# Training a Neural Network by Simulation for Dispatching Problems

Henri Pierreval

California State University, Chico
The McLeod Institute of Simulation Sciences
Chico, CA-95929-0410, USA.
henri@ecst.csuchico.edu

Université Claude Bernard, LISPI,
43 Bd du 11 Novembre, Bat. 710,
69622 Villeurbanne Cedex, France.
henri@lispsun1.univ-lyon1.fr

**Abstract:**

*Dispatching rules are often used in the dynamic scheduling of such systems as job shops and flow shops. Numerous rules exist, which perform differently on performance criteria. Rather than comparing and selecting a combination of rules using simulation, we suggest the use of a neural network trained by simulation. The neural network is faster than the simulation program and less memory consuming, so that it can be used either as a scheduling support tool or included in production management software. The building of such a neural network is illustrated through an example of a simplified flow shop.*

## 1. Introduction

The on-line scheduling of numerous manufacturing systems is based on dispatching rules (DRs). These rules, sometimes called scheduling rules [16], have been widely investigated. They are defined as: "rules used to select the next job to process from jobs awaiting service" [4]. DRs can be very simple or extremely complex. Examples of simple DRs are "select a job at random", or "select the job that has been waiting longest". A complex rule might be one that selects the job with the "shortest due date whose customer's inventory is less than a specified amount". Numerous DRs exist. The most well known include: SPT (Shortest Processing Time served first), LPT (Longest Processing Time served first), EDD (Earliest Due Date served first), FIQ (First In Queue served first) and FAS (First to Arrive in the System served first).

The performance of DRs has been widely studied using simulation (see for example: [11], [16], [4], [12], and [15]). Besides some general and common conclusions, all these studies stated that there is no DR which is globally better than the others. Certain ones, such as SPT, perform well on certain performance measures, such as the mean flowtime (i. e. the time spent by jobs in the system), but obtain poor results on others, such as the maximum lateness of jobs. The performance of DRs generally depends on the configuration of the shop (e.g. number of machines and size of the buffers) and on the production to be carried out (e.g. parts routing, number of operations, and operating times). These findings indicate that when a given system has to be scheduled, it is necessary to perform specific simulation experiments to choose the most suited DR. Moreover, several studies conclude that a combination of dispatching rules can perform better than applying the same one to each work center. Hence, the problem to address is to decide among the relevant rules, which combination is the most suited to the scheduling case. This problem occurs in the dynamic scheduling of such manufacturing systems as job shops. It also occurs if one wants to allow production management software the capability to automatically select appropriated DRs in order to establish priorities among concurrent jobs, according to the operating conditions of the shop floor.

The aim of this paper is to discuss the contribution of neural networks (NNs) and simulation to this selection problem. First, existing approaches based on simulation are introduced. Then, we suggest to use a NN that would automatically make a decision. We show how it is possible to train this NN by simulation. A simplified example, based on the dynamic scheduling of a flow shop illustrates the method.

## 2. Approaches based on simulation experiments

Due to the restricted assumptions of available mathematical methods, simulation remains the privileged technique to study the performance of DRs. A model describing the shop-floor and its configuration (e. g. number of machines available, shifts, mean time between breakdowns and quantity of raw material available) is developed. The production program (expressed in such

0-8186-2615-1/92 $3.00 © 1992 IEEE

terms as number of jobs, part routings, operating times and resources needed) is simulated with different combinations of DRs. The best combination, i. e. the one that obtains the best result on a given performance criteria (e. g. flowtime and tardiness of jobs) is selected and used in the factory. If major disturbances such as breakdowns or urgent orders occur, new simulations can be performed to select the new best suited combination. I has been suggested to simulate the flexible manufacturing system every $\Delta t$, to change the DRs when necessary [25].

Unfortunately, the approaches based on simulation generally require large computing times. If random variables are used in the model, long runs or several replications may be necessary to evaluate one combination of DRs. Moreover, simulation experiments have to be performed several times. In effect, each combination of DRs has to be simulated for each new scheduling problem. There is a new scheduling problem each time the production to be carried out is different (e. g. longest or shortest operating times) and each time the configuration of the shop floor changes (e. g. more or less operators and different shifts). A too large of an amount of computing time can proscribe the use of simulation as an aid for scheduling.

