fox & Smith (1984); Kim et al. (1998); Lee et al. (1997); Li & She (1994); Maimon (1987); Maley et al. (1988); Min et al. (1998); Nakasuka & Yoshida (1992); Pierreval & Mebarki (1997); Pierreval & Ralambondrainy (1990); PIRAMUTHU et al. (1993); PIRAMUTHU et al. (1994); Quiroga & Rabelo (1995); Rabelo & Alptekin (1989); Sabuncuoglu & Hommertzhaim (1992); Sarin & Salgame (1990); Shaw et al. (1992); Shaw & Whinston (1989); Shen & Chang (1988); Sun & Yih (1996); Thesen & Lei (1986); Wu & Wysk (1988).

of the systems reviewed uses CBR. It would therefore be interesting to test how good they are for scheduling problems.

3. Determination of the optimum number of training examples. None of the approaches reviewed calculate the number of examples required to optimally train the machine learning algorithm. Nor do they specify whether the test examples are the same, similar or very different to the training examples. Yet classification error of “scheduling knowledge”, and therefore the performance of a manufacturing system, depends to a great extent on the number of training examples that is considered. It is therefore necessary to study classification error as a function of the number of examples considered, and an adequate size of the training set must be chosen.
4. Selection of an adequate monitoring period. A study to determine the right monitoring period for each performance criterion is not generally done in the existing literature. However, the frequency of control attribute checking, to decide whether dispatching rules are to be changed or not, is a vitally important question that determines manufacturing system’s performance.

**Table 2.** Classification of references according to the machine learning algorithm applied

Machine Learning Algorithm	References
Is not used	Chandra & Talavage (1991); Pierreval & Mebarki (1997); Sabuncuoglu & Hommertzhaim (1992); Sarin & Salgame (1990); Thesen & Lei (1986).
Inductive learning	Nakasuka & Yoshida (1992); Pierreval & Ralambondrainy (1990); PIRAMUTHU et al. (1993); PIRAMUTHU et al. (1994); Shaw et al. (1992).
Neural Networks	Chen & Yih (1996); Min et al. (1998); Sun & Yih (1996).
Mixed	Chiu & Yih (1995); Cho & Wysk (1993); Kim et al. (1998); Lee et al. (1997); Li & She (1994); Rabelo & Alptekin (1989); Wu & Wysk (1988).
Others	Bowden & Bullington (1996); Quiroga & Rabelo (1995).

5. Determination of a mechanism or filter to smooth transitory states. On certain occasions the manufacturing system running on “scheduling knowledge” does not perform as it is expected to, and is worse than the alternative of using the best combination of dispatching rules constantly. This phenomenon is explained by the fact that the system reacts hastily to changes in control attributes that are only transitory in time. Thus, one proposal would be to use digital filters to smooth transitory scenarios in control attributes. This mechanism is not considered in most of the approaches that have been reviewed, or when it is considered, neither the different kinds of digital filters available nor their interrelationship with the monitoring period is analyzed.
6. Generation of new control attributes using an algorithm that can create attributes that are a combination of the initial ones. In some cases, it is necessary to check relationships of the following type in order to select the best dispatching rule: use of machine 1 is less than that of machine 2. To achieve these relationships, the arithmetical combinations of the basic initial attributes would need to be defined. However, these combinations are often not known at the outset, and can only be discovered in simple manufacturing systems after detailed examination of simulation results.
7. Incorporation of a simulator. The performance of the scheduling system could be enhanced if a simulator was used to determine the best rule from amongst those that the machine learning system considered the most important ones. On occasions, and given certain control attribute values, “scheduling knowledge” determines that there are two or more dispatching rules

