8^{th} *International Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 - Hammamet - Tunisia*
"Evaluation and optimization of innovative production systems of goods and services"

DISCOVERING DISPATCHING RULES FOR JOB SHOP SCHEDULING PROBLEM THROUGH DATA MINING

Atif SHAHZAD

IRCCyN
1, rue de la Noë
BP 92 101 - 44321 Nantes CEDEX 03 - France
atif.shahzad@irccyn.ec-nantes.fr

Nasser MEBARKI

IRCCyN
1, rue de la Noë
BP 92 101 - 44321 Nantes CEDEX 03 - France
nasser.mebarki@univ-nantes.fr

ABSTRACT: *A data mining based approach to discover previously unknown priority dispatching rules for job shop scheduling problem is presented. This approach is based upon seeking the knowledge that is assumed to be embedded in the efficient solutions provided by the optimization module built using tabu search. The objective is to discover the scheduling concepts using data mining and hence to obtain a rule-set capable of approximating the efficient solutions in a dynamic job shop scheduling environment. A data mining based scheduling framework is presented.*

KEYWORDS: *Priority dispatching rules, job-shop scheduling, Data mining, Simulation*

1 INTRODUCTION

The job shop scheduling problem (JSSP) with multiple precedence constraints is an optimization problem composed of resources, operations, and constraints. A job j consists of a sequence of n_j operations with each operation i as part of exactly one job. Each operation is executed on a resource k, with starting time $S_{ij} \geq 0$ during a processing time $p_{ij} > 0$ with precedence constraints. A job j has a release date denoted by r_j, completion time denoted by C_j and due date denoted by d_j. A resource can execute only one operation at a time. We assume that any successive operations of the same job are going to be processed on different machines. It is desired to find a feasible schedule, $\{S_{ij}\}$ which optimizes a set of given performance measures.

Scheduling problems in which number of jobs and their ready times are known and fixed are referred as static environment problems in contrast to the dynamic environment problems in which jobs are continually revealed during the execution process (S.French, 1982, 1982). Dynamic scheduling provides a solution by the use of priority dispatching rules (pdrs), aimed at selecting the next job to process from jobs awaiting service in queues (Chong et al., 2003, 2003), (Vieira et al., 2003, 2003). These pdrs are very simple heuristics providing approximate solution and are frequently used to compare the performance on various criteria such as flow time, tardiness, productivity of machines and work in progress etc. Composite dispatching rules are found to be performing much better than simple dispatching rules (Nhu Binh & Joe Cing, 2005, 2005). The major drawbacks of pdrs include their performance-dependence on the state of the system and non- existence of any single rule, superior to all the others for all possible states the system might be in (Geiger et al., 2006, 2006). Meta-heuristics (e.g., simulated annealing, tabu search etc) , on the other hand, have an advantage over pdrs in terms of solution quality and robustness, however they are usually more difficult to implement and tune, and too complex to be used in a real time system. Robust and better-quality solutions provided by meta-heuristics contain useful knowledge about the problem domain and solution space explored. In this paper, we propose to exploit the advantages of these two approaches by combining them using data mining.

In this article, we propose an approach that seeks this scheduling knowledge through data mining module to identify a rule-set by exploring the patterns in the solution set obtained by an optimization module based on Tabu Search, a very efficient meta-heuristic for JSSP in particular. The rule-set approximates the output of the optimization module when incorporated in a simulation model of the system. C5.0 algorithm (Quinlan, 2003, 2003) is used as a data mining algorithm for the induction of rule-set.

2 LITERATURE REVIEW

A drawback in using pdrs is that their performance is dependent on the state of the system and no single rule exists that is superior to all the others for all

possible states the system might be in (Geiger et al., 2006, 2006). This drawback would be eliminated if the best rule for each particular situation could be used (Pierreval & Mebarki, 1997, 1997), (Priore et al., 2006, 2006). Various methods are studied for selection of pdrs suitable for particular situations. Learning methods are extensively used to select a pdr suitable to the situation, utilizing the comparative results on pdrs generated by steady-state simulations or online simulation runs.

