The Usage of Artificial Neural Networks For Finite Capacity Planning

Ali Fuat Guneri and Alev Taskin Gumus

Mechanical Faculty
Industrial Engineering Department
Yildiz Technical University
Besiktas - Istanbul, Turkey
Corresponding author's e-mail: {Ali Fuat Guneri, guneri@yildiz.edu.tr}

In this study finite scheduling and artificial neural networks are applied for finite capacity planning. Utilisation of artificial neural networks on solving finite scheduling problems is examined. Also a model is developed by using multi layer perceptron (MLP) networks and carried out to solve a real world problem in a job shop scheduling system, in an automotive firm.

Significance: Finite capacity scheduling and its integration with the other functions of the companies is very important to manage planning activities in manufacturing. The usage of artificial neural networks in finite scheduling is studied in this paper.

Keywords: Finite capacity scheduling, Artificial neural networks, Multi layer perceptron.

(Received 13 March 2006; Accepted in revised form 27 February 2007)

1. INTRODUCTION

Scheduling may be defined as ''the allocation of resources over time to perform a collection of tasks'' (Pongcharoen et al., 2004). Also, scheduling has been defined as 'the art of assigning resources to tasks in order to insure the termination of these tasks in a reasonable amount of time' (Fonseca and Navaresse 2002). Production scheduling is one of the major issues in production planning and control of individual production units which lies on the heart of the performance of manufacturing organizations (Chan et al., 2006). Scheduling is a decision process about assigning resources in production cycles. The scheduling problem in manufacturing is sequencing and planning the operations under technological and physical constraints (Tardif and Spearman, 1997; Moon et al., 2002; Jansen and Mastrolilli, 2004). Scheduling problem is the allocation of resources to perform a set of activities in a period of time (Tavakkoli-Moghaddam et al. 2005).

Scheduling problems exist almost everywhere in real-world situations (Gen and Cheng, 1997; Watanabe et al., 2005) Many real scheduling problems in the manufacturing industries are quite complex and very difficult to be solved by conventional optimization techniques. Usually, these difficult-to-solve problems are characterized as combinatorial optimization problems subject to highly complex constraints (Pongcharoen et al., 2004; Watanabe et al., 2005). The combinatorial process is a necessary part when preparing the production scheduling that many these related problems are classified as NP-hard problems (Artiba and Elmaghraby, 1997; Chan et al., 2006).

The job shop scheduling is one of the most typical and complicated tasks in scheduling problems. The aim of job shop scheduling problem is to allocate n jobs to m machines in order to optimize a special factor. Recently, scheduling systems based on intelligence knowledge have been proposed and presented. Job shop is a production system with the capability of producing products with a number of jobs and different operation times for each job. Due to different operations on a product and machine requirements to process each step of production, it is so hard to find an efficient scheduling solution. Traditional approaches often consider small-sized problems with deterministic parameters. The real world of industry is deterministic free and production attributes are stochastic. Stochastic variables and constraints are not available in the application of traditional approaches (Tavakkoli-Moghaddam et al., 2005).

Job shop scheduling problem (JSSP) is a class of combinational optimization problems known as NP-Hard one (Tavakkoli-Moghaddam et al., 2005). It is usually very hard to find its optimal solution. Practically researchers turn to search its near-optimal solutions with all kind of heuristic algorithms (Yang and Wang, 2001). In this study, we seek the solution of finite capacity scheduling problem in a job-shop environment.

Capacity planning is the 'class of the problems related to the prediction of when in the future the capacity of an existing system will become insufficient to process the installation's workload with a given level of performance' (Sia and Ho, 1997). The planning and utilisation of production capacity is a major strategic decision in manufacturing. The strategic decisions of a company are concerned with acquiring the resources needed to survive and prosper over the long term (MirHassani et al., 2000).

Several approaches to finite capacity scheduling (or *Leitstands* as they are sometimes called) have been proposed (for example, Kanet and Sridharan, 1990; Jain et al., 1990; McCarthy and Barber, 1990). Acknowledging the fact that capacity of certain resources is limited means that a manufacturing firm would not accept all demand and would be willing to reject some if the product cannot be delivered at the requested time without hurting the overall profitability of the firm (Akkan, 1997).

In this paper finite scheduling, which is the basis of finite capacity planning is considered, and artificial neural networks are applied for finite capacity planning. The ANNs are used in finite capacity planning to determine an effective planning strategy. Also a model is developed by using multi layer perceptron (MLP) networks and carried out to solve an application problem in a job shop scheduling system.

2. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are built from intensive connections between basic calculation elements obtaining high performance by considering biological nervous system's structure. The ANN models have a great potential in various areas with parallel and high computation speed, especially in speech and image recognizing. The ANN models are based on various hypotheses that prudence to connect several computation elements with variable weights and realize intensive parallel processes (Simpson, 1990).

In general, an ANN model is formed from n layers, different numbers of computation elements that work like biological neurons and intensive connections between these computation elements along layers. The computation elements used in various ANN models are named as artificial neurons, knots, units or process elements.

The neurons are very simple functioned processors if considered alone that form neural networks. There are three main parts in a neuron structure. These are synapsis, collector and activation function, consecutively (Haykin, 1999). A neuron model is shown in Figure 1. As seen in this figure, the inputs of a neuron is multiplied by the weights on the synaptic connections and applied to the collector; and outputs are computated by using this total in the neuron activation function. Equation (1) computes the weighted total and equation (2) computes the output of neuron (Haykin, 1999):

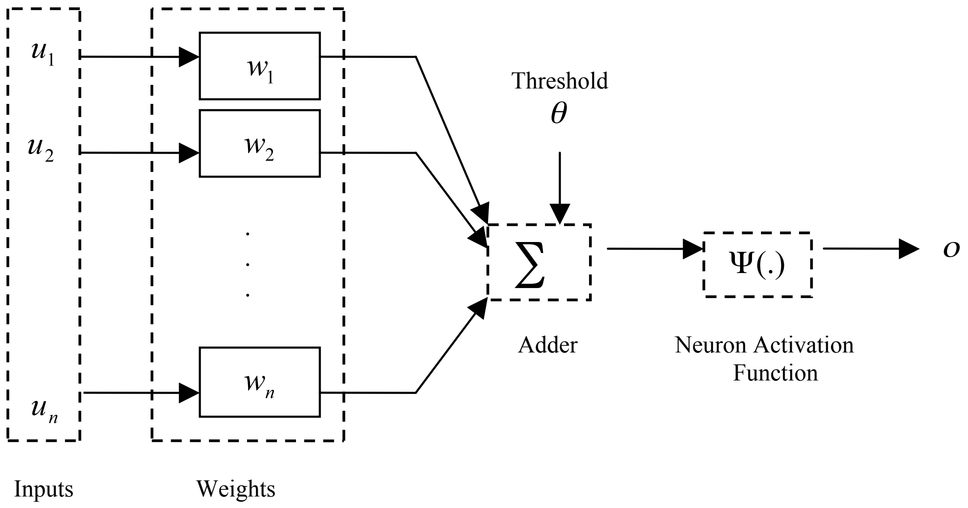

Figure 1. A neuron structure

$$S = w_1u_1 + w_2u_2 + ... + w_nu_n - \theta = \sum_{i=1}^{n} w_i u_i - \theta \qquad ... \qquad (1)$$

$$o = \Psi(S) \qquad ... \qquad (2)$$

The change in every input causes a certain change in the output of neuron, and the amplitude of this change is dependent to the connection benefits that determine the input's effect degree, the collector's threshold value and neuron activation type. Here the benefits shown by w_i are named as weight, θ is named as threshold and Ψ function is named as neuron activation function.

2.1 Multi Layer Perceptron Neural Networks

Multi Layer Perceptron (MLP) is the most common neural network model, consisting of successive linear transformations followed by processing with non-linear activation functions. MLP represents a generalisation of the single layer perceptron, which is only capable to construct linear decision boundaries and simple logic functions. However, by cascading perceptrons in layers complex decision boundaries and arbitrary Boolean expressions can be implemented. MLP is also capable to implement non-linear transformations for function approximations (Lippmann, 1987; Geva and Sitte, 1992; Hush and Horne, 1993; Haykin, 1999).

The network consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer. Each layer computes the activation function of a weighted sum of the layer's inputs. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The learning algorithm for multilayer perceptrons can be expressed using generalised Delta Rule and gradient descent since they have non-linear activation functions (Zurada, 1985; Rumelhart, 1987; Jondarr, 1996; Riedmiller and Braun, 1997).