**Table 3.** *The characteristics of scheduling systems*

References	Machine Learning Algorithm	Monitoring Period	Filter	Generation of New Attributes	Incorporation of a Simulator	Knowledge Refinement
Chandra & Talavage (1991)	No	No	No	No	No	No
Pierreval & Mebarki (1997)	No	No	No	No	No	No
Sabuncuoglu & Hommertzhaim (1992)	No	No	No	No	No	No
Sarin & Salgame (1990)	No	No	No	No	No	No
Thesen & Lei (1986)	No	No	No	No	No	No
Nakasuka & Yoshida (1992)	LADS	No	No	Yes	No	No
Pierreval & Ralambondrainy (1990)	GENREG	No	No	No	No	No
Piramuthu et al. (1993)	ID3; C4.5	No	Yes	No	No	Yes
Piramuthu et al. (1994)	C4.5	No	Yes	No	No	Yes
Shaw et al. (1992)	ID3	No	Yes	No	No	No
Chen & Yih (1996)	Backpropagation NN	No	No	No	No	No
Min et al. (1998)	Competitive NN	No	No	No	No	No
Sun & Yih (1996)	Backpropagation NN	No	No	No	No	No
Chiu & Yih (1995)	Inductive Learning; Genetic Algorithm	No	No	No	No	Yes
Cho & Wysk (1993)	Backpropagation NN; Simulator	Yes	No	No	Yes	No
Kim et al. (1998)	C4.5; Competitive NN	No	No	No	No	No
Lee et al. (1997)	C4.5; Genetic Algorithm	No	No	No	No	Yes
Li & She (1994)	Inductive Learning; Cluster Analysis	No	No	No	No	No
Rabelo & Alptekin (1989)	Expert System; Backpropagation NN	No	No	No	No	No
Wu & Wysk (1988)	Expert System; Inductive Learning; Simulator	Yes	No	No	Yes	Yes
Bowden & Bullington (1996)	Simulator; Genetic Algorithm	No	No	No	Yes	Yes
Quiroga & Rabelo (1995)	ID3; Fuzzy Logic; Backpropagation NN	No	No	No	No	No

that might in principle be the right one. In such cases, when “scheduling knowledge” decisions are not clear, incorporating a simulator would be very useful.

8. Refinement of the knowledge base. Once developed, the knowledge base is not static, so it would be interesting to establish a procedure that would automatically modify knowledge if important changes in the manufacturing system occur. The main aim of the refinement module is to discover deficiencies in the knowledge base and add training cases that cater to them. These deficiencies can occur in certain ranges of control attribute values. To solve this problem, such ranges have to be “covered” with new training cases, so that the new “scheduling knowledge” obtained is able to deal with these situations.

Table 3 provides a summary of the characteristics of each of the scheduling systems that have been reviewed above. Only the last five characteristics (of the eight listed above) are shown, as some of the systems include some of these

five characteristics and others include others. However, as none of the first three characteristics (comparison of the different methodologies for machine learning, use of CBR, and determination of the optimum number of training examples) are found in any of the systems, they do not appear in this table.

## 6. CONCLUSIONS

This paper provides a review of the literature on dynamic scheduling of FMSs using machine learning. A classification of general approaches to be found in the literature is first provided. Then, two ways of dynamically modifying dispatching rules in order to overcome their drawbacks when they are used statically are described. A review is then provided of the approaches available according to the machine learning algorithm that is used. Next, we indicate a number of limitations that would be desirable characteristics, but which are lacking in the approaches we have reviewed. Finally, the point is made that in future work it would be

interesting for a scheduling module to be designed that incorporates the eight characteristics that are listed, and for the effect of each of them on the performance of scheduling systems to be measured.