Inductive learning in production scheduling has primarily been devoted to issues such as selecting the best dispatching rule using simulated data. (Aytug et al., 1994, 1994) has presented a comprehensive review of different machine learning techniques with emphasis on inductive learning methods applied in scheduling. (Pierreval & Ralambondrainy, 1988, 1988) used the induction algorithm GENREG proposed by (Ralambondrainy, 1988, 1988) to obtain a rule set with best mean tardiness as target concept on simulation data for flow shop environment. (Nakasuka & Yoshida, 1989, 1989) employed a learning algorithm capable of automatically generating new useful attributes for on-line rule selection in a production line. (Shaw et al., 1992, 1992) developed pattern-directed scheduling to monitor the scheduling activity for changes in manufacturing patterns-combinations of various parameters that together represent a given state of the system. (Piramuthu et al., 1994, 1994) proposed a mechanism based on same principle to select among a given set of heuristics by using C4.5 algorithm (Quinlan, 2003, 2003) for decision tree generation. They observed that pattern-directed scheduling based decision trees were not able to improve upon results significantly. (Priore et al., 2006, 2006) compared inductive learning based on C4.5 algorithm with other machine learning techniques for a selected flexible manufacturing system.

(Geiger et al., 2006, 2006) proposed a system, named SCRUPLES that combines an evolutionary learning mechanism with a simulation model to discover new priority dispatching rules. (Li & Olafsson, 2005, 2005) used decision-tree induction in his proposed approach to discover the key scheduling concepts by applying data mining techniques on historic scheduling data and to generate scheduling rules. (Dimopoulos & Zalzala, 2001, 2001) used Genetic Programming to build sequencing policies that combine known sequencing rules. They do not, however, use fundamental attributes to construct these policies. (Koonce & Tsai, 2000, 2000) applied data mining on solutions generated by a genetic algorithm(GA) based scheduling and developed a rule set approximating the GA scheduler. The Attribute Oriented Induction approach was used to characterize the relationship between the operations' sequences and their attributes. (Huyet, 2006, 2006) proposed an evolutionary optimization and data mining based approach to produce the knowledge of systems behavior in a simulated job shop based production process. (Nhu Binh & Joe Cing, 2005, 2005) made use of Genetic Programming for evolving effective composite dispatching rules for solving the Flexible JSSP with recirculation, with the objective of minimizing total tardiness.

3 GENERAL APPROACH FOUND IN LITERATURE REVIEW

The review of the literature reveals that the usual practice involves the implementation of a pre-defined set consisting of a number of candidate rules in a discrete event simulation model of the system under consideration, and comparing their performance using simulation experiments under varying values of system attributes characterizing the system dynamics. A set of best performing rules under variety of system states are taken as training examples to be input to the learning system. Intelligent decisions are then made in real time based on this knowledge. Generally, examination of the simulation results will suggest changes to the selected rule-set as well, requiring repetition of at least a subset of the simulation experiments. This, of course, assumes that all the dispatching rules are known in advance and that the performance of these rules can accurately be simulated. Exception to this usual approach include (Geiger et al., 2006, 2006), (Li & Olafsson, 2005, 2005),(Koonce & Tsai, 2000, 2000) and (Huyet, 2006, 2006).

4 DATA MINING BASED APPROACH TO JOB SHOP SCHEDULING

By knowing the significance and interrelationships among a set of system attributes, a user is equipped to design an effective scheduling rule, particularly for a given environment. In addition, the knowledge about a set of good schedules helps to either improve an existing scheduling rule or discover a new one.

(Li & Olafsson, 2005, 2005) shows how data mining on production data can be used to capture both explicit and implicit knowledge to discover new dispatching rules in a single machine scheduling problem with an implicit assumption that it is worthwhile to capture the current practices from historical data and use it as training data for a learning algorithm. The set of efficient solutions obtained through a metaheuristic approach can be an alternative to the historic data. We consider tabu search to obtain this training data, instead. Tabu search (TS) algorithms are among the most effective approaches for solving JSSP (Jain & Meeran, 1998, 1998) using a memory function to avoid being trapped in a local optimum (Zhang et al., 2008, 2008). However, neighborhood structures and move evaluation strategies play the central role in the

effectiveness and efficiency of the tabu search for the JSSP (Jain et al., 2000, 2000). A brief description of data mining is given before the discussion of proposed approach.

