Expert Systems with Applications 40 (2013) 5341–5351

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A neuro-fuzzy model for a new hybrid integrated Process Planning and Scheduling system

Alper Seker [a,*], Serpil Erol [b], Reha Botsali [a]

[a] The Scientific and Technological Research Council of Turkey, Kavaklıdere, 221 06100 Ankara, Turkey
[b] Department of Industrial Engineering, Gazi University, Maltepe, 06570 Ankara, Turkey

ARTICLE INFO

Keywords:
Process Planning
Genetic Algorithm
Scheduling
Fuzzy Neural Network

ABSTRACT

In customized mass production, isolation of Process Planning (PP) and Scheduling stages has a critical effect on the efficiency of production. In this study, to overcome this isolation problem, we propose an integrated system that does PP and Scheduling in parallel and responds to fluctuations in job floor on time. One common problem observed in integration models is the increase in computational time in conjunction with the increase of problem size. Therefore in this study, we use a hybrid heuristic model combining both Genetic Algorithm (GA) and Fuzzy Neural Network (FNN). To improve GA performance and increase the efficiency of searching, we use a clustered chromosome structure and test the performance of GA with respect to different scenarios. Data provided by GA is used in constructing an FNN model that instantly provides new schedules as new constraints emerge in the production environment. Introduction of fuzzy membership functions in Artificial Neural Network (ANN) model allows us to generate fuzzy rules for production environment.

1. Introduction

Process Planning (PP) and Scheduling are production system components determining how and when to produce with respect to available resources. In today's manufacturing environment, generally PP and Scheduling are considered isolated activities from each other, and consequently these two production activities are carried out by different departments in a factory. This isolation creates a large time gap between PP and Scheduling, which in turn decreases the total production efficiency. The work done up to date in the field of Scheduling and PP mostly focuses on mass production environment. On the other hand, the isolation of PP from Scheduling and the resulting time gap between these activities is a critical problem that requires more attention. Being inspired by this fact, in this study our objective is to provide a model that does PP and Scheduling simultaneously in a customized mass production environment.

Integration of PP and Scheduling is an NP-hard (non-deterministic polynomial-time hard) problem and it is not possible to find the optimal solution in polynomial time. Generally, companies do not have the luxury of spending hours/days to iteratively plan production activities that dynamically change throughout the production process. Therefore, companies mostly prefer PP and Scheduling activities to be planned and coordinated by heuristics in reasonable time. To achieve coordination of PP and Scheduling activities, the necessity of reconfiguring PP and Scheduling departments arises. Tan and Khoshnevis (Tan & Khoshnevis, 2000) studied this issue, but their study has certain limitations. On the other hand, in the literature a more popular approach for this integration problem is to focus on the exchange of information between PP and Scheduling activities (Gaalman, Slomp, & Suresh, 1999; Gindy, Saad, & Yue, 1999; Guo et al., 2009; Shen, Wang, & Hao, 2006; Wang et al., 2009; Zhang, Saravanan, & Fuh, 2003).

Process plans usually provide inputs to scheduling just after the product design is completed and this corresponds to an earlier time than the start of production. In the meantime, job floor conditions change dynamically because of several reasons such as replacement of old machines, crises, strikes, disruptions in the supply chain, etc. A survey on this issue shows that 30% of the process plans need to be revised just before the production plans (Detand, Kruth, & Kempenaers, 1992). For detailed process plans and schedules, alternative production routes should be defined and chosen, followed by planning of operations for the chosen routes. The purpose of this study is to provide efficient production plans that are generated for the integrated problem of PP and Scheduling. Determining the alternative operations, machines, and sequence of operations during production are all parts of this integrated problem. As another objective, we do not want to sacrifice system flexibility and computational time for solving this integrated problem.

* Corresponding author. Address: The Scientific and Technological Research Council of Turkey, Atatürk Bulvarı No. 221, Kavaklıdere, 06100 Ankara, Turkey. Tel.: +90 312 468 53 00 4550.

E-mail addresses: alper.seker@tubitak.gov.tr (A. Seker), serpiler@gazi.edu.tr (S. Erol), rehabotsali@gmail.com (R. Botsali).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.03.043

Huang, Zhang, and Smith (1995) propose a nonlinear model for the integration problem, yet the assumption of one-way information flow from PP to Scheduling does not allow finding the optimal solution. As another drawback of their approach, some process plans created according to real-time production conditions may not be feasible. Later, based on Huang et al. (1995) approach, Tonshoff, Beckendorff, and Andres (1989) create process plans before manufacturing stage, they appoint a scheduling function for selecting the appropriate process plan according to the state of resources, and finally they apply active re-planning for fluctuations in the job floor. However, in this approach, repeating PP and Scheduling is very time-consuming operations and this makes their method not suitable for mass customization. In our study, by ensuring the information flow between PP and Scheduling and by using ANN, it is guaranteed to obtain effective production plans instantly as the shop floor conditions change. In addition, the use of ANN allows us to generate several feasible process plans to be stored in PP department for the case of manufacturing changes.

Kim and Egbelu (1999) compare a pre-processing algorithm, a mixed integer programming model, and a heuristic. They observe that the pre-processing algorithm takes shorter time than the mixed integer programming model, but longer than the heuristic. Increasing the number of jobs or process plans reduces the quality of heuristic's solution whereas increasing the number of machines has no effect. On the other hand, Weintraub et al. (1999) prove that scheduling with alternative routes has significant impact in meeting due dates in changing production environment. Considering both the positive and negative effect of available process plans in solution models, in this study we limit the number of process plans by eliminating non-promising routes in terms of processing time. Supporting our approach, Lee and Kim (2001) show that selecting process plans by a GA instead of using random combinations of process plans reduces the production time by 20%. Moon, Kim, and Hur (2002) prove that GA approach gives better results than "Tabu Search", a metaheuristic local search method that can be used for solving combinatorial optimization problems (Glover, 1989; Glover, 1990), in terms of calculation time for scheduling problem. Also GA gives better results than "Tabu Search" as problem size increases. They notice that population size and number of operations are main factors effecting GA's performance. Grabowik, Kalinowski, and Monica (2005) propose a new integration model that does PP several times to respond to fluctuations in job floor. Having alternative process plans before rescheduling increases the effectiveness of scheduling and flexibility of production system.

Iwata and Fukuda (1989) suggest that Scheduling and PP departments of a factory should be reorganized in order to get full use of Closed Loop approach (one of the approaches to the integration problem that require iterative PP and Scheduling). In addition to this, Closed Loop approach needs high capacity hardware and software. As production processes become more complex, this approach becomes unrealistic (Gindy et al., 1999). To overcome the deficiency of Closed Loop approach, in this study we use ANN, which eliminates the requirement of iterative PP and Scheduling. In our proposed model, both process plans and schedules are generated separately in an integration module and there is no need for reorganizing the factory.

