Original Article

Bulletin of Pure and Applied Sciences.
Vol. 38E (Math & Stat.), No.1, 2019. P.62-71

Available online at
www.bpasjournals.com

Print version ISSN 0970 6577
Online version ISSN 2320 3226
DOI: 10.5958/2320-3226.2019.00007.9

A FLOW SHOP SCHEDULING ALGORITHM BASED ON ARTIFICIAL NEURAL NETWORK

Harendra Kumar[1] & Shailendra Giri[2,]**

Author Affiliation:
[1]Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar, Uttarakhand 249404, India
E-mail: balyan.kumar@gmail.com
[2]Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar, Uttarakhand 249404, India
E-mail: shailly7777@gmail.com

***Corresponding Author:**
Shailendra Giri, Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar, Uttarakhand 249404, India

E-mail: shailly7777@gmail.com

Received on 17.11.2018, **Revised on 17.01.2019** **Accepted on 25.02.2019**

Abstract: In current years, artificial neural networks (ANNs) are of interest to researchers in many areas for different reasons. It has proved to be a good tool to solve many varieties of problems. This paper suggests an idea for the n jobs and m machines flow shop scheduling problems by applying the artificial neural network technique. The major purpose of this paper is to get the job sequence that minimizes the makespan. The performance of our suggested neural network system is compared with the immediate methods which are taken from different papers. A comparison of the procedure developed by us here with the other available methods available in the literature is also provided in this paper.

Keywords: Flow shop scheduling, artificial neural networks, sequence, makespan

1. INTRODUCTION

Scheduling is concerned as a decision- making operation to optimize one or distinct criteria in production and employment industries to allocate the resources to tasks through a specified time interval. The problem of flow shop scheduling (FSS) remains an attractive section of analysis for more than the last sixty years ever since Johnson [16] expects the two level scheduling problems for reducing the makespan. The development can be the maximization (profit) or minimization (cost) associated with the problem. The FSS target is to achieve an effective job sequence that reduces the makespan i.e. the necessary time for concluding all the jobs. A literature survey informs us that the fact-findings for flow-shop scheduling aim at minimizing the makespan. The issue of FSS originates if there is an uninterrupted progress of evolution jobs for several machines. To identify the flow shop a univalent flow of work is taken, i.e. all jobs through the machines have the uniform processing order. Ignall & Schrage [13] developed an ideal job sequence intended for the FSS problem consisting of 'm' machines and 'n' jobs. Palmer [25] suggests a slope grade sign to the sequence of jobs on the machines which depends on the working time. Garey et al. [10] describe that FSS allots with dealing of jobs via machines in an especial manner to improve a given benchmark. Johnny and Yih-Long [15] described a novel enhancement heuristic for explaining the FSS problem.