4.1 Data Mining

Data mining is an essential step in process of knowledge discovery from Data (KDD), however the two terms are often used interchangeably. Data mining is the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositoriesHan & Kamber, 2002 (2002). The data mining approach is particularly applicable for large, complex production environments, where the complexity makes it difficult to model the system explicitly.

From the viewpoint of our approach, data mining can specifically be considered as the analysis of a data set, referred as training data set in order to identify previously unknown and potentially useful hidden patterns and to discover relationships among the various elements of this data set. The aim is to classify the cases in another data set, referred as test data set, by mapping the newly discovered relationships on them. The discovery process can be termed as descriptive data mining, while the classification of test data set using the discovered relationships can be viewed as predictive data mining (Choudhary et al., 2009, 2009). A KDD process consists of an iterative sequence of Selection, Preprocessing,Data transformation, Data mining,Interpretation and Knowledge presentation (Fayyad et al., 1996, 1996).

We use decision tree based learning as the data mining step as it is simple to understand and interpret, requires little data preparation, able to handle both numerical and categorical data, uses a white box model, possible to validate a model using statistical tests, robust and performs well with large data in a short time. Induction of decision tree, or ID3 (Iterative Dichotomiser 3)(Quinlan, 1986, 1986) is one of the most powerful mining algorithm used in machine learning (Koonce & Tsai, 2000, 2000; Dudas et al., 2009, 2009; Priore et al., 2006, 2006). Revised versions of ID3 include GID3(Generalized ID3), ID4, ID5, C4.5 (Quinlan, 2003, 2003) and C 5.0 (Quinlan, 2003, 2003)(used for this study).

4.2 Proposed Approach

Construction of a proper training data set is a very crucial point in the entire KDD process. From the data mining perspective in JSSP, the target concept to be learned is to determine which job should be dispatched first within a set of schedulable jobs at a particular instant on the same machine. Extracting this knowledge from the training data set would allow us to dispatch the next job at any given time and thereafter to create dispatching lists for any set of jobs. In contrast to myopic nature of PDRs, metaheuristics such as tabu search can attack the problem more rigorously and intelligently due to relativly higher level of domain knowledge. As a consequence, the decisions made reflect the use of this useful knowledge. Given this target concept the propopsed framework (see Fig. 2)comprises following main phases:

1. A tabu search, (Nowicki & Smutnicki, 1996, 1996) based optimization module generates a set of efficient solutions for a set of problem instances. Since these efficient solutions are obtained through a series of some logical moves of the meta-heuristic, they have some general characteristics that may describe the relationship between operations and their sequential order in a particular solution. These characteristics are a form of scheduling knowledge, like dispatching rules. The aggregation of corresponding set of efficient solutions for each problem instance may then be used as the training data set for the data mining algorithm for discovery of scheduling knowledge.

2. Data preparation including aggregation, attribute construction, and attribute selection: Considerable amount of transformations are required before it is possible to mine any useful knowledge from the data obtained through the optimization module. A flat data file with columns representing the selected attributes of the data and each row representing a single piece of information independent(as much as possible) of the other rows: an example from which we can learn. We refer to each such example or a row in the file as an instance. There exist a strong relation among the sequencing of operations due to precedence constraints, however, considering only two (operations of the) jobs on the same machine among schedulable jobs (the predecessors of whom are already dispatched) at any instance for the comparison, reduces this dependency effect. Proper selection of attributes plays an important role to reduce this dependency as well. It is indeed unlikely that the attributes that are recorded as part of the available data are the attributes that are the most relevant or useful for data mining perocess. Thus, creation of new attributes must be considered. Attribute selection to eliminate certain redundant and irrelevant attributes is also critical to the effectiveness of the subsequent model induction. Both the creation of new attributes and selection of attributes are primarily linked with the objectives of the JSSP. Tardiness based objectives require different attributes to be taken into

