Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

USING SIMULATION AND NEURAL NETWORKS TO DEVELOP A SCHEDULING ADVISOR

Thanos Alifantis
Stewart Robinson

Operational Research and Systems Group
Warwick Business School
University of Warwick
Coventry, CV4 7AL UK

ABSTRACT

The research using artificial intelligence and computer simulation introduces a new approach for solving the job shop-scheduling problem. The new approach is based on the development of a neural network-scheduling advisor, which is trained using optimal scheduling decisions. The data set, which is used to train the neural network, is obtained from simulation experiments with small-scale job shop scheduling problems. The paper formulates the problem and after a review of the current solution methods it describes the steps of a new methodology for developing the neural network-scheduling advisor and collecting the data required for its training. The paper concludes by mentioning the expected findings that can be used to evaluate the degree of success of the new methodology.

1 INTRODUCTION

The scheduling of the general job shop is a well known NP hard problem which has attracted many researchers since it is very easy to be formulated but very difficult to be solved. Earlier research has proposed the use of heuristics that can be applied in order to approximate the optimal solution. In addition previous research has proposed interactive models and the use of knowledge based simulation in order to solve the problem efficiently. The purpose of this paper is to describe a new methodology for producing and testing neural network advisor for scheduling activities. Describing the development of a neural network model trained with optimal scheduling solutions that are obtained from simulation experiments the paper shows how simulation and artificial intelligence can be combined to create a learning environment. As the research is still in progress the paper is mainly dedicated to explain as clear as possible the steps in the methodology but it does not report empirical results of implementing it. Finally software guidelines for implementation of the methodology are provided.

2 DESCRIPTION AND FORMULATION OF THE PROBLEM

Hurrion (1978b) describes the general job shop scheduling problem as a series of jobs, each of which require a number of operations in a pre-defined order on a number of machines. The most common objective or performance measure is to minimise the mean processing time. The schedule consists of a number of decisions about which job in a queue in front of a machine should be processed first. A decision is taken at each time a new job from a queue should be loaded to a machine in order to be processed. The decision can be represented as a number that indicates the position of the job in the queue. So the whole schedule can be represented as a column vector **D** whose each row includes a number indicating the position in the queue of the job that must be processed at the particular decision point.

$$\mathbf{D} = \begin{bmatrix} d_1 \\ . \\ d_i \\ . \\ d_I \end{bmatrix} \tag{1}$$

The number of decision points and so the number of rows of the matrix **D** depends on the number of jobs and the number of machines that each job should pass through. In an example with 4 machines and 4 jobs which must be processed form all the machines the decision points are

4*4=16. Given the above description the problem can be represented as in expression 2.

$$\min_{\mathbf{D}} \left(\sum_{j=1}^{n} T_j \right) / n = f(\mathbf{D}) \tag{2}$$

Where:
T_j is the total processing time for the job j
n is the number of jobs
D is the schedule matrix
f is the function that relates the schedule with the processing time.

This above expression says that the objective of the problem is to minimise the mean processing time with respect to the schedule **D** i.e. by choosing the appropriate element from each row of matrix **S**.

$$\mathbf{S} = \begin{bmatrix} d_1^1 & . & d_1^* & . & d_1^K \\ . & . & . & . & . \\ d_i^1 & . & d_i^* & . & d_i^k \\ . & . & . & . & . \\ d_I^1 & . & d_I^* & . & d_I^k \end{bmatrix} \tag{3}$$

Matrix **S** in expression 3 represents the whole solution space. Each row of the matrix represents the alternative decisions that may be taken at the particular decision point. Each element of each row represents the position of each job in the queue in front of the machine that must be loaded. Element d* at each row of the matrix indicates the optimal decision at the particular decision point which must be chosen to minimise the mean processing time.

