



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY  
Volume 11 Issue 7 Version 1.0 May 2011  
Type: Double Blind Peer Reviewed International Research Journal  
Publisher: Global Journals Inc. (USA)  
ISSN: 0975-4172 & Print ISSN: 0975-4350

## Approach to Job-Shop Scheduling Problem Using Rule Extraction Neural Network Model

By Mahmood Al Bashir, Md. Zahidul Islam, Dr. A. K. M. Masud  
*Bangladesh University*

*Abstract-* This thesis focuses on the development of a rule-based scheduler, based on production rules derived from an artificial neural network performing job shop scheduling. This study constructs a hybrid intelligent model utilizing genetic algorithms for optimization and neural networks as learning tools. Genetic algorithms are used for obtaining optimal schedules and the neural network is trained on these schedules. Knowledge is extracted from the trained network. The performance of this extracted rule set is analyzed in scheduling a test set of 3x3 scheduling instances. The capability of the rule-based scheduler in providing near optimal solutions is also discussed in this thesis.

*GJCST Classification: 1.2.6*



*Strictly as per the compliance and regulations of:*



# Approach to Job-Shop Scheduling Problem Using Rule Extraction Neural Network Model

Mahmood Al Bashir, Md. Zahidul Islam, Dr. A. K. M. Masud

**Abstract**—This thesis focuses on the development of a rule-based scheduler, based on production rules derived from an artificial neural network performing job shop scheduling. This study constructs a hybrid intelligent model utilizing genetic algorithms for optimization and neural networks as learning tools. Genetic algorithms are used for obtaining optimal schedules and the neural network is trained on these schedules. Knowledge is extracted from the trained network. The performance of this extracted rule set is analyzed in scheduling a test set of 3x3 scheduling instances. The capability of the rule-based scheduler in providing near optimal solutions is also discussed in this thesis.

## I. INTRODUCTION

Scheduling requires the arrangement of activities under constraints to meet a specific objective. A complex decision making activity it is, because of different conflicting goals, precise or limited resources and the difficulty in accurately modeling real world scenarios.

In today's highly competitive manufacturing environment, there is a distinct need for an integrated global approach towards production planning and control. The planning functions include demand forecasting, capacity and materials planning, process planning and operation scheduling. Scheduling theory is concerned with the mathematical formulation and study of various scheduling models and development of associated solution methodologies. The deterministic job shop scheduling problem (JSSP) consists of a finite set of jobs to be processed on a finite set of machines. The basic entity in the scheduling process is an operation, which refers to the processing of a particular job step on a specified machine. Various performance measures are used to evaluate the optimality of schedules ranging from minimization of makespan, tardiness and process cost to maximization of throughput and optimum resource utilization. JSSP is a constrained optimization problem (COP), where the precedence constraints on the problem are given by a predetermined order of operations for each job; and capacity or disjunctive constraints require that each operation be processed by only one machine at any given time. A schedule is the feasible resolution of the precedence and capacity constraints in the COP [1].

*About*— Department of Industrial and Production Engineering (IPE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh

The current research uses Artificial Neural Networks (ANNs) as the machine learning tool of choice to study the scheduling process. ANNs are being recognized as a powerful and general technique for machine learning because of their non-linear modeling abilities. Further, their distributed architecture is more robust in handling the noise-ridden data. The hypothesis or model learned by the neural network is not explicitly stated, but is implicitly enumerated in the network architecture. However, ANNs can be made to yield comprehensible models by using rule extraction procedures. This thesis has three major objectives:

- To train an ANN on the schedules generated by a GA, to predict the priority of an operation in a schedule based on the job attributes.
- To capture the embedded knowledge by extracting symbolic rules and decision trees by using appropriate ANN rule extraction algorithms.
- A comparative evaluation of the predictive accuracy of the extracted rule set, the trained ANN, GA scheduler and other machine learning algorithms.