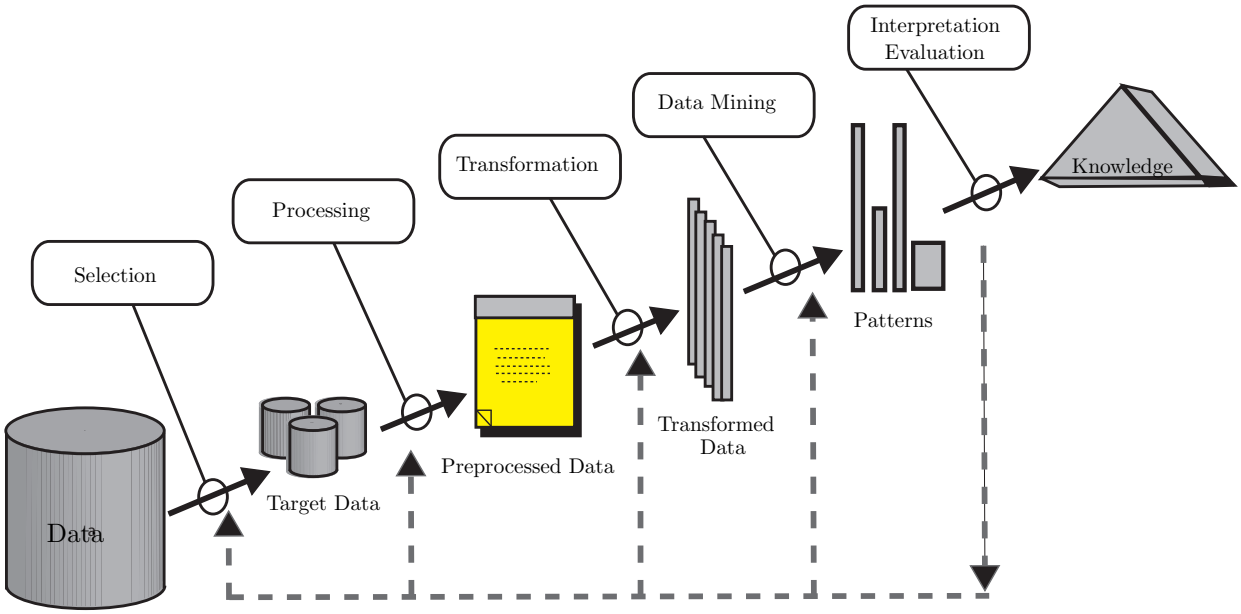


Figure 1: Knowledge Discovery Process

account while flowtime based objectives have different requirements. More detailed view of this is presented in next section.

3. Model induction and interpretation: After the data file has been constructed a learning algorithm is applied to infer a predictive model for the target concept. We focus on using decision trees and decision rules derived from such trees. The target class represents the relative order of any two jobs on a same machine. The decision tree obtained using the learning algorithm can be applied directly to the similar JSSP to validate the explored knowledge and as a predictive model to predict the target concept, which in our framework is a dispatching rule. The overall sequence of operations obtained by these rules is translated to a schedule using a schedule generator. As the sequences of operations on each machine are explicitly defined, the schedule, S_{ij} is indifferent of the approach used for building the schedule from the obtained sequence. Thus, the tree will, given any two jobs, predict which job should be dispatched first and can be thought of as a new, previously unknown, dispatching rule. In addition to the prediction, decision trees and decision rules reveal insightful structural knowledge that can be used to further enhance the scheduling decision.

Some aspects of this proposed framework require to be explored further for a better understanding and efficient manipulation of the entire process of knowledge extraction. We shall highlight some of these aspects in this section. Objectives of the scheduling problem play an important role in the process of selection and creation of attributes. For instance, flowtime based objectives consider different attributes whereas tardiness based objectives give importance to a different set of attributes, as mentioned earlier. Different feature selection algorithms may be employed for this purpose. For example, a trivial set of attributes that are found to be more relevant while considering makespan (i.e., $\max(S_{ij} + p_{ij})$), in deciding the execution order of a particular operation with respect to its competing operation include:

- Difference in processing time of the two operations to be compared, Δp
- Difference in remaining processing time for job, ΔRPT
- Earliest Possible Execution Time, ΔES
- Operation number
- Machine Load

It is generally required to discretize the attributes to obtain a more relevant set of rules for a variety of problem instances.