The development of ANN began close to 65 years ago, encouraged by affection to attempt both to understand the brain and to follow some of brain's strengths. The chronicle of neural networks starts in the early 1940's and thus closely concurrently with the record of programmable electronics computers. Arizono et al. [3] apply the stochastic NN model to solve a scheduling problem for reducing the entire flow time by applying the Gaussian machine model. Sabuncuoglu and Gurgun [28] develop a novel ANN procedure to explain the scheduling problem for the particular machine mean lateness and reducing the makespan for job and machine sequencing problem. Koulamas [18] presented a simple useful heuristic for the flow-shop makespan problem which is capable of formulating non-permutation schedules when it considers it appropriate. Jain and Meeran [14] describe that ANNs are being touted as the wave of the future in computing. El-Bouri et al. [8] examine a procedure related to the ANN based technique for particular job and machine sequencing problems. Lee and Shaw [23] consider the classic problem of jobs and machine for sequencing cognition with the advancement of two stage ANN. Feng et al. [9] suggest a new technique for the problem of job shop scheduling by using the application of ANN with an intention of a deterministic time-changing require pattern over an established planning horizon. Akyol [2] considers the advantage of the ANN technique over the pattern of six distinct algorithms tested to the ‘n’ jobs, ‘m’ machines real FSS problem including the intention of reducing makespan. Haq and Ramanan [12] analyse the sequencing of jobs that arrive in a flow shop in novel associations over time. Rouhani et al. [27] present a new technique for the scheduling problem of exponential type by adopting the perception of neural networks. Modrak and Pandian [24] present an algorithm dependent on changing an m-machine problem in to 2-machine problem. Ramanan et al. [26] use a neural network procedure with feed forward back propagation for solving the FSS problem with the intention to attain a sequence of jobs to decrease the makespan. Ahmad and Khan [1] present a technique for acquiring an optimal scheduling of n-jobs and m-machines FSS problem involving transportation time between jobs by using the concept of neural networks. Chaudhry and Khan [7] present a genetic algorithm based on spread sheet for the no wait FSS problem to minimize the makespan. Baskar and Anthony [5] propose an analytical method based on the Pascal’s Triangle for FSS problem to find a sequence of job having optimal or near optimal makespan. Kumar and Singhal [20] investigate an optimal scheduling FSS problem by using the genetic algorithm approach with lot splitting and sequence dependent setup time. Singhal and Hemrajani [30] propose an algorithm which is achieved by revising the NEH algorithm and gives minimum makespan. Krishna and Vijay [19] analyse and describe one of the determined methods for two machines scheduling problem and generate the job sequence by applying the Johnson’s method in multi stages. Sache [29] proposes an algorithm which consist of optimization of primary rate of starting time to reduce the entire weighted completion time of the jobs in an organization. Zini and Bernoussi [31] developed an algorithm related to particle swarm optimization for hybrid FSS problem to decrease the makespan. Baskar [4] tried to enhance the makespan by developing NEH algorithm. Recently, Kangarloo et al. [17] study the arrangement of flexible FSS problem with machine suitability, machine conditional setup time, machine breakdown to minimize the total weighted tardiness and earliness. Centobelli et al. [6] formulate a new flow shop scheduling algorithm in sequence to process a greater number of ranks by changing the FIFO logic for the drops events of a dealer company on a daily basis. Gonzalez-Neira et al. [11] make a review of the FSS problem in uncertainties and the role of FSS in manufacture logistics. Kumar et al. [21] propose a novel mixture optimization algorithm related to branch and bound procedure by applying genetic algorithm (GA) for solving the FSS problem. Lebbar et al. [22] analyse the blocking FSS problem with makespan condition and then present two mutative optimization approaches specifically the GA and simulated annealing GA.

The remainder part of the paper is organized as follows. In the second section we presents a brief review of the “problem description”. In section 3 the notations are given and in section 4 the detailed description of the proposed artificial neural network model is given. In section 5 computational results and comparisons are presented. Finally in section 6 conclusions are discussed.

2. PROBLEM DESCRIPTION

Many operational situations require the dealing of some jobs on certain machines and then the FSS problem arises. In flow shop scheduling, the job should be processed on the machine in the uniform technological order. In the FSS problem each job from the job set $I = \{1, 2, 3, ..., n\}$ has to be prepared on m machines set $J = \{1, 2, 3, ..., m\}$ in the sequence specified through the catalogue of the machines.

Generally when n jobs and m machines flow shop scheduling problem is considered, it will give $(n!)^m$ possible sequence to find the makespan, which is very difficult, time consuming and some time impossible to calculate. Then after some betterment of the above problem, it simplifies and reduces to $n!$ sequences. As specified in numerous investigations for this kind of difficulty such as of the exponential kind, it is essential to develop simple and less complex ways or algorithms to determine it. According to these occurrences and for decreasing the difficulty of problem issues, the goal of the present paper is to acquire the best sequence of jobs in order that the makespan (C_{max}) of the problem is decreased. In this paper, a heuristic algorithm is developed to find the best job sequence for flow shop scheduling problems by using single layer neural network.

3. NOTATIONS

Following notations are used throughout this paper:

M_m : Machine m, $m = 1, 2, 3, ...$

J_n : Job n, $n = 1, 2, 3, ...$

x_i : Input Neurons (Units), $i = 1, 2, 3, ...$

y_j : Output Neurons (Units), $j = 1, 2, 3, ...$

w_{ij} : Weights on connection links from inputs neurons to output neurons

P_{ij} : Processing time of job i on machine j

y_{-in} : Net input to output unit neurons Y

y_k : Net output from output neuron

F(x) : Activation function

exp : Exponential

RPD_i : Relative percent deviation of problem i

C_{max} : Makespan (total completion time)

C_{RMS_i} : Makespan obtained by reference method of problem i

C_{PMS_i} : Makespan obtained by proposed algorithm of problem i.