## II. APPROACH

The deterministic job shop scheduling problem (JSSP) is one of the classical problems in scheduling literature. JSSP consists of a finite set of  $n$  jobs to be processed on a finite set of  $m$  machines and is denoted as an  $n \times m$  problem. The routing of a job is a predetermined sequence of operations. Each operation is processed on a specified machine and has a predetermined processing time. The job routings and the associated processing times are given by a definite process plan. JSSP is a constrained optimization problem (COP) where the precedence constraints on the problem are given by the job routings. Capacity or disjunctive constraints require that each operation be processed by only one machine at any given time. Other assumptions include the following:

- Machine repetitions by a job are not allowed.
- Machine absences are not allowed (i.e., each job is processed on every machine).
- Uninterrupted processing of operations without preemption.
- No machine breakdowns throughout the scheduling process.

- Transportation time between machines is zero.
- The job shop is static and deterministic in nature i.e., there is no randomness involved in determining all the necessary parameters for defining the job shop problem.

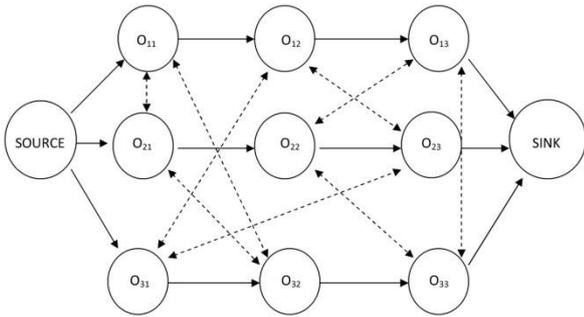


Figure1: The disjunctive graph representation of the 3 x 3 problem. Adapted from Yamada and Nakano [2]

In the above figure, the conjunctive constraints are given by complete arrows and the dashed arrows indicate the disjunctive constraints. Two fictitious nodes, source and sink nodes are added to the graph to represent the starting and ending operations. Panwalker and Iskander [3] provide a comprehensive survey of scheduling heuristics. The main drawback in using these elementary rules is that different rules perform best in different scenarios and no single rule dominates the rest across all scenarios. To improve performance, probabilistic combinations of the elementary priority rules are often employed for determining priority. Blackstone *et al.* [4] provide a detailed comparison of several elementary dispatching rules and their combinations. Lawrence [5] compares the performance of ten individual priority dispatch rules with a randomized combination of these rules. Superior results were delivered by the combination method, but it required substantially more computing time. Kaschel *et al.* [6] provide empirical results of the performance of priority rules in scheduling several benchmark instances. The authors compare single priority rules, simple combinations of them and combinations of priority rules by the Analytic Hierarchy Process method. AHP is a statistical decision making tool capable of generating weighted combinations of priority rules and provided the best results in this study. Most symbolic Artificial Intelligence (AI) approaches cast the JSSP as a constraint satisfaction problem. A CSP specifies a set of decisions to be made and a set of constraints to determine the validity of such decisions. The general procedure to solve a CSP is to reduce the search space by utilizing a constructive search strategy. Such a strategy incrementally builds a solution by assigning values to variables and checking for constraint violations. If any violations are found, a backtracking strategy is employed to undo previous variable

assignments. The procedure is repeated with a fresh set of variable assignments. The Intelligent Scheduling and Information System constructed by Fox [7] is a good example of the AI scheduling system. There are many variations of the generic constraint satisfaction procedure. Fox and Sadeh [8] provide a comparative summary of a variety of constraint satisfaction approaches applied to a set of benchmark scheduling instances.

Neural network scheduling systems offer an alternate AI-based scheduling paradigm. Cheung [9] provides a comprehensive survey of the main neural network architectures used in scheduling. These are: searching network (Hopfield net), probabilistic network (Boltzmann machine), error-correcting network (multilayer perceptron), competing network and self-organizing network. Jain and Meeran [10] also provide an investigation and review of the application of neural networks in JSSP.