Distributed integration is another solution approach for the integration of PP and Scheduling. Similar to Closed Loop approach, Scheduling and PP departments of a factory should be reorganized to apply distributed integration effectively (Haddadzade, Razfar, & Farahnakian, 2009). However, distributed integration is not accepted as an efficient method since it responds to changes in job floor through continuous feedback between PP and Scheduling, which requires longer computation time and more effort (Kempenaers, Pinte, & Detand, 1996).

During literature review, we observe that most of the earlier studies on integration problem suggest a solution model that uses a single algorithm, but as the solution and search space increase, the computational time increases dramatically. Therefore, in this study we propose a solution model that uses appropriate algorithms (shortest-path, GA, and ANN) to exploit specific structures embedded in the integrated problem.

As another observation, most studies in the literature ignore the fluctuations in job floor especially after scheduling. In addition to this, the majority of the available studies do not consider the fact that internal and external fluctuations make the available schedules infeasible before production. In this study, we propose a solution model using ANN. In the proposed system, ANN is trained by the outputs of PP and Scheduling integration module and by this way the production system is able to respond the fluctuations in job floor and regenerate new schedules on time. In the next section, we provide a mathematical programming model for the integrated PP and Scheduling problem and explain the main steps of our solution model. In Sections 3 and 4, the details of GA and ANN used in this study are given respectively. In Section 5, we analyze the experimental results. In Section 6, we discuss how fuzzy logic can be used in this model. Finally in Section 7, we provide our conclusions and suggestions for future studies.

2. Integration model for mass customization

The integrated PP and Scheduling problem is NP-Hard and it is not practical to use exact optimization algorithms to find the optimal solution. Yet, a mathematical programming model for this integrated problem with the objective of minimizing makespan would be as indicated in that thesis (Seker, 2013). However, it is not practical (and mostly not possible) to find the optimal solution for a mathematical model dynamically each time as shop floor conditions change. In this study we provide a new model that keeps process plans and schedules up to date according to changing customer demands and production conditions (most prominent features of mass customization production environment). The advantages of this model compared to the existing models in literature are:

1. There is no need for reorganizing PP and Scheduling departments. The communication between these departments is kept at optimal level (No iterative Scheduling and Process Planning).
2. Both schedules and process plans are stored in separate departments.
3. By using ANN, the model creates feasible and efficient process plans and schedules to respond the changes in production environment dynamically.

The main steps of the hybrid algorithm used in this model are:

Step 1 At time $T1$, for each job alternative process plans are created according to the job floor conditions.

Step 2 Under the terms of the job floor conditions at the time of $T2$ in PP department, alternative process plans are ranked by respective makespan lengths. Selected process plans with shorter makespans are sent to scheduling department. In Scheduling department, by using GA, schedules are generated subjected to makespan minimization objective. At the same time, best routes (process plan) used for scheduling is stored at PP department.

Step 3 At time T3, process plan fed back from Scheduling department to PP department is regenerated by using ANN. During this period, in Scheduling department optimal schedules (operation sequences) are generated and then this operation sequences are used for creating neural networks.

Step 4 At time T4, as customer demand and job floor conditions change, new operation sequences (schedules) are created using previously generated neural networks.

Step 5 At time T5, detailed schedules and process plans are available and the system is ready to dynamically generate new schedules and process plans for changing conditions.

3. Integration problem and application of GA approach

We apply the proposed model to a set of process networks and related data reflecting mass customization's wide range of operation and process flexibility. These flexibilities are measured by the following formulations:

$$OF = TM/MO \tag{1}$$

$$PF = \text{Number of alternative routes} \tag{2}$$

where: *OF*: Operation flexibility, *TM*: Total machine number for each job, *MO*: Maximum operation number for each job, *PF*: Process flexibility

For each flexibility type there are five different levels:

OF: very low (VL) (1–3,6), low (L) (3,6–6,2), medium (M) (6,2–8,8), high (H) (8,8–11,4), very high (VH) (11,4–13), *PF*: very low (VL) (1–3), low (L) (3–5), medium (M) (5–7), high (H) (7–9), very high (VH) (9–10)

In Tables 1a and 1b, flexibility rates for 18 jobs in our test data are shown. Problems with different flexibility levels are constructed by using different combinations of jobs and later they will be used to test our GA's performance.

For the test data, based on available job floor conditions, alternative routes (process plans) are generated and ranked for each job. The ranking of process plans is done with respect to the minimum possible completion time of process plans. The problem of finding minimum possible completion time (*CT*) for a given process plan (*P*) is described as follows:

$$\min\ CT \tag{3}$$

$$CT = \sum_{i=1}^{N_p} \min\{t_{ki} : k \in S_i\} \tag{4}$$

N_p is the total number of operations for process plan *P*, S_i is the set of machines that can be used for the *i*th operation of process plan *P*, t_{ki} is the processing time required by machine *k* for the *i*th operation of process plan *P*.

The above problem is a simple shortest-path problem and can be solved easily. For all the routes of each job, we solve the shortest-path problem and rank the routes with respect to their minimum possible completion times. In Eq. (4), by replacing the function $\min\{t_{ki}{:}k \in S_i\}$ with $average\{t_{ki}{:}k \in S_i\}$, we also rank the process plans with respect to average completion times. The ranking of process plans is used in the next step of our solution model where PP and scheduling is integrated.

Table 1b
Flexibility rates of jobs (between Jobs 10 and 18).

Job	10	11	12	13	14	15	16	17	18
MO	7	5	5	9	6	5	10	7	5
TM	25	32	33	65	32	28	77	43	39
OF	3.57	6.4	6.6	7.22	5.33	5.6	7.7	6.14	7.8
PF	3	3	5	6	6	6	5	8	6
Flexibility	ÇD, D	O, D	O, O	O, O	D, O	D, O	O, O	D, Y	O, O

In PP and Scheduling integration module, we develop a GA to generate production schedules. To operate Genetic Algorithms, an initial population is created (Chen & JIA, 2007). In our GA, for all population members, total completion time is calculated and the individual with the minimum completion time is chosen as the best solution. GA uses a 2-chromosomed structure as shown in Fig. 1. The operations data in Chromosome 2 is clustered as alternative process plans (routes). This clustering method eases crossover operation and at the same time GA gives better and faster results for the scheduling problem. In Fig. 1, each gene in Chromosome 1 carries the job and machine information of the respective operation that is determined by the route data in Chromosome 2. The *i*th gene of Chromosome 2 represents the alternative route for job *i*. Depending on the ranking of process plans, for each job we use the best two routes in our GA. For this reason, each gene of Chromosome 2 can either show first route or second route. By this way, it is aimed to eliminate non-promising regions of the solution space and obtain acceptable schedules faster as discussed in Section 5. To give an example for decoding, in Table 2, a sample problem having five different jobs where each job has at most three different alternative routes is given. In Fig. 1, job process plan assignments of two different schedules generated for the data given in Table 2 are given. In these schedules, jobs with the same process plans are determined and the following crossover algorithm is applied as shown in Fig. 2:

Step 1 Genes having the same route for the same job in Chromosome 2 of randomly chosen Parent1 and Parent2 are chosen and marked. (In this example, jobs 1 and 3 are marked and they are referred as common jobs).