A scheduling problem with similar distribution of processing time, due date setting method and job routings is to be used as a test data set for the scheduling knowledge discovered.

5 EXPERIMENTAL SETTINGS

A set of 6×6 similarly sized instances of a static job shop problem with different seed values are used as test data set. All jobs are available simultaneously

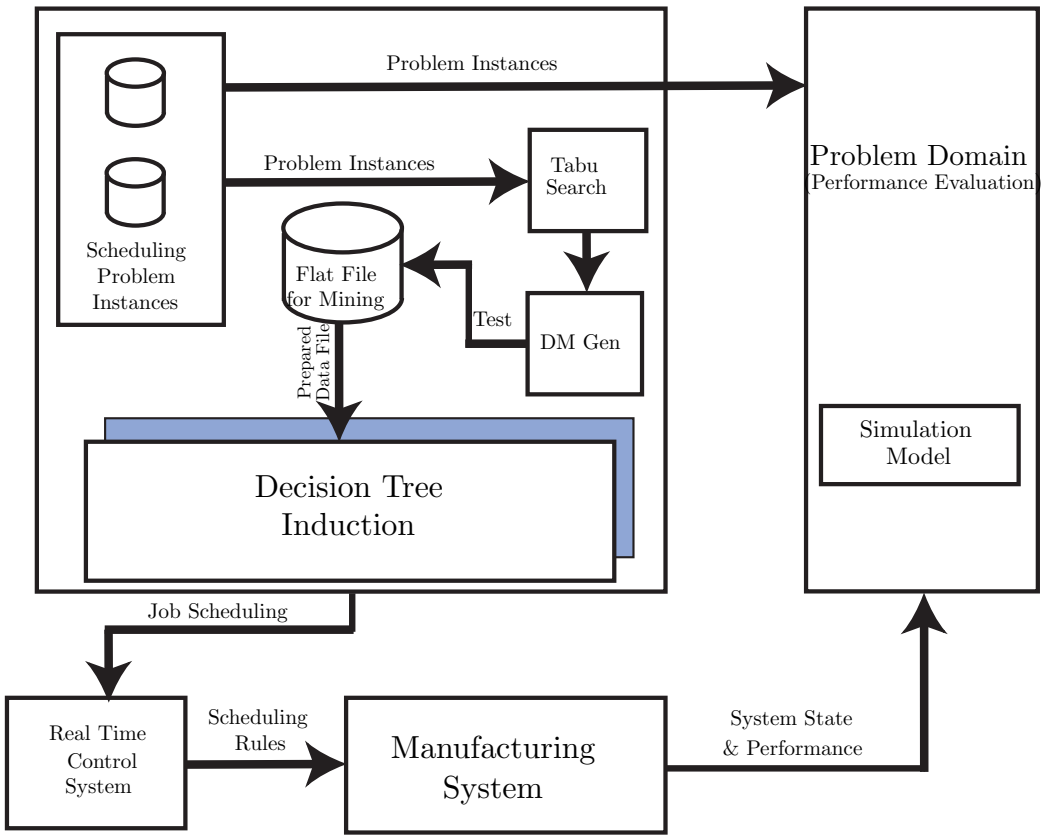


Figure 2: A General Framework

at time zero. Discrete uniform distribution between 1 and 10 is used to generate the operation processing times. The job due dates are determined using two parameters τ and ρ, where τ determines the expected number of tardy jobs (and hence the average tightness of the due dates) and ρ the due date range. Once these parameters have been specified, the job due dates are generated from the discrete uniform distribution

$$d_j = UNIFORM(\mu - \frac{\mu\rho}{2}, \mu + \frac{\mu\rho}{2})$$

where $\mu = (1-\tau)E[C_{max}]$ is the mean due date. $E[C_{max}]$ denotes the expected makespan for the problem instance and is calculated by estimating the total processing time required by all operations and dividing it by number of machines. Note that this assumes no idle time on machines, and hence will be an optimistic estimate of C_{max}. We consider $\tau = 0.3$ and $\rho = 0.5$. L_{max}(Maximum Lateness) is used as the scheduling objective.

