

## NEURAL NETWORK MODELING AND SIMULATION OF THE SCHEDULING

---

Ricardo Lorenzo Avila Rondon

*Facultad de Ingeniería, Universidad de Holguín, Cuba. ricardo@cadcam.uho.edu.cu*

Adriano da Silva Carvalho

*Faculdade de Engenharia, Universidade do Porto, Portugal. asc@fe.up.pt*

Guillermo Infante Hernández

*Facultad de Ingeniería, Universidad de Holguín, Cuba. guillermo@cadcam.uho.edu.cu*

*A three layer feed forward neural network was constructed and tested to analyze the scheduling process on single machine. The operating variables studied are the operation, processing time, setup time, deadline time, due date time, priority, machine, and fabric color. These variables were used as input to the constructed neural network in order to predict the scheduling completion time as the output on a single machine. Three layer feed forward network trained with error back propagation learning rule are used. The constructed network was found to be precise in modeling the scheduling for the operating conditions studied and also, in predicting the scheduling for the new input data which are kept unaware of the trained neural network.*

### 1. INTRODUCTION

Scheduling is a decision making process of allocating limited resources (machines at a workshop, runways at an airport, crews at construction site) to activities (operation in a workshop, takeoffs and landings, stages at a construction project) over time (Pinedo, 2005). Generally resources are identified with machine and activities with tasks or jobs. A schedule is a job sequence determined for every machine of the processing system. Scheduling in the context of manufacturing involves finding a sequential allocation of competing resources that optimizes a particular objective function, subject to certain constraints.

Job shop scheduling has generated a great of research (Johnson 1954), (Adams et al. 1988), (Dell'Amico and Trubian 1993), Foo et al. 1995, Yamada and Nakano 1996). In job-shop scheduling the parts may flow in different directions, also a machine can be visited more than once by the same part, and it's objective might be to minimize the makespan or maximize machine utilization, subject to constraints such as the number of machines, plant capacity, labour availability, and so on.

Another mode of processing in correspondence with the way parts visit machines is known as flow shop, when all parts flow in one direction.

The scheduling models are specified (Pinedo, 2005) according to three field classification  $\alpha|\beta|\gamma$  where:  $\alpha$  specifies the machine environment,  $\beta$  specifies the job characteristics, and  $\gamma$  specifies the optimality criterion.

## 2. SCHEDULING

### 2.1. Machine environment

a) Single stage systems.

- If there is a single or dedicated machine ( $m = 1$ ), each job should be processed by that machine exactly once. Here  $\alpha = \mathbf{1}$ , and  $p_j$  is the processing time of job  $j$ .
- If there are several parallel machines  $\{M_1, M_2, \dots, M_n\}$ , each job can be processed by any machine. Here  $\alpha = \mathbf{P}$ , and  $p_{ij}$  is the processing time of job  $j$  on machine  $i$ .

b) Multistage systems

In this case each job should be processed on each machine from the set  $\{M_1, M_2, \dots, M_n\}$ . Here  $\alpha = \mathbf{F}$  (for flow shop, a job  $j$  is processed first on machine 1, then on machine 2, and finally on machine  $n$ ), or  $\alpha = \mathbf{J}$  (for job shop, each job has its own route to follow), or  $\alpha = \mathbf{O}$  (for open shop, each job can be processed by the machines in an arbitrary order).

### 2.1 Job environment

There are  $n$  jobs  $N = \{1, \dots, n\}$ . For job  $j$  there may be given also. Figure 1.

- $w_j$  - weight (the importance or priority of the job)
- $p_j$  - process time (the time the job is processed on the system)
- $r_j$  - release time (the time the job arrives at the system)
- $s_j$  - start time (the time the job starts on the system)
- $c_j$  - completion time (the time the job ends on the system)
- $d_j$  - due date (the time the job is promised to the customer)

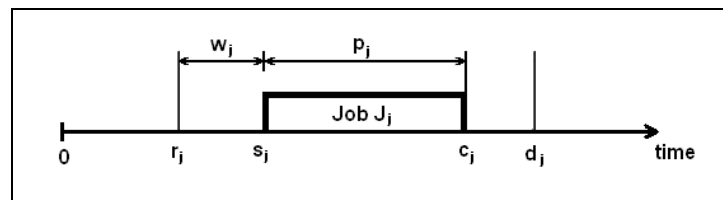


Figure 1. Graphics representation of job parameters

### 2.2. Optimality criterion

The objective is to construct a schedule that minimizes a given objective function  $F$ .