In general form of an MLP network, the x_i inputs are fed into the first layer of $x_{h,1}$ hidden units. The input units are simply 'fan-out' units: no processing takes place in these units. The activation of a hidden unit (neuron *j*) is a function f_j of the weighted inputs plus a bias, as given in Eq.3.

$$x_{pj} = f_j\left(\sum_i w_{ji} x_{pi} + \theta_j\right) = f_j(y_{pj}) \quad \dots \qquad (3)$$

where w_{ji} is the weight of input *i* to neuron *j*, x_{pi} is input *i*, that is, output *i* from the previous layer, for input pattern *p* and θ_j is the threshold value. The output of the hidden units is distributed over the next layer of $x_{h,2}$ hidden units until the last layer of hidden units, of which the outputs are fed into a layer of x_o output units (Kröse and Van der Smagt, 1993; Vysniauskas, 1993; Taskin and Guneri, 2005).

3. FINITE CAPACITY PLANNING

Scheduling may be defined as on resource sequencing of operations according to certain target criteria (such as minimizing the number of delayed works or the number of preparations) and determining the start and ending times of the assigned operations. Taking the limitations into the account (such as resource limitations, delays in material supply, regular maintenance, failures etc.) turns the already formed Gantt scheme into a more applicable one, while this process performed in consideration of the system limitations is named as "finite capacity scheduling – FCS"(Zozom, 1998). Finite Capacity Planning (FCP) is defined as establishing the approximate start and ending times of the manufacturing works and capacity loads of the resources based on the outcome of the work scheduling. Medium and long term manufacturing needs should be taken into account for the product groups rather than the product itself. As a result, it is not necessary that the routes be detailed as is the case in the finite capacity scheduling.

Finite planning can be defined as determining the manufacturing schedule in the resources so that the manufacturing can be applied and the functions supporting it can operate synchronously. This module consists of two sub-modules namely finite capacity scheduling and capacity planning. Finite capacity scheduling guides the manufacturing in the workshop by forming detailed schedules. Short term material requirements are determined using the output of this sub-module. Finite capacity planning issues this schedule making use of the approximate data and other schedules for the long term. It provides decision support for medium and long term capacity plans and constructs the material requirements which have a longer supply period than the FCS (Sen, 2000).

In conclusion, this process is called as "Finite Capacity Planning" as it takes into account the real capacities of the entire resources, the time they end and the format of using such resources in each operation.

4. DEVELOPING A FINITE CAPACITY SCHEDULING SYSTEM USING MULTI LAYER PERCEPTRON

4.1 Job Shop Environment

The specifications of the processed workshop level are as follows (Feng et al., 2003):
There are m machines $\{M_1, M_2, ..., M_m\}$ and there are n jobs $\{J_1, J_2, ..., J_n\}$. There is a certain order that the jobs should follow. Each machine has a certain capacity $\{C_1, C_2, ..., Cm\}$. Furthermore, the amounts to be manufactured, i.e. the demands are definite $\{D_1, D_2, ..., D_n\}$.

Each machine can process only and only one job at a certain time. Each job can be processed in a machine at a certain time. For the jobs and machines given, the processing orders of the machines and the jobs are independent from each other. When a job is processed in a machine, the operation is not interrupted. The transfer format of the jobs should be determined during the operation, because, a job should be forwarded to the following machine when the procedure is over.

4.2 Problem Solving Process

FCS process is given in Figure 2. The manufacturing data first enters into the systems. The data are evaluated and the changes are made accordingly. Then training process runs. Before producing the last manufacturing schedule, first manufacturing schedule is provided and revised (Feng et al., 2003).

Before running the multi layer perceptron networks, three technical problems should be solved:

1. Data organization
2. Control of the local minimum solution and
3. Correction of mistakes in the first schedule.

These problems are analyzed below:

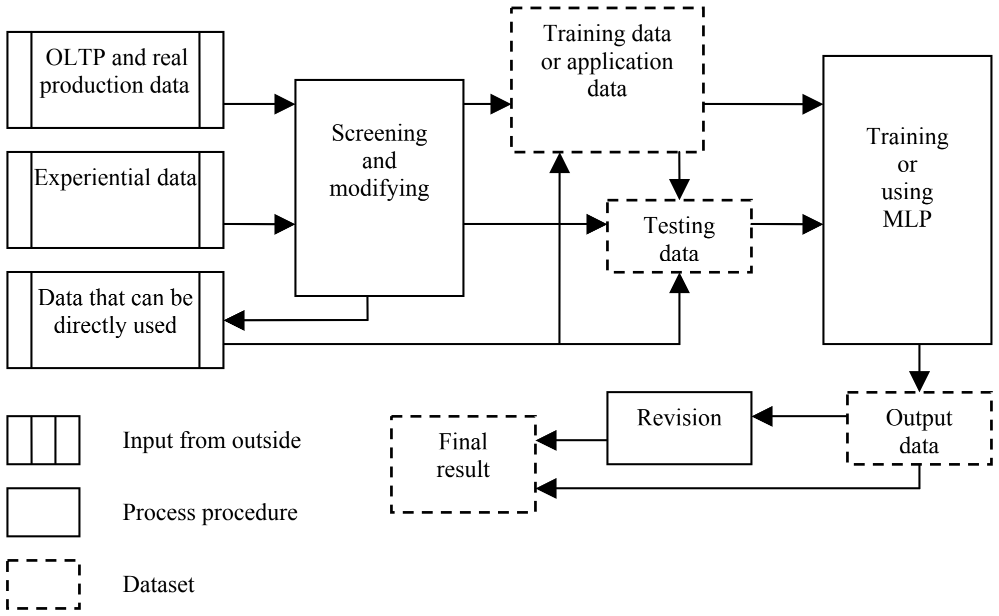

Figure 2. The production scheduling process

4.2.1 The Sample Data

One of the difficulties faced in application of the multiple layer perceptron method is related to organizing the sample data for training and choosing an algorithm suitable for a given problem. Because of this, the first problem to be overcome is deciding as to how to organize the sample data. After assessing the operations in the firm, it was determined that two types of resource data should be acquired (Feng et al., 2003): Corporate database and existing manufacturing data, and experiential data.

After completion of the preliminary processing, the input resource data (database and existing manufacturing data or experimental data) can be used as training data (Feng et al., 2003).

4.2.2 Data Encoding

The input sample data set should include the processing time, processing order, demand and the machines' capacity information of all jobs. The output schedule provides the processing order on each relevant machine for all the jobs. The processing time shows how long a specific process of a job will last on a machine, and the processing order shows the order that the job will move from one machine to another (Feng et al., 2003). The branching tree method is employed to encode this type of process. To improve the encoding process, the hybrid bit encoding method is employed. The hybrid bit encoding method represents the processing time and processing order simultaneously with a single bit. The details are as follows (Feng et al., 2003):

1. *The format of sample data:* Input: X (X_1, X_2, X_3, X_4, ..., X_{4n}), where X_{4n-3} is the processing order of the *i*th job and X_{4n-2} is the processing time of the *i*th job. Also X_{4n-1} is the demand information of the *i*th job and X_{4n} is the machine capacity of the *i*th job will be processed on, here $i \in \{1, 2, ..., n\}$, n is the number of jobs.

2. *Sample encoding:* Assume that a job shop is equipped with three machines to work on 2 jobs. The processing order, processing time and demand information for each job are shown in Table 1, and the capacities for each machine are shown in Table 2.

The input sample can be represented as (1 2 3, 3 3 2, 50 75 100, 50 75 100; 2 3 1, 3 1 4, 75 100 50, 50 75 100). The input X_1 = 1 2 3 means job 1 is processed orderly on machine 1, 2 and 3. X_2 = 3 3 2 means the processing time for job 1 on machine 1, 2 and 3. X_3 = 50 75 100 states demand for job 1 on machine 1, 2 and 3. X_4 =50 75 100 shows the capacities of machine 1, 2 and 3. The inputs for X_5, X_6, X_7 and X_8 show the processing order, processing time, demand and machine capacities for job 2. The output from a trained MLP is able to provide the required information regarding what load should be scheduled for each of the three machines for finite capacity scheduling. So, the output Y_j (Y_1, Y_2, ..., Y_m) shows the processing order of a waiting job on machine j (j = 1, 2, ..., m).

