Mehmet Bayram Yildirim Industrial Engineering

Machine number, priority rule, and due date determination in flexible manufacturing systems using artificial neural networks

Mehmet Bayram Yildirim
Wichita State University, bayram.yildirim@wichita.edu

Tarik Cakar
Sakarya University, Esentepe, Adapazari, 54040, Turkey

Ufuk Doguc
Sakarya University, Esentepe, Adapazari, 54040, Turkey

Jose Ceciliano Meza
Wichita State University, jlcecilianomeza@gmail.com

Machine Number, Priority Rule, and Due Date Determination in Flexible Manufacturing Systems Using Artificial Neural Networks

Mehmet Bayram Yildirim[†1], Tarik Cakar[‡], Ufuk Doguc[‡] and Jose Ceciliano Meza[†]

[†]Department of Industrial and Manufacturing Engineering, Wichita State University
Wichita, KS, 67220-0035, USA

[‡]Industrial Engineering Department, Sakarya University
Esentepe, Adapazari, 54040, Turkey

[1] The corresponding Author, Tel: +1-316-978 3426, Fax: +1-316-978 3742, Email: Bayram.Yildirim@wichita.edu

Machine Number, Priority Rule, and Due Date Determination Using Artificial Neural Networks

Abstract:

When there is a production system with excess capacity, i.e., more capacity than the demand for the foreseeable future, upper management might consider utilizing only a portion of the available capacity by decreasing the number of workers or halting production on some of the machines/production lines, etc., while preserving the flexibility of the production system to satisfy demand spikes. To achieve this flexibility, upper management might be willing to attain some pre-determined/desired performance values in a production system having identical parallel machines in each work center. In this study, we propose a framework that utilizes parallel neural networks to make decisions on the availability of resources, due-date assignments for incoming orders, and dispatching rules for scheduling. This framework is applied to a flexible manufacturing system with work centers having parallel identical machines. The artificial neural networks were able to satisfactorily capture the underlying relationship between the design and control parameters of a manufacturing system and the resulting performance targets.

Keywords: Artificial Neural Networks, Priority Rules, Due-Date Assignment, Flexible Manufacturing System, Inverse Scheduling

1. Introduction

In the global competitive market, several factors might affect the management of demand and capacity. For example, seasonality in demand, a new competitor in the global market, or an economic or political crisis might force underutilization of the available capacity. In any of these cases, upper management might not have the luxury of running production at capacity for a long period of time. One alternative might be to reduce short-run operational capacity by shutting certain production lines/work stations/machines while preserving the flexibility to satisfy demand spikes. This could lead to opportunities for larger orders if the incoming orders are satisfied while achieving some critical and conflicting objectives such as faster delivery speed, greater reliability, higher customer satisfaction, and minimum cost. To achieve these objectives, management might need a decision-support tool that will provide the optimal resource structure and scheduling policy. For a given resource structure and scheduling policy, computing the performance of a production system is straightforward. However, to employ ”what-if” analysis requires simulation. If the goal is to maintain certain performance measures at predetermined levels to accommodate the unexpected demand, then a what-if approach, which might require extensive simulation, may not be feasible at the operational level. Therefore, an intelligent decision system is necessary to support management's operational decisions in the short run.

Our goal is to utilize neural networks and simulation as tools for scheduling jobs on the shop floor of a flexible manufacturing system setting in order to achieve the desired system goals set by upper management; specifically we will determine the number of machines in each work center (WC) (design decision), the priority rule used on the shop floor and the due date (scheduling decision) for incoming orders.

2. Literature Review

A flexible manufacturing system (FMS) is a computer-controlled production system with a set of connected CNC machines in which routing, loading, and scheduling operations are controlled by a central computer. Fry and Smith (1989) consider the most significant objectives for an FMS as meeting the due dates; maximizing the system and machine utilization rates; minimizing the level of work in process inventory; maximizing the production rate; minimizing the setup, preparation, and team change times; minimizing the mean flow time; maximizing the capability of performing jobs at parallel work stations; and operating according to the pre-determined capacity. In an FMS system, design decisions include selection of hardware, machine types, part types, the material handling system, tools, apparatuses and pallets, and staffing. Control process decisions address short- to medium-range operational issues such as general planning, system setup, and scheduling. Mellichamp and Wahab (1987) identify testing on a small-scale physical model of the system, using analytical methods and simulation as approaches for performance analysis of design alternatives. Chan and Rathmill (1985) propose a three-step approach that includes planning, design, and system setup in an FMS. Stecke (1983) shows that the system performance in an FMS literally depends on selected job loading and control strategies. Edghill and Cewsswell (1986) conclude that generalized control mechanisms will result in performance levels inferior to mixed scheduling policies.