6 COMPUTATIONAL RESULTS

Performance of the proposed system is measured in relative terms and is indicated by two measures namely η_1 and η_2. η_1 indicates performance of obtained rule-set and the set of pdrs relative to tabu search algorithm used and is given as,

$$\eta_1 = \frac{L_{max}(r \cup RS, I) - L_{max}(TS, I)}{L_{max}(r \cup RS, I)}$$

η_2 refers the performance of obtained rule-set relative to considered set of pdrs and is given as,

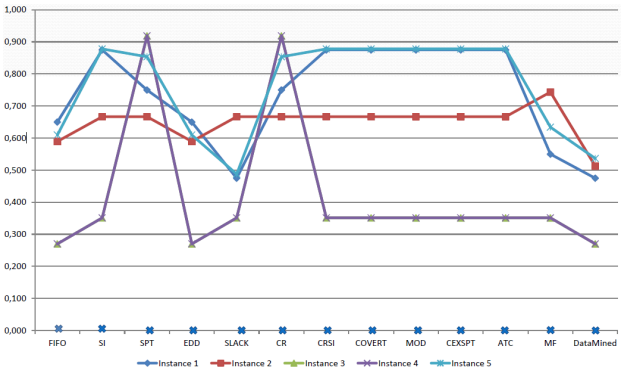
$$\eta_2 = \frac{L_{max}(r, I) - L_{max}(RS, I)}{L_{max}(r, I)}$$

where

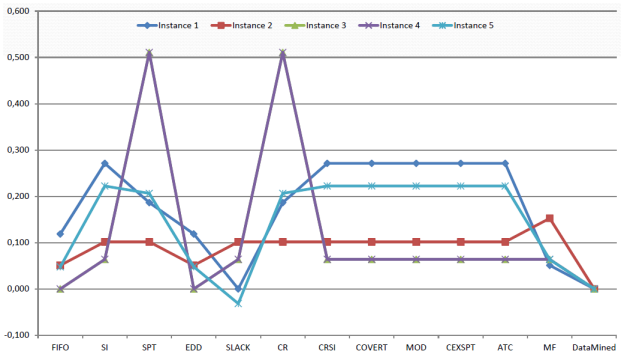
$$r \in \{FIFO, SI, SPT, EDD, SLACK, CR, CR/SI, COVERT, CEXSPT, ATC, MF\}$$

, TS: Tabu Search, RS: Rule-Set obtained and I: set of problem instances used.

It is observed that the results of mined rule-set is always superior or at least comparable to the best performing rule for the L_{max} measure. For the five instances, plot of η_2, shown in Fig. 3(b), reflects the fact that the rule-set is consistently a better performer among all the pdrs used with slack rule as the closest competitor. This is due to the nature of selected attributes used in the algorithm. There is a considerable room of improvement in performance of rule-set, as can be seen from plot of η_1 shown in Fig. 3(a), that may be obtained through better attribute selection.



(a) η_1



(b) η_2

Figure 3: System Performance

7 CONCLUSIONS AND PERSPECTIVES

It is not always possible to obtain or to implement the optimal solutions for a complex dynamic real-world sized JSSP due to constantly varying conditions. However, through this approach several alternative solutions could be proposed that are sufficiently efficient. This approach focuses on the identification of the critical parameters and states of a particular dynamic scheduling environment that contribute to the construction of some efficient solution. The proposed methodology is based upon the implicit assumption about the ability of tabu search to move intelligently in the solution space while providing the opportunity, at the same time, to learn the embedded knowledge about the thinking lines behind these intelligent moves. We believe that it it is possible to more effectively benefit benefit from it by making an analysis of long term memory of TS algorithm. It is also good to know how the obtained knowledge can be used in-process to reorient the tabu search for some large size instances of the problem. Feature selection for different objectives and their combinations has to be explored in much more detail to obtain compact and efficient rule set.

References

H. Aytug, et al. (1994). 'A review of machine learning in scheduling'. *IEEE Transactions on Engineering Management* **41**(2):165–171.

C. S. Chong, et al. (2003). 'Simulation Based Scheduling for Dynamic Discrete Manufacturing'. In *Proceedings of the 2003 Winter Simulation Conference*, pp. 1465–1473.

