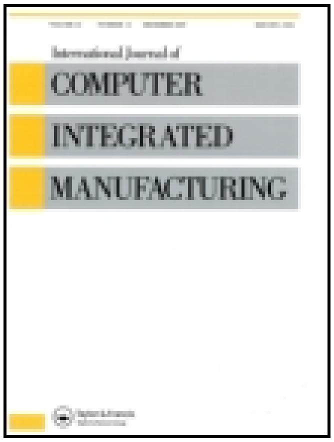

International Journal of Computer Integrated Manufacturing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcim20

An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop

R. Shafaei [a] , M. Rabiee [a] & M. Mirzaeyan [b]

[a] Faculty of Industrial Engineering , K. N. Toosi University of Technology , Tehran, Iran

[b] Faculty of Technical and Vocational Training , Shahid Beheshty , Karaj, Iran

Published online: 29 Jul 2011.

To cite this article: R. Shafaei , M. Rabiee & M. Mirzaeyan (2011) An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop, International Journal of Computer Integrated Manufacturing, 24:10, 888-899, DOI: 10.1080/0951192X.2011.597430

To link to this article: http://dx.doi.org/10.1080/0951192X.2011.597430

International Journal of Computer Integrated Manufacturing
Vol. 24, No. 10, October 2011, 888–899

An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop

R. Shafaei[a], M. Rabiee[a]* and M. Mirzaeyan[b]

[a]*Faculty of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran;* [b]*Faculty of Technical and Vocational Training, Shahid Beheshty, Karaj, Iran*

(*Received 25 November 2010; final version received 12 June 2011*)

In a flow shop scheduling problem, no-wait processing is becoming an important subject where the subsequent operations are performed with no-waiting time in between. In addition, it is important to estimate the schedule performance in terms of given criteria. This enables managers to conduct a reliable logistic planning while estimating robust job completion times. In this study, six heuristic algorithms are used to solve a no-wait two stage flexible flow shop with minimising makespan. Furthermore, an adaptive neuro fuzzy inference system (ANFIS) with neural network and fuzzy theory is applied for estimating the makespan. The robustness of the proposed ANFIS using the six heuristic scheduling algorithms is testified by comparing its performance with that of the actual schedule performance. This is followed by concluding remarks and potential area for further researches.

Keywords: scheduling; no-wait; ANFIS; estimation; makespan

1. Introduction

Production scheduling plays a crucial role in both manufacturing systems and service industry as it leads to efficient utilisation of resources while meeting customer needs (Wang 2003, Stadtler 2005). Among the scheduling problems, flow shop scheduling problem (FSSP) has been widely studied by researchers. This has been considered as a difficult class of problems to solve (Baker 1974, Garey and Johnson 1979, Mohammadi *et al*. 2010, Solimanpur and Elmi 2011).

In practice, one of the most applicable flow shop problems is a flexible flow shop, also known as a hybrid flow shop or FSSP with parallel machines, i.e. flexible flow line. This has been studied by many researchers; among them include those investigated by Zandieh *et al*. (2006), Desprez *et al*. (2009), Mirsane *et al*. (2009) and Luo *et al*. (2010). For a literature review in this area, the readers are referred to Richard and Zheng (1999), Ribas *et al*. (2010), and Ruiz and Vazquez-Rodriguez (2010). In the flexible FSSP, there are a set of n jobs $j = \{1,2 \ldots n\}$ that need to be processed at k working stages ($i = \{1,2 \ldots k\}$). Each stage might have a number of identical machines performing in parallel (m_i). When there is more than one parallel machine at each stage, two major decisions need to be taken: (1) allocation of the machines to the jobs at each stage and (2) jobs sequencing on each machine.

In a flow shop, the scheduling problems are also classified into two categories, namely with and without waiting time in operation intervals. In a flow shop with waiting time, the jobs are processed from one machine to the next allowing waiting time in between, whereas in a no-wait flow shop system, the jobs are processed from one machine to the next without waiting time. Classical FSSP (i.e. the flow shop with waiting times) has been paid more attention by researchers compared with no-wait flow shop problem.

In a no-wait flow shop, it is assumed that there are n jobs each consisting k operations owning a pre-determined processing order through machines. Each job is to be processed without pre-emption and interruption on or between k machines. That is, once a job is started on the first machine, it has to be continuously processed through subsequent machines without interruption. In addition, each machine can handle not more than one job at a time and each job has to visit each machine exactly once. Therefore, when needed, the start of a job on the first machine must be delayed in order to meet the no-wait requirement (Tasgetiren *et al*. 2007). As an example, in a chemical industry, if the waiting time is allowed between each subsequent stage, it may lead to the change in the material property (e.g. degrading the polymer). Therefore, in such industries, once an operation is completed, the subsequent operation shall be started without any delay. The problem studied in

*Corresponding author. Email: meysam_rabiee@yahoo.com

ISSN 0951-192X print/ISSN 1362-3052 online

http://dx.doi.org/10.1080/0951192X.2011.597430
http://www.tandfonline.com

this article is a no-wait flexible flow shop scheduling as it has a variety of practical applications in both production industry and service companies. In the remaining part of this section, the research conducted in this area is reviewed and potential subjects for further research is highlighted.

The no-wait FSSP has been studied in the past several decades. In a no-wait FSSP with a single objective, it is proven that the problem is strongly nondeterministic polynomial time-hard (NP-hard) when the number of machines is more than two (Rock 1984). Reddi and Ramamurthy (1972) and Wismer (1972) have formulated the no-wait FSSP as an asymmetric travel salesman problem. Bonney and Gundry (1976) and King and Spachis (1980) have developed heuristics to minimise makespan. Rajendran and Chaudhuri (1990) have also proposed a tighter lower bound on total flow time. In addition, they proposed two efficient heuristic algorithms to minimise the total flow time. Kalczynski and Kamburowsk (2007) also proved that in a no idle, no-wait flow shop with makespan performance measure, where at each stage the number of machines is more than one machine but less than the number of jobs (e.g. two machines with four jobs), the problem is non-deterministic and NP-Complete. Aldowaisan and Allahverdi (1998) also proposed a heuristic algorithm to solve no-wait flow shops with setup times and minimum total flow time. The proposed algorithms obtained optimal solutions for two specific cases. In addition, the results of the investigation indicate that in general the proposed heuristic algorithm yields good solutions.