### III. METHODOLOGY

The knowledge base for the learning task was provided by the genetic algorithm's solution to the job shop problem. For this purpose, a well-known 3x3 problem instance, ft03 devised by has been chosen as the benchmark problem. This test instance has three jobs, each with three operations to be scheduled on three machines and has a known optimum makespan of 11 units. The data for the instance is shown in Table 1 using the following structure: machine, processing time.

Table 1 The ft03 instance

Job	Operation		
	1	2	3
1	1,1	2,4	3,8
2	2,7	3,4	1,11
3	1,12	2,5	3,2

The schedules obtained by the GA contain valuable information relevant to the scheduling process. The learning task was to predict the position of an operation in the sequence, based on its features or attributes. Based on a study of operation attributes commonly used in priority dispatch rules, the following attributes have been identified as input features: operation, process time, remaining time and machine load. These input features have been clustered into different classes using the concept hierarchy for 6 x 6 job shop problems developed by Koonce and Tsai [11]. *Operation:* Each job has three operations that must be processed in a given sequence. The *Operation* feature identifies the sequence number of the operation ranging between 1 and 3. This feature has been clustered into three classes as: *First, Middle* and *Last*. *ProcessTime* and *RemainingTime:* The *ProcessTime* feature represents the processing time for

the operation. The *RemainingTime* feature denotes the sum of processing times for the remaining operations of that job and provides a measure of the work remaining to be done for completion of the job. For the benchmark ft03 instance, the processing times ranged from 1 to 4 units, while the remaining times ranged from 0 to 18 units. Based on the data, three classes (clusters) for these features were identified as follows. The ranges were split into three equal intervals. The first interval was classified as *Short*, second as *Medium*, and last interval was labeled as *Long*. Table 2 shows the classification of the *ProcessTime* and *RemainingTime* for these intervals.

Table 2 ProcessTime, RemainingTime feature classes

Attributes	Short	Middle	Long
ProcessTime	[1, 4]	[5, 8]	[9, 12]
RemainingTime	[0, 6]	[7, 12]	[13, 18]

*Machine Load*: The *Machine Load* feature determines the machine loading and was clustered into two classes: *Light* and *Heavy*. This feature represents the capacity or utilization of machines in units of time and Table 3 shows the classification of machine loading for the ft03 instance.

Table 3 MachineLoad feature classification

Attributes	Light	Heavy
Time	Less than 7	Greater than 7

The machine processing times range between 1 and 12 units, with an average of 6.5 units. All the machines having processing times less than 7 units were classified as *Light*, while those having processing times greater than 7 units were labeled as *Heavy*.

*Priority*: The target concept to be learned was the priority or position in the sequence. Since an operation can be positioned in any one of the 9 locations available in the sequence, it may be difficult to discover an exact relationship between the input features and the position. However, if the problem was modified to predict a range of locations for the operation, the learning task becomes easier. The target feature priority, thus determines the range of positions in the sequence where the operation can be inserted. The possible range of positions have been split into 4 classes and assigned class labels as shown in Table 4.

Table 4 Assignment of class labels to target feature

Range of Position	Priority
0 – 1	One
1 – 3	Two
4 – 6	Three
7 – 9	Four

There are three aspects related to development of a neural network model. The first is the choice of the training, cross-validation (CV) and testing data sets and their sizes, the second is the selection of suitable architecture, training algorithm and learning constants, and the third is the determination of the termination criteria. Unfortunately, there are no definitive heuristics or formulae to determine these parameters. Considerable experimentation was necessary to achieve a good network model of the data. The software NeuroShell Predictor developed by Ward System Group, Incorporated was used for development and testing of the neural network model.

*Training, Cross-validation and Test Datasets*: The 144 optimal schedules obtained by the GA represent a total number of 108 operations (12 schedules x 9 operations/schedule). Assignment of input features and target classes was done for each operation according to the classification scheme described in the previous subsection. Sample data for the classification task is shown in Table 5.