ANN : Artificial Neural Network

FSS : Flow Shop Scheduling

4. DESCRIPTION OF THE PROPOSED ARTIFICIAL NEURAL NETWORK APPROACH

The simplified patterns of the biological nervous structures are neural networks and therefore have drawn their inspiration from the type of computing achieved by the human brain. A neural network is an accumulation of highly interconnected processing units (neurons) that have the capability to understand and thereby gain knowledge and make it accessible for use.

During the preceding decade, for determining optimization problems many neural network models are developed. During the latest years, high-tech enhancements in software system and apparatus have supported the development of utilized equipments like ANN, which are applied in determining many type of difficult problems. The applications of ANN encompass several zones of scheduling, like, finding the sequence of jobs, mentioning perfect job sequence, makespan and checking the strategy for sharing of jobs etc.

A feed forward ANN with single layer is assumed in this research. In this the network through connection links signals are moved in the direction of output layers from the input layers and instruction is completed by classifying the connection loads. This paper elaborates one of the main points of the definition of input and output layers. In this study to perform the ANN analysis, we describe the structure of a new ANN model for FSS problems to find a job sequence with the objective of minimizing the makespan. In the proposed model, we have solved *n* jobs on *m* machines FSS problem. In accession to the architecture of ANN model the manner of setting the measures of the weights (preparation) is an important discriminative characteristic of various neural nets. For convenience, we shall distinguish two kinds of preparing supervised & unsupervised for a neural network in addition there are nets whose weights are fixed without a reiterative training process. There is some ambiguity in the tagging of training methods as supervised or unsupervised. In the current study, supervised learning is applied to prepare the network model. In supervised training every output archetype that is applied to lead the network is related with an output pattern, which is the objective or the desired pattern. The supervised learning principle for ANN model is alike the preceptor learning algorithm. In this method weights are arranged according to a learning algorithm. An artificial neuron admits inputs from the input neurons then by trained model it gives the final output. First of all by supervised learning process we find the weights on each connection link, which is shown in figure 1.

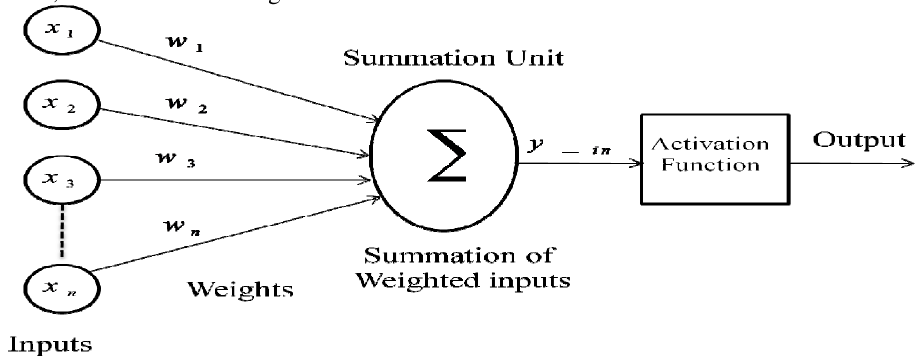

Figure 1: The artificial neural network model

To find the output for the n^{th} job from the job set $\mathrm{I} = \{1, 2, 3, ..., \mathrm{n}\}$ has to be prepared on *m* machines set $\mathrm{J} = \{1, 2, 3, ..., \mathrm{m}\}$ in the sequence specified by the indexing of the machines generates an $\mathrm{n}\times\mathrm{m}$ dimensional problem. All *n* jobs have to be processed on each machine one by one. The processing times which are taken by each job on all machines will become the input values for our proposed neural network model, i.e. the processing time of each job on all *m* machines will become the input values for the input neurons $x_1, x_2, x_3, ..., x_m$ respectively of neural network model for that job. After that to find the net input $\mathrm{y}_{-\mathrm{in}}$ from neurons $x_1, x_2, x_3, ..., x_m$ to neuron *Y*, we require weights on each connection links which are connected from input neurons to output neuron *Y*. For finding the weight on each connection link, we arrange the input values (i.e. the processing time on all machines for each job) in increasing order then gives the priority starting from 1 and divides the priority by 100 from which we get the weights for each connection link from neurons $x_1, x_2, x_3, ..., x_m$ to neuron *Y*. Now, we multiply each neurons input value with each weight value on the connection link respectively and sum up them, then this gives the net input to neuron *Y* from the neurons $x_1, x_2, x_3, ..., x_m$. To find the output from neuron *Y*, we take the logistic sigmoid function as the activation function. Then we calculate the output value from the neuron *Y* with the help of this activation function. The architecture diagram of the above mentioned proposed model is displayed in figure 2. We repeat this process for all jobs until there is no job left for sequencing. After getting the output values for all jobs, we arrange these output values in decreasing and increasing order and find two job sequences then we will choose that sequence which produces the minimum makespan.