Sabuncuoglu (1998) is one who firstly presented a review of the literature and research directions of scheduling approaches using neural network. His study deals with scheduling problems involving artificial neural network (ANN) applications. The objective is to review the literature of applying neural networks to various scheduling problems ranging from a single machine scheduling to satellite broadcasting scheduling. Both the theoretical developments and computational experiences to the date of the article publication were discussed, and potential areas for further research were highlighted.

Sabuncuoglu and Gurgun (1996) combined neural network and algorithmic approaches to solve the single machine scheduling problem with mean tardiness and the job-shop scheduling problem with minimum makespan. Chen and Muraki (1997) offered a standard back-propagation neural network for online rescheduling on the basis of pre-processed information about the shop status.

In a no-wait hybrid flow shop, there are two main subjects to be addressed. The first is to apply an efficient algorithm in order to obtain the best performance measure. Second is to make sure that the schedule to be developed will have satisfactory performance in advance of scheduling. Assume that there are a number of jobs to be processed in a hybrid flow shop. If the problem is solved using any specific algorithm, then its expected performance measure can be calculated. In case if the performance obtained using the best algorithm, available, is not satisfactory, then the problems shall be changed in terms of the number of jobs, the job mix, etc. Having made any change in the problem, the revised problem shall be solved again. This process is repeated till a satisfactory performance measure is obtained. In such an environment, since for each scenario the problem is solved, the process is normally time consuming. To overcome this problem, researchers intend to develop an intelligent system to anticipate the performance measure without solving the problem. The model is firstly trained by solving a number of problems upon which the performance of the subsequent problems can be anticipated directly.

Makespan estimation has been paid attention in the literature on due date assignment (Cheng and Gupta 1989). One of the most popular approaches used in estimating the makespan is neural networks. Philipoom *et al.* (1994) compared a non-linear regression analysis with neural networks in due-date assignments of job scheduling problem. Here, the performance of six regression-based due-date assignment rules, found in the literature, was compared with the due dates predicted by the neural network-based model. The results of the study for a simple flow shop revealed that in terms of absolute-deviation (MAD) and standard-deviation-of-lateness (SDL) criteria, the neural network outperforms all other six conventional rules. Ivanescu *et al.* (2002) also developed a regression-based model to estimate makespan in a batch process shop. The results indicate that the performance of the regression model is good when high variety in the job mix and high uncertainty in the processing times exist. Raaymakers and Weijters (2003) studied the performance of neural network and regression model in estimating makespan in bath process industries. The results confirmed superiority of the proposed neural network to the regression model. Li *et al.* (2007) also developed a genetic algorithm (GA)-based neural network for makespan estimation in a batch processing industry. Their research showed that combined GA with neural network approach is more efficient and accurate in makespan estimation than back propagation neural network.

The aim of this article is to develop an intelligent system whereby the performance measure associated with any specific problem is anticipated without solving the problem. For this purpose, an adaptive neuro fuzzy inference system (ANFIS) based intelligent system is developed. The system consists of a couple of phases namely training and test. In the training phase, the system is learned by solving a number of problems in different sizes. Then the system is able to anticipate the performance of any specific problems without solving the problem. In addition, the performances of a number of heuristic algorithms are studied and the most efficient algorithm/s performing with ANFIS is introduced.

To the authors' knowledge, so far no research has been conducted in forecasting makespan performance in a no-wait two stage hybrid flow shop. In this study, we develop an ANFIS-based approach, which is a combination of neural network and fuzzy theory, to estimate the makespan performance of the scheduling problems addressed above.

The remainder of this article is organized as follows. In Section 2, the problem studied in this research is described in detail. In Section 3, the framework of the proposed approach with an illustrative example for predicting makespan is described. In Section 4 the result of the numerical experiments is presented. Finally, Section 5 presents the summary of the research and concluding remarks followed by recommendation for further research.

2. Problem description

The problem studied in this article is a no-wait two stage flexible FSSP. The performance of the proposed heuristic algorithm is studied in terms of minimum makespan. The structure of the problem studied is as follows. A set of n jobs $J = \{j_1, j_2, \ldots, j_n\}$are to be processed in a flexible shop. Each job consists of two operations to be processed in a two stages namely S_1 and S_2. No-waiting time is allowed between the two operations. Stage S_1 and S_2 have m_1 and m_2 identical machines, respectively. The processing times of job j are p_{1j} and p_{2j}, respectively (Rabiee *et al.* 2010). The problem is shown by $F_2(m_1,m_2)|\text{no-wait}|C_{\max}$. In this study, the operations set-up times are assumed to be independent of the job sequences, and hence is added to the operation times. This problem is schematically depicted in Figure 1.

As illustrated above, each job has $m_1 \times m_2$ possible schedules and hence n jobs have $n!$ possible schedules. Therefore, in total there are $n! \times m_1^n \times m_2^n$ possible solutions for this problem. The two stage no-wait hybrid flow shop problems are NP-hard in the strong sense (Chen and Muraki 1997). In this article, we propose an intelligent system to anticipate the performance of the problem addressed above. In addition, the performances of a number of heuristic algorithms applied in the proposed intelligent system are studied. The framework of the proposed ANFIS-based intelligent system is described in the following section.

3. Adaptive neuro fuzzy inference system

3.1. *Structure of ANFIS*

This study addresses an ANFIS model which a combination of neural network and fuzzy rules. The aim is to take the advantage of the capabilities of both Fuzzy systems, which a rule-based approach and neural network which focus of the network training. The ANFIS architecture applied in this research is based on the Takagi–Sugeno (TSK) model, (Takagi and Sugeno 1985), which has been applied in estimating values of factors in various fields such as estimation of utility functions, occurrence of an accident, medical diagnostics and performance estimation (Haykin 1999, Fu *et al.* 2007). Figure 2 shows the structure of the model developed by Takagi and Sugeno (1985).