Flexibility in an FMS makes it difficult to use analytical methods to find the optimal solution to planning, scheduling, and control problems (Mahmoodi et al., 1999). When analytical methods are used, the original system is usually over simplified. Even in this case, the resulting problem requires extensive computational power because most of the scheduling problems in a manufacturing system are NP-hard problems (Chryssolouris et al., 1990). Therefore, methods like simulation, artificial neural networks (ANNs), simulated annealing, and metaheuristics are typically utilized to address such complexities. For example, Sridharan and Babu (1998) use simulation to investigate the effect of scheduling rules on the FMS performance. Sabuncuoglu and Karabuk (1998) recommend artificial intelligence algorithms for solving scheduling problems in an FMS environment. Fonseca and Navaresse (2002) conclude that ANNs can be used as a valid alternative to a simulation approach. Some of the other approaches used in FMS design and control are genetic algorithms for FMS scheduling (Jawahar et al., 1998), a mathematical modeling approach to optimally solve job shop problems (Gomez et al., 2005), artificial neural networks (Vujosevic, 1994), a hybrid dynamic programming approach based on a genetic algorithm in FMS scheduling (Yang, 2001), an inductive learning and neural network method for multi-purpose FMS scheduling (Kim et al., 1998), a multi-objective simulated annealing approach (Loukil et al., 2005), and a multi-objective scheduling framework for hierarchical FMS control (Tung et al., 1999). Unless simplistic assumptions are made to model and solve these complex problems, the number of control variables in designing and controlling any automated manufacturing system is usually very large. Hence, solving design and control problems simultaneously in an FMS setting might require a vast amount of data and/or an extensive amount of computational effort.

An alternative might be utilizing a combination of the above-mentioned methods to determine a good solution efficiently. One way to speed up this process is to utilize ANNs and simulation together. Neural networks have several applications including scheduling and production system design on FMS problems (Chryssolouris et al., 1990 and Feng et al., 2003).

Vujosevic (1994) proposes an FMS design framework integrating simulation and a rule-based initial processor that compares simulation results with target values. Given a set of targeted performance measures, Chryssolouris et al. (1990) determine the number of machines in each WC in an FMS. Cakar et al. (2005) extend this work to also select the priority rule used in scheduling. Phillom et al. (1994) propose a method to determine the due date for incoming orders using ANNs along with alternative due-date selection rules. We synthesize the approaches of Chryssolouris et al. (1990), Phillom et al. (1994), and Cakar et al. (1998) to determine the number of machines and the priority rule to be used in scheduling the jobs at each WC, while simultaneously assigning due dates. In section 3, first, we will describe the framework in designing and controlling a manufacturing system having parallel identical machines in each work center. Then the detailed description on how parallel artificial neural networks are trained is followed by how the optimal number of machines in each work center to achieve performance goals is determined. In section 4, we will present an experimental FMS and describe the production environment. Then, we will apply the framework presented in section 3 to this production setting to describe the steps of the framework.

3. A Framework for Solving the FMS Design Problem

In this paper, our goal is to design a flexible machine system to attain the performance goals supported by management. It is assumed that management will set the target performance measures' values by utilizing their experience in the production facility. Furthermore, management's goal is not to minimize the total manufacturing cost; a more expensive option in order to bring more flexibility to accommodate possible demand spikes may be chosen.

It is assumed that the production system has multiple work centers, each having parallel identical machines. The routings for parts through the FMS system are assumed to be known.

The parts are processed at each work center after a setup is performed. The setup time and processing time are assumed to be random variables. The control variables in achieving these objectives are the number of machines in each WC, the priority rule to be used in scheduling, and the due-date assignment for incoming orders.

Let P denote the number of potential dispatching rules that can be utilized on the shop floor and K be the number of due-date assignment alternatives. We provide a short description of the framework to achieve the performance requirements as follows:

Step 1: Obtain the training and test data sets for neural networks by simulating the production system for K priority rules and P different due-date assignment scenarios in parallel.

Step 2: Train and test KP parallel Back Propagation Artificial Neural Networks (BPANNs).

Step 3: Input performance measures (management goals) to the KP-trained BPANNs to obtain solution alternatives.

Step 4: Validate the results by re-simulating the manufacturing system. Obtain performance measures of the solution alternatives (which are the output of the BPANNs).

Step 5: Select the best alternative using the simulation results.

Note that the results obtained from ANNs are an approximation to the simulation results for a non-deterministic manufacturing system. As a result, one needs to validate the configurations given by ANNs by re-simulating the manufacturing system.

The proposed approach is closely related to an inverse scheduling framework. The goal is not to optimize any scheduling criteria; it is to determine a design and control scheme that minimizes the deviation from managerial goals. In the literature, it has been shown that using inverse optimization, one can find a solution with zero deviation for linear programs (Ahuja and Orlin, 2001). Koulamas (2005) shows that some of the inverse scheduling problems can be formulated as a linear programming problem. Furthermore, he characterizes some instances of inverse scheduling problems that can be solved in polynomial time where the corresponding forward problem is NP-hard. In our case, the mathematical model can be classified as a multi-objective stochastic nonlinear mixed-integer problem. As a result, we do not expect to obtain an optimal solution. We will discuss the framework presented above in more detail in sections 3.1 and 3.2.

