Proceedings of the
International Conference on Mechanical Engineering 2005
(ICME2005) 28- 30 December 2005, Dhaka, Bangladesh

ICME05-AM-44

ARTIFICIAL NEURAL NETWORK APPROACHES TO MANUFACTURING SYSTEMS ANALYSIS

M. Reaz-us Salam Elias[1] and M. A. A. Hasin[2]

[1]Dept. of Mech. Engg., DUET, Gazipur.
[2]Dept. of IPE, BUET, Dhaka.

ABSTRACT
The power of artificial neural network (ANN) can be utilized to analyze manufacturing systems and to solve various difficult problems, such as, group technology, process planning, production scheduling, combinatorial optimization problems, control chart pattern recognition, etc. This paper presents a brief review on applications of ANN approaches to manufacturing systems analysis, particularly in the area of production scheduling and proposes an ANN model to map Johnson's *n* jobs on two machines problem in order to obtain an optimal sequence for a flow shop scheduling.

Keywords: Manufacturing systems, Neural networks, Flow Shop Scheduling, Johnson's algorithm

1. INTRODUCTION
Manufacturing industries, in recent years, have been experiencing an unprecedented degree of changes – global competition, shortened product life cycle, changes in management, increasing requirements for quality, increasing customer expectations, fast-paced advances in technology, and rapidly expanding options in manufacturing processes and materials [2]. Decision-making process in such manufacturing system environment is becoming increasingly difficult and overwhelming to humans. Since Artificial Neural Networks (ANN) are capable of learning and adapting to new environment with little human intervention and can mimic human thought processes [3], much effort has been concentrated to solve complex problems of manufacturing systems using ANN.

The objective of this paper is to focus on ANN approaches to several manufacturing systems analysis, with special emphasis on production activity scheduling (PAS). Since an effective production scheduling can reduce production costs by 10 to 15%, doubling the profit margin of a company, reduce inventory costs by 8 to 10% and increase "on-time" delivery to customers by 30% [2], PAS plays an important role in manufacturing systems analysis. We also provide an ANN model to find optimal job sequence for Johnson's *n* jobs on two machines problems. The organization of this paper is as follows. Section 2 describes an overview of an artificial neuron and neural networks architecture; section 3 provides neural networks applications in manufacturing systems with particular focus on production scheduling; section 4 discusses flow shop scheduling and Johnson's problem; section 5 provides an ANN model; section 6 provides its verification; section 7 gives conclusion and future directions.

2. ARTIFICIAL NEURAL NETWORKS
Inception of artificial neural networks (ANNs) are motivated by biological nervous systems. It is generally understood that all biological neural functions, including memory, are stored in the neurons and in connections between them, and learning is viewed as building of new connections between neurons or modification of existing connections [5]. An Artificial neuron (figure 1) has also a set of synapses or connecting links, each of which is characterized by a weight or strength of its own. An adder sums the input signals, weighted by the respective synapses of the neuron. An activation function limits the amplitude of the output of a neuron. Mathematically,

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \text{ and the output } y_k = \varphi(v_k)$$

Here, v_k = activation potential of neuron k and x_1, x_2, …x_m = input signals, and w_{k1}, w_{k2},…w_{km} are synaptic weights of neuron k. x_0 =1 and w_{k0} = bias of neuron k.

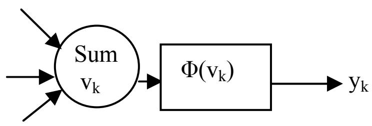

Fig 1. Model of a neuron

Layers of neurons form neural networks (figure 2). A neural network is comprised of a system of individual neurons with weighted interconnections. Neural network architecture consists of an input layer to receive input data and an output layer to present the result of its operation. Layers that lie in between are called hidden layers. Neurons (also known as Processing Elements (PEs) or Nodes) in each layer are interconnected depending on the topology of the network [12].