As illustrated above, it is assumed that the fuzzy inference system consists of five layers of adaptive network with two inputs x and y and output z. For the first-order of TSK fuzzy model, a typical rule set with two fuzzy if-then rules is defined as follows:

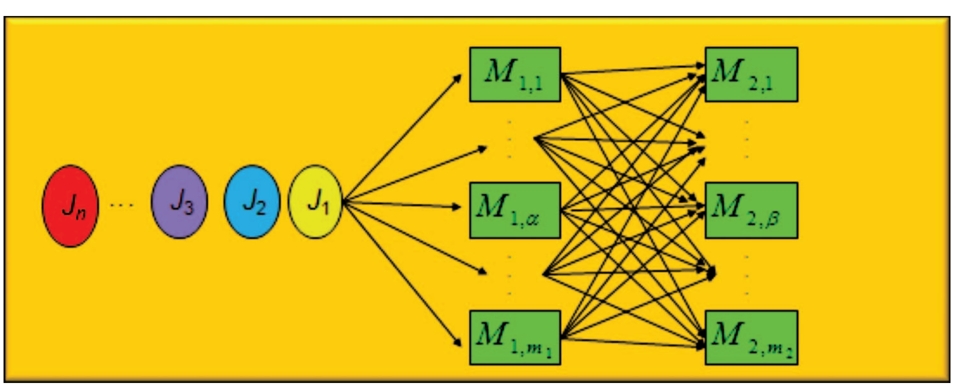

Figure 1. Schematic of the problem.

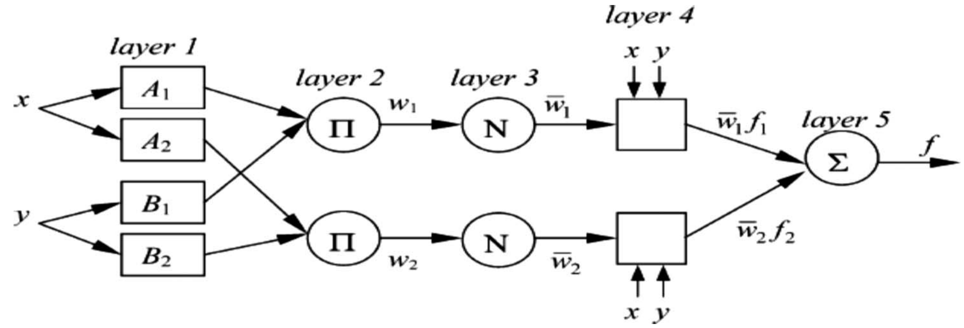

Figure 2. Schematic diagram of ANFIS.

$$\text{Rule1}: \text{ If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then } f_1 = p_1x + q_1y + r_1$$

$$\text{Rule2}: \text{ If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ then } f_2 = p_2x + q_2y + r_2$$

The entire system architecture consists of five layers, namely, fuzzification layer, product layer, normalised layer, de-fuzzification layer and total output layer.

[Layer 1]: For a node like i, this layer is an adaptive node with a node function explained as below:

$$O_{1,i} = \mu A_i(x), i = 1, 2$$
$$O_{1,j} = \mu B_{j-2}(y), j = 3, 4$$

where x (and y) is the input to node, i and A_i (and B_j) is a linguistic label (such as 'small' or 'large') connected to this node. $O_{1,i}$ is then the membership grade of a fuzzy set A (= A_1,A_2, B_1 and B_2). Gaussian parameterised membership function is usually used as the input membership function which guarantees a smooth transition between 0 and 1. It is defined as follows:

$$\mu A(X) = \exp\{-(\frac{x - c_i}{a_i})^2\}$$

where $\{a_i, c_i\}$ is the parameter set.

[Layer 2]: The output of this layer is the product of all incoming signals and represents the firing strength of a rule defined as follows:

$$O_{2,i} = \omega = \mu A_i(x) \times \mu B_i(y), i = 1, 2$$

[Layer 3]: The outputs of this layer are the normalisation of incoming firing strengths:

$$O_{3,i} = \bar{\omega} = \frac{\omega_i}{(\omega_1 + \omega_2)}, i = 1, 2$$

[Layer 4]: Every node i in this layer is an adaptive node with a node function:

$$O_{4,i} = \bar{\omega}_i f_i = \bar{\omega}(p_i x + q_i y + r_i)$$

where $\bar{\omega}$ is a normalised firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Linear parameters in this layer are referred to consequent parameters.

[Layer 5]: The single node in this layer computes the overall output as the summation of all incoming signals:

$$O_{5,i} = \sum \bar{\omega}_i f_i.$$

It can be observed that there are two adaptive layers with square nodes in this ANFIS architecture. They are the first layer and the fourth layer. In the first layer, there are two modifiable parameters $\{a_i, c_i\}$ which are related to the input membership functions. These parameters are called premise parameters. In the fourth layer, there are also three modifiable parameters $\{p_i, q_i, r_i\}$ pertaining to the first-order polynomial. These parameters are called consequent parameters.

3.2. *Hybrid learning algorithm*

It is observed that given the values of premise parameters, the overall output can be expressed as linear combinations of the consequent parameters. More precisely, the overall output of the ANFIS model can be written as the following equation: (Jang 1993)

$$\begin{aligned} f &= \sum f_i = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \overline{w_1} f_1 + \overline{w_2} f_2 \\ &= (\overline{w_1}) p_1 + (\overline{w_1}) q_1 + (\overline{w_1}) r_1 + (\overline{w_2}) p_2 + (\overline{w_2}) q_2 + (\overline{w_2}) r_2 \end{aligned}.$$

The above-mentioned non-linear and linear parameters in premise and consequent parts are adjusted by a hybrid learning algorithm, based on a collection of process data. One of the hybrid learning algorithms, proposed by Jang (1993), includes the gradient descent method (GDM) and the least squares estimate (LSE).

Each epoch of the hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, functional signals go forward till layer 4 and consequent parameters are optimised by the LSE, under the condition that premise parameters are fixed. In the backward pass, the error rates propagate backward and the premise parameters are updated by the GDM accordingly.

3.3. An ANFIS to makespan estimation

Here the steps of ANFIS developed for makespan estimation are explained as follows:

Step 1. Load data sets such as training data and test data.