Table 1. Processing order, processing time and demand information for two jobs

Jobs	Processing Order-Processing Time (Day)-Demand		
J1	1-3-50	2-3-75	3-2-100
J2	2-3-75	3-1-100	1-4-50

Table 2. Capacity information for three machines

Machine	Capacity (Pieces/Day)
M1	50
M2	75
M3	100

4.2.3 MLP Architecture And Implementation

Parameters: The relevant parameters used in the FCS algorithm are as follows (Feng et al., 2003):
η : learning rate, η >0
β : momentum parameter, 0< β <1
α : oscillation parameter, 0< α <1
θ : valve value
w: matrix of weight values
w(t): weight value between neurons after the *t*th change
Δw: weight change
y: ideal target output of object tier layer
y_i: ideal target output of unit i
$\hat{y}$: real target output
$\hat{y}_i$: real target output of unit i
z: real output of hidden layer
$z_{i:}$: real output of unit i in hidden layer
n: the number of neurons in the input layer
h: the number of neurons in the hidden layer
m: the number of neurons in the output layer
s: training sample assemble
λ : real constant
ψ (*): monotonically increasing real function that is independent of the mapping f(*) which it approaches
ε : a positive constant
g_i(*): real continuous function

System Structure: The MLP has an input layer, a hidden layer and an output layer to implement the mapping y = f(x). The system structure is shown as Figure 3.

In this structure, the input unit transfer each component of the input vector x to the computing units in the hidden layer. The disposal units in the hidden layer implement the following input/output relation (Feng et al., 2003):

$$Z_k = \sum_{j=1}^{n} \lambda^k \psi(x_j + k\varepsilon) + k, \quad k = 1, 2, ..., h. \quad ... \quad (4)$$

Here, Z_k (6) is computed by (6),

$$Z_k = \frac{1}{1+\exp[-(\sum_{j=1}^{n} W_{kj}X_j + \theta_k)]} \quad \ldots \quad (5)$$

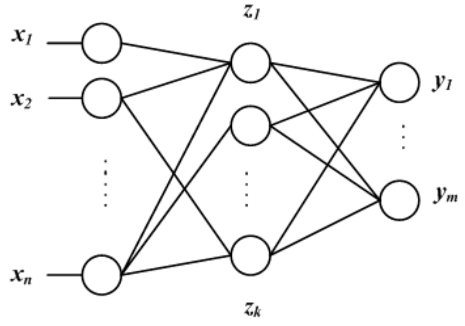

Figure 3. MLP neural network

where W_{kj} is the synaptic weight between the hidden unit k and the input unit j, θ_k is the value of disposal unit k.

The m output unites are provided with the following input/output function:

$$\hat{y}_i = \sum_{k=1}^{h} g_i(z_k), \quad i = 1, 2, ..., m \quad \ldots \quad (6)$$

generally,

$$g_i(z_k) = W_{ik}Z_k \quad \ldots \quad (7)$$

(1) Learning Algorithm

The error function is shown as below,

$$e(w) = \frac{1}{2}\sum_{k=1}^{s} \| f(x_k) - \hat{y}_k \|^2 \quad \ldots \quad (8)$$

For the output layer there is an equation shown by (10),

$$\frac{\partial e(w)}{\partial w_{ik}(t)} = \eta \sum_{r=1}^{s} [(y_i^r - \hat{y}_i^r) z_k^r] \quad i = 1, 2, ..., m; k = 1, 2, ..., h \quad \ldots \quad (9)$$

where r is the sequence number of the sample in the sample assemble, y_i^r the output of component i of the ideal value of the sample, $\hat{y}_i^r$ the output of component i of the actual value of sample r, and y_k^r the output of component k of the actual hidden layer value of sample r (Haykin, 1999). For the hidden layer,

$$\frac{\partial e(w)}{\partial w_{kj}(t)} = \eta \sum_{r=1}^{s} \left[\sum_{i=1}^{m} \left[(y_i^r - \hat{y}_i^r) w_{ik}(t) \right] z_k^r (1 - z_k^r) x_j^r \right],$$

$$j = 1, 2, ..., n+1; w_{k(n+1)}(t) = \theta_k(t), \quad x_{n+1}^r = 1 \quad \ldots \quad (10)$$

(2) Strategy for Adjusting the Weights

$$w(t) + \alpha (w(t^*) - w(t))$$

$$\text{if } w(t+1) = w(t^*) \quad \ldots \quad (11)$$

$$w(t+1) = w(t) - \eta \frac{\partial e(w)}{\partial w(t)} + \beta[w(t) - w(t-1)] = w(t) + \nabla w(t), \quad \ldots \quad (12)$$

where α is a random number $\in (0,1)$.