The work flow of the developed neural network for the FSS problem is displayed in the following figure 2:

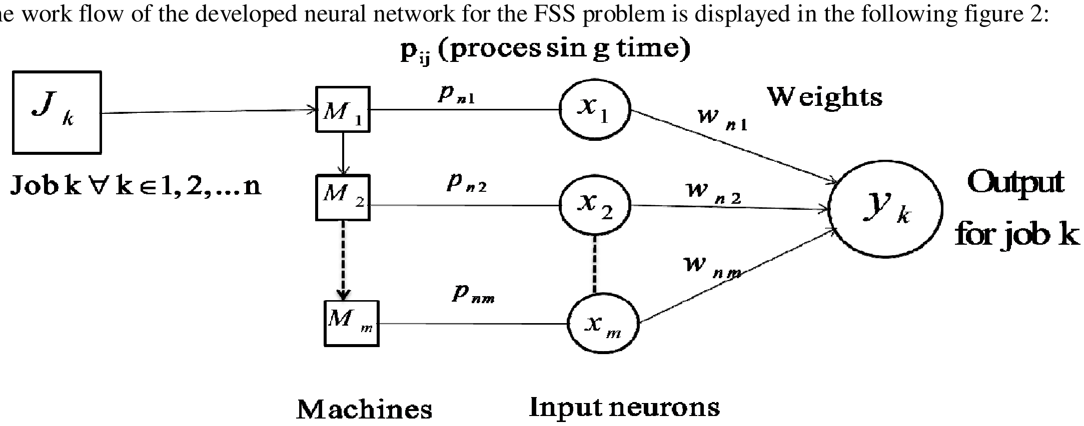

Figure 2: Architecture diagram of proposed neural network

The behaviour of a neuron can be captured by a simple pattern as displayed in figure 3. Every element of model allows a direct comparison to the real components of a biological neuron and therefore is termed an artificial neuron.

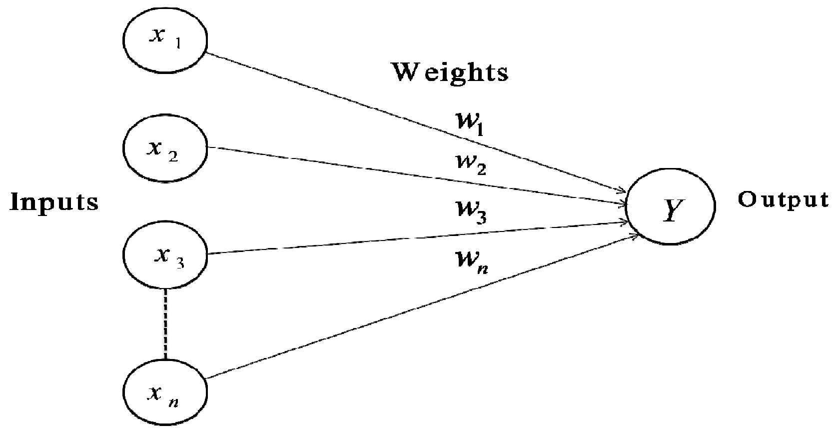

Figure 3: A simple artificial neural network

Let us consider a neuron *Y* in figure 3 that receives input from neurons $X_1, X_2, X_3, ..., X_n$. The activations (output signals) of these neurons are $x_1, x_2, x_3, ..., x_n$ respectively and the weights on the connections from $X_1, X_2, X_3, ..., X_n$ to neuron Y are $w_1, w_2, w_3, ..., w_n$ respectively. The net input y_{-in} to neuron *Y* is the sum of the weighted singles form neurons $X_1, X_2, X_3, ..., X_n$, i.e.

$$y_{-in} = w_1x_1 + w_2x_2 + w_3x_3 + ... + w_nx_n \tag{1}$$

The activation function of neuron *Y* is calculated by using its net input $y_k = f\left(y_{-in}\right)$. In the present study, we are using the logistic sigmoid function as an activation function.

$$f(x) = \frac{1}{1+\exp(-x)} \tag{2}$$

Algorithm: To get the sequence of *n* jobs set $I = \{1, 2, 3, ..., n\}$ on *m* machines set $J = \{1, 2, 3, ..., m\}$ to minimize the makespan.