Step 2 Two offspring are created with empty chromosomes. In chromosome 1 of child 1, genes containing the operation information of common jobs are directly copied from chromosome 1 of parent 1 without losing their original position in the chromosome. The remaining empty genes of chromosome 1 of child 1 are filled with the genes of chromosome 1 of parent 2 where these genes contain the operation information of non-common jobs. If an overlapping occurs with previously placed genes of parent 1, genes of parent 2 are shifted until there is no overlapping. Size of chromosome 1 belonging to child 1 is adjusted by adding extra genes or deleting empty genes if necessary. Chromosome 2 of child 1 gets route information of common jobs from parent 1 and route information of non-common jobs from parent 2.

Table 1a
Flexibility rates of jobs (between Jobs 1 and 9).

Job	1	2	3	4	5	6	7	8	9
MO	6	6	6	10	8	6	11	9	6
TM	13	34	78	29	73	52	52	36	76
OF	2.17	5.67	13	2.9	9.13	8.67	4.73	4	12.7
PF	2	4	5	4	4	6	10	8	7
Flexibility	ÇD, ÇD	D, D	ÇY, O	ÇD, D	Y, D	O, O	D, ÇY	D, Y	ÇY, Y

Chromosome order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Chromosome 1 (Job#,machine)	(2,5)	(1,8)	(3,8)	(4,7)	(5,7)	(4,3)	(1,8)	(3,3)	(2,4)	(5,8)	(3,5)	(1,8)	(5,7)	(3,1)	(4,8)	
Chromosome 2 (Operations)	F	A	D	D	A	E	D	E	G	C	F	C	D	G	C	
Chromosome 2 (After clustering)	1	2	1	3	2											
	Job1	Job2	Job3	Job4	Job5											

Parent 2

Chromosome order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Chromosome 1 (Job#,machine)	(3,7)	(2,8)	(1,7)	(5,7)	(2,1)	(4,8)	(5,8)	(2,5)	(4,3)	(5,7)	(3,1)	(3,5)	(1,7)	(4,8)	(3,1)	(1,8)
Chromosome 2 (Operations)	D	C	A	A	E	D	C	F	E	D	E	F	D	C	G	C
Chromosome 2 (After clustering)	1	1	1	2	1											
	Job1	Job2	Job3	Job4	Job5											

Fig. 1. Double chromosome structure for integration module and clustering approach.

Table 2
Sample problem for decoding. (a) Alternative routes for each job. (b) Alternative machines for each operation.

Job #	Alternative routes		
	#1	#2	#3
1	A, D, C	B,E	A, B, F
2	C, E, F	F, G	–
3	D, E, F, G	B, C, D	–
4	B, F, G	A, B, E	D, E, C
5	A, C, D	A, B. G	–

Operation	Alternative machines	
	#1	#2
A	7	8
B	5	8
C	8	–
D	8	7
E	1	3
F	5	–
G	4	1

Step 3 Step 2 is repeated for child 2 where common job genes are copied from parent 2 and non-common job genes are copied from parent 1.
Step 4 Two off-springs with same or different chromosome size are created.

At each iteration of GA, the following mutation operations are applied after crossover operations:

Step 1 Machine mutation: Chromosome 1 of any individual in population is chosen. Random gene in Chromosome 1 is selected and corresponding operation's available machines are replaced (Fig. 3).
Step 2 Operation mutation: Chromosome 1 of any individual in population is chosen. Random two genes in Chromosome 1 is selected and interchanged (Fig. 4).
Step 3 New Chromosome 1 is created by considering operation sequence of a job and corresponding machine for the operation.
Step 4 Chromosome 2 remains the same.

4. Creating artificial neural networks with schedules obtained from Genetic Algorithms

Schedules obtained via GA described in the previous section are effective as long as the production conditions do not change. However, this is not realistic in real life. In order to strengthen our solution model against changing production conditions, we need a replanning module besides the integration module. For this reason, we introduce ANN into the solution model. (See Table 3)

During the run of GA, besides final schedules, production data such as "processing time", "machine load", "remaining processing time", "previous job sequence", and "previous machine sequence" are also calculated and stored for each operation (Table 4). Our experiments show that among available production data, "processing time", "machine load", and "remaining processing time" are most critical and they should be selected as inputs to ANN system after scheduling. This is also proved by our global sensitivity analysis which is given in the next section. The available data is used for training, testing and validation functions of ANN. ANN is trained and tested for several conditions such as cancelation of orders, changes in due dates of jobs, failure of a machine during operation, changes on number of machines that are very common in mass customization production environment. With this ANN based re-planning module, changes in production environment are considered for each phase of production.

The performance of ANN is measured by how well it predicts the inputs not used during training. Generalization issue is a common problem for training neural networks. To train and validate ANN, we divide the input data into two groups as training and validation input data, which is the common procedure for constructing ANN. These divided inputs are used to train the networks, verify and test the performance of networks, and realize the validation test, which determines how well new inputs are predicted.

For ANN system, the purpose is to get maximum correlation between predictions of neural networks and the target input which is the sequence of operations. In this study, we focus on Multilayer Perceptron (MLP) networks and Radial Basis Functions (RBF) networks. MLP is a supervised ANN and RBF is hybrid one, containing both supervised and unsupervised approaches. MLP is the most commonly used network type in the literature, yet its iterative training requirement makes it work slower than other network

Crossover

Parent1 - Chromosome 1	(2,5)	(1,8)	(3,8)	(4,7)	(5,7)	(4,3)	(1,8)	(3,3)	(2,4)	(5,8)	(3,5)	(1,8)	(5,7)	(3,1)	(4,8)	
Parent1 - Chromosome 2	1	2	1	3	2											
Parent2 - Chromosome 1	(3,7)	(2,8)	(1,7)	(5,7)	(2,1)	(4,8)	(5,8)	(2,5)	(4,3)	(5,7)	(3,1)	(3,5)	(1,7)	(4,8)	(3,1)	(1,8)
Parent2 - Chromosome 2	1	1	1	2	1											
Child 1																
Genes from Parent1		(1,8)	(3,8)				(1,8)	(3,3)			(3,5)	(1,8)		(3,1)		
Genes from Parent2		(2,8)		(5,7)	(2,1)	(4,8)	(5,8)	(2,5)	(4,3)	(5,7)				(4,8)		
Child 1 - Chromosome 1	(2,8)	(1,8)	(3,8)	(5,7)	(2,1)	(4,8)	(1,8)	(3,3)	(5,8)	(2,5)	(3,5)	(1,8)	(4,3)	(3,1)	(5,7)	(4,8)
Genes from Parent1	1		1													
Genes from Parent2		1		2	1											
Child 1 – Chromosome 2	1	1	1	2	1											
Child 2																
Genes from Parent1	(2,5)			(4,7)	(5,7)	(4,3)			(2,4)	(5,8)			(5,7)		(4,8)	
Genes from Parent2	(3,7)		(1,7)								(3,1)	(3,5)	(1,7)		(3,1)	(1,8)
Child 2 - Chromosome 1	(3,7)	(2,5)	(1,7)	(4,7)	(5,7)	(4,3)	(2,4)	(5,8)	(5,7)	(4,8)	(3,1)	(3,5)	(1,7)	(3,1)	(1,8)	
Genes from Parent1		2		3	2											
Genes from Parent2	1		1													
Child 2 – Chromosome 2	1	2	1	3	2											

Fig. 2. Representation of crossover operation.