5. AN INDUSTRIAL IMPLEMENTATION

5.1 The Data

The data used in this study is gained from an international automotive firm's Istanbul factory in Turkey. Here, six components from hood will be taken into consideration and the finite capacity scheduling of this six jobs on five machines will be the desired result. These five machines are as below:

M1: Press of 60 tons for edge cutting,
M2: Hydraulic press of 60 tons for U bending,
M3: Eccentric press of 80 tons for notching,
M4: Press of 80 tones for notching,
M5: Eccentric press of 200 tones for sheet forming,

There are some factors to be considered before scheduling. The first of these is the processing order defined for jobs. It means that the state of each job will be processed on which machine (from technological restrictions). The processing orders of the six jobs are shown in Table 3. The another factor should be considered is the processing time, as seen from Table 4.

Table 3. The processing orders of the six jobs

JOB	PROCESSING ORDER				
J1	M1	M2	M3	M4	M5
J2	M2	M3	M1	M4	M5
J3	M5	M4	M1	M3	M2
J4	M2	M3	M4	M5	M1
J5	M4	M1	M5	M2	M3
J6	M3	M5	M2	M1	M4

Table 4. The processing times of the six jobs on each machine

	PROCESSING TIME (DAY)				
JOB	**M1**	**M2**	**M3**	**M4**	**M5**
J1	3	3	2	4	2
J2	4	3	1	2	2
J3	1	1	2	3	4
J4	3	2	4	3	1
J5	2	3	4	5	1
J6	6	2	1	1	4

Also, the machine capacities and the demand for the jobs are factors to be considered in finite scheduling. The information about these factors shown in Table 5 and 6. As known, the existing capacity and demand must be evaluated for FCS. In this study, the machine capacities are evaluated by assuming that each product will use each source at the same amount.

Table 5. Capacities for each machine

MACHINE	CAPACITY (PIECES/DAY)
M1	300
M2	300
M3	300
M4	300
M5	200

Table 6. Demand for each job

JOB	DEMAND (PIECES)
J1	200
J2	300
J3	325
J4	150
J5	75
J6	250

5.2 The ANN Model for Implementation

There is a finite scheduling problem with six jobs and five machines to find optimal processing orders for each job on each machine by considering capacity constraints. There are the meanings of the symbols that take part in the inputs and outputs of the ANN below:

Input Layer:

Processing Orders of the Jobs: {J1, J2, J3, J4, J5, J6}
Processing Times to Process the Jobs on the Machines: {MJ1, MJ2, MJ3, MJ4, MJ5, MJ6}
Demand for the Jobs: {D1, D2, D3, D4, D5, D6}
Capacities of the Machines: {C1, C2, C3, C4, C5}

Output Layer:

The Processing Orders for Each Machine After Finite Capacity Scheduling: {M1, M2, M3, M4, M5}

The tangent hyperbolic function is used in the hidden layers of the modeled ANN. The output layer has linear activation function. The network has 23 inputs, 2 hidden layers with 12 neurons and 5 outputs. The mathematical function that the network has offered can be obtained as below.

The mathematical function that the network has offered:

The mathematical function that obtained has shown by (14), (15) and (16). Here x means input, u means the output of the first layer, z means the output of the second layer and z means the last output. Furthermore w_1, w_2, w_3 are the weight matrixes of the first, second and output layers', and b_1, b_2, b_3 are the threshold matrixes of the first, second and output layers'.

$$u = \frac{1}{1+\exp[-(w_1 \times x_1 + b_1)]} \quad \ldots \quad (13)$$

$$z = \frac{1}{1+\exp[-(w_2 \times u + b_2)]}$$

$$= \frac{1}{1+\exp[-(w_2 \times \frac{1}{1+\exp[-(w_1 \times x_1 + b_1)]} + b_2} \quad \ldots \quad (14)$$

$y = w_3 \times z + b_3$

$$y = \frac{w_3}{1+\exp[-(w_2 \times \frac{1}{1+\exp[-(w_1 \times x_1 + b_1)]} + b_2]} + b_3 \quad \ldots \quad (15)$$

Figure 4 gives the training graph of the network developed.