Function f in (2) is the relationship which determines the mean processing time given a schedule **D**. Since each scheduling decision – element of matrix **D** is not taken at the same time the problem has a dynamic nature. So modelling the job shop problem discrete event simulation can be used to represent function f and to predict the mean processing time for each alternative schedule **D**.

3 CURRENT SOLUTION METHODS

Since the job shop scheduling problem is a well-known NP hard problem a realistic size of the problem can not be solved by searching and evaluating all the alternative solutions (matrix **S**). Since there is not any optimisation algorithm, which can solve a NP hard problem in a polynomial time, the problem has to be solved using algorithms, which approximate the optimal solution, by using various heuristics. These methods can not guaranty optimality but usually can be applied to produce a good near optimal solution. As a consequence there is a vast literature which reports various promising heuristic methods to solve the problem. An alternative to heuristic technique for solving the job shop-scheduling problem is Hurrion's visual interactive simulation approach. Hurrion (1976) with his seminal work proposed to visualise the problem and each time which a decision is required to decide which job should be processed taking into account information provided form a visual display. The implicit assumption on Hurrion's approach is that the visual display can provide the information that can guide the human decision-maker to take an efficient decision.

The visual interactive approach has been extensively applied in many simulation applications including the job shop-scheduling problem since 1978 (Hurrion 1986). The only problem of this approach is that the user should run the model interactively in order to generate a sensible schedule for the job shop problem. The fact that the model is interactive i.e. it should stop at each decision point makes the simulation slow and requires the involvement of the human decision-maker that must control the problem. It seems that Flitman (1986), Hurrion & Flitman (1987), Liang (1992) and Curram (1997) recognise this problem and based on the original visual interactive approach they improved the methodology by constructing an expert system which can control the model and can be used to replace the human decision maker whose involvement is no longer necessary during the simulation run. The expert system in Hurrion & Flitman (1996) was constructed applying rule induction to data collected from human experts who were asked to input decisions in the model when this was required. The fact that the expert system was constructed based on human decision making means that the system if realistic will represent the human expert but there is not any guaranty that the decision produced from it will be optimal or near optimal. The implicit assumption is that the human experts make optimal decision and that the expert system represents realistically the human experts.

This paper based on the original visual interactive approach proposes a methodology for constructing an artificial intelligence scheduling advisor using data of optimal decisions. The artificial intelligence program is planned to include a neural network model which based on the attributes of the system can indicate which job in the queue should loaded to the machine first.

4 METHODOLOGY

Having explained the current approaches to solve the general job shop-scheduling problem this section is dedicated to explain the steps of the methodology (Figure 1). Using visual interactive simulation (step 1), small-scale job shop scheduling problems are optimized (step 2a) with respect to the appropriate schedule. The optimal solutions together with the system attributes are recorded (step2b) and used as input data to train a neural network model (step3). Since neural networks can learn, and generalize from a limited number of examples it is expected that they can learn from a limited number of optimal scheduling solution and be capable of indicating the optimal solution in any other circumstances (Gupta & Lam 1996). Linking the simulation of the job shop with the neural network-scheduling advisor (step4) the capability of neural networks to generalize can be tested. This can be done by running the simulation for job shops where the optimal schedule and the value of the objective function (mean processing time) is already known.