Usually function  $F$  depends on job completion times  $C_j$ ,  $j = 1, \dots, n$ , where  $C_j$  is the completion time of the last operation of job  $j$ .

Common objective functions are:

Makespan:  $C_{\max} = \max \{C_j, | j = 1, \dots, n \}$

Total completion time:  $\sum C_j = \sum_{j=1}^n C_j$

Total weighted completion time:  $\sum w_j C_j = \sum_{j=1}^n w_j C_j$

Other objective functions depend on due date  $d_j$ . It is defined for each job  $j$ :  
 Lateness of job  $j$ :  $L_j = c_j - d_j$ . The latest time at which a job  $j$  can be completed.  
 Earliness:  $E_j = \max \{0, d_j - c_j\}$ . The amount of time between the  $d_j$  and  $c_j$ .  
 Tardiness:  $T_j = \max \{0, c_j - d_j\}$ . The amount of time between the  $c_j$  and  $d_j$ .

### 2.3. Approaches to Scheduling Problems

The input size of a typical scheduling problem is bounded by the number of job  $n$ , the number of machines  $m$ . An algorithm is said to be polynomial if its running time is bounded by a polynomial in input size. Polynomial algorithms are sometimes called efficient or simply good. The class of all polynomial solvable problems is called class  $\mathcal{P}$ .

Another class of optimization problems is known as NP-hard problem (NP stand for non-deterministic polynomial). In this case, complete enumeration of all the solutions to identify the optimal one is not practical. Very often technological constraints demand that each job should be processed through the machines in a particular order.

There are many techniques applied to find a solution to the different scheduling models, all of them are in correspondence with model classification, problem size, optimality criteria, and so on.

Given a scheduling problem we first need to determine its complexity status. This is done either by designing a polynomial time algorithm for its solution or by proving that the problem is NP-hard. Once it is known the problem is NP-hard, it is necessary to determine the solution type: exact or approximate. It is unlikely that an exact solution can be found by a polynomial time algorithm. In any case, for problems of practical interest only small size instances can be handled by exact methods.

In order to find a good solution within an acceptable amount of time, two types of algorithms can be developed: approximation algorithms and heuristic algorithms.

An algorithm is called an approximation algorithm if it is possible to establish analytically how close the generated solution is to the optimum (either in the worst case or on average). Approximation algorithms produce solutions in polynomial time, but for the price of loss of optimality. The solutions found are guaranteed to be within a fixed percentage of the actual optimum.

The performance of a heuristic algorithm is usually analyzed experimentally, through a number of runs using either generated instances or known benchmark instances. Heuristic algorithms can be very simple but still effective. Most of modern heuristics are based on various ideas of local search (neighborhood search (Hurink, 1998), tabu search, simulated annealing, genetic algorithms (Dorigo and Stutzle, 2004), (Pinedo, 2005), and so on). Heuristic algorithms produce feasible solutions which are not guaranteed to be close to optimum.

### 2.4. Case study

The case study considered in this paper is similar to the paper mill mentioned by (Pinedo, 2005). The input to textile mill is cotton, polyester or other fiber the output is finished rolls of cotton fabric, or wool fabric, or synthetic material. At the heart of the textile mill are its textile machines, which are very large and represent a significant capital investment. The end of this process produces various types of fabric which are characterized by their basis weights, grades and colors.

Master production plans for these machines are typically drawn up on an annual basis. The projected schedules are cyclic with cycle times weekly. A particular type of fabric may be produced in every cycle, or every other cycle, or even less often, always depending on demand.

Every time the machine switches over from one combination of grade of fabric to another there is a setup cost involved. During the changeover the machine keeps on producing fabric. Since the fabric produced during a changeover does not meet any of the required specifications, it is either sold at a steep discount or considered wasted into the production system.

The production plan tries to maximize production, while minimizing inventory costs. Maximizing production implies minimizing changeover times. This means longer production runs, which in turn result in higher inventory cost. The overall production plan is a trade off between setup and inventory cost.