Furthermore, simulation programs are often developed using simulation packages such as Slam-II, Siman, GPSS-V and Simula, which require a large amount of memory. This may be an inconvenience for their inclusion in a production management system.

Due to these problems, several approaches have been proposed to provide assistance in the selection of DRs. They use simulation results and are based either on numerical/symbolic learning, or on multivariate statistical techniques.

(1) Numerical/Symbolic learning.

The use of numerical/symbolic algorithms has been discussed in [18], [26], [17], and [20]. In this approach simulation is used to provide a large number of observations related to the behavior of the system when different combinations of DRs are applied. These observations are used to build a training set from which a learning algorithm will learn, so as to generate production rules. These rules contain the knowledge needed to select among the DRs. A typical example of a production rule is:

if <conditions about the configuration of the system>
and <performance criterion of interest>
then <combination of DRs to use>.

(2) Statistical approaches.

Multivariate statistical techniques, such as discriminant analysis methods [18], regression [13], segmentation [6], and clustering [5], [8], are also relevant to our purpose. Simulation is used to provide observations, which are treated using a statistical technique. This provides a model capable of proposing DRs in the output using data about the system given in the input. Let us note that such approaches as [8] are very close to learning approaches in their principles.

In this paper we are interested in the NN approach. Due to their capability to deal with non linear problems, NNs provide powerful learning capability. Moreover, these tools can deal both with quantitative and qualitative variables in the input and the output. These advantages allow NNs to compete very well with current learning and statistical algorithms as pointed out in [1], [24], and [23]. Compared to an expert system, NNs do not require an expertise that is difficult to obtain and capture in our case. The main principles of the NN approach are presented in the next section.

## 3. Neural network approach

In the following, we will concentrate on the back propagation paradigm that is frequently used in practice. However, other NN approaches are also suited to our purpose [14].

A neural network is composed of connected cells, organized into layers (Figure 1). Weights are associated with the connections in order to allow the network to compute outputs according to values presented to the input cells. These weights are computed during the learning process, using a set of examples which is presented to the network. The accuracy of the results are strongly dependant on the architecture of the network and on a set of parameters (e. g. gain) utilized by the backpropagation algorithm. Once a NN has been trained, it is able to compute the outputs of new cases. If we apply these generalization capabilities to the selection of DRs, a NN is able to give very quickly the best combination of DRs to apply for each new scheduling case. Moreover, the run time component of a NN (which does not include the learning algorithms and only allows the computation of output) generally does not require a large memory allocation. Hence, it can be incorporated as a part of an other software, even on a PC-like computer. For example, a NN may be embedded in production management software; several DRs may be implemented in the production management software, and chosen by the NN according to the scheduling case (see [3], for example). Such NNs can also be directly used in the plant as scheduling decision support systems when the decision must be made very quickly (e. g. flexible manufacturing systems).

In the production management area, the application of back propagation algorithms has been proposed, for example, to detect the critical situations in a manufacturing line [7], to model manufacturing systems in order to provide auto-organizing systems [10], and to be embedded in a scheduling expert system in order to improve its performance [22]. Following previous works based on a symbolic/numerical learning approach [18], [19], our purpose is here to build a NN capable of providing in the output the best suited combination of DRs from three kinds of data in the input: the system configuration, the characteristics of the production program, and the performance criteria to optimize, as depicted in Figure 1.