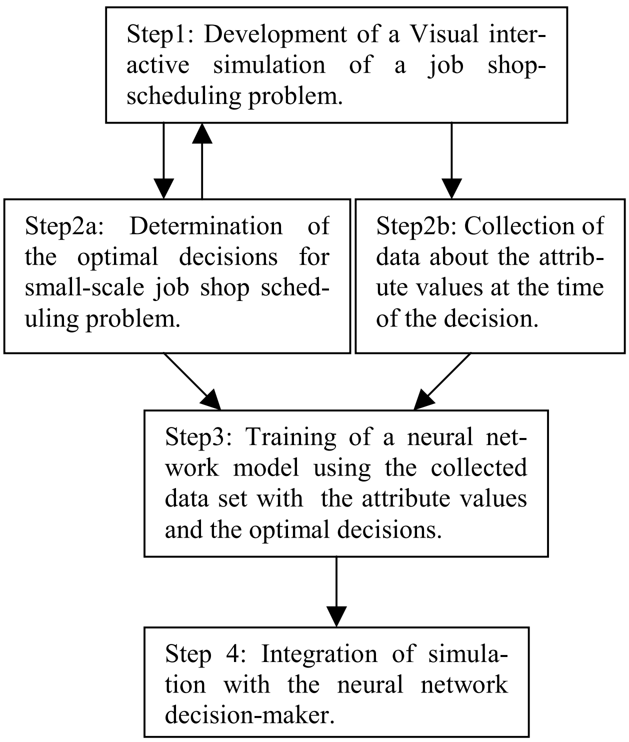

Figure 1: Steps of the Methodology

4.1 Visual Interactive Simulation

A visual interactive simulation of a job shop must be developed. The simulation model should specify the number of machines in the job shop, the number of jobs that must be processed, the number of operations for each job, the order with which the operations should be performed and the number of machine that should perform each operation. During the development of the system the above information can be specified in the model sampling from appropriate statistical distributions. In addition the model should be interactive and it should stop the simulation each time at which a decision is required i.e. each time which a new job from a queue should be loaded to a machine. The model should also be visual and at each decision point should report and record information about the attributes of the system at the particular time. For example some attributes of the system in a problem with 4 jobs and 4 machines at the time t are the following:

```
Number of jobs waiting in queue 1
Number of jobs waiting in queue 2
Number of jobs waiting in queue 3
Number of jobs waiting in queue 4
      ..............
```

4.2 Data Collection: Determining the Optimal Inputs in Small Scale Job Shop Problems

In order to construct a neural network-scheduling advisor to indicate the position of the job in the queue that must be processed the neural network should be trained using a data set of decisions with the associated attribute variables. The data set should have the form of two matrices the first should include the decisions and the second the value of each attribute the time at which the decision is required. The values for the attributes of the system can be collected directly form the simulation that generates them the difficult part is the information about the optimal decision i.e. the position in the queue of the job which must be processed.

The optimal decision at each decision point for small-scale problems (for example 4 machine 4 jobs) can be found using a search procedure that can be linked with the simulation software. The optimal decisions for larger problems are not possible to be found since as it has already been explained the problem is NP hard and so ordinary search is very slow. The hope and the main assumption of this methodology is that the attributes of the system are a good predictor for the optimal strategy for small and large job shop scheduling problems. According to this assumption the neural network can be trained using data from small job shop scheduling problems at which the optimal solution can be found searching the whole solution space. Although the neural network will be trained with data from small-scale problems it is expected that it will be possible to indicate the optimal solution for large-scale problems given the value of each attribute.

An iterative search procedure in a Visual Basic front end can be used to search for the optimal schedule in some small-scale problems. The procedure searching all the possible solutions can find and record in a reasonable time the solution schedule which minimises the mean processing time. The front end will also record in a text file the attribute of the system the time that the decision was required.

The data set should have the form of two matrices **D*** and **A** (expression 4). Matrix **D*** is a column vector that should indicate the optimal schedule. Each element in vector **D*** should be a number that indicates the position of the job in a queue, which must be processed next. In addition each line of matrix **A** includes the value of each attribute of the system the time at which each decision was taken. In a system that has J attributes matrix **A** will have J columns.

$$\mathbf{D^*} = \begin{bmatrix} d_1^* \\ . \\ d_i^* \\ . \\ d_I^* \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{1,1} & . & a_{1,j} & . & a_{1,J} \\ . & . & . & . & . \\ a_{i,1} & . & a_{i,j} & . & a_{i,J} \\ . & . & . & . & . \\ a_{I,1} & . & a_{I,j} & . & a_{I,J} \end{bmatrix} \tag{4}$$

It is proposed that not one but many job shop scheduling problems with different size should be used in order to make sure that the data set is as generic as possible. What is more given that the problem includes some sources of uncertainty such as the processing time for each operation of each job the simulation should be replicated a number of times.