## REFERENCES

- Aytug, H., Bhattacharyya, S., Koehler, G.J., & Snowdon, J.L. (1994). A review of machine learning in scheduling. *IEEE Transactions on Engineering Management* 41(2), 165–171.
- Baker, K.R. (1984). Sequencing rules and due-date assignments in a job shop. *Management Science* 30(9), 1093–1103.
- Basnet, C., & Mize, J.H. (1994). Scheduling and control of flexible manufacturing systems: A critical review. *International Journal of Computer Integrated Manufacturing* 7(6), 340–355.
- Blackstone, J.H., Phillips, D.T., & Hogg, G.L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research* 20(1), 27–45.
- Bowden, R., & Bullington, S.F. (1996). Development of manufacturing control strategies using unsupervised machine learning. *IIE Transactions* 28, 319–331.
- Chandra, J., & Talavage, J. (1991). Intelligent dispatching for flexible manufacturing. *International Journal of Production Research* 29(11), 2259–2278.
- Chaturvedi, A.R., Hutchinson, G.K., & Nazareth, D.L. (1993). Supporting complex real-time decision making through machine learning. *Decision Support Systems* 10, 213–233.
- Chen, C.C., & Yih, Y. (1996). Identifying attributes for knowledge-based development in dynamic scheduling environments. *International Journal of Production Research* 34(6), 1739–1755.
- Chiu, C. (1994). *A learning-based methodology for dynamic scheduling in distributed manufacturing systems*. PhD Thesis. School of Industrial Engineering, Purdue University, West Lafayette, Indiana.
- Chiu, C., & Yih, Y. (1995). A learning-based methodology for dynamic scheduling in distributed manufacturing systems. *International Journal of Production Research* 33(11), 3217–3232.
- Cho, H., & Wysk, R.A. (1993). A robust adaptive scheduler for an intelligent workstation controller. *International Journal of Production Research* 31(4), 771–789.
- Choi, R.H., & Malstrom, E.M. (1988). Evaluation of traditional work scheduling rules in a flexible manufacturing system with a physical simulator. *Journal of Manufacturing Systems* 7(1), 33–45.
- De, S., & Lee, A. (1998). Towards a knowledge-based scheduling system for semiconductor testing. *International Journal of Production Research* 36(4), 1045–1073.
- Denzler, D.R., & Boe, W.J. (1987). Experimental investigation of flexible manufacturing system scheduling rules. *International Journal of Production Research* 25(7), 979–994.
- Dong, Y., & Kitaoka, M. (1994). Implementation of a case-based production scheduling system in C language. *Computers & Industrial Engineering* 27(7), 265–268.
- Egbelu, P.J., & Tanchoco, J.M.A. (1984). Characterization of automated guided vehicle dispatching rules. *International Journal of Production Research* 22(3), 359–374.
- Evert, B. (1980). *Cluster Analysis*. Heinemann, New York.
- Fox, M.S., & Smith, S.F. (1984). ISIS-A knowledge-based system for factory scheduling. *Expert Systems* 1(1), 25–49.
- Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.
- Gunasekaran, A., Martikainen, T., & Yli-Olli, P. (1993). Flexible manufacturing systems: An investigation for research and applications. *European Journal of Operational Research* 66, 1–26.
- Han, M., Na, Y.K., & Hogg, G.L. (1989). Real-time tool control and job dispatching in flexible manufacturing systems. *International Journal of Production Research* 27, 1257–1267.
- Harmonosky, C.M., & Robohn, S.F. (1991). Real-time scheduling in computer integrated manufacturing: A review of recent research. *International Journal of Computer Integrated Manufacturing* 4(6), 331–340.
- Henneke, M.J., & Choi, R.H. (1990). Evaluation of FMS parameters on overall system performance. *Computer Industrial Engineering* 18(1), 105–110.
- Hooke, R., & Jeeves, T.A. (1961). Direct search solution of numerical and statistical problems. *Journal of the Association of Computer Machines* 8, 212–229.
- Hutchison, J., Leong, K., Snyder, D., & Ward, F. (1989). Scheduling for random job shop flexible manufacturing systems. *Proc. of the Third ORSA/TIMS Conf. on Flexible Manufacturing Systems*, 161–166.