Chromosome 1 (Job #,machine)	(2,5)	(1,8)	(3,8)	(4,7)	(5,7)	(4,3)	(1,8)	(3,3)	(2,4)	(5,8)	(3,5)	(1,8)	(5,7)	(3,1)	(4,8)
Chromosome 2 (Clustering)	1	2	1	3	2										

Mutation

*Selected gene (corresponding operation is D and available machines are 7 and 8)

Chromosome 1 (Job #,machine)	(2,5)	(1,8)	(3,8)	(4,7)*	(5,7)	(4,3)	(1,8)	(3,3)	(2,4)	(5,8)	(3,5)	(1,8)	(5,7)	(3,1)	(4,8)

After Mutation

Chromosome 1 (Job #,machine)	(2,5)	(1,8)	(3,8)	(4,8)*	(5,7)	(4,3)	(1,8)	(3,3)	(2,4)	(5,8)	(3,5)	(1,8)	(5,7)	(3,1)	(4,8)

*New gene after machine mutation

Chromosome 2 (Clustering)	1	2	1	3	2

Chromosome 2 remains the same

Fig. 3. Machine mutation operator.

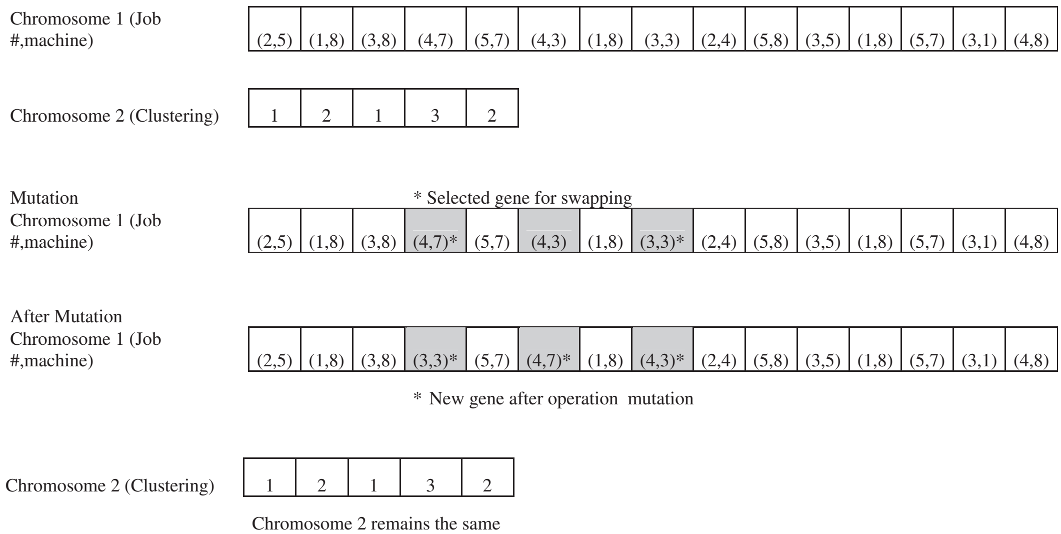

Fig. 4. Operation mutation operator.

Table 3
Schedules with different routes. (a) Schedule 1 (b) Schedule 2.

Job #	Route
1	1
2	2
3	1
4	3
5	2
Job #	**Route**
1	1
2	1
3	1
4	2
5	1

types. However, MPL networks are generally smaller and once they are trained, they work faster and give better results compared to other network structures. RBF networks are larger than MLP networks and usually have worse performance especially when the number of available input variables is large. This is caused by sensitiveness of RBF networks to ineffective input variables. RBF networks are not suitable for modeling the data with categorical inputs since its basic functions are located in continuous space but categorical inputs are located in discrete space (Haykin, 1994).

Generally MLP, which is a back-propagation network type, includes one input layer, one or more hidden layers, and one output layer. The number of neurons in input and output layers depends on the number of input and output data in the problem. Some theorems have been formulated for the assessment of the ideal number of neurons in hidden layers for the case of one three-layer networks, such as Kolmogorov's Theorem (Kolmogorov, 1957) and Fletcher and Goss's Theorem (Fletcher & Goss, 1993). Also for some cases, rule-of-thumb methods are used for determining the number of neurons in the hidden layer, such as the number of neurons in hidden layer are 2/3 of the number of neurons in input layer (Boger & Guterman, 1997), the number of hidden layer neurons shall be less than twice of the number of neurons in input layer (Berry & Linoff, 1997) and the size of the hidden layer neurons is between the input layer size and the output layer size etc. (Blum, 1992). However these methods are not always true since the input layer and the output layer cannot decide the size of the hidden layer neurons alone. There are also other factors, such as the complexity of the activation function, the training algorithm, the architecture of neural network and the training samples database. In this study a structured trial and error method is used for creating a neural network's layer approximation. The network for our problem consists of four neurons in input layer and one neuron in output layer. There is only one hidden layer and different functions are used for hidden and output layer of network structure (Table 5).

For RBF network type, although there are three layers which are input, hidden and output layers, transmission from input layer to hidden layer is provided by nonlinear radial basis activation functions. Transmission between hidden layer and output layer is a linear transmission. In RBF free parameters are the central vectors, output layer weights and the width of the radial functions. Given an input vector x, the output of the RBF network is the activity

Table 4
Sample data after integration module.

Chromosome number	Job	Operation sequence	Machine	Prev. job order	Prev. machine	Starting time	Finish time	Process time	Machine load	Job remaining time
0	5	14	2	−1	−1	0	14	14	159	92
17	1	0	14	−1	−1	0	13	13	55	99
4	17	3	1	1	−1	8	21	13	139	76
5	11	7	3	−1	1	8	26	18	65	89
7	6	0	8	−1	6	11	28	17	98	103
29	0	1	15	11	−1	13	31	18	87	160
20	5	15	8	0	7	28	62	34	81	78
35	0	4	13	29	−1	31	61	30	54	142
...	...	...	...	...	...	...	...	...	...	...

Table 5
Comparison of different scenarios.