The various steps involved in the proposed neural network based algorithm are listed as below:

Step 1: Input: number of jobs, number of machines, processing times.

Step2: Generate a $n \times m$ dimensional input matrix by taking the processing times of each job on each machine.

Step 3: Compute the input values x_j of *j*-th neuron for *i*-th job by:

$$x_j = p_{ij},\ j = 1, 2, 3, ..., m$$

Step 4: Find the weights on connection links as:

(a). Arrange the input values x_j in increasing order of their processing times.

(b). Define their priorities starting from 1 to m.

(c). Divide each priority by 100.

Step 5: Calculate the net input y_{-in} by:

$$y_{-in} = \sum_{n=1}^{n} x_n w_n$$

Step 6: Find the output values y_k by using following function:

$$y_k = f\left(y_{-in}\right),\ k = 1, 2, 3, ..., n$$

Where, $f(x) = \dfrac{1}{1+\exp(-x)}$

Step 7: Find the sequence of the jobs by arranging the output values y_k in increasing order.

Step 8: Find the absolute completion time c_{max} for the obtained sequence.

Step 9: End

5. COMPUTATIONAL RESULTS AND COMPARISONS

Consider a FSS problem with 5 machines and 5 jobs. There are 5 jobs to be scheduled and the completion times and processing times are as displayed in Table-1 and Table-2 respectively. In Table-1, for all the combinations of the jobs, we assimilate the absolute completion times for pair's *ij* and *ji*.

Table 1: Completion times of jobs for pairs *ij* and *ji*

Pairs	Completion times	Pairs	Completion times
(1,2)	43	(2,1)	41
(1,3)	40	(3,1)	42
(1,4)	38	(4,1)	36
(1,5)	44	(5,1)	46
(2,3)	42	(3,2)	43
(2,4)	41	(4,2)	39
(2,5)	37	(5,2)	38
(3,4)	45	(4,3)	43
(3,5)	42	(5,3)	44
(4,5)	38	(5,4)	39

Table 2: Processing time for 5 jobs and 5 machines problem

Machines			Jobs		
	J1	J2	J3	J4	J5
M1	7	8	1	5	3
M2	9	6	4	7	5
M3	3	5	2	4	8
M4	5	7	3	1	6
M5	4	2	8	2	7

The processing times of job 1 on machines 1-5 are 7, 9, 3, 5 and 4 respectively. Here the input value of neurons x_1, x_2, x_3, x_4 and x_5 are 7, 9, 3, 5, and 4 respectively. The results of the step -4 are given in table-3.

Table 3: Priorities and weights of neurons

Step	Neurons	x_3	x_5	x_4	x_1	x_2
Step-4: (a)	Processing times	3	4	5	7	9
Step-4: (b)	Priorities	1	2	3	4	5
Step-4: (c)	Weights	$w_3 = 0.01$	$w_5 = 0.02$	$w_4 = 0.03$	$w_1 = 0.04$	$w_2 = 0.05$

Figure 4 depicts the architecture diagram for the weight of each neuron for job 1.

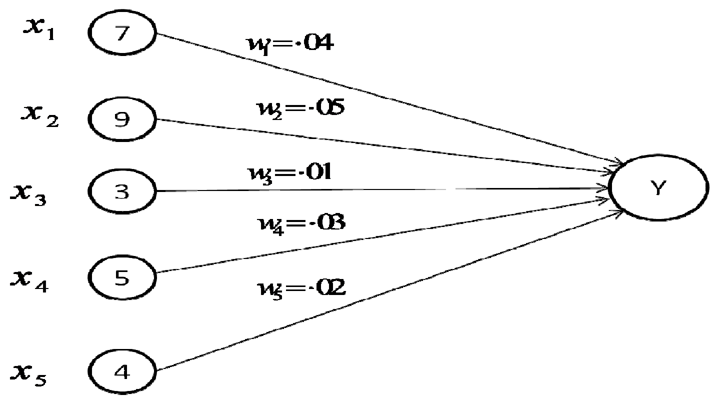

Figure 4: Weight of each neuron for job 1

After applying the steps 5-6, the final output y_1 for the job 1 is 0.729129. Similarly the final outputs values (y_2, y_3, y_4 and y_5) for the job 2, job 3, job 4 and job 5 are 0.727114, 0.668226, 0.672631 and 0.729129 respectively. The jobs are sorted in decreasing order of their final outputs values. The present method yields the sequence 5-1-2-4-3 with makespan $c_{\max} = 49$. The results of the above problem are compared by two well known algorithms, *Aslan's frequency* and *Aslan's point* algorithms reported in the literature**.** From table 4, the present neural based algorithm outperforms the results received by *Aslan's frequency* and *Aslan's point* algorithms [32].