- Ishii, N., & Talavage, J. (1991). A transient-based real-time scheduling algorithm in FMS. *International Journal of Production Research* 29(12), 2501–2520.
- Jain, A.S., & Meeran, S. (1998). Job-shop scheduling using neural networks. *International Journal of Production Research* 36(5), 1249–1272.
- Jeong, K.-C., & Kim, Y.-D. (1998). A real-time scheduling mechanism for a flexible manufacturing system: Using simulation and dispatching rules. *International Journal of Production Research* 36(9), 2609–2626.
- Kanet, J.J., & Adelsberger, H.H. (1987). Expert systems in production scheduling. *European Journal of Operational Research* 29, 51–59.
- Kim, Y.-D. (1990). A comparison of dispatching rules for job shops with multiple identical jobs and alternative routings. *International Journal of Production Research* 28(5), 953–962.
- Kim, M.H., & Kim, Y.-D. (1994). Simulation-based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing Systems* 13(2), 85–93.
- Kim, C.-O., Min, H.-S., & Yih, Y. (1998). Integration of inductive learning and neural networks for multi-objective FMS scheduling. *International Journal of Production Research* 36(9), 2497–2509.
- Kimemia, J., & Gershwin, S.B. (1985). Flow optimization in flexible manufacturing systems. *International Journal of Production Research* 23, 81–96.
- Kiran, A., & Alptekin, S. (1989). A tardiness heuristic for scheduling flexible manufacturing systems. *Fifteenth Conf. on Production Research and Technology: Advances in Manufacturing Systems Integration and Processes*, 559–564.
- Kouvelis, P. (1992). Design and planning problems in flexible manufacturing systems: A critical review. *Journal of Intelligent Manufacturing* 3, 75–99.
- Kusiak, A., & Chen, M. (1988). Expert systems for planning and scheduling manufacturing systems. *European Journal of Operational Research* 34, 113–130.
- Lashkari, R.S., Dutta, S.P., & Padhye, A.M. (1987). A new formulation of operation allocation problem in flexible manufacturing systems: Mathematical modelling and computational experience. *International Journal of Production Research* 25, 1267–1283.
- Lee, C.-Y., Piramuthu, S., & Tsai, Y.-K. (1997). Job shop scheduling with a genetic algorithm and machine learning. *International Journal of Production Research* 35(4), 1171–1191.
- Li, D.-C., & She, I.-S. (1994). Using unsupervised learning technologies to induce scheduling knowledge for FMSs. *International Journal of Production Research* 32(9), 2187–2199.
- Maimon, O.Z. (1987). Real-time operational control of flexible manufacturing systems. *Journal of Manufacturing Systems* 6(2), 125–136.
- Maley, J.G., Ruiz-Mier, S., & Solberg, J.J. (1988). Dynamic control in automated manufacturing: A knowledge integrated approach. *International Journal of Production Research* 26(11), 1739–1748.
- Min, H.-S., Yih, Y., & Kim, C.-O. (1998). A competitive neural network approach to multi-objective FMS scheduling. *International Journal of Production Research* 36(7), 1749–1765.
- Minton, S. (1993). *Machine Learning Methods for Planning*. Morgan Kaufmann, San Mateo, California.
- Montazeri, M., & Van Wassenhove, L.N.V. (1990). Analysis of scheduling rules for an FMS. *International Journal of Production Research* 28(4), 785–802.
- Nakasuka, S., & Yoshida, T. (1992). Dynamic scheduling system utilizing machine learning as a knowledge acquisition tool. *International Journal of Production Research* 30(2), 411–431.
- Panwalkar, S.S., & Iskander, W. (1977). A survey of scheduling rules. *Operations Research* 23(5), 961–973.
- Pierreval, H., & Mebarki, N. (1997). Dynamic selection of dispatching rules for manufacturing system scheduling. *International Journal of Production Research* 35(6), 1575–1591.
- Pierreval, H., & Ralambondrainy, H. (1990). A simulation and learning technique for generating knowledge about manufacturing systems behaviour. *Journal of the Operational Research Society* 41(6), 461–474.