OF	PF	No. of Job	Jobs	Makespan			CPU time			Deviation from the best sol		
				ARA	BRA	BRA_O	ARA	BRA	BRA_O	ARA	BRA	BRA_O
D	L	4	1–4	169	169	169	19.18	4.52	4.51	0	0	0
M	H	4	5–8	110	104	104	80.34	116.62	187.71	0.06	0	0
M	L	4	9–12	100	85	85	21.79	30.09	29.95	0.18	0	0
M	M	15	1–15	203	170	200	814.98	431.37	90.11	0.19	0	0.18
M	M	18	1–18	240	177	200	1596.69	1180	1284.92	0.36	0	0.13
M	L	3	1–3	169	169	169	5.57	3.81	3.51	0	0	0
M	L	6	1–6	169	169	169	33.82	6.9	6.7	0	0	0
M	M	10	1–10	197	169	169	984.85	171.96	76.23	0.16	0	0
M	M	13	1–13	218	169	169	296.84	244.67	189.21	0.29	0	0

vector y which is given by Eq. (5). In this study Gauss function (6) is used for hidden layer activation function and Identity function is used for output layer activation function.

$$Y(x) = \sum_{i=1}^{N} W_{ij} A_i(x) \tag{5}$$

$$Q(r) = \exp\frac{\|x - c_i\|^2}{2s_i^2} \tag{6}$$

W_{ij} the weight between ith neuron of hidden layer W and jth neuron of output layer, $A_i(x)$ Activation function (For this problem it is Gauss function), x Input vector, c_i Center, $x - c_i$ the standard Euclidean distance, si width, $Q(r)$ Gaussian basis function for the hidden units, $Y(x)$ Activity vector

For performance measurement of the selected networks, the correlation coefficient between the target data and the predictions of networks are calculated. Correlation coefficients can take any value between -1 and 1 where 1 means perfect accuracy for neural networks. However, since target data may include measurement errors, to get 1 for correlation coefficient is not a desirable situation. To get 1 for correlation coefficient means that the network shows maximum performance for training of the data, but poor performance for the test and validation of data. Therefore, while evaluating the performance of network structures, correlation coefficients of test and verification data should be used instead of correlation coefficients of training data. Another point to consider is when the network has a correlation degree smaller than one, this does not necessarily show that the network is trained badly since some of the errors originated from data collection process can be corrected by network during the prediction phase. Actually this means that the used neural network structure is more conservative and preferable. In this study, since the data is created by a heuristic approach (GA), ANN's performance shows a proof for the accuracy of GA's solution. Therefore, there is a two-way control mechanism for the integration module and re-planning module.

5. Experimental design and results

Genetic Algorithm is coded in C++ and implemented on a computer with a 3.10 Ghz Intel Core i5-2400 CPU. All alternative process plans created in PP module are directly used as inputs for the integration module. The final schedule for all jobs and the optimal process plan for each job are created by the integration module. GA is run for PP data of three different scenarios which are: (1) all feasible routes found according to the minimum processing time method in PP module (ARA), (2) the best two routes found according to the minimum processing time method in PP module (BRA), and (3) the best two routes found according to the average processing time method in PP module (BRA_o). The results are compared for different performance criteria such as Makespan length (time to complete all jobs), CPU time (computational time for final schedule) and deviation from the best solution (percentage difference of makespans from the best makespan) (Table 6).

In Table 5, to show the performance and efficiency of all three approaches, nine test-bed problems are constructed and their

Table 6
First 10 network having best performances.

Seq.	Network	Training perf.	Testing perf.	Training error	Testing error	Training algorithm	Error function	Hidden layer activation fun.	Output layer activation fun.
1	MLP 4-44-1	0.966629	0.968571	0.002807	0.002714	BFGS 578	SOS	Tanh	Tanh
2	MLP 4-25-1	0.947420	0.953585	0.004378	0.003973	BFGS 375	SOS	Tanh	Identity
3	MLP 4-34-1	0.957677	0.958605	0.003542	0.003556	BFGS 531	SOS	Tanh	Identity
4	MLP 4-42-1	0.955279	0.957221	0.003739	0.003670	BFGS 515	SOS	Logistic	Tanh
5	MLP 4-46-1	0.954130	0.958972	0.003832	0.003536	BFGS 526	SOS	Logistic	Identity
6	MLP 4-34-1	0.949904	0.953418	0.004177	0.003988	BFGS 283	SOS	Tanh	Tanh
7	MLP 4-45-1	0.960674	0.964954	0.003300	0.003037	BFGS 487	SOS	Logistic	Logistic
8	MLP 4-47-1	0.958058	0.959122	0.003512	0.003515	BFGS 408	SOS	Tanh	Exponential
9	MLP 4-29-1	0.948332	0.952953	0.004304	0.004028	BFGS 396	SOS	Logistic	Tanh
10	MLP 4-41-1	0.954478	0.956190	0.003806	0.003757	BFGS 423	SOS	Logistic	Logistic

Table 7a
Fuzzy data used in building Neural Networks.

Operation				Process time			Remaining processing time			Machine loading	
First	Middle	Later	Last	Short	Medium	Long	Short	Middle	Long	Heavy	Light
0	0	0.5	0.5	0.957	0.043	0	0	0.935	0	0.426	0.5738
0	0	0.25	0.75	0	0	1	0	0.891	0	0	1
1	0	0	0	0.609	0.391	0	0	0.783	0.217391	0.131	0.8689
1	0	0	0	1	0	0	0	0.804	0	0.82	0.1803
0	0	0	1	0.522	0.478	0	0	0.174	0	0	1
0	0	1	0	0.261	0.739	0	0	0.5	0	1	0
0	0.25	0.75	0	0	0	1	0	0.87	0.130435	0	1
1	0	0	0	0.957	0.043	0	0	0	0.306122	0.459	0.541
0	0	0	1	0	0	1	0	0.783	0.217391	0	1
1	0	0	0	0	0	1	0	0.63	0.369565	0.295	0.7049
1	0	0	0	0.957	0.043	0	0	0.63	0	0	1
0.25	0.75	0	0	0.522	0.478	0	0	0.913	0	0.951	0.0492
1	0	0	0	0.522	0.478	0	0	0	0.571429	0.262	0.7377
0	0	0.75	0.25	1	0	0	0	0.478	0.521739	0.41	0.5902
0	0	0	1	0.522	0.478	0	0.213	0	0	0	1
0.75	0.25	0	0	0	0.739	0.261	0	0.565	0	0.18	0.8197
1	0	0	0	0.609	0.391	0	0	0.348	0	0.672	0.3279
0	0	0	1	0	0.043	0.957	0	0.848	0	0	1
0	1	0	0	1	0	0	0	0.522	0	0.639	0.3607
0	0	0.25	0.75	0	0.13	0.87	0	0.652	0	0.59	0.4098
0	0	0.5	0.5	0	0.739	0.261	0	0.717	0.282609	1	0
1	0	0	0	0	0.478	0.522	0	0	0.938776	1	0
1	0	0	0	0.957	0.043	0	0	0.913	0.086957	0.754	0.2459
0	0	0.5	0.5	0	1	0	0	0.043	0	0.016	0.9836

Table 7b
Fuzzy data used in building Neural Networks.