The input variables are number of jobs, number of machines in stage one, number of machines in stage two, summation of processing time in stage one and summation of processing time in stage two. The output variable is completion time or makespan. Since, the range of the input variables is considerably different, the input variable is normalised as follows:

$$x'_j = \frac{x_j - x_{\min j}}{x_{\max j} - x_{\min j}}$$

where x_{minj}, x_{maxj} are the minimum and maximum values of the jth input variable.

Step 2. In this stage, the network is trained to obtain the network parameter values such as initialise and the number of membership functions and the trend of the membership functions, as well as input and output parameters. In this study, a sub-clustering has been used to train the network. The details will be explained later.

Step 3. Select the ANFIS parameter optimisation method for the neuro fuzzy inference system.

Step 4. Decide on the number of training epochs of the FIS.

Step 5. Start the training of the FIS.

Step 6. Validate the network output data by comparing them with those of actual data.

3.4. Network structure of proposed ANFIS

As described before, a clustering technique is used to train the network. The main parameter affecting the sub-clustering model includes range of influence, squash factor, acceptance ratio and rejection ratio. The values of parameters will be given as default values except range of influence, which is considered equal to 0.01. In addition, the Gaussian Bell shape membership function will be chosen as a default membership function for each input parameter. There are two parameter optimisation methods. The first method is hybrid method which considers mixed least squares and back propagation gradient descent technique. The second method is simple back propagation. The Error Tolerance is used as a stop criterion. Summary of value of the parameters used in this study is shown in Table 1.

A numerical example is used to demonstrate the concept of the proposed algorithm. Table 2 shows the details.

According to Table 2, five problems each one consisting of a number of input variables, namely number of jobs (N), number of machines in each stage (M_1 and M_2) and total processing time in each stage ($\sum_{j=1}^{n} p_{1j}$ and $\sum_{j=1}^{n} p_{2j}$) are created. The results given in the right hand side of the Table 2 shows the

Table 1. Effective parameters value in FIS results.

Structure of FIS	Range of influence	Squash factor	Accept ratio	Rejection ratio	Membership function	Train FIS method	Epoch	Error tolerance
Sub-clustering	0.01	1.25	0.5	0.15	Gaussian Bell shape	hybrid	5	0

Table 2. Sample problems and the results using heuristic rules.

Problem number	Input variables					Dispatching rules					
	N	M_1	M_2	Sum(P_{1j})	Sum(P_{2j})	MDA	SPT	LPT	Johnson	MD	NEH
1	3	1	2	23	22	31	31	22	27	33	27
2	3	2	1	22	15	26	26	27	23	26	23
3	4	2	2	28	27	23	25	30	23	25	22
4	4	2	1	29	33	38	38	42	38	42	38
5	5	2	2	36	35	32	32	31	25	31	25

performance of the heuristic algorithms in terms of makespan.

Table 3 shows the normalized value of the data presented in Table 2 in a range between 0 and 1.

Table 4 shows the results obtained using ANFIS and multiple linear regression (MLR). According to Table 4, for each heuristic method four of five problems are randomly selected for training procedure [e.g. problem numbers 1, 2, 3 and 4 for minimum deviation algorithm (MDA)] and the remaining problem is considered as the test problem (e.g. problem number 5 for MDA).

The actual and normalised actual values of $C_{\max}$ for problem number 5 are 32 and 0.6, respectively. Y represents the results using MLR with five variables (i.e.N, M_1, M_2, $\sum_{j=1}^{n} p_{1j}$ and $\sum_{j=1}^{n} p_{2j}$).

MLR and ANFIS show makespan estimation of the test problem using MLR and adaptive fuzzy neural network inference system, respectively. For instance, for MDA the results show that the predicted values of ANFIS and MLR are 30.5 and 28.25, respectively, compared with the actual value of 32. This reveals that ANFIS obtained more accurate result than MLR.

4. Results

4.1. Simulation process

This section describes the results of the computational experiments performed to evaluate the performance of the proposed ANFIS method compared with that of the regression analysis using six algorithms, namely MDA (Xie and Wang 2005), LPT (largest processing time), SPT (shortest processing time), Johnson rule (Johnson 1954), MD (minimum deviation) (Narasimhan and Panwalkar 1984) and Nawaz, Enscore and Ham (Nawaz *et al.* 1983). The simulation model based on the proposed algorithm was developed using Visual Basic language, and the ANFIS model was developed in MATLAB 2008a language. In addition, the regression analysis is performed using MINITAB software. The simulation model was run in a personal computer with 3.4 GHz, 895 MB memory. The performance measures used in this study are MSE, root mean squared error (RMSE) and R-squared, which are defined as follows.

In statistics, the mean square error, i.e. MSE, of an estimator is a way to quantify the difference between an estimated and the real value of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the square of the 'error'. The error is the amount by which the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator does not account for information that could produce a more accurate estimate (Lehmann and Casella 1998).

The MSE is the second moment (against the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, MSE is equal to variance. Like the variance, MSE has the same unit of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields RMSE, which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard error.

The MSE of an estimator $\widehat{\theta}$ with respect to the estimated parameter θ is defined as follows:

$$\mathrm{MSE}(\widehat{\theta}) = E[(\widehat{\theta} - \theta)^2].$$

The MSE is equal to the sum of the variance and the squared bias of the estimator

$$\mathrm{MSE}(\widehat{\theta}) = \mathrm{Var}(\widehat{\theta}) + (\mathrm{Bias}(\widehat{\theta}, \theta)).$$

The MSE thus assesses the quality of an estimator in terms of its variation and unbiasedness. Since MSE is an expectation, it is a scalar, and not a random variable.

RMSE is a frequently used measure of the difference between values predicted by a model or an estimator and the values actually observed from the thing being modelled or estimated. These differences are also called residuals. The RMSD of an estimator $\widehat{\theta}$ with respect to the estimated parameter θ is defined as the square root of the mean squared error:

Table 3. Sample problems with normalised input variables and results in a range between 0 and 1.

Problem number	Input variables					Dispatching rules					
	N	M_1	M_2	Sum(P_{1j})	Sum(P_{2j})	MDA	SPT	LPT	Johnson	MD	NEH
1	0	0	1	0.07143	0.35	0.53333	0.46154	0	0.26667	0.47059	0.3125
2	0	1	0	0	0	0.2	0.07692	0.25	0	0.05882	0.0625
3	0.5	1	1	0.42857	0.6	0	0	0.4	0	0	0
4	0.5	1	0	0.5	0.9	1	1	1	1	1	1
5	1	1	1	1	1	0.6	0.53846	0.45	0.13333	0.35294	0.1875

Table 4. Makespan estimation results using ANFIS and MLR.