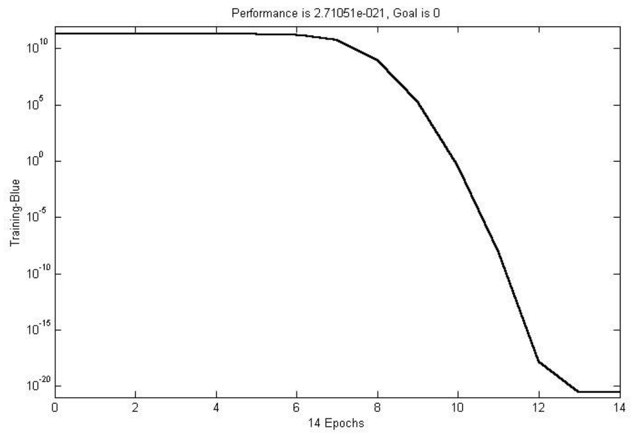

Figure 4. The training graph of developed ANN model

The training data were applied to MLP in Matlab 6.0 Neural Network Toolbox. To structure the ANN model for the FCS problem the parameters below are used:

Maximum training number	: 20000
Sample number	: 20
Training coefficient	: 0.1
Minimum error	: 0
Network structure	: 23-12-12-5
Training number	: 14
Training time	: 15 s

As a result, the processing orders of the jobs on machines are obtained by using ANNs in FCS:

M1: J1, J4, J3, J6, J5, J2;

M2: J2, J3, J1, J5, J4, J6;

M3: J5, J3, J1, J6, J2, J4;
M4: J4, J6, J1, J3, J2, J5;
M5: J6, J5, J4, J2, J3, J1.
where the term M1: J1, J4, J3, J6, J5, J2 means that machine 1 is firstly occupied by job 1, then jobs 4, 3, 6, 5, 2.

6. CONCLUSION

In this paper, the usage of ANNs is examined to realize finite capacity planning. The ANNs are used in finite capacity planning to determine an effective planning strategy.

The results obtained from the network are as an optimal order for FCS. The multi layer perceptron neural networks are used and the weights for each layer obtained by the least squares method in this study. There is a finite scheduling problem with six jobs and five machines to find optimal processing orders for each job on each machine by considering capacity constraints. The tangent hyperbolic function is used in the hidden layers of the modeled ANN. The output layer has linear activation function. The network has 23 inputs, 2 hidden layers with 12 neurons and 5 outputs. As a result, the processing orders of the six jobs and five machines are obtained by using multi layer perceptron neural networks in FCS. The implementation of this study can be adapted to each sector.

As a further work, the inputs applied for the neural network can be expanded for more detailed FCS procedures. For example; some additional inputs can be used as waiting times, tardiness and earliness, transportation, etc., and also it can be assumed that all jobs use different amount of all sources.

7. REFERENCES

1. Akkan, C. (1997). Finite-Capacity Scheduling-Based Planning for Revenue-Based Capacity Management. European Journal of Operational Research, 100: 170-179.
2. Chan, F.T.S., Au, K.C. and Chan, P.L.Y. (2006). A Decision Support System for Production Scheduling in an Ion Plating Cell. Expert Systems with Applications, 30: 1–12.
3. Feng, S., Li, L.X., Cen, L., Huang, J. (2003). Using MLP Networks to Design a Production Scheduling System. Computers and Operations Research, 30: 821-832.
4. Fonseca, D.J. and Navaresse, D. (2002). Artificial Neural Networks for Job Shop Simulation. Advanced Engineering Informatics, 16: 241–246.
5. Geva, S., Sitte, J. (1992). A Constructive Method for Multivariate Function Approximation by Multilayer Perceptrons. IEEE Transactions on Neural Networks, 3(4): 621-624.
6. Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, Prentice Hall, 2nd Edition, New Jersey.
7. Hush, D.R., Horne, B.G. (1993). Progress in Supervised Neural Networks. IEEE Signal Processing Magazine: 8-39.
8. Jain, S., Barber, K. and Osterfeld, D. (1990). Expert Simulation for On-Line Scheduling. Communications of the ACM, 33/10: 54-60.
9. Jansen, K. and Mastrolilli, M. (2004). Approximation Schemes for Parallel Machine Scheduling Problems with Controllable Processing Times. Computers & Operations Research, 31: 1565–1581.
10. Jondarr, C.G.H. (1996). Back Propagation Family Album. Technical Report C/TR96-05. Dept. of Computing, Macquarie University.
11. Kanet, J.J. and Sridharan, V. (1990). The Electronic Leitstand: A New Tool for Shop Scheduling. Manufacturing Review, 3(3): 161-170.
12. Kröse, B.J.A., Van der Smagt, P.P. (1993). An Introduction to Neural Networks. Univ. of Amsterdam.
13. Lippmann, R.P. (1987). An Introduction to Computing with Neural Nets. IEEE ASSP Magazine: 4-22.
14. McCarthy, S.W. and Barber, K.D. (1990). Medium to Short Term l-Mite Capacity Scheduling: A Planning Methodology for Capacity Constrained Workshops. Engineering Costs and Production Economics, 19: 189-199.
15. MirHassani, S.A., Lucas, C., Mitra, G., Messina, E. and Poojari, C.A. (2000). Computational Solution of Capacity Planning Models Under Uncertainty. Parallel Computing, 26: 511-538.
16. Moon, C., Kim, J. and Hur, S. (2002). Integrated Process Planning and Scheduling with Minimizing Total Tardiness in Multi-Plants Supply Chain. Computers & Industrial Engineering, 43: 331-349.
17. Pongcharoen, P., Hicks, C. and Braiden, P.M. (2004). The Development of Genetic Algorithms for the Finite Capacity Scheduling of Complex Products, with Multiple Levels of Product Structure. European Journal of Operational Research, 152: 215-225.
18. Riedmiller, M., Braun, H. (1997). RPROP - A Fast Adaptive Learning Algorithm. Proceedings of ISCIS VII.
19. Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1987). Learning Internal Representations by Error Propagation in Parallel Distributed Processing. ed. D.E. Rumelhart and J.L. McClelland, Vol.1, Cambridge, MA: MIT Press, pp. 318-362.