Table 4: Comparison of results obtained by present algorithm with earlier literature results

Present algorithm		**Aslan's frequency algorithm method**		**Aslan's point algorithm method**	
Sequence	Makespan	Sequence	Makespan	Sequence	Makespan
5-1-2-4-3	49	4-2-1-3-5	59	4-2-1-3-5	59

The above table indicates that the outcomes of the present algorithm are quite better in comparison to two other techniques for finding the best jobs sequence which yields the minimum makespan. In order to estimate the achievement of the present neural based algorithm with existing algorithms, another comparison measure, relative percent deviation (RPD) is also adopted. The relative percent deviation is calculated by using the following formula.

$$\mathrm{RPD}_i = \frac{C_{\mathrm{RMS}_i} - C_{\mathrm{PMS}_i}}{C_{\mathrm{PMS}_i}} \times 100, \quad \text{where}$$

C_{RMS_i} = Makespan obtained by reference method solution of problem *i*.

C_{PMS_i} = Makespan obtained by proposed algorithm of problem *i*.

The proposed heuristic is tested against the solution of the different approaches which are taken from different papers (as listed below) to determine the FSS problem with the intention of minimizing the makespan. The outline of the results are specified in Table 5.

Table 5: Comparison of the results obtained by present algorithm with the existing methods

Problem	Reference	Used techniques	By reference method		By proposed method		RPD
			Sequence	Make span	Sequence	Make span	
1.	Derya Eren Akyol [2]	ANN	4-3-1-2	54	1-2-3-4	54	0
2.	Ahmad and Khan [1]	ANN	1-2-3	112	1-3-2	105	6.67
3.	Kumar and Singhal [20]	GA	F-C-D-E-B-A-H-G	261	G-F-C-A-D-H-E-B	229	13.97
4.	Christos Koulamas [18]	HFC Algorithm	2-3-1	22	3-1-2	22	0
5.	Singhal & Hemrajani [30]	Improved NEH	5-4-3-1-2	58	5-4-3-2-1	52	11.53
6.	Saeed Rouhani et. al. [27]	ANN	3-6-4-2-5-1	586	3-1-6-4-5-2	567	3.35
7.	Ramanan et. al. [26]	ANN	4-3-2-5-1	849	3-1-2-4-5	849	0
8.	Chaudhry &. Khan [7]	GA	5-3-2-1-4	28	4-1-3-2-5	25	12
9.	Basker & Anthony [5]	Pascal's triangle	3-1-2-4	30	4-3-2-1	30	0
10.	Krishna & Vijay [19]	Modified Johnson's method	5-4-2-1-3	77	2-1-4-3-5	76	1.31

The results with reference to table 5 show that the present Neural Network algorithm is quite better than the other algorithms used for finding the best job sequence which yields the minimum makespan. The graphical comparison of proposed method with the existing references methods for makespan is also displayed in figure 5. From figure 5, it can be inferred that the recommended method has performed quite better than the other reference methods. The outcomes of the numerical illustrations indicate that the model presented in this paper is suitable for arbitrary number of jobs and machines.