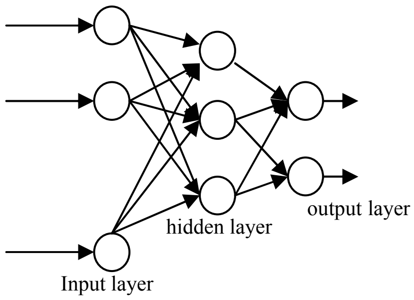

Fig 2. Neural network architecture

Neural networks may be classified as *feedforward* network and *feedback (or recurrent)* network depending on the directions of flow of signals. In feedforward network (e.g. multi-layer perceptrons (MLP)), signals proceed in only one direction from input layers through hidden layers to output layers. Feedforward networks are again classified as fully connected or partially connected between the layers of neurons. In recurrent networks (e.g. Hopfield network), signals may propagate from input layer to output layer and also from output layer to input layer.

3. APPLICATION OF NEURAL NETWORKS IN MANUFACTURING SYSTEMS ANALYSIS

History of ANN may be traced back from 1943. McCulloch and Pitts provided first ANN model based on threshold (i.e. step) activation function [6]. Since then many researchers have contributed a lot to enrich the field of ANN. Dr. John Hopfield is one of such pioneers, who first proposed and applied ANN approaches to solve Traveling Salesperson Problem (TSP) – a NP (Non Polynomial) complete, constraint satisfaction problem- in which a salesperson has to make a closed tour through a certain number of cities, visiting each only once, while minimizing the total distance traveled. It took almost 40 years (from McCulloch and Pitt's neuron (1943) to Hopfield network (1982)) for neural networks to be applied actually to optimization problems like TSP. Hopfield's NN model and its application to optimization problem has encouraged many other researchers to apply ANN to solve complex manufacturing systems problems.

Zhang and Huang [3] have made a state-of-the-art survey on applications of neural networks in manufacturing systems. They have shown seven specific manufacturing areas where neural networks have been used as a tool for analyzing manufacturing systems. These are: (i) group technology (ii) engineering design (iii) monitoring machining processes, tool conditions and fault diagnosis (iv) process modeling and control (v) quality assurance (vi) scheduling (vii) process planning.

3.1 Application of ANNs in Production Scheduling

Scheduling is the allocation of resources over time to perform a collection of tasks [1]. Production scheduling is a crucial stage in manufacturing and plays an important decision making role in manufacturing. An effective scheduling can reduce both production and inventory costs and also increase "on-time" delivery to customers.

Scheduling problem is scientifically proven to be one of the difficult problems (NP-complete, i.e., amount of computation time grows exponentially with the growth of problem size) that can be met in optimization theory [2]. Usually there are *n* jobs to be processed on *m* machines in a predetermined sequence of workflows. Workflows may be unidirectional (known as flow shop scheduling) or may not be unidirectional (known as job shop scheduling). The commonly used performance criteria for production scheduling problem include makespan (total time to completely process all jobs), flow-time (average time of jobs in shop), average work-in-process (WIP) inventory, etc. In the notation n/m/A/B, n refers to number of jobs, m refers to number of machines, A is the flow of pattern (e.g. J for job shop, F for flow shop), B is the optimization criteria (e.g. minimizing makespan, minimizing actual flow time, etc). For example, $6/5/J/C_{max}$ represents 6 jobs to be processed on 5 machines in a job shop scheduling where C_{max} refers to minimization of makespan.

Traditional procedures of solving scheduling problems may be classified into three categories: (i) deterministic approaches, such as branch and bound (ii) heuristic procedures, and (iii) integer linear programming [1]. Recently neural networks and several other artificial intelligence (AI) techniques have been applied to solve production scheduling problems.

Neural network techniques in production scheduling can be classified into two structures/ types: stand-alone ANNs and hybrid ANNs [2]. Most stand-alone ANNs are based on (a) Hopfield model or its extensions or (b) Multi-layer Perceptron (MLP) based on back error propagation.