Table 5 Sample data for the classification task

Pattern_ID	Operation	Process Time	Remaining Time	Machine Load	Priority
1	First	Short	Short	Low	1
2	Mid	Mid	Mid	Low	2
.	.	.	.	.	.
.	.	.	.	.	.
1,295	Last	Mid	Mid	High	3
1,296	Last	Long	Long	High	4

The entire data set included 24 distinct input patterns with different target feature values (priority), constituting a total of 1,296 patterns (exemplars). This classification data set was split into training, cross validation and testing data sets with 70%, 15% and 15% memberships.

The neural network can be considered an implicit model of the training data. The goal of the rule extraction algorithms is to translate this implicit model into explicit symbolic form. In this work, the extracted knowledge is captured in two symbolic representations: decision trees and propositional rules. The possible rule space for these procedures is derived in the following way. The number of classes in the input features of the classification problem (Operation, ProcessingTime, RemainingTime and the MachineLoad features) are three, three, three and two respectively. Hence, the number of possible rule antecedent combinations (patterns) in the rule space is  $3 \times 3 \times 3 \times 2 = 54$ .

In this rule set, referred to as NN-Rule set, the keyword "Any" denotes all possible values of an attribute. The first rule in NN-Rule set implies that the *First* operation for a job with a *Short* processing time, a *Short* remaining time and processed on a machine of

any load class (i.e., *Light* or *Heavy*) has an associated priority of *one* (highest priority) for scheduling in the sequence.

A candidate rule is verified based on the convergence of the forward and backward passes. A run constitutes one forward and one backward pass. The procedure terminates when the difference between validity intervals between two consecutive runs becomes less than a predefined tolerance (a small threshold). Generally, the number of runs required for convergence can be considered a function of the specified validity intervals and the parameters of the neural network (weights and biases). A contradiction in the above procedure implies that the candidate rule incorrectly describes the behavior of the neural network.

Such a rule is expunged from the candidate rule set and the procedure is repeated with the other candidate rules. This process continues until the candidate rule set is exhausted. A tolerance of 0.01 was chosen for termination. All the 36 rules in the NN-Rule set were verified by this procedure. Among the 1296 data, 1000 data were used to train the network and the rest were used to test the network. A total of 80 hidden neurons were trained with a optimal of 16 hidden neurons. The network performance was at an optimum rate of 0.8102. The lack of the training can be attributed to the amount of data used in the training procedure. Had more data were used, the best network performance could have been better.

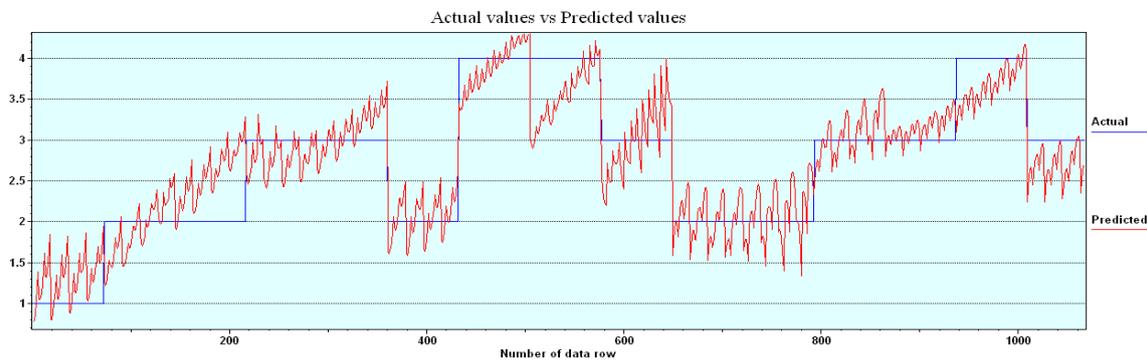


Figure 2: Actual vs. Predicted value graph in training phase (Enlarged in appendix C)