Algorithms	Train problem numbers	Test problem number	Multiple linear regression model	C_{max}	Normalised C_{max}	Normalised MLR predicted	Normalised ANFIS predicted	MLR predicted	ANFIS predicted
MDA	1–2–3–4	5	$Y = 0.95 - 0.75 \times x_2 - 0.73 \times x_3 + 0.88 \times x_5$	32	0.60000	0.35556	0.50000	28.25	30.5
SPT	1–2–3–4	5	$Y = 0.795 - 0.717 \times x_2 - 0.692 \times x_3 + 1.025 \times x_5$	32	0.53846	0.41026	0.50000	30.3333	31.5
LPT	1–2–4–5	3	$Y = 0.25 - 1.1434 \times x_1 - 0.57 \times x_3 + 0.913 \times x_5$	30	0.40000	0.15652	0.33964	25.1304	28.7928
Johnson	2–3–4–5	1	$Y = 0.25 - 1.1434 \times x_1 - 0.57 \times x_3 + 0.913 \times x_5$	27	0.26667	−0.08333	0.27077	20.917	29.77
MD	2–3–4–5	1	$Y = 0.0588 - 0.688 \times x_3 - 0.187 \times x_4 + 1.149 \times x_5$	33	0.47059	−0.22059	0.50000	21.245	33.5
NEH	1–3–4–5	2	$Y = 1.265 - 0.429 \times x_2 - 0.976 \times x_3 + 0.328 \times x_4$	23	0.06250	0.83594	0.09748	29.375	23.56

$$\mathrm{RMSE}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{E[(\hat{\theta} - \theta)^2]}.$$

Statistical measure of how well a regression line approximates real data points; an *R*-squared of 1.0 (100%) indicates a perfect fit. The formula for r(X, Y) is defined as follows:

$$\mathrm{r}(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\mathrm{StdDev}(X) \times \mathrm{StdDev}(Y)}$$

where Cov (X, Y) is covariance of X and Y, and StdDev(X) represents Standard Deviation of X. In order to investigate the performance of the proposed model, its performance was compared with that of a MLR model defined below:

MLR attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y. The population regression line for p explanatory variables $x_1, x_2 \ldots, x_p$ is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$. This line describes how the mean response μ_y changes with the explanatory variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation∂. The fitted values $b_0, b_1, \ldots, b_p$ estimate the parameters β_0, $\beta_1, \ldots, \beta_p$ of the population regression line.

Since the observed values for y vary around their means μ_y, the multiple regression model includes a couple of variables, namely Residual and Fit. The model is expressed as Data = Fit + Residual, where the 'Fit' term represents the expression $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$.

The 'Residual' term represents the deviations of the observed values y from their means μ_y, which are normally distributed with mean 0 and variance∂. The notation for the model deviations is ε. Formally, the model for MLR, given 108 observations, associated with 108 problems was solved in this study. The maximum completion time is estimated using the following equation:

$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \varepsilon$ for $i = 1, 2 \ldots, 108$

where $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ represent regression parameters and x_1, x_2, x_3, x_4, x_5 show variables which are defined in previous section and ε indicates the notation for the model deviations.

4.2. *Data analysis*

In this study, the performance of the ANFIS model is compared with that of the regression analysis. The details of the problems solved in this study are illustrated in Table 5.

Table 5. Parameters of the sample problems.

Scales	Factors	Number of levels	Levels
Small	NM (number of machine)	3	$m_1 = 3, m_2 = 4$
			$m_1 = 2, m_2 = 2$
			$m_1 = 3, m_2 = 2$
Large		3	$m_1 = 8, m_2 = 10$
			$m_1 = 10, m_2 = 10$
			$m_1 = 12, m_2 = 10$
Small	NJ (number of jobs)	9	8,10,14,16,20,24,28,30,36
Large		9	40,42,44,72,80,88,108,120,132
Small and large	DT (distribution of processing times)	2	U (4,40)
			N (80,25)
Small and large	A (algorithm)	6	A_1:MDA
			A_2:LPT
			A_3:SPT
			A_4:Johnson
			A_5:MD
			A_6:NEH

According to Table 5, four parameters are considered to define the problems. These are number of machines in the both stages, number of jobs, processing time distribution and the heuristic sequencing algorithms. The problems solved in this study are classified into two categories, namely small and large scale problems in terms of the number of machines at each stage. For each scale, three types of the problems, as shown in row one and two of Table 5, are considered. In addition, for each scale, the problems are defined in nine sub-categories in terms of the number of jobs. Besides that two different processing time distributions, namely Uniform and normal distributions, are defined. By taking into account of all combinations of the problems, 108 different problems can be established in total. In this study, the performance of ANFIS and MLR algorithms is investigated using six different heuristic algorithms. For each case, all 108 problems were solved. Among these problems, 90% of the problems were solved to train the model. Then for the last 10% of the problems, the performance measures namely MSE and RMSE and R-square are estimated based on the experiences the model gained during the training phase. In order to compare the estimated performance measure with that of the actual value, each problem is solved and the estimated value of the performances was compared with those of the actual value. For each problem, in order to obtain stable results, the problems were solved five times, each one with different seed numbers upon which the random numbers are created. The results are illustrated in Tables 6, 7 and 8.

Tables 6, 7 and 8 indicate the simulation results using ANFIS and MLR model in terms of MSE, RMSE and R-square, respectively. Each model was studied using a number of heuristic algorithms. The results show that in the training phase, ANFIS always obtains more accurate makespan estimation than MLR. Similarily, this approach outperforms MLR for most of the test problems.