3.1. Training Artificial Neural Networks Using Simulation

In this study, we utilize a framework where a BPANN is used along with simulation to determine a production layout and scheduling plan to achieve system performances that are as close as possible to management goals. Simulation is used in both training the BPANNs and testing the results. Simulation experiments programmed in SIMAN (1990) are designed to determine the effect of the scheduling and priority rule decisions on performance criteria. Input for the simulation module is the number of machines at each WC, the priority rule used in scheduling, information on incoming orders, and the due date of each order. Output is the performance measures. Alternative configurations are obtained by manipulating the number of machines and job priority rules at each WC. The input and output of the simulation models are used to train the BPANNs. These outputs are also used as a test set for BPANNs to determine if the BPANN proposes reliable results or not. Each BPANN has a unique momentum coefficient and learning rate, usually found by intensive experimentation. The learning rate determines the size of the node weight adjustments during training to control the rate at which BPANN's attempt to learn. Although a higher learning rate results in faster learning, it can also lead to training instabilities and divergence. The momentum factor is the coefficient that smoothes learning to achieve overall algorithm stability and support fast learning. The parallel use of *KP* BPANNs provides *KP*-alternative solutions and increases the possibility of having good-solution configurations. After the BPANNs are successfully trained and tested, they can be utilized to find the system configuration that will provide performance values that are as close as possible to the targeted ones.

3.2. Determining Optimal Number of Machines in Each Work Center to Achieve Performance Goals

After training and testing is completed, the trained *KP* BPANNs determine the number of machines in each WC, the due-date assignment rule, and a priority rule, in order to schedule incoming orders to achieve performance measures set by management goals. Since the production system/data is not deterministic, these *KP*-solution alternatives are re-simulated to obtain performance measures for each alternative. The alternative with the minimum Mean Absolute Percentage Error (MAPE) is selected, where

$$MAPE = \frac{\sum_{i=1}^{N} \left| \frac{PM_i^T - PM_i^O}{PM_i^T} \right|}{N}.$$

In this equation, PM_i^T is the target performance value of the i^{th} performance measure, PM_i^O is the obtained performance value of the i^{th} performance measure, and N is the number of performance measures.

Note that the total number of alternatives for a sample with M work centers, K possible due dates, P alternative scheduling rules, and possibly L identical machines in each WC is $(L)^M*K*P$. It would be very difficult to simulate all of these alternatives to find the best solution in a reasonable amount of CPU time. Even when all of the input (data) is deterministic, this problem is a multi-objective nonlinear mixed-integer program. For example, if completion time is one of the sub-objectives in a multi-objective programming model for a job shop, the multi-objective problem is NP-complete, since the job-shop problem with minimum completion time objective, in its general form is NP-complete.

4. Computational Experimentation

In this section, we will apply the proposed framework described in section 3 to a hypothetical flexible manufacturing system. First, we will discuss the experimental setup, which includes information about the production system and part types. Then, we will present upper management's targets for this production facility. Finally, we will describe how we have utilized parallel BPANNs to solve such a complex FMS design problem.

4.1. Experimental Setup

In this study, we consider an experimental FMS with four WCs, a loading/unloading robot, and an automated guided vehicle (AGV). The layout of the FMS is shown in Figure 1. Each WC has, at most, five identical parallel machines. Production requirements are driven by downstream assembly or machining operations. We consider five different part types having equal arrival probability and different production requirements. Parts are transferred to work stations in batches of ten. If a production requirement is more than ten components, then the demand is split into transfer batches. Each transfer batch is defined as a job. The inter-arrival rate for jobs is ten time units and exponential. The experimental FMS that we consider is similar to the one by Cakar et al. (1998).

For Figure 1 see page : 20.

In Table 1, we present the routes for each part, batch sizes, setup time (S.T.), and processing time (P.T.) of each part type at each WC. Setup time and processing time for each operation at the WCs are assumed to be normally distributed (N(x, y) with mean x and standard deviation y). In Table 1, for example, Part 4, having a batch size of 420, is routed through WC_1 and WC_3 and, thus, has two operations. The setup time for part 1 at WC_1 is normally distributed with a mean of 4 and a standard deviation of 2. Similarly the processing time at that WC is N(6, 1).

For Table 1 see page : 17

Jobs are scheduled in a work center according to one of the following priority rules: Shortest Processing Time (SPT), Longest Processing Time (LPT), or First-Come-First-Served (FCFS). Furthermore, the due date for each job is determined using the total work content method (Baker, 1984) where the due date of a job is *k* times its total processing time in the system. *k* can take values of 2, 3, 4, 5, 6, 7, 8, or 9.

The performance measures of interest are mean flow time ($\overline{F}$), mean tardiness ($\overline{T}$), maximum completion time (*Cmax*), and machine utilization rates (μ_i) at each WC_i. Note that the framework presented in section 3 is independent of the dispatching rule/scheduling heuristic that is utilized. Furthermore, this framework can be adapted to any due-date assignment rule, as long as the decisions about due dates can be discretized.