4.3 The Neural Network Model

Having collected the data set this can be used to train the neural network model. It is expected that a 3 layers feed forward network should be computationally sufficient to build a model to predict the optimal policy given the value of each attribute. In a system with 5 attributes the functional form of the model can be represented geometrically as in figure 2 and mathematically as in expression 5.

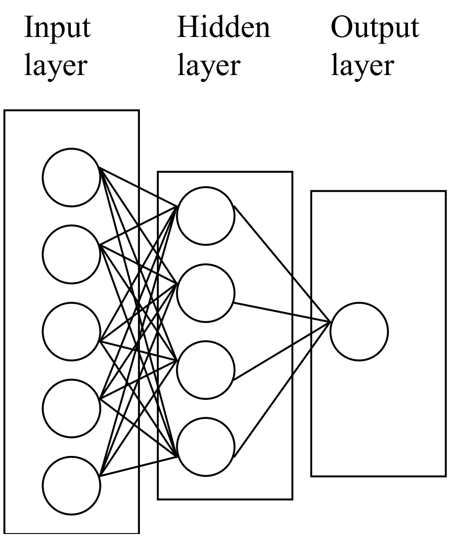

Figure 2: Feed-froward 3 layers neural network structure

$$Y = \sigma((\sigma(\mathbf{XW}_1 + \mathbf{B}_1))\mathbf{W}_2 + \mathbf{B}_2) \tag{5}$$

Where:
Y is a number that indicates the position in the queue of the job that must be processed next.
X is an input row vector with the attributes of the system at the time i.
$\mathbf{W_1}$ is a matrix with the weights that link the input and hidden layer.
$\mathbf{B_1}$ is a matrix with the bias term that is added to the input vector .
$\mathbf{W_2}$ is a matrix with the weights, which connect the hidden and output layer.
$\mathbf{B_2}$ is a matrix with the bias term which is added to the output
σ (x) is the function that transforms the data to have range between zero and one: $\sigma(X) = 1/(1+e^{-X})$

4.4 Model Integration and Validation

Having constructed the neural network advisor which, given the attributes of the system, can indicate which job from the queue should be loaded to the machine which is available the scheduling advisor can then be linked with the simulation. So each time a decision point is reached the simulation, stops and the neural network model, is invoked using OLE technology. The link enables the automatic input of the value of each attribute from the simulation to the neural network that can produce a decision. Inputting the decision back to the simulation software the OLE technology reactivates the simulation which continues the run. When the simulation has reached the end of the run the mean processing time provides an indicator about the performance of the neural network. Comparing this performance with the best possible performance (which is known from the data collection process) can show how much valid is the neural network advisor.

5 IMPLEMENTATION

The visual interactive simulation that is described in the methodology can be implemented in any simulation software such as Witness. The fact that the job shop-scheduling problem is one of the major applications of computer simulation encourages the software vendors to make their software appropriate for developing job shop simulations. The solution search procedure is recommended to be implemented using Visual Basic since it is the most appropriate language for linking and integrating applications. In addition the most simulation packages are compatible with Visual Basic and they can communicate with it during the simulation run (Robinson et al). Since the type of neural network described in the methodology is one of the most commonly applied there is no need to write a specific application and so it can be developed in one of the commercial neural network packages such as Neuralyst or Braincel.

6 DISCUSSION AND EXPECTED FINDINGS

The implementation of the above methodology has not been completed yet so this paper avoids reporting the result of implementing it. Later paper will report results such as the time needed for the search algorithm to find the optimal solution for small scale job shop problems and the number of problems needed to be solved in order to collect a sufficient data set which would produce an accurate neural network advisor. In addition and perhaps most important it will report the relative significance and importance of the attributes of the system for determining the optimal solution. Finally the results of experiments with large scale job shop problems using the neural network advisor to solve them are going to be reported in the later paper.