In order to study the efficiency of the algorithms in proposed ANFIS, a comparative performance measure namely relative percentage deviation (RPD) is used. This metric is calculated using the below equation:

$$\mathrm{RPD} = \frac{|\mathrm{Method}_{\mathrm{sol}} - \mathrm{Best}_{\mathrm{sol}}|}{\mathrm{Best}_{\mathrm{sol}}} \times 100$$

where $\mathrm{Method}_{\mathrm{sol}}$ is the value of a method and $\mathrm{Best}_{\mathrm{sol}}$is the best value obtained among the algorithms. Also, $\overline{\mathrm{RPD}}$ is defined according to the below equation:

$$\overline{\mathrm{RPD}} = \frac{\sum_{i=1}^{\text{number of run}} \mathrm{RPD}}{\text{number of run}}.$$

As already discussed, ANFIS obtained superior performance in estimating makespan than the regression model. Hence, further study was performed to evaluate the performance of the heuristic algorithms using ANFIS. Table 9 shows the results in detail.

According to the results, Johnson's rule (Johnson 1954) and MD obtained better performance than the others. Johnson's rule (Johnson 1954) outperformed the other algorithms in terms of MSE and RMSE in the training phase, whereas MD is dominant in terms of MSE and RMS in the test phase. In addition, with respect to R-square, all of the algorithms have quite similar performance in the training phase while in the test problem Johnson's rule (1954) has the best performance.

In general, it can be concluded that the proposed ANFIS with Johnson (1954) and MD dispatching rules

Table 6. The simulation results in terms of MSE using ANFIS and regression model.

		MDA		LPT		SPT		Johnson		MD		NEH	
		ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR
Train	1	8.01E-15	0.00852	8.60E-15	0.00825	7.81E-15	0.19618	7.70E-15	0.00868	8.02E-15	0.00714	7.91E-15	0.00815
	2	8.01E-15	0.00852	8.58E-15	0.00795	7.80E-15	0.00792	7.68E-15	0.00875	7.58E-15	0.00675	8.31E-15	0.00840
	3	8.02E-15	0.00852	8.60E-15	0.00808	7.43E-15	0.00820	7.68E-15	0.00875	8.37E-15	0.00749	7.63E-15	0.00838
	4	7.74E-15	0.00846	8.99E-15	0.00819	8.41E-15	0.20178	7.43E-15	0.00874	7.60E-15	0.00717	7.88E-15	0.00826
	5	7.74E-15	0.00848	9.04E-15	0.00838	8.61E-15	0.00836	7.70E-15	0.00875	7.80E-15	0.00777	7.48E-15	0.00832
	$Best_{sol}$	7.74E-15	8.46E-03	8.58E-15	7.95E-03	7.43E-15	7.92E-03	7.43E-15	8.68E-03	7.58E-15	6.75E-03	7.48E-15	8.15E-03
	Average	7.90E-15	8.50E-03	8.76E-15	8.17E-03	8.01E-15	8.45E-02	7.64E-15	8.73E-03	7.87E-15	7.26E-03	7.84E-15	8.30E-03
	Std dev	1.50E-16	2.80E-05	2.30E-16	1.65E-04	4.86E-16	1.05E-01	1.18E-16	2.95E-05	3.29E-16	3.85E-04	3.16E-16	1.03E-04
Test	1	0.00091	0.00287	0.00085	0.00785	0.00893	0.00837	0.01016	0.00726	0.00065	0.00759	0.00123	0.00376
	2	0.00091	0.00283	0.00081	0.00848	0.00895	0.00831	0.00025	0.00046	0.00057	0.004	0.00046	0.00413
	3	0.00079	0.00276	0.00077	0.00576	0.00159	0.00318	0.00016	1.45E-09	0.00051	0.00643	0.00077	0.00367
	4	0.00048	0.00378	0.00075	0.00371	0.00092	0.00061	0.0009	0.00633	0.00075	0.0044	0.00063	0.00526
	5	0.00059	0.00353	0.00043	0.00062	0.00071	0.00075	0.00025	0.00614	0.0007	0.0066	0.00081	0.00455
	$Best_{sol}$	4.84E-04	2.76E-03	4.28E-04	6.21E-04	7.08E-04	6.08E-04	1.59E-04	1.45E-09	5.10E-04	4.00E-03	4.57E-04	3.67E-03
	Average	7.34E-04	3.16E-03	7.20E-04	5.28E-03	4.22E-03	4.25E-03	2.34E-03	4.04E-03	6.36E-04	5.81E-03	7.81E-04	4.27E-03
	Std dev	1.92E-04	4.67E-04	1.68E-04	3.21E-03	4.32E-03	3.88E-03	4.38E-03	3.50E-03	9.77E-05	1.54E-03	2.88E-04	6.52E-04

Table 7. The simulation results in terms of RMSE using ANFIS and regression model.

		MDA		LPT		SPT		Johnson		MD		NEH	
		ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR
Train	1	8.95E-08	0.09228	9.27E-08	0.09084	8.84E-08	0.44292	8.77E-08	0.09318	8.96E-08	0.08452	8.89E-08	0.09027
	2	8.95E-08	0.09230	9.26E-08	0.08914	8.83E-08	0.08897	8.76E-08	0.09352	8.71E-08	0.08216	9.12E-08	0.09166
	3	8.95E-08	0.09231	9.27E-08	0.08987	8.62E-08	0.09057	8.77E-08	0.09354	9.15E-08	0.08655	8.73E-08	0.09156
	4	8.80E-08	0.09198	9.48E-08	0.09047	9.17E-08	0.44920	8.62E-08	0.09349	8.72E-08	0.08467	8.88E-08	0.09087
	5	8.80E-08	0.09207	9.51E-08	0.09153	9.28E-08	0.09143	8.78E-08	0.09356	8.83E-08	0.08813	8.65E-08	0.09121
	$Best_{sol}$	8.80E-08	9.20E-02	9.26E-08	8.91E-02	8.62E-08	8.90E-02	8.62E-08	9.32E-02	8.71E-08	8.22E-02	8.65E-08	9.03E-02
	Average	8.89E-08	9.22E-02	9.36E-08	9.04E-02	8.95E-08	2.33E-01	8.74E-08	9.35E-02	8.87E-08	8.52E-02	8.85E-08	9.11E-02
	Std dev	8.44E-10	1.52E-04	1.22E-09	9.15E-04	2.71E-09	1.95E-01	6.76E-10	1.58E-04	1.84E-09	2.26E-03	1.78E-09	5.69E-04
Test	1	0.03014	0.05357	0.02917	0.08859	0.09456	0.09151	0.10086	0.08519	0.02557	0.08713	0.03511	0.06132
	2	0.03014	0.05323	0.02856	0.09211	0.09463	0.09117	0.01593	0.02155	0.02381	0.06326	0.02137	0.06427
	3	0.02813	0.05255	0.02773	0.0759	0.03993	0.0564	0.01265	0.00004	0.02258	0.08021	0.02776	0.06056
	4	0.02241	0.06148	0.02734	0.06089	0.03042	0.02467	0.037	0.07954	0.02741	0.06632	0.02514	0.07253
	5	0.02424	0.05945	0.02075	0.02492	0.02667	0.02748	0.01581	0.07835	0.02643	0.08125	0.02851	0.06746
	$Best_{sol}$	2.20E-02	5.25E-02	2.07E-02	2.49E-02	2.66E-02	2.47E-02	1.26E-02	3.81E-05	2.26E-02	6.33E-02	2.14E-02	6.06E-02
	Average	2.69E-02	5.61E-02	2.67E-02	6.85E-02	5.72E-02	5.82E-02	3.50E-02	5.29E-02	2.52E-02	7.56E-02	2.76E-02	6.52E-02
	Std dev	3.65E-03	4.10E-03	3.40E-03	2.73E-02	3.44E-02	3.27E-02	3.74E-02	3.93E-02	1.96E-03	1.03E-02	5.05E-03	4.91E-03