4.2. Training and Testing the Neural Network

The neural network that is being trained by the simulation experiments has 7 input nodes, 30 hidden nodes, and 4 output nodes (see Figure 2). We have 8x3=24 parallel neural networks based on three different scheduling rules (SPT, FCFS, and LPT) and 8 possible values of *k*.

For Figure 2 see page : 21

We train and test each BPANN on 100 test problems. In Table 2, we present the results on the proposed BPANNs. In this table, SPT_k, $LPT_{k,}$ and $FCFS_k$ denote the priority rules SPT, LPT, and FCFS and the due-date determination coefficient *k*. We summarize the error in both training and test phases. In Table 2, we summarize the learning rate, momentum coefficient, number of iterations, and error rates in the training set and test set for each of the 24 neural networks. Each BPANN has four output nodes. As a result, the total number of experiments for each priority rule and *k* combination is 400. The error rate is calculated using the number of output nodes, which yields an incorrect result. For example, BPANN#07 using LPT_7 (i.e., the priority rule for scheduling is LPT and the due-date determination coefficient is 7) has six output nodes, which yielded incorrect results. Thus, the error rate in the training set for LPT_7 is 1.50%. This also results in an error rate of 9% in the test set for LPT_7. However, the success rate is over 98% for all BPANNs. This is ensured by additional training until the success rate is over 98%. The average number of iterations to train the BPANNs is 810,417. The average errors in the test set and training set are 0.06% and 0.94%, respectively. After training and testing is complete, we use the BPANNs to find the configuration that will yield results that are as close as possible to the targeted ones.

For Table 2 see page : 18

4.3. Using BPANNs to Achieve Targeted Goals

In the experimental FMS considered, it is assumed that management goals are a mean flow time ($\overline{F}$) of 700 time units, a mean tardiness ($\overline{T}$) of 18, a maximum completion time (Cmax) of 16,000, and machine utilization rates ($\mu_1, .., \mu_4$) of 0.24, 0.34, 0.27, and 0.45 for WC1 through WC4, respectively.

For Figure 3 see page : 22

When these performance values are inputted to the parallel BPANNs (see Figure 3), each trained BPANN gives an alternative solution. We then validate the results obtained from the BPANNs by using the solutions given in Table 3 as input to the simulation model. After the simulation runs, we compare the resulting simulation performance measures with those for which we aimed. We sort the alternative solutions in increasing order based on the *MAPE* value and select the alternative with the minimum *MAPE* value. For the experimental setting described above, the average *MAPE* is 158.5%. Furthermore, it is observed that FCFS is the most efficient priority rule. Four out of the six best solutions utilize the FCFS priority rule in scheduling to achieve the targeted performance rules. On the average, FCFS finds a solution with *MAPE* of 102.3%. This is 181.3% with LPT and 259% with SPT priority rules. As can be seen in Table 3, the best solution alternative for this example is 2-2-3-2, FCFS, *k*=5 (i.e., two machines in WC1, two machines in WC2, three machines in WC3, and two machines in WC4; the FCFS priority rule used in scheduling; and the due date for the current batch as current time plus five times the total processing time of the job when the TWK method is utilized). Note that the objective function that is minimized has seven components, some of which might be conflicting. Furthermore, the data is not deterministic. As a result, for the set of targeted performance values aimed at by management, having FCFS as the most efficient dispatching rule on the shop floor should not be a surprise.

For Table 3 see page : 19.

5. Conclusion

In this paper, we have utilized artificial neural networks to model a highly nonlinear system, a multi-objective nonlinear optimization problem in a manufacturing system having identical parallel machines in each work center, in which the goal is to obtain a solution that is as close as possible to management goals. Management can utilize the proposed framework to effectively manage the capacity of the production plant while achieving certain performance measures. This framework will be beneficial especially in environments where a trained workforce is abundant, or workers can be trained in a short amount of time to perform the required operations. Results from the case study indicate that the artificial neural networks were able to satisfactorily capture the underlying relationship between the design and control parameters of a manufacturing system and the resulting performance targets.

Note that the framework presented in this paper not only applies to systems having excess capacity due to low demand but also to systems in which there is enough demand that makes the capacity a bottleneck. However, in this case, determination of the number of machines might become irrelevant. The framework can then be utilized to find the priority rule and due-date determination coefficient in order to have a solution as close as possible to the targeted ones.

This framework can be extended to any other input data (i.e., generated by different probability distributions), due-date assignment rule, and alternative dispatching rules (e.g., ATC, etc.). Based on the number of factors to be considered, the computational time to train-test-generate alternative solutions might increase significantly. However, this framework will still reduce the number of alternatives to be simulated significantly, and thus it will still be computationally more efficient compared to a standalone simulation approach.

Acknowledgment: This research was partially supported by the National Science Foundation under Grant No. EPS-0236913 and matching support from the State of Kansas through Kansas Technology Enterprize Corporation. We also thank to the anonymous referees for their valuable comments.