A. Choudhary, et al. (2009). 'Data mining in manufacturing: a review based on the kind of knowledge'. *Journal of Intelligent Manufacturing* **20**(5):501–521. 10.1007/s10845-008-0145-x.

C. Dimopoulos & A. M. S. Zalzala (2001). 'Investigating the use of genetic programming for a classic one-machine scheduling problem'. *Advances in engineering software* **32**:489–498.

C. Dudas, et al. (2009). 'Information Extraction from Solution Set of Simulation-based Multi-objective Optimisation using Data Mining'.

U. Fayyad, et al. (1996). 'From data mining to knowledge discovery: An overview' .

C. D. Geiger, et al. (2006). 'Rapid Modeling and Discoveriny of Priority Dispatching Rules: An Autnomous Learning Approach' **9**(1):7–34.

J. Han & M. Kamber (2002). *Data Mining Concepts and Techniques*. Diane Cerra, 2nd edn.

A. L. Huyet (2006). 'Optimization and analysis aid via data-mining for simulated production systems'. *European Journal of Operational Research* **173**(3):827–838. doi: DOI: 10.1016/j.ejor.2005.07.026.

A. S. Jain & S. Meeran (1998). 'A State-of-the-Art Review of Job-Shop Scheduling Techniques' pp. 1–48.

A. Jain, et al. (2000). 'New and stronger job-shop neighbourhoods: A focus on the method of Nowicki and Smutnicki (1996)'. *Journal of Heuristics* **6**(4):457–480.

D. A. Koonce & S.-C. Tsai (2000). 'Using data mining to find patterns in genetic algorithm solutions to a job shop schedule'. *Comput. Ind. Eng.* **38**(3):361–374. 361517.

X. Li & S. Olafsson (2005). 'Discovering Dispatching Rules using Data Mining'. *Journal of Scheduling* **8**(6):515–527.

S. Nakasuka & T. Yoshida (1989). 'New framework for dynamic scheduling of production systems'. In *Industrial Applications of Machine Intelligence and Vision, 1989., International Workshop on*, pp. 253–258.

H. Nhu Binh & T. Joe Cing (2005). 'Evolving dispatching rules for solving the flexible job-shop problem'. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3, pp. 2848–2855 Vol. 3.

E. Nowicki & C. Smutnicki (1996). 'A fast taboo search algorithm for the job shop problem'. *Manage. Sci.* **42**(6):797–813. 256054.

H. Pierreval & N. Mebarki (1997). 'Dynamic selection of dispatching rules for manufacturing system scheduling'. *International Journal of Production Research* **35**(6):1575–1591.

H. Pierreval & H. Ralambondrainy (1988). 'Generation of Knowledge About the Control of a Flow Shop using Data Analysis Oriented Learning Techniques and Simulation'. Tech. Rep. 897.

S. Piramuthu, et al. (1994). 'Learning-based scheduling in a flexible manufacturing flow line'. *IEEE Transactions on Engineering Management* **41**(2):172–182.

P. Priore, et al. (2006). 'A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems'. *Engineering Applications of Artificial Intelligence* **19**(3):247–255.

J. Quinlan (1986). 'Induction of decision trees'. *Machine learning* **1**(1):81–106.

J. Quinlan (2003). *C4. 5: programs for machine learning*. Morgan Kaufmann.

H. Ralambondrainy (1988). 'The algorithm GENREG for generating rules from symbolic or numerical data'. Tech. rep. INFO:INFO-OH 1988-10 RR-0910.

S.French (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop*. Ellis Horwood Ltd.

M. J. Shaw, et al. (1992). 'Intelligent scheduling with machine learning capabilities: The induction of scheduling knowledge'. *IIE Transactions* **24**(2):156–168.

G. E. Vieira, et al. (2003). 'Rescheduling manufacturing systems: A framework of strategies, policies, and methods'. *Journal of Scheduling* **6**(1):39–62.

C. Zhang, et al. (2008). 'A very fast TS/SA algorithm for the job shop scheduling problem'. *Computers and Operations Research* **35**(1):282–294.