Table 8. The simulation results in terms of R-square using ANFIS and regression model.

		MDA		LPT		SPT		Johnson		MD		NEH	
		ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR	ANFIS	MLR
Train	1	1	0.82321	1	0.83204	1	0.83536	1	0.81678	1	0.85404	1	0.82363
	2	1	0.82316	1	0.83222	1	0.83565	1	0.81559	1	0.85322	1	0.82354
	3	1	0.82308	1	0.82938	1	0.83015	1	0.81471	1	0.84638	1	0.83875
	4	1	0.82437	1	0.82515	1	0.82448	1	0.81572	1	0.84815	1	0.83162
	5	1	0.82402	1	0.81986	1	0.82358	1	0.81414	1	0.85386	1	0.83257
	$Best_{sol}$	1	8.24E-01	1	8.32E-01	1	8.36E-01	1	8.17E-01	1	8.54E-01	1	8.39E-01
	Average	1	8.24E-01	1	8.28E-01	1	8.30E-01	1	8.15E-01	1	8.51E-01	1	8.30E-01
	Std dev	0	5.90E-04	0	5.25E-03	0	5.75E-03	0	1.01E-03	0	3.59E-03	0	6.48E-03
Test	1	0.85931	0.73244	0.94799	0.79025	0.77283	0.79081	0.73601	0.81201	0.88965	0.80005	0.94328	0.83515
	2	0.85931	0.74049	0.94762	0.79065	0.76959	0.79004	0.96993	0.96975	0.85161	0.84384	0.90575	0.87903
	3	0.87401	0.73867	0.937	0.83447	0.78737	0.63522	0.98327	0.83587	0.86646	0.78453	0.89467	0.88014
	4	0.93338	0.69525	0.91289	0.85263	0.69894	0.84362	0.90971	0.83331	0.888	0.82799	0.94588	0.91098
	5	0.92654	0.72778	0.58058	0.56796	0.71255	0.5741	0.9735	0.84113	0.87134	0.81449	0.87517	0.85248
	$Best_{sol}$	9.33E-01	7.40E-01	9.48E-01	8.53E-01	7.87E-01	8.44E-01	9.83E-01	9.70E-01	8.90E-01	8.44E-01	9.46E-01	9.11E-01
	Average	8.91E-01	7.27E-01	8.65E-01	7.67E-01	7.48E-01	7.27E-01	9.14E-01	8.58E-01	8.73E-01	8.14E-01	9.13E-01	8.72E-01
	Std dev	3.66E-02	1.84E-02	1.60E-01	1.15E-01	3.97E-02	1.16E-01	1.04E-01	6.32E-02	1.58E-02	2.32E-02	3.09E-02	2.90E-02

Table 9. The makespan estimation in terms of $\overline{\text{RPD}}$ based on heuristic algorithms using ANFIS.

		MDA	LPT	SPT	Johnson	MD	NEH
MSE	Train	5.06713	16.46994	6.50641	1.53815	4.68411	4.23520
	Test	165.78020	151.94710	1176.57500	307.99620	116.08900	162.15120
RMSE	Train	2.49930	7.90693	3.15711	0.76360	2.29425	2.08289
	Test	56.64094	53.59907	217.54900	66.11512	44.16483	57.34902
R-square	Train	0	0	0	0	0	0
	Test	7.60330	10.17122	22.39373	5.23687	9.36919	5.24530

is effective method in predicting the performance measure of a no-wait two stage hybrid flow shop. This enables both researchers and practitioners to predict the job completion time before the schedule is constructed. Such an estimate can be used as a basis to agree on reliable due dates with the customers in advance of production while helping the practitioners to anticipate the schedule performance and make subsequent logistic planning, e.g. material procurement, accordingly.

5. Conclusion

In this study, ANFIS was developed to predict the performance/s of the problems in a no-wait two stage flexible flow shop. A MLR model was used for bench marking purpose. The simulation model was developed in two phases, namely training and test phases. In the training phase, about 90% of the problems were solved to obtain effective information upon which the performance of the proposed approach can be predicted. The results of the simulation study reveal that the proposed approach outperforms MLR. Therefore, it can be concluded that in a no-wait two stage flexible flow shop, the proposed ANFIS with Johnson (1954) and MD rules can be used as a reliable approach in estimating the job completion time of the problem studied. The proposed approach can help the practitioners in estimating the performance of the schedule before it is created. As a further study, it is suggested to extend this study towards anticipating other performance measure such as mean flow time, mean tardiness and mean lateness. Furthermore, it is recommended to applying this intelligent system in the other manufacturing system with more variables and objectives. In addition, combining ANFIS with evolutionary algorithms is worthy to be investigated.

References

Aldowaisan, T. and Allahverdi, A., 1998. Total flow time in no-wait flow shops with separated setup times. *Computer & Operation Research*, 25, 757–765.