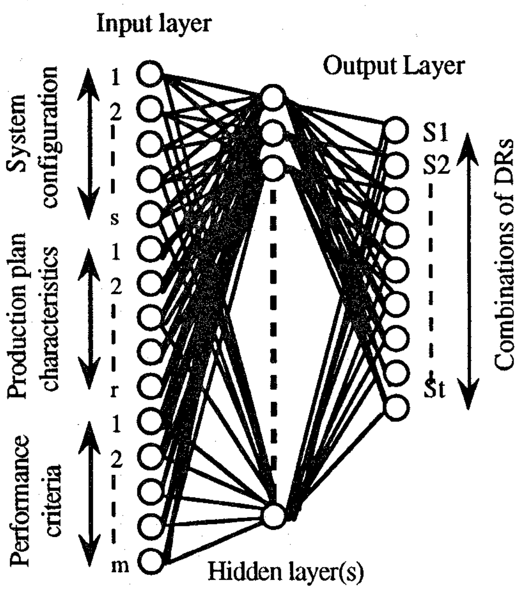


**Fig. 1. Neural network for selecting dispatching rules**

The NN must be trained with a sufficient number of examples of the form:

<system configuration>
∧ <production program characteristics>
∧ <performance criterion>
⇒ <rank(combination#1), rank(combination#2), ..., rank(combination#n)>. (1)

The right hand side of the examples gives the rank of each combination regarding its performance on the performance criterion chosen in the left hand side. In the next section, this method is illustrated through an example of a simplified flow shop.

## 4. Example of a simplified flow shop

The present scheduling problem is based on an study from Barrett and Barman [2]. More details about the system and the assumptions for the simulation can be found in their paper. The system is a simplified two work-center flow shop: WC1 and WC2. Each work center has two machines capable of doing the same operations. Some jobs need rework, done at WC2. Figure 2 presents an overview of the system. Each activity duration is characterized by an exponential distribution (arrival rate of jobs, processing times, rework). The processing times are also submitted to variation, either high or low. The production program to be carried out is characterized here by the mean arrival rate of jobs, the mean expected processing time on WC1, the mean expected processing time on WC2, and the processing time variation. Each machine is out of service once for each 50 jobs during 0.5 time units.

The aim of the original study was to evaluate the influence of combinations of dispatching rules (i. e. scheduling heuristics) at each work center, using a simulation model. Each couple (S-WC1, S-WC2), contains first a rule for WC1, and second a rule for WC2. The dispatching rules tested are the ones previously introduced: FIQ, EDD, SPT and LPT, plus FAS for WC2 (FAS is equivalent to FIQ for WC1). The evaluation of these strategies is done according to several system performance criteria, which are: flowtime, lateness, waiting time, tardiness, earliness and number of jobs in process.

We have used the same model as Barrett and Barman except that we have done more simulations in order to take into account more variation of the parameters. We have not studied the effect of changing the system configuration,

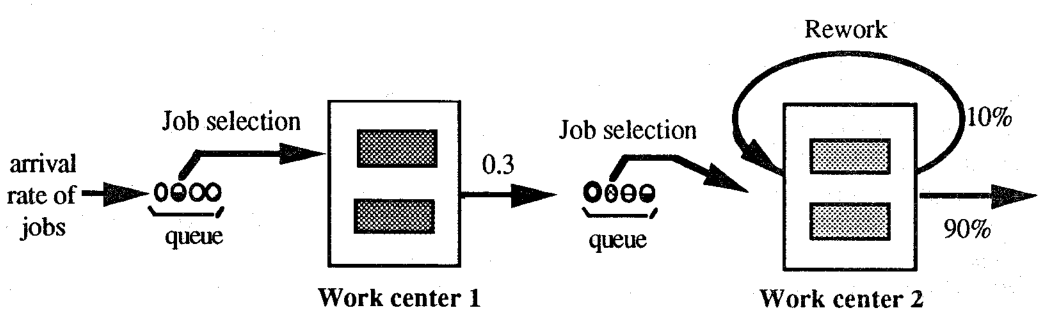


**Fig. 2. Simplified Flow Shop (Barrett and Barman)**

so that the NN does not take it into account (it would have been possible for example, to study the effect of changing the probability of a rework, the number of machines, and the number of jobs before each machines is out of service).