The efficacy of the rule extraction task was evaluated along the following dimensions:

- **Comprehensibility and Expressive Power:** The propositional rules in the NN-Rule set developed by the rule extraction procedures used easily understood feature and class labels for describing the extracted knowledge. The antecedent of each rule in the rule set was a simple conjunction of the input features. The number of input features in the antecedent of a rule provides an indirect measure of the comprehensibility and expressive power of the rule set. Table 5.3 shows the rules in the NN-Rule set (shown on the right) segregated according to the number of features in their respective antecedents (shown on the left). Approximately, a third of the rules had fewer than four antecedents (maximum) increasing the comprehensibility of the developed rule set.

- **Accuracy and Fidelity:** The rule set accurately mimicked the behavior of the trained neural network in classifying all the patterns in the rule space. Hence, the fidelity of the extraction process was maximum.

To schedule the 3 x 3 benchmark instance (ft03), a priority index was assigned to each of the 36 operations. This was accomplished by matching the features of the operation with the antecedent of the induced rules in the NN-Rule set. The consequent of the matched rule represented the priority index of that operation. This algorithm chooses from among the available operations based on the priority index. Also, the operations were locally left-shifted to improve the makespan of the generated schedule. The Gantt chart was used as a tool for visualizing the developed schedules. A similar procedure was utilized for scheduling the problem with the ID3-Rule set and the Shortest Processing Time (SPT) heuristic.

After the completion of the training phase the rest of the data were used to test the neural network to find the optimum level of usage it can offer. With the rest 296 data the network was run to test the difference between the actual and predicted value. An average error of .03182 was originated which yielded the network to be efficient enough to be used comprehensively. The

Table 6 Number of features in the rule antecedent for the NN-Rule set

Number of features in the rule antecedent	Number of rules (Total 36)
1	None
2	2
3	11
4	23

data obtained from a manufacturing facility in Bangladesh is also decided to be tested in this network with a view to finding the average error the network shows and the stability of the network.

#### IV. CONCLUSION

This paper presents a novel knowledge-based approach for the job shop scheduling problem by utilizing the various constituents of the soft computing paradigm. The ability of a genetic algorithm (GA) to provide multiple optimal solutions was exploited to generate a knowledge base of good solutions. A neural network was successfully trained on this knowledge base. Then, rule extraction algorithms were employed to induce decision tree and propositional rule representations describing the behavior of the trained neural network. The rule extraction task was successful in generating a rule set which completely and accurately mimicked the behavior of the trained neural network. The scheduler developed from this rule set can be utilized to schedule any 3 x 3 job shop scenario. Also, the developed system provides knowledge in the form of comprehensible rules which can effectively aid a human in the scheduling task.

A test problem set consisting of 10 randomly generated 3 x 3 scenarios was used to evaluate the performance of the developed rule-based scheduler. The makespans produced by the GA were considered to be the known optimal solutions for these scenarios. The rule-based scheduler had a deviation of 4.6 time units (8.4 %) from the optimum (i.e., average makespan of the GA) on the test problem set. Also, the rule-based scheduler performed better than the Shortest Processing Time (SPT) heuristic in all ten cases. Though the rule-based scheduler could not match the performance of the genetic algorithm, it is computationally less intensive than the GA and offers a more comprehensible scheduling approach. It also provides an attractive alternative to simple heuristics like SPT for scheduling 6 x 6 job shop problems.

A comparative evaluation of the rule-based scheduler with other schedulers developed from different machine learning methodologies was also undertaken. Two schedulers developed by other researchers using the Attribute-Oriented Induction (AOI) data mining methodology and another scheduler based on the ID3 decision tree induction algorithm were used for comparison. Among these schedulers, the rule-based scheduler developed in the current work had the closest average makespan to that of the genetic algorithm. However, statistical analysis revealed no significant differences in the performance of these schedulers on the test problem set.