20. Sen, R. (2000). Combining Infinite Capacity Scheduling and Finite Capacity Scheduling: An Experimental Investigation of an Alternative Scheduling Procedure. PhD. Thesis, The Graduate School of Clemson University.
21. Sia, C.-L. and Ho, Y.-S. (1997). Predictive Capacity Planning: A Proactive Approach. Information and Software Technology, 39: 195-204.
22. Simpson, P.K. (1990). Artificial Neural Systems. Pergamon Press.
23. Tardif, V. and Spearman, M.L. (1997). Diagnostic Scheduling in Finite-Capacity Production Environments. Computers Ind. Engng, 32 (4): 867 878.
24. Taskin, A. and Guneri, A.F. (2005). Economic Analysis of Risky Projects by Artificial Neural Networks. Applied Mathematics and Computation, In Press, Corrected Proof.
25. Tavakkoli-Moghaddam, R., Jolai, F., Vaziri, F., Ahmed, P.K. and Azaron, A. (2005). A Hybrid Method for Solving Stochastic Job Shop Scheduling Problems. Applied Mathematics and Computation, 170: 185–206.
26. Vysniauskas, V., Groen, F.C.A., Kröse, B.J.A. (1993). The Optimal Number of Learning Samples and Hidden Units in Function Approximation with a Feedforward Network. Technical Report CS-93-15, Univ. of Amsterdam.
27. Watanabe, M., Ida, K. and Gen, M. (2005). A Genetic Algorithm with Modified Crossover Operator and Search Area Adaptation for the Job-Shop Scheduling Problem. Computers & Industrial Engineering, 48: 743–752.
28. Yang, S. and Wang, D. (2001). A New Adaptive Neural Network and Heuristics Hybrid Approach for Job-Shop Scheduling. Computers & Operations Research, 28: 955-971.
29. Zozom, A. (1998). A Finite Capacity Job Shop Planning and Scheduling System. PhD. Thesis, The Graduate Faculty of North Carolina State University.
30. Zurada, J.M. (1995). Introduction to Artificial Neural Systems. PWS Publishing, Boston.

BIOGRAPHICAL SKETCH

Ali Fuat Guneri is an Assistant Professor in the Department of Industrial Engineering at Yildiz Technical University, Istanbul. He received his PhD in Industrial Engineering from Yildiz Technical University. He is interested in production planning and control, decision analysis and technology management.

Alev Taskin Gumus is a Research Assistant in the Department of Industrial Engineering at Yildiz Technical University. Dr. Taskin Gumus received her MSc. Degree in Industrial Engineering from Yildiz Technical University, in 2003, MBA degree from Istanbul Technical University, in 2004, and PhD degree in Industrial Engineering fom Yildiz Technical University, in 2007. Her post-doctoral studies mainly focus on supply chain management, production and inventory systems, artificial neural networks, and fuzzy logic applications in industrial engineering and management science.