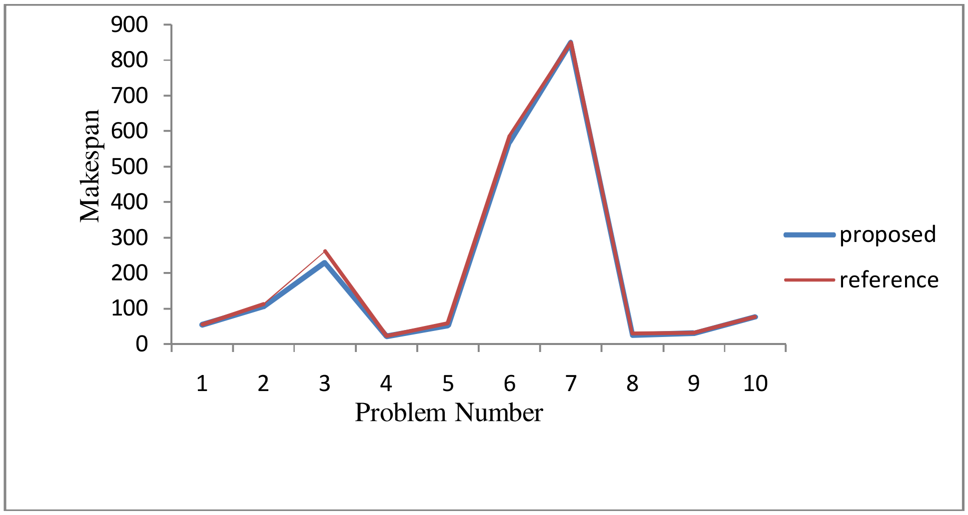

Figure 5: Graphical comparison of the results

6. CONCLUSION

ANN has been effectively applied for various complex and difficult FSS problems. In this paper ANN based method is designed to determine the FSS problem of n jobs on m machines to get the best job sequence which yields the minimum makespan. The main goal of this study is to establish a new and another approach by applying a neural network for the evaluation of the makespan of FSS. The new proposed method takes a very different approach from the other discussed techniques which are taken from different papers as mentioned in the bibliography of this paper. From the outcomes, it is clear that the performance of the new proposed method for best job sequence which provides the minimum makespan is better than the other methods evaluated in different papers as stated above. Hence this suggested technique has a wide opportunity in production where n jobs are needed to be appointed on m machines for larger production, effective planning of assets and maintaining proper command over the industry. This study shows the applicability of ANN to real scheduling issues. The future directions of this research are to occupy other new techniques that will enhance the exactness of the outcomes. For designing the production schedules various ANN frameworks may be applied to anticipate the makespan. Continuations of the suggested method, involving distinct neural network frameworks can be developed for solving complex types of scheduling problems having distinct performance measures.

REFERENCES

[1]. Ahmad, Q. S. and Khan, M. H. (2012). Application of neural network to scheduling problem including transportation time, *International Journal of Computer Application*, Vol.54, No.5, 8-10.

[2]. Akyol, D. E. (2004). Application of neural networks to heuristic scheduling algorithms, *Computers & Industrial Engineering*, Vol.46, No.4, 679-696.

[3]. Arizono, I., Yamamoto, A. and Ohta, H. (1992). Scheduling for minimization total actual flow time by neural networks, *International Journal of Production Research*, Vol.30, No.3, 503-511.

[4]. Baskar, A. (2015). Minimizing the makespan in permutation flow shop scheduling problem using simulation, *Indian Journal of Science and Technology*, Vol.8, No.22, 1-7.

[5]. Baskar, A. and Anthony, X. M. (2012). A new heuristic algorithm using Pascal's Triangle to determine more than one sequence having optimal / near optimal makespan in flow shop scheduling problems, *International Journal of Computer Applications*, Vol.39, No.5, 9-15.

[6]. Centobelli, P., Converso, G., Murino, T. and Santillo, L. C. (2016). Flow shop scheduling algorithm to optimize warehouse activities, *International Journal of Industrial Engineering Computations*, Vol.7, No.1, 49-66.

[7]. Chaudhry, I., A. and Khan, A., M. (2012). Minimizing makespan for a no wait flowshop using genetic algorithm, *Sadhana, Indian Academy of Sciences*, Vol. 37, No. 6, 695-707.

[8]. El-Bouri, A., Balakrishnan, S. and Popplewell, N. (2000). Sequencing jobs on a single machine: A neural network approach, *European Journal of Operational Research*, Vol.126, No.3, 474-490.

[9]. Feng, S., Ling, L., Ling, C. and Huang, J. (2003). Using MLP networks to design a production scheduling system, Computer & Operations Research Vol.30, No.6, 821-832.

[10]. Garey, M. R., Johnson, D. S. and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling, *Mathematics of Operation Research*, Vol.1, No.2, 117-129.

[11]. Gonzalez-Neira, E. M., Montoya-Torres, J. R. and Barrera, D. (2017). Flow shop scheduling problem under uncertainties: Review and trends, *International Journal of Industrial Engineering Computations*, Vol. 8, No. 4, 339-426.