Hopfield neural network is a recurrent network (each neuron has feedback from each of other neurons, except to its own) with only one layer. The core of the Hopfield neural network is to minimize an energy function (which should obey the stability conditions of Lyapunov). Effective mapping of production scheduling problems into Hopfield NN depends on constructing an appropriate cost or energy function that includes both *strong* constraints (for example sequence constraints, resource constraints) as well as *weak* constraint, i.e. the objective function (for example, minimization of makespan or minimization of amount of time the jobs spend in the shop, etc.). Examples of using Hopfield model for solving scheduling problems include Foo and Takefuji [7], Zhou *et al* [10], Arizono *et al* [11], Satake *et al* [13], etc.

Multilayer perceptron (MLP) (figure 2) was first developed by Frank Rosenblatt. MLP trained by backpropagation algorithm has been applied to solve production scheduling problems. Jain and Meeran [4], Feng *et al* [9], are among of many other researchers who applied backpropagation MLP to solve medium ($6/5/J/C_{max}$) to large ($30/10/J/C_{max}$) problems. Willems and Rooda [14] mapped integer linear programming (ILP) of job shop scheduling problems into a three layer ANN model with feedback connections to implement sequence constraints and resource constraints. Yang and Wang [15] proposed a two layer ANN that mapped ILP of job shop scheduling. They have also combined heuristics to ANN in order to minimize the makespan.

Hybrid ANNs are the combinations of ANN with expert systems or genetic algorithms to overcome the weakness of stand-alone ANNs.

4. FLOW SHOP SCHEDULING

This shop contains *m* different machines and each job consists of *m* operations, each of which requires a different machines. Each operation after the first has exactly one direct predecessor and each operation before the last has exactly one direct successor. Such type of structure is known as *linear* precedence structure. The flow of work is unidirectional. The machines in a flow shop are numbered 1, 2, 3, … , *m*; Each job can be treated as if it had exactly *m* operations, for in cases where fewer operations exist, the corresponding processing time can be taken to be zero [1]. The operations of job *j* are correspondingly numbered (*j,* 1), (*j,* 2), … , (*j, m*).

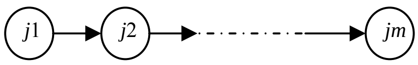

Fig 3. Workflow in a "pure" flow shop

4.1 Johnson's Problem

The two-machine flow shop problem with the objective of minimizing makespan is known as Johnson's problem. The results originally obtained by Johnson are now standard fundamentals in the theory of scheduling [1]. In the formulation of this problem, N is a set of *n* jobs, N = {1, 2, 3, …., j, …n}. M is a set of machines and M = {1, 2}. Job j is characterized by processing time t_{j1} required on machine 1 and t_{j2} required on machine 2 after operation on machine 1 is complete. An optimal sequence can be obtained by the following rule for ordering pairs of jobs: (Johnson's rule) job *i* precedes job *j* in an optimal sequence if $\min\{t_{i1}, t_{j2}\} \le \min\{t_{i2}, t_{j1}\}$.

4.2 Sample problem

Table 1: depicts a five-job problem taken from [1].

Table 1. Processing time of five jobs in two machines

Job *j*	1	2	3	4	5
t_{j1}	3	5	1	6	7
t_{j2}	6	2	2	6	5

The number of possible sequences is 5! = 120. One may apply exhaustive method to enumerate all possibilities and find an optimal sequence to obtain optimal makespan. If one more job is added, the number of possible sequences is 720 and enumerating all possibilities would be time-consuming, not logical and inefficient. Johnson's algorithm provides an elegant way to find an optimal sequence so as to minimize makespan.

In this paper, a fully-connected feedforward neural network model has been proposed to map Johnson's $n/2/F/C_{max}$ problem.