7 SUMMARY

The paper has described the steps of a methodology to develop a job-shop-scheduling advisor. Recommending the training of a neural network model using optimal scheduling decisions obtained by simulation experiments the methodology introduces a new approach for solving the job shop-scheduling problem.

REFERENCES

Carvalho, M. Dougherty, M. Fowkers, A. Wardman, M.R. Forecasting Travel Demand : A comparison of logit and artificial neural network methods. *Journal of Operational Research Society*. 1998; 49717-722.

Curram, S. *Representing intelligent decision making in discrete event simulation*. PhD Thesis University of Warwick ; 1997.

Flitman, A. *Towards the application of artificial intelligence techniques for discrete event simulation*. PhD Thesis University of Warwick; 1986.

Flitman, A.M. Hurrion, R. D. Linking discrete-event simulation models with expert systems . *Journal of Operational Research Society*. 1987; 38(8):723-733.

Gupta, A. Lam, M.S. Estimating Missing Values Using Neural Networks. *Journal of Operational Research Society.* 1996 Feb; 47(2):229-238.

Hurrion, R. *The design use and required facilities of an interactive visual simulation language to explore production planning problems*. PhD thesis University of London; 1976.

Hurrion, R. Visual interactive simulation system an aid to decision making. *Omega*. 1978a; 6.

Hurrion, R. An investigation of visual interactive simulation methods using the job-shop scheduling problem. *Journal of Operational Research Society*. 1978b; 29.

Hurrion, R. An interactive visual simulation system for industrial management. *European Journal of Operational Research*. 1980; 31.

Hurrion, R. Visual interactive modeling. *European Journal of Operational Research*. 1986; 23.

Liang Moskowitz Yih. Integrating neural networks and semi-markov processes for automated knowledge acquisition an application to real time scheduling. *Decision Science.* 1992; 23.

Maire, F. Rule Extraction by Backpropagation of Polyhedra. *Neural Networks* . 1999; 12717-725.

Pierreval Ralambondrainy. A simulation learning technique for generating knowledge about manufacturing systems behavior. *Journal Of Operational Research Society*. 1990; 41.

Robinson, S. Edwards, J. Yongfa, W. An expert systems approach to simulating human decision maker. *Proceedings of the 1998 Winter Simulation Conference*. 1998.

Vaughn, M.L. Derivation of the multilayer perceptron weight constraints for direct network interpretation and knowledge discovery. *Neural Networks*. 1999; 12.

AUTHOR BIOGRAPHIES

THANOS ALIFANTIS is a PhD researcher at Warwick Business School in the UK. He holds a BSc in Economics and Operational Research from Lancaster University. He is investigating the use of artificial intelligence for representing human decision-makers in simulation models His research interests are in econometrics, optimization, simulation, heuristic techniques especially simulated annealing and taboo search for timetabling problems, data envelopment analysis, visual basic programming, game theory, direct rule extraction from artificial neural networks, analysis of the structure of feed forward neural networks and expert systems. His email and web address are `a.alifantis@warwick.ac.uk` and `<http://www.warwick.ac.uk/staff/A.Alifantis/>`

STEWART ROBINSON lectures in Operational Research and Systems at the Warwick Business School in the UK. He holds a BSc and PhD in Management Science from Lancaster University. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. His research interests are in finding ways to improve the use of operational research, and specifically simulation, within industry. He is author/co-author of two books on simulation. Current work involves an investigation into the use of expert systems to represent a human decision-maker in simulation models, the analysis of complex output behaviours from simulations, and developing an understanding of quality in relation to operational research and simulation studies. His email and web addresses are `stewart.robinson@warwick.ac.uk` and `<www.btinternet.com/~stewart.robinson1/SLRHOME.HTM>`