Membership values for each sequence position					
Seq1	Seq2	Seq3	Seq4	Seq5	Seq6
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
0.8888889	0.1111111	0	0	0	0
0.7777778	0.2222222	0	0	0	0
0.6666667	0.3333333	0	0	0	0
0.5555556	0.4444444	0	0	0	0
0.4444444	0.5555556	0	0	0	0
0.3333333	0.6666667	0	0	0	0
0.2222222	0.7777778	0	0	0	0
0.1111111	0.8888889	0	0	0	0
0	1	0	0	0	0
0	0.8888889	0.1111111	0	0	0
0	0.7777778	0.2222222	0	0	0
0	0.6666667	0.3333333	0	0	0
0	0.5555556	0.4444444	0	0	0
0	0.4444444	0.5555556	0	0	0
0	0.333333	0.666667	0	0	0
0	0.222222	0.777778	0	0	0
0	0.111111	0.888889	0	0	0
0	0	1	0	0	0

process and operational flexibilities are defined. Genetic parameters in integration module are: Population size: 150, Number of Generations: 200, Tournament size: 3, Replacement percentage: 0, 2. The termination condition of GA is the maximum number of generations, which is set to 200.

Results displayed in Table 5 show that when routes are selected according to the minimum process plans, instead of using all feasible routes (ARA), using the best two routes of each job (BRA) generally gives better results in terms of makespan length and CPU time. We can explain this by the effect of solution space size. As the size of solution space increases by including all feasible routes in GA, the chance of converging to a local optimal solution decreases. Depending on this result, we conclude that in the optimal schedule the probability of having best routes used (process plans with minimum time requirement for each job) is quite high. Therefore using selected best routes instead of using all routes in scheduling process is more practical.

When BRA and BRA_o are compared, for the tasks with less jobs, BRA_o generally shows better CPU time performance, yet BRA_o shows worse performance than BRA approach in terms of makespan length. Again this result supports us to use best two routes with minimum processing times as a good strategy for integrating Scheduling and PP.

After this integration phase, by using ANN's training, testing and validation functions, first 10 best performing network structures are created and used for the predictions of new schedules in re-planning module (Table 6). Best performing network structures are all from MLP networks. This shows again that MLP networks are superior to RBF networks for most problems in different fields.

6. Rebuilding of neural network data via the implementation of fuzzy logic

During the research, while evaluating the results of ANN it has been observed that new operation sequences predicted by networks are close to the values of target output (operation sequence) but cannot be obtained discretely. In other words, since the sequences cannot be obtained as an integer, it cannot be used efficiently for the Re-planning Module. To solve this problem, inputs are recreated by using fuzzy logic. Training data is classified by using fuzzy membership function before it is submitted to the use of ANN. To calculate the fuzzy membership values, a new code is created in MATLAB and the output of this code is used as new training data set (Tables 7a and 7b). In Table 7b, the priority of each operation is shown by sequence number. If sequence number increases, the priority of that operation decreases.

Table 8
Global Sensitivity Analysis.

	Processing time			Machine load		Remaining process time			Operation sequence			
	Short	Medium	Long	Heavy	Light	Short	Middle	Long	First	Middle	Later	Last
1.MLP 12-45-16	24.1	17.6	19.7	15.8	28.2	19.7	10.8	34.2	28	13.8	21.2	43.5
2.MLP 12-62-6	29.8	14.6	13.7	19.5	22.4	13.5	33.4	9.03	22.5	10.2	10.9	21.3
3.MLP 12-57-6	61.8	51.4	57	48.6	53.6	47.4	143.9	17.8	184.6	47.4	55.8	109.1
4.MLP 12-61-6	14.9	10.6	10.6	12.1	14	9.5	24.3	8.7	15.7	10	9.4	12.9
5.MLP 12-47-6	36.2	31	26.9	49.1	62.7	13.4	56.3	10.1	26.8	9.8	19.6	41.3
6.MLP 12-58-6	42.6	19.5	41.1	16	38.7	17.4	78.5	28.5	97.1	28.6	34.5	92.5
7.MLP 12-57-6	28.1	23.6	22.1	22.3	31.8	11.5	29.5	8.5	28.5	14.9	17.6	29.1
8.MLP 12-44-6	2184	2484	1385	780.2	1986	2727	1353	62.6	1503	23.4	60.8	2204
9.MLP 12-56-6	20.2	19.5	19.5	14.4	21.1	8.9	20.9	8.1	21.1	13.2	14.8	22.4
10.MLP 12-60-6	28.5	30.9	29.6	19.2	70.5	29.6	22.5	111.4	128	38	28.5	137

Table 9
First 10 network having best performances with fuzzy data.

Seq	Network	Training perf	Testing perf.	Training Error	Testing Error	Training algorithm	Error function	Hidden layer activation fun.	Output layer activation fun.
1	MLP 12-45-6	0.936091	0.921104	0.031423	0.037871	BFGS 440	SOS	Logistic	Identity
2	MLP 12-62-6	0.938784	0.925186	0.030059	0.035868	BFGS 386	SOS	Logistic	Identity
3	MLP 12-57-6	0.957086	0.946883	0.021221	0.025629	BFGS 878	SOS	Tanh	Identity
4	MLP 12-61-6	0.932456	0.922572	0.034200	0.038044	BFGS 355	SOS	Exponential	Tanh
5	MLP 12-47-6	0.940730	0.924414	0.029226	0.036686	BFGS 461	SOS	Logistic	Identity
6	MLP 12-58-6	0.946348	0.935161	0.026621	0.031763	BFGS 407	SOS	Exponential	Identity
7	MLP 12-57-6	0.942229	0.929762	0.028362	0.033846	BFGS 429	SOS	Logistic	Identity
8	MLP 12-44-6	0.935665	0.923301	0.030440	0.034629	BFGS 365	SOS	Logistic	Exponential
9	MLP 12-56-6	0.938199	0.927488	0.030600	0.035069	BFGS 402	SOS	Logistic	Tanh
10	MLP 12-60-6	0.945065	0.934681	0.027240	0.031921	BFGS 380	SOS	Exponential	Identity