The couples of DRs were compared using N=198 instances of production characteristics (i. e. mean arrival rate of jobs, etc.), randomly selected. Classical statistical problems, such as truncations procedures and run lengths, were studied on pilot runs. For each of the 198 instances, 19 simulations (number of possible couples (S-WC1, S-WC2) except LPT-LPT which give very poor results), were performed in order to evaluate the performance of each couple and ranked regarding their results on the performance criteria, so as to constitute examples as (1), which constitute the training set. Only the mean tardiness performance criterion is considered here, because it is the one that differentiates the most the DRs (for example the strategy SPT-SPT obtains always the best results on the mean flowtime criterion). This training set was used to design and train the NN as presented in the next section.

## 5. Results

Several NNs, as in Figure 3, were developed and tested. Rather than coding the best combination with 1 and the others with 0, the desired outputs were computed as the rank obtained by the combination divided by 19. The best combination chosen by the network is given by the output cell with the lowest value. The difference between the first and the second heuristic gives an idea of the "certainty" of the result.

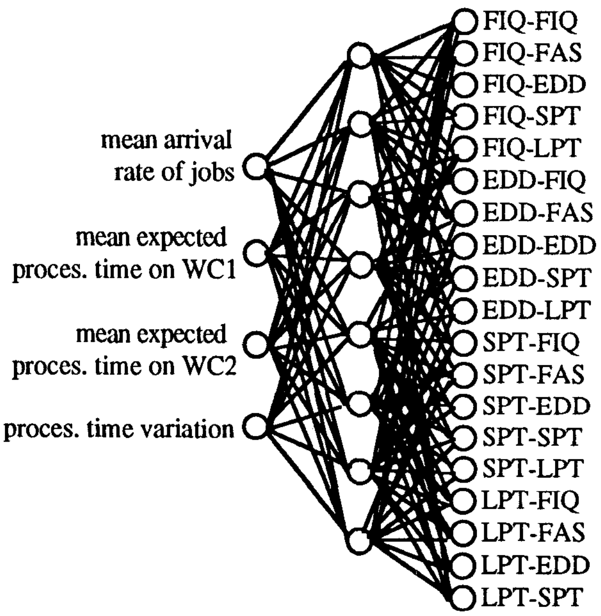


**Fig. 3. Example of network for the mean tardiness**

Seventy five percent of the training set was used for learning; the other twenty five percent was kept to test the generalization capability of the NN. Several experiments were carried out to select and configure a network. Four hidden layers were utilized.

The resulting NN was able to recognize 94 percent of the examples already learned (learning capabilities) and also 94 percent of new examples (generalization capabilities). This result is quite encouraging since only a simple basic back propagation algorithm was used (e. g. no bias, etc.). The NN is able to select a combination for new cases in less than one second, which is a lot less than the time required to run 19 simulation runs.

## 6. Conclusion

In this paper, we have proposed a neural network approach to the selection of dispatching rule problems. Simulation is used to constitute a training set. Once the network has learned, it is able to select for itself the best suited dispatching rules for new cases. Such a network can be used either as a tool to assist scheduling in a plant, or can be embedded in production management or CIM software. This last possibility is a basis for our future work, which will also include the development and test of a network on a more complex and realistic plant (e. g. a flexible manufacturing system).


## Acknowledgments

We are grateful to Professor Barrett, who allowed us to use the data of the flow shop example.


## References


[1] Badran F., Thiria S. and Fogelman Soulie F. (1988), "Comparaison analyse de donnee et reseaux multicouches", *preprint EHEI.*

[2] Barrett R. T. and Barman S. (1986), "A SLAM II Simulation study of a simplified flow shop", *Simulation* 47(5), 181-189.

[3] Bensana, Bel G. (1988) "OPAL: a multi-knowledge approach for industrial job shop scheduling", *International Journal of Production Research*, 26(5), 795-820.

[4] Blackstone J. H., Phillip D. T. and 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.

[5] Bonneau F., Proth J. M. (1985), "Application de règles de gestion à un système de fabrication: classification des objectifs atteints en vue de leur utilisation", *INRIA research report n° 372*, March.