### 2.5. Mathematical Formulation

Consider a single machine and  $n$  jobs as it is depicted in (Pinedo, 2005). Job  $j$  has a processing time  $p_j$ , a release date  $r_j$  and a due date  $d_j$ . If  $r_j = 0$  and  $d_j = \infty$ , then the processing of a job  $j$  is basically unconstrained. It is clear that the makespan  $C_{\max}$  in a single machine environment does not depend on the schedule. For various other objectives certain priority rules generate optimal schedules. If the objective to be minimized is the total weighted completion time and the processing of the job  $j$  is unconstrained, then the Weighted Shortest Processing Time first (WSPT) rule, which schedules the jobs in decreasing order of  $w_j/p_j$ , is optimal. If the objective is the maximum lateness  $L_{\max}$  and the jobs are all released at time 0, then the Earliest Due Date first (EDD) rule, which schedules the jobs in increasing order of  $d_j$ , is optimal.

Other objectives, such as the total tardiness  $\sum T_j$  and the total weighted tardiness  $\sum w_j T_j$  are much harder to optimize than the total weighted completion time or the maximum lateness. A heuristic for the total weighted tardiness objective is described in (Dorigo and Stutzle, 2004), (Glover, 1989), (Pinedo, 2005).

In general for this case,  $n$  jobs have to be sequentially processed on a single machine. Each job  $j$  has a processing time  $p_j$ , a weight  $w_j$ , and a due date  $d_j$  associated, and the jobs become available for processing at time zero. The tardiness of a job  $j$  is defined as  $T_j = \max \{0, C_j - d_j\}$ , where  $C_j$  is the completion time of job  $j$  in the current job sequence. The goal is to find a job sequence which minimizes the sum of the weighted tardiness given by

$$\sum_{i=1}^n w_i T_i$$

When the jobs in a single machine problem have different release date  $r_j$ , then the problems tend to become significantly more complicated.

### 2.6. Experimental data

It was considered the follow real data: operation, processing time, setup time, deadline time, duedate time, priority, machine, and color. Initially the data are in worksheet format file. The data are arranged in a 75 by 6 matrix corresponding to one work day in a factory. For convenience the matrix is divided into two data sets, the first 55 rows as training set and the rest as a testing set.

### 3. NEURAL NETWORK MODELING

Neural network consists of a number of interconnected artificial neurons which are divided into three groups, namely: input, hidden and output layers. To construct the network, following steps are to be followed. First, weighting factors are divided from back propagation learning algorithm. The weighted inputs are summed and transferred through the sigmoid function of neurons to obtain outputs. The output is propagated along the outgoing of connecting weights to become the input of neurons on next layer. In the present study a multiple layer feed forward network with two layers was constructed. Multiple layer networks can approximate any function very well. Feed forward ANN allows signals to flow only in one direction, i.e., from input to output.

The feed forward ANN adjust the transfer function that is associated with the inputs and outputs. In the present study, initially a network with five hidden layer was constructed and was trained to simulate the scheduling process for various operating conditions. The detailed structure of the network and the training strategy of the constructed neural network are shown in Figure 2 and Figure 3 respectively. Figure 3 shows the feed forward network with two hidden layers. P1 is the input vector to the hidden layer, W1 and b1 represents the weight and bias of the hidden layer.

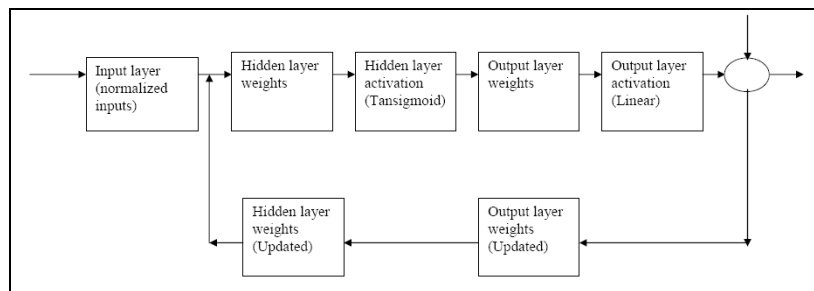


Figure 2. Training strategy of the constructed feed forward artificial network

The information from the hidden layer is transferred to the output layer as shown in Figure 3. The terms P2 represent the output vector and can be determined from the weight, W2 and bias b2 of the output layer. In the present research a *tansig* function and a *purelin* function was used as the propagation functions in the hidden layer and in the output layer respectively.