REFERENCES

Ahuja, R.K. and Orlin. J.B. (2001). Inverse Optimization. Operations Research 49, 771-783.

Baker, K. R. (1984). Sequencing Rules and Due Date Assignments in a Job Shop. Management Science, 30, 1093-1104.

Cakar, T., Cil, I., and Kurt, A. (1998). Flexible Manufacturing Design and Determination of Priority Rules by Using Artificial Neural Networks, 8th International Machine Design and Production Conference-UMTIK98, September 9-11, Ankara, Turkey.

Cakar, T., Yildirim M. B., and Barut M. (2005). A Neuro-genetic Approach to Design and Planning of a Manufacturing Cell. Journal of Intelligent Manufacturing, 16, 453 – 462.

Chan,W., and Rathmill, K. (1985). An Integrated FMS Design Procedure. Annals of Operational Research, 3, 207-237.

Chryssolouris, G. E., Lee, M., Pierce, J., and Domroese, M. (1990). Use of Neural Networks for the Design of Manufacturing Systems. Manufacturing Review, 3, 187-194.

Dagli, C. (1994). Artificial Neural Networks for Intelligence Manufacturing, Chapman and Hall.

Edghill, J. P., and Cresswell, C. (1986). FMS Control Strategy–A Survey of the Determining Characteristics. Proceedings of 4th International Conference on Flexible Manufacturing Systems, 305-315.

Feng S., Li, L., Cen, L., and Huang, J. (2003). Using MLP networks to design a production scheduling system. Computers & Operations Research, 30, 821-832

Fry and Smith (1989). FMS Implementation Procedure: A Case Study. IIE Transactions, 21(3), 228-239.

Fonseca, D. J., and Navaresse, D. (2002). Artificial Neural Networks For Job Shop Simulation. Advanced Engineering Informatics, 16, 241-246.

Gomes, M.C., Barbosa-Póvoa, A.P., and Novais, A.Q. (2005). Optimal Scheduling for Flexible Job Shop Operation. International Journal of Production Research, 43, 2323-2353

Jawahar, N., Aravindan, P., and Ponnambalam, P. G. (1998). Genetic Algorithm for Scheduling Flexible Manufacturing System. International Journal of Advanced Manufacturing Technology, 14(8), 588-607.

Kim, C. O., Min, H. P., and Yih, Y. (1998). Integration of Inductive Learning and Neural Networks for Multi-Objective FMS Scheduling. International Journal of Production Research, 36(9), 2497-2509.

Koulamas, C. (2005) Inverse scheduling with controllable job parameters, International Journal of Services and Operations Management, 1, 35 - 43

Loukil, T., Teghem, J. and Tuyttens, D. (2005) Solving multi-objective production scheduling problems using metaheuristics European Journal of Operational Research, 161, 42-61.

Low, C., Yip, Y. and Tai-Hsi Wu, T. (2006) Modelling and heuristics of FMS scheduling with multiple objectives. Computers & Operations Research, 33, 674-694

Mahmoodi, F., Mosier, C. T., and Morgan, J. R. (1999). The Effects of Scheduling Rules and Routing Flexibility on the Performance of a Random Flexible Manufacturing System. The International Journal of Flexible Manufacturing Systems, 11, 271-289.

O'Kane, J. F. (2000). A Knowledge-Based System For Reactive Scheduling Decision-Making in FMS, Journal of Intelligent Manufacturing, 11, 461-474,

Philipoom, P. R., Rees, L. P., and Wiegmann, L. (1994). Using Neural Networks to Determine Internally Set Due-Date Assignment for Shop Scheduling. Decision Science, 25(5/6), 825-851.

Sabuncuoglu, I., and Karabuk, P. (1998). A Beam Search-Based Algorithm and Evaluation of Scheduling Approaches For Flexible Manufacturing Systems. IIE Transactions, 30, 179-191.

Saygin, C., and Kilic, P. E. (1999). Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing Systems. International Journal of Advanced Manufacturing Technology, 15(4), 268-280.

SIMAN IV Reference Guide (1990), Systems Modeling Corporation, Sewicldey, PA.

Sridharan, R., and Babu, A. P. (1998). Multi-Level Scheduling Decisions in a Class of FMS Using Simulation Based Metamodels. Journal of the Operational Research Society, 49, 591-602.

Tung, L. F., Li, L, and Rakesh, N. (1999). Multiple-Objective Scheduling for the Hierarchical Control of Flexible Manufacturing System. International Journal of Flexible Manufacturing Systems, 11(4), 379-409.

Vuyosevic, R. (1994). Visual Interactive Simulation and Artificial Intelligence in Design of Flexible Manufacturing Systems. International Journal of Production Research, 32(8), 1955-1971.

Yang, J. B. (2001). GA-Based Discrete Dynamic Programming Approach for Scheduling in FMS Environments. IEEE Transactions on Systems, Man, and Cybernetics, 31(5), 824-835.