- Piramuthu, S., Raman, N., & Shaw, M.J. (1994). Learning-based scheduling in a flexible manufacturing flow line. *IEEE Transactions on Engineering Management* 41(2), 172–182.
- Piramuthu, S., Raman, N., Shaw, M.J., & Park, S. (1993). Integration of simulation modeling and inductive learning in an adaptive decision support system. *Decision Support Systems* 9, 127–142.
- Quiroga, L.A., & Rabelo, L.C. (1995). Learning from examples: A review of machine learning, neural networks and fuzzy logic paradigms. *Computers & Industrial Engineering* 29, 561–565.
- Rabelo, L.C., & Alptekin, S. (1989). Integrating scheduling and control functions in computer integrated manufacturing using artificial intelligence. *Computers & Industrial Engineering* 17, 101–106.
- Rachlin, J., Kasif, S., Salzberg, S., & Aha, D.W. (1994). Towards a better understanding of memory-based reasoning systems. *Proc. of the Eleventh International Conf. on Machine Learning*, 242–250.
- Ramasesh, R. (1990). Dynamic job shop scheduling: A survey of simulation studies. *OMEGA: The International Journal of Management Science* 18(1), 43–57.
- Russel, R.S., Dar-El, E.M., & Taylor, B.W. (1987). A comparative analysis of the COVERT job sequencing rule using various shop performance measures. *International Journal of Production Research* 25(10), 1523–1540.
- Sabuncuoglu, I., & Hommertzheim, D.L. (1992). Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. *International Journal of Production Research* 30(5), 1059–1079.
- Sarin, S.C., & Salgame, R.R. (1990). Development of a knowledge-based system for dynamic scheduling. *International Journal of Production Research* 28(8), 1499–1512.
- Shanker, K., & Rajamarthandan, S. (1989). Loading problem in FMS: Part movement minimization. *Proc. of the Third ORSA/TIMS Conf. on Flexible Manufacturing Systems*, 99–104.
- Shanker, K., & Tzen, Y.J. (1985). A loading and dispatching problem in a random flexible manufacturing system. *International Journal of Production Research* 23, 579–595.
- Shaw, M.J., Park, S., & Raman, N. (1992). Intelligent scheduling with machine learning capabilities: The induction of scheduling knowledge. *IIE Transactions* 24(2), 156–168.
- Shaw, M.J., & Whinston, A.B. (1989). An artificial intelligence approach to the scheduling of flexible manufacturing systems. *IIE Transactions* 21(2), 170–183.
- Shen, S., & Chang, Y.-L. (1988). Schedule generation in a flexible manufacturing system: A knowledge-based approach. *Decision Support Systems* 4, 157–166.
- Stecke, K.E. (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science* 29(3), 273–288.
- Stecke, K.E., & Solberg, J. (1981). Loading and control policies for a flexible manufacturing system. *International Journal of Production Research* 19(5), 481–490.
- Sun, Y.-L., & Yih, Y. (1996). An intelligent controller for manufacturing cells. *International Journal of Production Research* 34(8), 2353–2373.
- Tang, L.-L., Yih, Y., & Liu, C.-Y. (1993). A study on decision rules of a scheduling model in an FMS. *Computer in Industry* 22, 1–13.
- Thesen, A., & Lei, L. (1986). An expert system for scheduling robots in a flexible electroplating system with dynamically changing workloads. *Proc. of the Second ORSA/TIMS Conf. on Flexible Manufacturing Systems*, pp. 555–566. Elsevier Science Publishers, Amsterdam.
- Vepsalainen, A.P.J., & Morton, T.E. (1987). Priority rules for job shops with weighted tardiness costs. *Management Science* 33(8), 1035–1047.
- Wilson, J.M. (1989). An alternative formulation of the operation-allocation problem in flexible manufacturing systems. *International Journal of Production Research* 27, 1405–1412.
- Wu, S.-Y.D., & Wysk, R.A. (1988). Multi-pass expert control system—a control/scheduling structure for flexible manufacturing cells. *Journal of Manufacturing Systems* 7(2), 107–120.
- Wu, S.-Y.D., & Wysk, R.A. (1989). An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing. *International Journal of Production Research* 27(9), 1603–1623.
- Zweben, M., & Fox, M.S. (1994). *Intelligent Scheduling*. Morgan Kaufmann, San Mateo, California.

---

**Paolo Priore** majored in industrial engineering at University of Oviedo, Spain and graduated in 1991. He gained his Ph.D. from the University of Oviedo in 2001. He is currently an Assistant Professor at the School of Industrial Engineering at Oviedo University. His research interests include machine learning applications in production problems, simulation, and manufacturing. Dr. Priore is a member of EUROMA (European Operations Management Association).

**David de la Fuente** majored in industrial engineering at University of Madrid, Spain and graduated in 1983. He gained his Ph.D. from the University of Oviedo in 1987. Currently, he is a Professor at the School of Industrial Engineering at Oviedo University. His present research interests are in the areas of artificial intelligence, forecasting, and manufacturing. Dr. de la Fuente is a member of APICS.

**Alberto Gomez** graduated from the University of Oviedo, Spain in 1994, where he majored in the field of industrial engineering. He is currently an Assistant Professor at the School of Industrial Engineering at Oviedo University. His research interests include applications of genetic algorithms in production problems, and information systems.

**Javier Puente** graduated in industrial engineering in 1994 at University of Oviedo, Spain. He is currently an Assistant Professor at the School of Industrial Engineering at Oviedo University. His current research interests are in the areas of fuzzy logic in production problems, and total quality management.