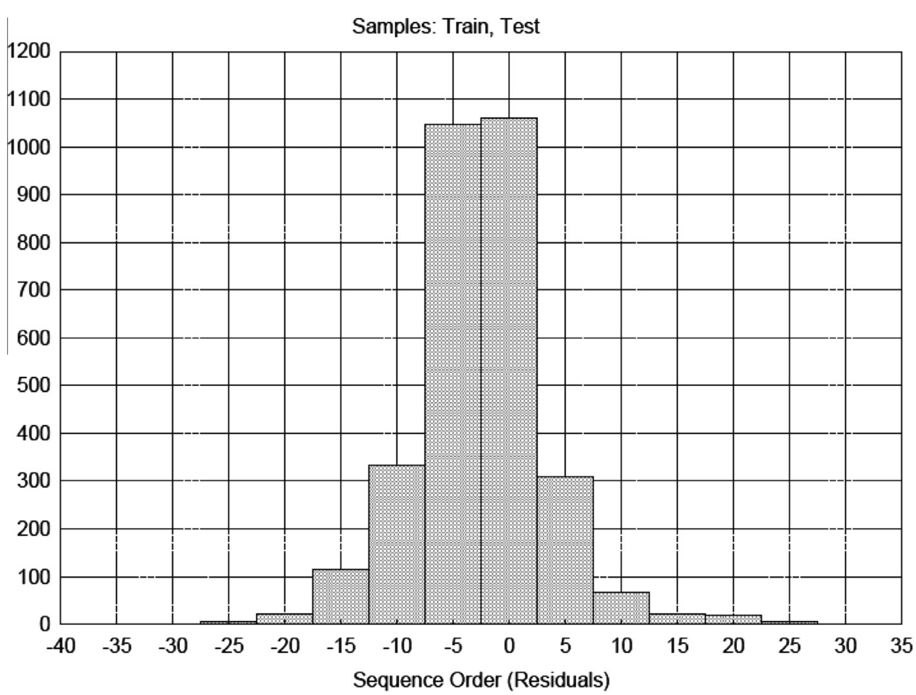

Fig. 5. Residuals.

Also to test the effectiveness of using "processing time", "machine load", and "remaining processing time" data to predict the operation sequence, we do global sensitivity analysis as seen in Table 8. In global sensitivity analysis, for a variable having a network error value less than one means that the respective variable is unnecessary and should not be included in the network. Table 8 shows that for all three variables (processing time, machine load, and remaining processing time), network error value is greater than one, which means we choose the critical inputs for ANN.

With this new input data, all ANN procedures are reapplied and new network structures are created (Table 9). For this new network structures, predictions and global sensitivity analysis are repeated.

In Fig. 5, to show how accurate the networks created for predicting the operation sequence, residuals for training and testing

Table 10
Comparing Neural Networks predictions with prior Scheduling results.

Neural Network ANN's predictions						MLP 12-45-6 Target output					
Seq1	Seq2	Seq3	Seq4	Seq5	Seq6	Seq1	Seq2	Seq3	Seq4	Seq5	Seq6
1.0026	0.0046	−0.0293	−0.0402	0.0390	0.0103	1	0	0	0	0	0
1.0105	−0.0042	−0.005	0.0075	0.0063	0.0026	1	0	0	0	0	0
0.911	0.1407	0.0663	−0.004	−0.0456	−0.0396	1	0	0	0	0	0
0.9860	0.0054	−0.0549	0.0489	0.0025	0.0089	1	0	0	0	0	0
0.9878	−0.0019	0.0008	−0.0023	0.0030	0.0109	1	0	0	0	0	0
0.9990	−0.0119	−0.0081	0.0014	−0.0087	0.0196	1	0	0	0	0	0
0.9990	0.0028	0.0047	−0.0058	−0.0008	0.0165	1	0	0	0	0	0
0.9922	−0.0042	0.0062	−0.0072	−0.0034	0.002	1	0	0	0	0	0
1.1546	−0.0127	−0.0407	−0.0353	−0.0464	−0.0362	1	0	0	0	0	0
0.9825	−0.0051	0.0176	−0.0072	−0.0157	0.0079	1	0	0	0	0	0
0.8330	0.1549	0.0221	0.0051	−0.0036	0.0277	0.8888	0.1111	0	0	0	0
0.8126	0.1707	0.0036	−0.0269	0.0058	0.0186	0.7777	0.2222	0	0	0	0
0.6743	0.3423	−0.0197	−0.0135	−0.0015	0.0007	0.6666	0.3333	0	0	0	0
0.5342	0.4920	0.1196	−0.1336	0.0254	−0.0128	0.5555	0.4444	0	0	0	0
0.5331	0.3978	0.0183	−0.1279	0.0758	0.0648	0.4444	0.5555	0	0	0	0
0.3894	0.5842	−0.0111	0.0134	0.0046	−0.0126	0.3333	0.6666	0	0	0	0
0.2320	0.6336	0.15586	0.0000	−0.0102	0.0133	0.2222	0.7777	0	0	0	0
0.01627	0.78798	0.2035	−0.0231	0.00946	−0.0083	0.11111	0.88889	0	0	0	0
0.0063	1.0679	−0.0284	0.01835	−0.0312	0.01165	0	1	0	0	0	0

Table 11
Comparing Neural Networks predictions with prior Scheduling results after changes in production conditions in mass customization production environment.

Neural Network: ANN's predictions						MLP 12-45-6 Sequences before changes					
Seq1	Seq2	Seq3	Seq4	Seq5	Seq6	Seq1	Seq2	Seq3	Seq4	Seq5	Seq6
129015	0.41643	−2.1826	−0.9878	0.42306	1.91768	1	0	0	0	0	0
0.08107	−0.1494	1.73562	0.63879	−1.6145	0.23557	1	0	0	0	0	0
0.91100	0.14070	0.06638	−0.0040	−0.0456	−0.0396	1	0	0	0	0	0
−0.5682	0.88199	−0.6140	−0.3910	0.50653	1.05761	1	0	0	0	0	0
−0.1066	0.19757	0.41999	−0.0952	0.29994	0.27535	1	0	0	0	0	0
2.22743	−0.3952	−1.6614	−0.3714	−0.1531	1.21154	1	0	0	0	0	0
0.97866	−0.3995	1.25160	0.28078	−1.3725	0.19123	1	0	0	0	0	0
0.99220	−0.0041	0.00623	−0.0072	−0.0034	0.00200	1	0	0	0	0	0
1.15465	−0.0126	−0.0406	−0.0353	−0.0464	−0.0361	1	0	0	0	0	0
0.98253	−0.0051	0.01765	−0.0072	−0.0156	0.00790	1	0	0	0	0	0
0.83307	0.15495	0.02217	0.00518	−0.0035	0.02774	0.8888	0.1111	0	0	0	0
0.35245	0.27284	1.03958	0.45310	−0.0953	−0.9081	0.7777	0.2222	0	0	0	0
0.67438	0.34239	−0.0197	−0.0135	−0.0014	0.00076	0.6666	0.3333	0	0	0	0
0.53423	0.49202	0.11961	−0.1335	0.02549	−0.0127	0.5555	0.4444	0	0	0	0
0.53315	0.39786	0.01836	−0.1278	0.07589	0.06481	0.4444	0.5555	0	0	0	0
0.38949	0.58421	−0.0110	0.01349	0.00463	−0.0126	0.3333	0.6666	0	0	0	0
0.23209	0.63361	0.15586	−0.0004	−0.0102	0.01332	0.2222	0.7777	0	0	0	0
0.01627	0.78798	0.20350	−0.0231	0.00946	−0.0083	0.11111	0.88889	0	0	0	0
0.00630	1.06790	−0.0283	0.01835	−0.0311	0.01165	0	1	0	0	0	0

samples are studied and it is observed that the data has a normal distribution with mean zero. This proves that noise of target variable for the network predictions is normally distributed. In addition, since the noises spread over a wide area as shown in Fig. 5, our Residuals Analysis shows that the data is created correctly and accurately.