5. THE PROPOSED ANN MODEL

The proposed NN has two layers. Input layer has n neurons (Figure 4). Connection weights for interconnections are assumed to be 1. Each neuron in input layer represents a job. Output layer has 2 neurons - each representing machine 1 and machine 2 respectively. Each neuron receives two 'encoded' inputs. The encoding 'jmt_{jm}' contains three information - job *j* ε N, machine *m* ε M and t_{jm} is the processing time of job j on machine m. For example, '517' represents job number 5 on machine 1 and processing time of job 5 on machine 1 is 7 units. A neuron in an ANN can act as controlling device [12] that can alter its output according to activation function; therefore the activation function of each neuron at input layer may be formulated as follows:

$$\varphi_j(\cdot) = \min_j (t_{j1}, t_{j2}) \qquad j \in N$$

This function embedded in each neuron at input layer

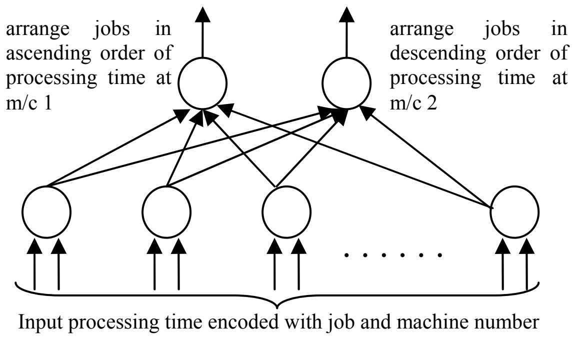

Fig 4. ANN architecture to find optimal sequence of jobs for Johnson's *n* jobs-2 machine problem

will pass the job with minimum of processing time to the output layer.

First neuron at the output layer receives inputs from first layer but concerns only with machine 1. This neuron will filter only those inputs encoded with $j1t_{j1}$ and arrange the output in *ascending* order of processing time t_{j1}. Second neuron at the output layer receives inputs from the first layer but concerns only with machine 2. This neuron filters those inputs encoded with $j2t_{j2}$ and gives the output in *descending* order of processing time t_{j2}.

6. VERIFICATION OF THE PROPOSED MODEL

Figure 4 illustrates the application of proposed NN model on a $5/2/F/C_{max}$ problem. Each node at input layer represents job. There are five neurons at input layer. The encoded inputs contain job number, machine number and respective processing times. (Table 1 gives the processing times for each job on each machine.)

Each neuron at input layer fires for minimum processing time. For example, first neuron at input layer gives the output '113', since processing time (t_{j1}= 3) for job 1 on machine 1 is less than processing time (t_{j2} = 5) for same job on machine 2. In similar fashion, second, third, fourth and fifth neuron fire for '222', '311', '416', '525' respectively. If a job has same processing time on both machines (for example, job 4 has same processing time on both machines), assigned neuron for that job fires for machine 1.

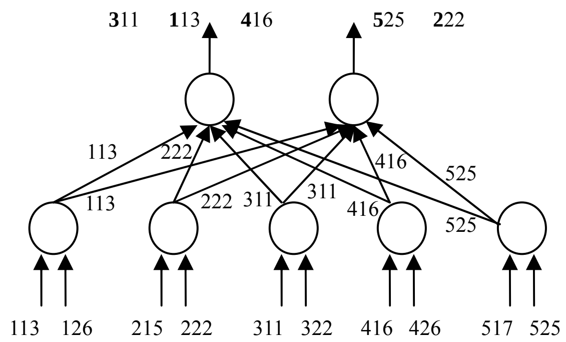

Fig5. ANN architecture to find optimal sequence of jobs for 5/2/F/Cmax. The optimal sequence of job is 3-1-4-5-2

First neuron at output layer processes its input arrange its output in ascending order of processing time on machine 1. Second neuron at output layer processes its input and arranges its output in descending order of processing time on machine 2. The optimal sequence of jobs should be read from left to right (figure 5) and it is 3-1-4-5-2; first on machine 1, then on machine 2. The Gantt chart (figure 6) now can be built up according to this sequence. Makespan has been found to be 24.