The training strategy of the network is shown in Figure 2. As shown in Figure 2, the input vectors and the corresponding output vectors are used to train the network until it approximates the propagation function. Thus the bias and the weights can be obtained from the training procedure which is based on the experimental data.

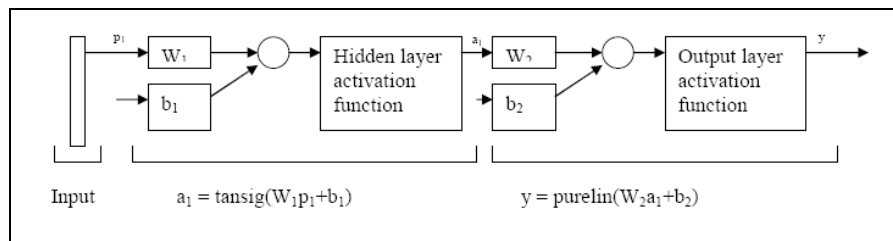


Figure 3. Two layer network structure and the flow of information within the network

In the present study, the operation, processing time, setup time, due date time, priority, and fabric color treated are as the input vector and the corresponding completion time was defined as the output vector to train the neural network.

The neural network toolbox Version 4 of MATLAB, Mathworks Inc., was used for simulation. The input vectors and the target vector was normalized before training such that they fall in the interval of 0 to 1, so that their standard deviation and mean will be below the value of 1. The experimental data used for targets and the experimentally determined target values were pre-processed so that the mean and the standard deviation is 0 and 1.

The experimental conditions and the corresponding experimentally determined completion time were set as the input and the target vectors, respectively. The neural network was trained in a batch mode. The training was made using the Levenberg-Marquardt's training strategy.

Training the neural networks by the Levenberg-Marquardt algorithm is sensitive to the number of neurons in the hidden layer. The more the number of neurons, the better is the performance of the neural network in fitting the data. However, too many neurons in the hidden layer may result in the over fitting. During the training process, the number of neurons in the hidden layer was changed during the training process while optimizing the transfer function for the given input and output vectors. In order to avoid the problems due to the over fitting, a Bayesian regularization technique in combination with the Levenberg-Marquardt's algorithm was used during the ANN training process.

The Bayesian algorithm works best when the networks input and output are scaled within range of -1 to +1. After several trials, the neural network with 10 neurons in the hidden layer was found to be excellent in representing the scheduling process. In the hidden layer, initially three types of transfer functions namely the exponential sigmoid, tangent sigmoid and linear functions were tested while training the neural network. The linear function was used at the output layer. A tansigmoid function in the hidden layer and the linear function at the output layer are found to be excellent in predicting the scheduling completion time irrespective of the initial operating variable condition. The training is stopped until the convergence is reached and the network is set ready for the prediction.

The convergence is reached when the sum of the squared errors and the weights and biases reach some constant values. The normalized targets were converted back to the original target values. The training program was terminated when the neural network has truly converged. The network is converged if the sum of the squared errors and the sum of the squared weights are nearly constant over several iterations.

The details of the completely trained neural network used in the present study to design the scheduling completion time are given in table 1. Figure 4 shows a comparison between actual targets and predicted completion time by ANN. From the Figure 4, it can be observed that the developed neural network was found to be excellent in predicting scheduling completion time for various operating conditions.

*Table 1.* Details of the trained neural network used to predict the completion time in the scheduling on single machine

Type	Value/comment
Layer 1	6 neurons
Layer 2	5 neurons
Layer 3	1 neuron
Number of data used for training	75 (one machine, one work day)
Function in hidden layer	Tansigmoid
Function of output layer	Linear

### 3.1 Results

The accuracy of the newly constructed neural network was verified using regression analysis and correlation coefficient. The correlation coefficient between the experimentally determined  $c_{\text{experiment}}$  and the  $c_{\text{ann}}$  determined by the neural network was found to be 0.886 with slope equal 0.95. The high correlation coefficient confirms that the newly constructed ANN was highly precise in predicting the scheduling completion time solutions.