The similar performance of the current approach compared to the AOI data mining methodology proves the feasibility of neural network

based data mining. Unlike the rule set derived in this work, those induced from AOI and ID3 methods were insufficient to describe any randomly generated 3 x 3 scenario. Also, the decision tree induction algorithm utilized for knowledge extraction from the neural network is similar to the ID3 algorithm. The deviation in their performance of the rule-based scheduler and the ID3-based scheduler is mainly attributable to the robustness of the neural networks in handling noisy data sets.

In summary, this research was able to successfully develop a rule-based scheduler, which provides a close approximation to the performance of a GA scheduler for the 3 x 3 job shop scheduling problems.

This research focused primarily on deriving production rules from a neural network performing job shop scheduling. There is a definite scope for improvement in the current research along the following directions.

#### *Use of multiple data sets*

The knowledge base for training the neural network in the current approach was derived from solutions to a single benchmark 3 x 3 problem. This knowledge base can be augmented with near-optimal solutions to randomly generated 3 x 3 scenarios provided by a GA. This can lead to an improvement in the generalization capabilities of the trained neural network.

#### V. ACKNOWLEDGEMENT

The authors are heartily thankful to their supervisor, Professor Dr. A. K. M. Masud, Professor and Head, Department of Industrial and Production Engineering, Bangladesh University of Engineering and Technology, whose spirit to work, guidance and support from the initial to the final level enabled them to develop an understanding of the subject. Above all and the most needed, he provided them unflinching encouragement and support in every possible ways.

The authors express their profound thanks and gratefulness to ShuvaGhosh, Assistant Professor, Department of Industrial and Production Engineering, Bangladesh University of Engineering and Technology for his advice, supervision, and crucial contribution, which made the thesis possible to initiate.

Finally, the authors would like to thank everybody who were important and helped them out in every process to the successful realization of thesis.

#### REFERENCES RÉFÉRENCES REFERENCIAS

1. Baker, K. (1974). Introduction to sequencing and scheduling. New York, NY: John Wiley & Sons, Inc.
2. Yamada, T., & Nakano, R. (1995). Job shop scheduling by simulated annealing combined with deterministic local search. *Metaheuristics*

- International Conference. Hilton, Breckenridge, Colorado, USA, 344-349.
3. Panwalker, S. S., & Iskander, W. (1977). A survey of scheduling rules. *Operations Research*, 25, 45-61.
  4. Blackstone Jr., 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, 27-45.
  5. Lawrence, S. (1984). Supplement to resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques. Graduate School of Industrial Administration. Carnegie-Mellon University, Pittsburgh, USA.
  6. Käsichel, J., Teich, T., Köbernik, G., & Meier, B. (1999). Algorithms for the job shop scheduling problem: A comparison of different methods. *European Symposium on Intelligent Techniques*. Greece, June 3-4.
  7. Fox, M. S. (1987). Constraint-directed search: A case study of job-shop scheduling. *Research Notes in Artificial Intelligence*. London: Pitman Publishing.
  8. Fox, M. S., & Sadeh, N. (1990). Why is scheduling difficult? A CSP perspective. In Aiello, L. (ed), *ECAI-90 Proceedings of the 9th European Conference on Artificial Intelligence*. Stockholm, Sweden. August 6-10, 754-767.
  9. Cheung, J. Y. (1994). Scheduling. In Dagli, C. H. (ed), *Artificial Neural Networks for Intelligent Manufacturing*. London: Chapman and Hall, Chapter 8, 159-193.
  10. Jain, A. S., & Meeran, S. (1998). Job-shop scheduling using neural networks. *International Journal of Production Research*, 36(5), 1249-1272.
  11. Koonce, D. A., & Tsai, S. C. (2000). Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. *Computers & Industrial Engineering*, 38 (3), 361-374.