		Operation 1			Operation 2			Operation 3		
Part	Demand	WC	S.T.	P.T.	WC	S.T.	P.T.	WC	S.T.	P.T.
1	400	1	N(8,2)	N(7,3)	2	N(3,2)	N(8,1)	4	N(4,2)	N(10,2)
2	380	3	N(3,1)	N(7,1)	4	N(7,2)	N(7,3)	-	-	-
3	350	2	N(6,2)	N(9,2)	4	N(6,2)	N(6,1)	-	-	-
4	420	1	N(4,2)	N(6,1)	3	N(3,1)	N(8,1)	-	-	-
5	425	2	N(4,2)	N(7,2)	3	N(3,1)	N(9,2)	4	N(8,2)	N(7,1)

Table 1. Routes, Demand, and Processing Time for Each Part

BPANN No.	Priority Rule and *k*	Learning Rate	Momentum Coefficient	Number of Iterations	Error Rate for Training Set		Error Rate for Test Set	
					#	%	#	%
#01	LPT2	0.42	0.68	575,000	0	0	2	0.50
#02	LPT3	0.40	0.70	625,000	0	0	4	1.00
#03	LPT4	0.42	0.68	450,000	0	0	5	1.25
#04	LPT5	0.40	0.70	575,000	0	0	6	1.50
#05	LPT6	0.42	0.68	2,350,000	0	0	8	2.00
#06	LPT7	0.40	0.70	1,500,000	6	1.5	9	2.25
#07	LPT8	0.40	0.70	1,000,000	0	0	4	1.00
#08	LPT9	0.42	0.68	635,000	0	0	7	1.75
#09	SPT2	0.40	0.70	625,000	0	0	0	0.00
#10	SPT3	0.40	0.70	520,000	0	0	2	0.50
#11	SPT4	0.40	0.70	825,000	0	0	2	0.50
#12	SPT5	0.40	0.70	1,100,000	0	0	0	0.00
#13	SPT6	0.42	0.68	750,000	0	0	2	0.50
#14	SPT7	0.40	0.70	500,000	0	0	6	1.50
#15	SPT8	0.42	0.68	525,000	0	0	5	1.25
#16	SPT9	0.40	0.70	1,125,000	0	0	7	1.75
#17	FCFS2	0.42	0.68	610,000	0	0	0	0.00
#18	FCFS3	0.40	0.70	550,000	0	0	3	0.75
#19	FCFS4	0.42	0.68	625,000	0	0	2	0.50
#20	FCFS5	0.40	0.70	1,250,000	0	0	0	0.00
#21	FCFS6	0.40	0.70	910,000	0	0	1	0.25
#22	FCFS7	0.40	0.70	600,000	0	0	2	0.50
#23	FCFS8	0.43	0.67	425,000	0	0	8	2.00
#24	FCFS9	0.40	0.70	800,000	0	0	5	1.25

Table 2. Training Parameters and Test Results of BPANNs

BPANN OUTPUT				PERFORMANCE VALUES									
CONFIGURATION				**Target Values**		**700**	**18**	**16000**	**0.24**	**0.34**	**0.27**	**0.45**	
WC #1	**WC #2**	**WC #3**	**WC #4**	**PRIORITY RULE**	***k***	$\overline{F}$	$\overline{T}$	C_{max}	μ_1	μ_2	μ_3	μ_4	***MAPE***
2	2	3	2	FCFS	5	713	15	16202	0.20	0.37	0.17	0.46	12
2	2	3	2	LPT	4	732	19	16160	0.41	0.73	0.52	0.92	56
2	2	3	2	FCFS	4	713	15	16202	0.41	0.75	0.52	0.93	59
3	2	3	2	LPT	5	651	3	16000	0.28	0.74	0.53	0.92	61
2	2	4	2	FCFS	7	719	0	16248	0.40	0.74	0.38	0.93	62
2	2	4	2	FCFS	9	719	0	16248	0.40	0.74	0.38	0.93	62
2	2	3	2	SPT	4	732	19	16160	0.41	0.94	0.52	0.92	65
3	2	3	2	SPT	5	651	3	16000	0.28	0.94	0.52	0.92	69
2	2	3	2	LPT	9	732	0	16160	0.41	0.73	0.52	0.92	70
2	2	3	2	SPT	9	732	0	16160	0.41	0.73	0.52	0.92	70
1	2	3	2	FCFS	6	707	1	16200	0.80	0.73	0.52	0.93	92
1	2	3	2	FCFS	3	707	46	16200	0.80	0.73	0.52	0.93	101
1	2	2	1	LPT	2	789	192	31838	0.43	0.38	0.40	0.93	189
2	2	3	2	FCFS	2	713	203	16202	0.41	0.75	0.52	0.92	203
1	2	2	2	SPT	2	729	178	19394	0.71	0.65	0.69	0.81	205
1	3	5	2	FCFS	8	954	226	16088	0.79	0.49	0.32	0.93	227
2	3	3	1	LPT	3	658	280	31738	0.22	0.25	0.26	0.93	244
3	3	3	2	SPT	7	785	284	15939	0.28	0.50	0.53	0.93	251
3	3	3	2	LPT	7	885	284	15939	0.28	0.50	0.53	0.93	253
2	3	4	2	LPT	8	772	316	16039	0.41	0.50	0.40	0.93	277
2	3	4	2	SPT	8	672	316	16039	0.41	0.73	0.52	0.92	292
2	4	3	2	LPT	6	853	339	15838	0.43	0.37	0.53	0.93	300
2	3	3	1	SPT	3	558	480	31739	0.63	0.25	0.26	0.93	426
2	4	3	2	SPT	6	853	839	15838	0.43	0.37	0.53	0.85	694