[12]. Haq, A. N. and Ramanan, T. R. (2006). A bicriterion flow shop scheduling using artificial neural network, *Int. J. Adv. Manuf. Technol*, Vol.30, No.11-12, 1132-1138.

[13]. Ignall, E. and Schrage, L. (1965). Application of the branch and bound technique to some flowshop scheduling problems, *Operation Research*, Vol.13, No.3, 400-412.

[14]. Jain, A. S. and Meeran, S. (1998). Job shop scheduling using neural networks, *International Journal of Production Research*, Vol.36, No.5, 1249-1272.

[15]. Johnny, C. H. and Yih-Long (1991). A new heuristic for the n jobs, m machines flow shop problem, *European Journal of Operational Research*, Vol.52, No.2, 194-202.

[16]. Johnson, S. M. (1954). Optimal two and three stage production schedules with setup times included, *Naval Research Logistics Quarterly*, Vol.1, No.1, 61-68.

[17]. Kangarloo, N., Rezaeian, J. and Khosrawi, X. (2016). JIT Scheduling problem on flexible flow shop with machine breakdown, machine eligibility and setup times, *Journal of Mathematics and Computers Science*, Vol.16, No.1, 50-58.

[18]. Koulamas, C. (1998). A new constructive heuristic for the flow shop scheduling problem, *European Journal of Operational Research*, Vol.105, No.1, 66-71.

[19]. Krishna, V. G. and Vijay, P. (2013). A proposed method for permutation flow shop scheduling, *International Journal of Emerging Technology and Advanced Engineering*, Vol.3, No.4, 209-213.

[20]. Kumar, G. and Singhal, S. (2013). Genetic algorithm optimization of flow shop scheduling problem with sequence dependent setup time and lot splitting, *International Journal of Engineering Business and Enterprise Applications (IJEBEA)*, Vol.4, No.1, 62-71.

[21]. Kumar, H., Kumar, P. and Sharma, M. (2018). An effective new hybrid optimization algorithm for solving flow shop scheduling problems, *Malaya Journal of Matematik*, Vol. 5, No. 1, 107-113.

[22]. Lebbar, G., Barkany, A. E., Jabri, A. and Abbassi, I. E. (2018). Hybrid metaheuristics for solving the blocking flowshop scheduling problem, *International Journal of Engineering Research in Africa*, Vol. 36, No. 1, 124-136.

[23]. Lee, I. and Shaw, M. J. (2000). A neural-net approach to real time flow-shop sequencing, *Computer & Industrial Engineering*, Vol.38, No.1, 125-147.

[24]. Modrak, V. and Pandian, R. S. (2010). Flow shop scheduling algorithm to minimize completion time for *n*-jobs *m*-machine problem, *Technical Gazette*, Vol.17, No.3, 273-278.

[25]. Palmer, D. (1965). Sequencing jobs through a multi stage processing the minimum total time a quick method of obtaining a near optimum, *Operation Research*, Vol.16, No.1, 45-61.

[26]. Ramanan, T. R., Sridharan, R., Shashikant, K. S. and Haq, A. N. (2011). An artificial neural network based heuristic for flow shop scheduling problems, *J. Intell. Manuf.*, Vol.22, 279-288.

[27]. Rouhani, S., Fathian, M., Jafari, M. and Akhavan, P. (2010). Solving the problem of flow shop scheduling by neural network approach, Springer-Verlag Berlin Heidelberg, CCIS 88, 172-183.

[28]. Sabuncuoglu, I. and Gurgun, B. (1996). A neural network model for scheduling problems, *European Journal of Operational Research*, Vol.93, No.2, 288-299.

[29]. Sache, M. R. G. (2014). Neural network for solving job shop scheduling problem, *IOSR Journal of Computer Engineering*, Vol.16, No.6, 18-25.

[30]. Singhal, E. and Hemrajani, N. (2013). An improved NEH algorithm applied to permutation flow shop scheduling, *International Journal of Engineering Science and Research*, Vol.2, No.5, 1164-1170.

[31]. Zini, H. and El-Bernoussi, S. (2015). A discrete particle swarm optimization with combined priority dispatching rules for hybrid flow shop scheduling problem, *Applied Mathematical Sciences*, Vol.9, No.24, 1175-1187.

[32]. Aslan, D. (1999). Model development and application based on object oriented neural network for scheduling problem. DEU Res. Fund. Proj., Nr. 0908.97.07.01. University of Dokuz Eylul, Izmir.