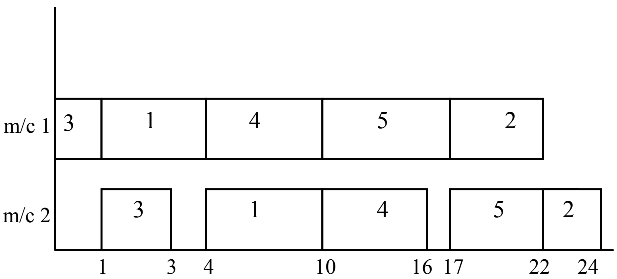

Fig 6. Gantt chart for optimal sequence of jobs 3-1-4-5-2

7. CONCLUSIONS

In this paper, various area of application of ANN in manufacturing systems analysis has been focused and an ANN model has been proposed to solve Johnson's *n* jobs on two machines problem. In case a flow shop scheduling, the proposed architecture will find the optimal sequence of any number of jobs on two machines accurately. The architecture also implements the Johnson's algorithm for *n* jobs on two machines efficiently.

There are limitations also. Interconnection weights of the proposed ANN model are set to a fixed value of 1. In order to build prior information into the network, such restriction has been imposed on the weight connections [5]. However, usually NN has dynamic interconnection weights. The model is dedicated to solve job sequences on two machines only. If one or more machines are added, the model would not be able to provide an optimal sequence. Further research may be conducted to search the ways to overcome such kind of limitations.

8. REFERENCES

1. Baker, R. K, 1974, *Introduction to Sequencing and Scheduling*, John Willey & Sons.
2. Metaxiotis, K. and Psarras, John, 2003, "Neural Networks in Production Scheduling: Intelligence Solutions and Future Promises," Applied Artificial Intelligence, vol. 17, 361 – 373.
3. Zhang, H. C. and Huang, S. H., 1995, "Application of neural networks in manufacturing: a state-of-the-art survey", International Journal of Production Research, vol. 33, no. 3, 705-728.
4. Jain, A. S. and Meeran, S., 1998, "Job-Shop Scheduling using Neural networks," International Journal of Production Research, vol. 36, no. 5, 1249-1272.
5. Hagan, M. T., Demuth, H. B. and Beale, M., *Neural network Design*, PWS Publishing Company.
6. Haykin, S., *Neural Networks A comprehensive Foundation*. Pearson Education Ltd.
7. Foo, Y. S. and Takefuji, Y., 1988, "Stochastic neural networks for solving job-shop scheduling: Part 1. problem representation," *Proc. IEEE 2nd Int. con. on Neural Networks*, vol II, 275-282.
8. Hong, W., 2005, "Flexible flow shop scheduling: optimum, heuristic and artificial intelligence solutions," Expert Systems, vol. 22, No. 2, 78 – 85.
9. Feng, S., Li L., Cen, L. and Huang J., 2003, "Using MLP networks to design a production scheduling system," Computers & Operations Research, vol. 30, 821- 832.
10. Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Olsen, D. E., 1991, "A neural network approach to job shop scheduling," IEEE Transaction on Neural Networks, vol. 2, no. 1, 175 – 179.
11. Arizono, I., Yamamoto, A. an Ohta, H., 1992, "Scheduling for minimizing total actual flow time by neural networks," International Journal of Production Research, vol. 30, no. 3, 503 – 511.
12. Rao, Harish A and Gu, P., 1994, "Expert self-organizing neural network for the design of cellular manufacturing systems," Journal of Manufacturing Systems.
13. Satake, T., Morikawa, K., and Nakaruma, N., 1994, "Neural network approach to minimizing the makespan of the general job-shop," International Journal of Production Economics, vol. 3, 67-74.
14. Willems, T. M. and Rooda, J. E., 1994, "Neural networks for Job-shop scheduling," Control Engineering Practice, vol. 2, No. 1, 31-39.
15. Yang, S. and Wang, D., 2000, "Constraint satisfaction adaptive neural network and heuristics combined approaches for generalized job-shop scheduling," IEEE transactions on Neural Networks, vol. 11, No. 2, 474 – 486.