By using created networks in Re-planning module, the predictions for new schedules are made for the case where no input is available and job floor conditions are not changed. The system's performance is measured by comparing new schedules with the schedules created in integration module (Table 10). By this comparison, it is shown that Re-planning module is able to produce accurate predictions. Another case study is presented by using the changing job floor conditions. For this case, values of input variables are changed and a new data set is created. With these inputs, new schedules are produced by using pre-created network structures and changes in operation sequences and priorities are observed (Table 11).

7. Conclusions

Process Planning (PP) and Scheduling are critical interrelated activities that affect the efficiency of overall production process. The relationship between PP and Scheduling requires manufacturing companies to synchronize these activities during planning stage, yet most companies do PP and Scheduling sequentially instead of doing simultaneously. In this study, we propose an integrated solution model that does PP and Scheduling at the same time. Our model also allows us to generate efficient production plans without much effort as the production constraints change.

In our solution model, alternative process plans are created with pre-defined optimization criteria and constraints. Best alternative process plans are chosen to be used for the integration module where PP and Scheduling done simultaneously. By allowing having alternative process plans (routes) for the jobs, we preserve system flexibility. On the other hand, by eliminating non-promising process plans, we narrow down the solution space of the integration problem, which consequently decreases computational time. Although we integrate PP and Scheduling, the proposed model still can be used for the factories having both Scheduling and PP departments since process plans and schedules are created separately. This makes reorganization of the factory unnecessary.

Our solution model also allows production planners to modify existing production plans as shop floor conditions change. We achieve this by introducing ANN in the solution model. To construct the final ANN, we use the data provided by the integration module. ANN is able to predict the operation sequence accurately and quickly for the case of changing production environment and customer demands. ANN's accuracy also proved by comparing the predictions of ANN with GA's solutions for scheduling in integration module (Table 10). Introduction of fuzzy membership functions in Artificial Neural Network (ANN) model allows us to generate fuzzy rules for the production environment. As a result of producing schedules quickly and accurately, we think that the proposed FNN approach is as a promising approach for both increasing the production efficiency and decreasing the computational time of the integration problem. For future research, the integration problem can be extended by adding new constraints such as machine, tool and fixture changes or considering setup and transportation time separately from processing time of machines.

References

Berry, M. J. A., & Linoff, G. (1997). *Data mining techniques*. NY: John Wiley & Sons.

Blum, A. (1992). *Neural networks in C++*. NY: Wiley.

Boger, Z., & Guterman, H. (1997). Knowledge extraction from artificial neural network models. In *IEEE systems, man, and cybernetics conference*. Orlando, FL, USA.

Chen, K. J., & JIA, B. P. (2007). A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval. *Expert Systems with Applications*, 1004–1010.

Detand, J., Kruth, J. P., & Kempenaers, J. (1992). A computer aided process planning system that increases the flexibility of manufacturing. In *IPDES workshop, (4–5) July*. Senlis, France.

Fletcher, D., & Goss, E. (1993). Forecasting with neural networks: an application using bankruptcy data. *Information and Management, 24*, 159–167.

Gaalman, G. J. C., Slomp, J., & Suresh, N. C. (1999). Towards an integration of process planning and production planning and control for flexible manufacturing systems. *International Journal of FMS, 11*(1), 5–17.

Gindy, N., Saad, S., & Yue, Y. (1999). Manufacturing responsiveness through integrated process planning and scheduling. *International Journal of Production Research, 37*(11), 2399–2418.

Glover, F. (1989). Tabu search – Part 1. *ORSA Journal on Computing, 1*(2), 190–206.

Glover, F. (1990). Tabu search – Part 2. *ORSA Journal on Computing, 2*(1), 4–32.

Grabowik, C., Kalinowski, K., & Monica, Z. (2005). Integration of the CAD/CAPP/PPC systems. *Journal of Materials Processing Technology, 164–165*(2), 1358–1368.

Guo, Y. W. et al. (2009). Optimization of integrated process planning and scheduling using a particle swarm optimization approach. *International Journal of Production Research, 47*(14), 3775–3796.

Haddadzade, M., Razfar, M. R., & Farahnakian, M. (2009). Integrating process planning and scheduling for prismatic parts regard to due date. In *Proceedings of world academy of science, engineering March 23–25(51)* (pp. 64–66). Hong Kong, China.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York, USA: Macmillan Publishing.

Huang, S. S., Zhang, H. C., & Smith, M. L. (1995). A progressive approach for the integration of process planning and scheduling. *IIE Transactions, 27*, 456–464.

Iwata, K., & Fukuda, Y. (September 1989). A new proposal of dynamic process planning in machine shop. In *Proceedings of CIRP international workshop on computer aided process planning, 21–22* (pp. 73–83). Germany: Hanover University.

Kempenaers, J., Pinte, J., & Detand, J. (1996). A collaborative process planning and scheduling system. *Advances in Engineering Software, 25*(1), 3–8.

Kim, K., & Egbelu, P. (1999). Scheduling in a production environment with multiple process plans per job. *International Journal of Production Research, 37*(12), 2725–2753.

Kolmogorov, A. N. (1957). On the representations of continuous functions of many variables by superpositions of continuous functions of one variable and addition. *Doklady Akademii Nauk USSR, 114*(5), 953–956.

Lee, H., & Kim, S. (2001). Integration of process planning and scheduling using simulation based genetic algorithms. *International Journal of Advanced Manufacturing Technology, 18*(8), 586–590.

Moon, C., Kim, J., & Hur, S. (2002). Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain. *Computers and Industrial Engineering, 43*(1), 331–349.

Seker, A. (January 2013). *A hybrid approach combining artificial neural networks and genetic algorithm to integrate process planning and scheduling for mass customization* (pp. 21–22). Gazi University Institute of Science and Technology, Unpublished PHD Thesis.

Shen, W., Wang, L., & Hao, Q. (2006). Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews, 36*(4), 563–577.

Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling – A review. *Journal of İntelligent Manufacturing, 11*(1), 51–63.

Tonshoff, H. K., Beckendorff, U., & Andres, N. (September 1989). FLEXPLAN: A concept for intelligent process planning and scheduling. In *CIRP international workshop, 21–22* (pp. 87–106). Hannover, Germany.

Wang, J. et al. (2009). Reducing tardy jobs by integrating process planning and scheduling functions. *International Journal of Production Research, 47*(21), 6069–6084.

Weintraub, A. et al. (1999). Scheduling with alternatives: A link between process planning and scheduling. *IIE Transactions, 31*, 1093–1102.

Zhang, Y., Saravanan, A., & Fuh, J. (2003a). Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research, 41*(3), 611–628.