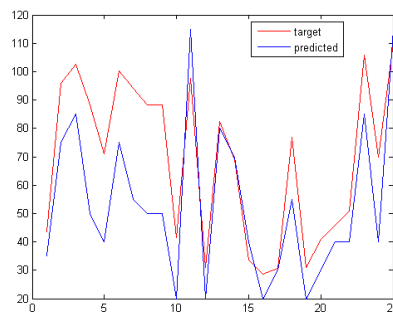


Figure 4. Comparison between actual targets and predictions

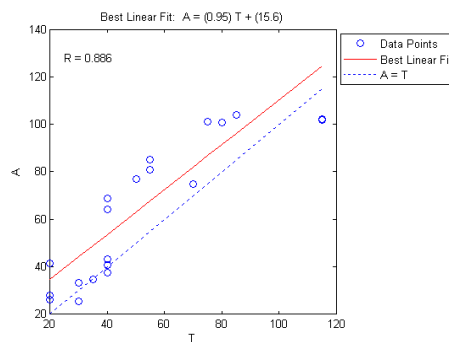


Figure 5. Regression graphic to judge scheduling ANN

From design point of view it would be helpful to use know the amount of data at any time irrespective of the operating conditions. Thus the constructed network was used to simulate the production system for new operating conditions that are kept unaware of the trained neural network. For simulation purpose, new inputs which are not used while training was feed to the trained neural network and the corresponding completion time were determined from the neural network. Figure 5 shows the plot of predicted  $c_{\text{ann}}$  using the trained network and the  $c_{\text{experiment}}$  calculated experimentally for various initial operating conditions. From Figure 5, it can be observed that the newly constructed neural network was precise in predicting the scheduling completion time with a higher correlation coefficient of 0.886.

#### **4. CONCLUSIONS**

In this study we presented an artificial neural network for the one-machine scheduling problem. This shows the developed neural network model can be precise in predicting the scheduling. Another advantage of the newly constructed neural network model over the theoretical models is its accuracy to predict the completion time for any initial production data. The present study and some of our previous works suggested the strong influence of the operating variables on the scheduling completion time. It is a highly complicate process to propose a generalized expression correlating the operating variables involved in the production system with the scheduling priority. The constructed network was trained considering all the operating variables of the system making the neural network precise enough to predict the scheduling for any operating condition. Though the lab scale study is limited to the range of operating variables studied, it is always possible to introduce new inputs to train the network whenever new experimental data are available. The future work is aimed to extend this idea for the logical link between process planning and scheduling systems.

#### **5. REFERENCES**

1. Adams, J., Balas, E., Zawack, D. The shifting bottleneck procedures for job shop scheduling. *Management Science* 1988; 34 (3): 391–401.
2. Dell'Amico, M., Trubian, M. Applying tabu searches to the job shop scheduling problem. *Annals of Operations Research* 1993; Vol. 41: 231-252.
3. Dorigo, M., and Stutzle, T. *Ant Colony Optimization*. A Bradford Book. The MIT Press. ISBN 0-262-04219-3, 2004.
4. Foo, Y.P.S., Takefuji, Y. Integer linear programming neural networks for job-shop scheduling. In: *Proceedings of Joint International Conference on Neural Networks* 1988; vol. 2: 341–348.
5. Glover, F. Tabu search. Part I. *ORSA J. Computing* 1989; Vol. 1: 190-206.
6. Hurink, J.L. Memorandum No. 1449. An exponential neighborhood for a one-machine batching problem. ISSN 0169-2690. July 1998.
7. Johnson, S. M. Optimal two-and three stage production schedules with setup times included. *Nav. Res. Logistics Q.*, 1954; 1(1): 61–68.
8. Pinedo, M. L. *Planning and Scheduling in Manufacturing and Services*. Springer Series in Operation Research. ISBN 0-387-22198-0, 2005.
9. Van Laarhoven, P.J.M., Aarts, E.H.L. Job shop scheduling by simulated annealing. *Operations Research* 1992; Vol. 40, no 1: 113-125.