[6] Canals D. (1986), *Ordonnancement d'atelier par simulation: Etude des règles de priorité et aide au lancement*, Phd Thesis, ENSAE, Toulouse, France, Jully (in French).

[7] Chappaz E. (1991), "A neural network for optimization of manufacturing lines", in *Proc. 10th MICAD conf. on CADCAM computer graphics and computer aided technologies*, Paris, France, February, pp. 624-634.

[8] Chu C. and Portmann M. C. (1991), "Application of the artificial memory approach to scheduling problems", *presented at the 11th European Congress on O. R., EURO XI*, Aachen, Germany, July, and submitted to the *Journal of Intelligent Manufacturing*.

[9] Clerc, Baptiste P., Guivarch (1988), "An application of expert system to scheduling", *research paper INSA LYON*, Lyon, France.

[10] Guillard S., Baptiste P. and Favrel J. (1991), "Modelling and simulation for self-organization in modern production workshops", in *Proc. of the CAPE'91 Int. IFIP conf. on Computer Application in Production Engineering*, Bordeaux, France, September, pp. 705-712.

[11] Hershauer J. C. and Ebert R. J. (1975), "Search and simulation selection of a Jobshop sequencing rules", *Management Science*, 21(7), 833-843.

[12] Kiran A. S. and Smith M. L. (1982), "Simulation studies in job shop scheduling", *Research paper*, Dep. of Industrial Engineering, Texas University, U. S. A..

[13] Kleijnen J. P. C. (1987), *Statistical tools for simulation practitioners*, Marcel Dekker Inc., New York, U. S. A.

[14] Lippmann R. P. (1987), "An introduction to computing with neural nets", *IEEE ASSP Magazine*, April.

[15] Montazeri M. and Van Wassenhove L. N. (1990), "Analysis of scheduling rules for an FMS", *International Journal of Production Research*, 28(4), 785-802.

[16] Panwalkar S. S. and Iskander W. (1977), "A survey of Scheduling rules", *Operation Research*, 25(1), 45-61.

[17] Park S. C., N. Raman and M. J. Shaw (1989), "Heuristic learning for pattern directed scheduling in a flexible manufacturing system", in *Proc. third ORSA/TIMS Conf. on Flexi. Manuf. Sys.*, Elsever Science Pub., pp. 369-376.

[18] Pierreval H. (1988), "Data analysis oriented techniques for learning about manufacturing control with simulation", in: *Proceedings of the 2nd European Simulation Multiconference: factory of the future*, Nice, France, June 1988, pp. 61-66.

[19] Pierreval H. and Ralambondrainy H. (1988), "Generation of knowledge about the control of a flow shop using data-analysis oriented techniques and simulation", *INRIA research report n° 897*, Rocquencourt, Le Chesnay, France, Septembre 1988.

[20] Pierreval H. and Ralambondrainy H. (1990), "A simulation and learning technique for generating knowledge about manufacturing systems behavior", *Journal of the Operational Research Society*, 41(6), 461-474.

[21] Pierreval H. (1991), "Rule-based simulation metamodels", presented to *IFORS conf. SPC1*, Bruges, Belgium, 1991 and submitted to *Eur. J. Op. Res.*.

[22] Rabelo L. C. and Alptekin S. (1989), "Synergy of neural networks and expert systems for FMS scheduling", in *Proc. third ORSA/TIMS Conf. on Flexi. Manuf. Sys.*, Elsever Science Pub., pp. 361-366.

[23] Shavlick J. W., Mooney R. J. and Towel G. G. (1991), "Symbolic and neural algorithms: an experimental comparison", *Machine learning*, vol. 6, pp. 11-143.

[24] White H. (1989), "Neural Networks, Learning and statistics", *AI Expert*, December, 48-52.

[25] Wu S. Y. D. and 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), 1603-1624.

[26] Yoshida T. and S. Nakasuka (1989), "A dynamic scheduling for flexible manufacturing systems: hierarchical control and dispatching by heuristics", in *Proc. 28th IEEE conference on decision and control*, December.