Baker, K.R., 1974. *Introduction to sequencing and scheduling*. New York: John Wiley & Sons.

Bonney, M.C. and Gundry, S.W., 1976. Solutions to the constrained flow shop sequencing problem. *Operation Research Quarterly*, 24, 869–883.

Chen, W. and Muraki, M., 1997. An action strategy generation framework for on-line scheduling and control system in batch processing with neural networks. *International Journal of Production Research*, 35, 3483–3506.

Cheng, T.C.E. and Gupta, M.C., 1989. Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research*, 38, 156–166.

Desprez, C., Chu, F., and Chu, C., 2009. Minimising the weighted number of tardy jobs in a hybrid flow shop with genetic algorithm, *International Journal of Computer Integrated Manufacturing*, 22 (8), 745–757.

Fu, J.Y., Liang, S.G., and Li, Q.S., 2007. Prediction of wind-induced pressures on a large gymnasium roof using artificial neural networks. *Computers and Structures*, 85, 179–192.

Garey, M.R. and Johnson, D.S., 1979. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.

Haykin, S., 1999. *Neural networks: a comprehensive foundation*. New York: Macmillan Publishing.

Ivanescu, C.V., Fransoo, J.C., and Bertrand, W.M., 2002. Makespan estimation and order acceptance in batch process industries when processing times are uncertain. *OR Spectrum*, 24, 467–495.

Jang, J.S.R., 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23 (3): 665–685.

Johnson, S.M., 1954. Optimal two and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61–68.

Kalczynski, P.J. and Kamburowsk, J., 2007. On no-wait and no-idle flow shops with makespan criterion. *European journal of Operational research*, 178, 677–685.

King, J.R. and Spachis, A.S., 1980. Heuristics for flow shop scheduling. *International Journal of Production Research*, 18 (3), 345–357.

Lehmann, E.L. and Casella, G., 1998. *Theory of point estimation*. 2nd ed. New York: Springer.

Li, S., *et al.*, 2007. A GA-based NN approach for makespan estimation. *Applied Mathematics and Computation*, 185, 1003–1014.

Luo, X., *et al.*, 2010. Optimal resource allocation for hybrid flow shop in one-of-a-kind production. *International Journal of Computer Integrated Manufacturing*, 23 (2), 146–154.

Mirsanei, H.S., *et al.*, 2009. A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 21 (4), 1–14.

Mohammadi, M., Fatemi Ghomi, S.M.T., and Jafari, N., 2011. A genetic algorithm for simultaneous lotsizing and sequencing of the permutation flow shops with sequence-dependent setups. *International Journal of Computer Integrated Manufacturing*, 24 (1), 87–93.

Narasimhan, S.L., and Panwalker, S.S., 1984. Scheduling in a two-stage manufacturing process. *International Journal of Production Research*, 22, 555–564.

Nawaz, M., Enscore, E.E.J, and Ham, I., 1983. A heuristic algorithm for the m machine, n job flowshop sequencing problem. *Omega-International Journal Manage Science*, 11 (1), 91–95.

Philipoom, P.R., Rees, L.P. and Wiegemann, L., 1994. Using neural networks to determine internally-set due-date assignments for shop scheduling. *Decision Sciences*, 25, 825–851.

Raaymakers, W.H.M and Weijters, A.J.M.M., 2003, Makespan estimation in batch process industries: a comparison between regression analysis and neural networks. *European Journal of Operational Research*, 145, 14–30.

Rabiee, M., Shafaei, R., and Mirzaeian, M., 2010. Generating an Efficient Schedule in a No-Wait Two Stage Flexible Flow Shops with Maximizing Utilization, *Proceedings of international conference on industrial engineering and business management* (ICIEBM2010). 555–561.

Rajendran, C. and Chaudhuri, D., 1990. Heuristic algorithms for continuous flow shop problem. *Naval Research Logistics*, 37, 695–705.

Reddi, S.S. and Ramamoorthy, C.V., 1972. On the flow shop sequencing problem with no-wait in process. *Operational Research Quarterly*, 23 (3), 323–331.

Ribas, I., Leisten, R., and Framinan, J.M., 2010. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37, 1439–1454.

Richard, L. and Zhang, W., 1999. Hybrid flow shop scheduling – a survey. *Computer and industrial engineering*, 37, 57–61.

Rock, H., 1984. The three-machine no-wait flow shop problem is NP-complete. *Journal of Associate Computer Machinery*, 31, 336–345.

Ruiz, R. and Vazquez-Rodriguez, J.A., 2010. The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 20, 1–18.

Sabuncuoglu, I., 1998. Scheduling with neural networks – a review of the literature and new research directions. *Production Planning and Control*, 9 (1), 2–12.

Sabuncuoglu, I. and Gurgun, B., 1996. A neural network model for scheduling problems. *European Journal of Operational Reseach*, 93, 288–299.

Solimanpur, M. and Elmi, A., 2011. A tabu search approach for group scheduling in buffer constrained flow shop cells. *International Journal of Computer Integrated Manufacturing*, 24 (3), 257–268.

Stadtler, H., 2005. Supply chain management and advanced planning-basics, overview and challenges. *European Journal of Operational Research*, 163, 575–588.

Takagi, T., and Sugeno, M., 1985. Fuzzy identification of systems and its applications, to modelling and control. *IEEE Transactions on System, Man, Cybernetics*, 15 (1), 116–132.

Tasgetiren, M.F., *et al.*, 2007. A discrete differential evolution algorithm for the no-wait flow shop scheduling problem with total flow time criterion. *Proceedings of the 2007 IEEE symposium on computational intelligence in scheduling* (CISched2007), Hawaii, USA, 251–258.

Wang, L., 2003. *Shop scheduling with genetic algorithms*. Beijing, China: Tsinghua Univ. Press and Springer.

Wismer, D.A., 1972. Solution of the flow shop sequencing problem with no intermediate queues. *Operations Research*, 20, 689–697.

Xie, J. and Wang, X., 2005. Complexity and algorithms for two-stage flexible flow shop scheduling with availability constraints. *Computer and Mathematics with Application*, 50, 1629–1638.

Zandieh, M., Fatemi Ghomi, S.M.T., and Moattar Husseini, S.M., 2006. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematic and Computation*, 180, 111–127.