Table 3. Alternative Solutions Given by Artificial Neural Networks Sorted with Respect to Mean Absolute Percentage Error

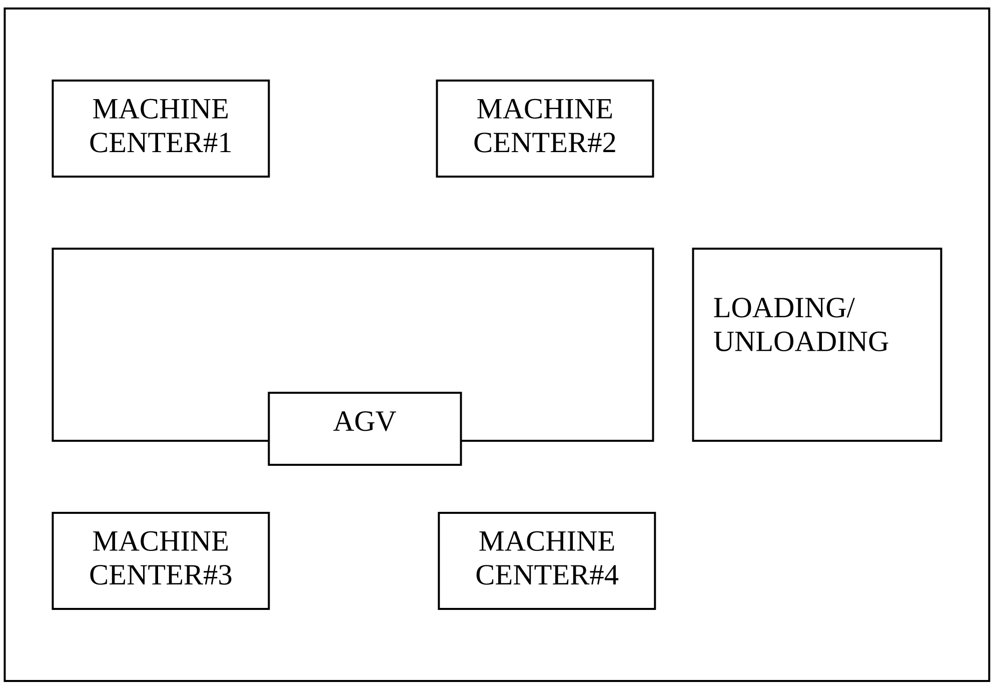
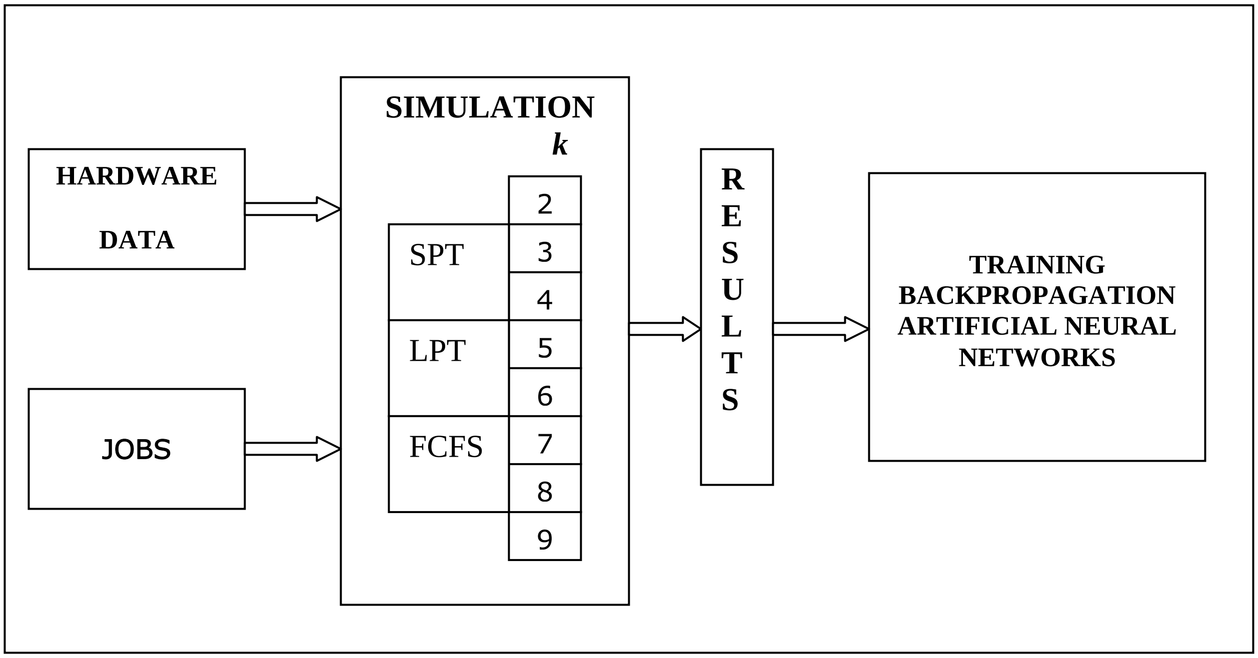
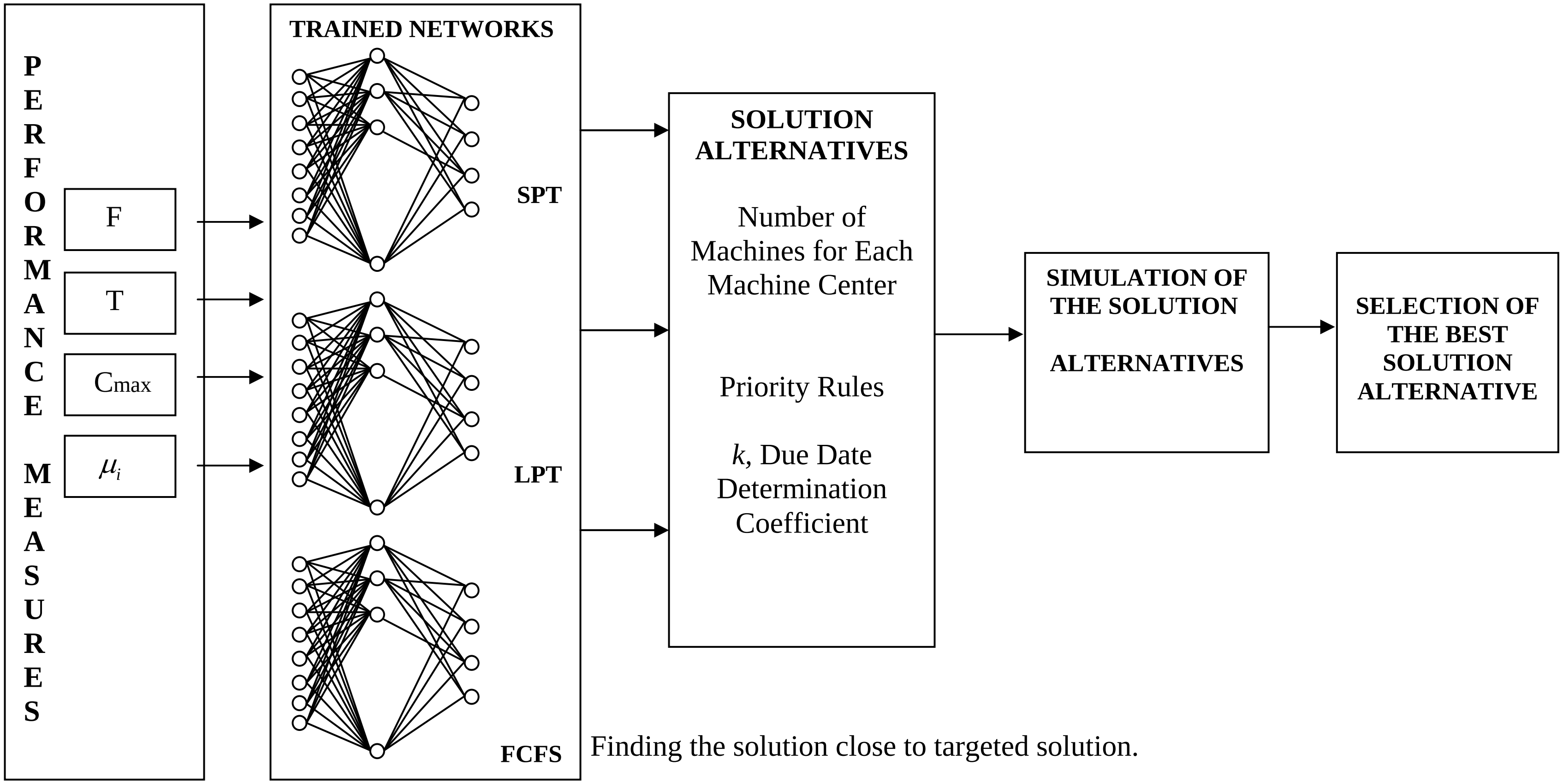


Figure 1. The experimental flexible manufacturing system.


HARDWARE
DATA
JOBS
SIMULATION
k
2
SPT
3
4
LPT
5
6
FCFS
7
8
9
RESULTS
TRAINING
BACKPROPAGATION
ARTIFICIAL NEURAL
NETWORKS



Finding the solution close to targeted solution.