Mehmet Bayram Yildirim Industrial Engineering

A Neuro-genetic Algorithm for Parallel Machine Scheduling to Determine the Number of Machines and Priority Scheduling Rules.

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

Mehmet Barut
Wichita State University, Mehmet.Barut@wichita.edu

Tarik Cakar
Sakarya University, tcakar@sakarya.edu.tr

A Neuro-genetic Algorithm for Parallel Machine Scheduling to Determine the Number of Machines and Priority Scheduling Rules.

Mehmet Bayram Yildirim
Industrial and Manufacturing Engineering Department
Wichita State University
Wichita, KS, 67260-0035, USA
e-mail: bayram.yildirim@wichita.edu

Mehmet Barut
FREDS Department
Wichita State University
Wichita, KS, 67260-0035, USA
e-mail: Mehmet.Barut@wichita.edu

Tarik Cakar
Industrial Engineering Department
Sakarya University
54040 Esentepe Adapazari, Turkey
e-mail: tcakar@sakarya.edu.tr

Abstract: In this paper, we propose a neuro-genetic artificial neural network framework to achieve certain targeted productivity measures/ performance values in a flow shop with parallel processors (resources) at each stage. The performance measures that we consider are flow time, number of tardy jobs, total tardiness and machine utilizations. In order to achieve these goals, the management has to make decisions on the availability of resources, in our setting, the number of identical machines in each work station and the dispatching rule to be utilized in the shop floor to achieve performance values as close as to the targeted ones.

Introduction:

In the current competitive global market, demand for a certain product might change based on several factors. For example, seasonality of demand, a new competitor or lower than expected sales due to an economic crisis in the market might force an under utilization of the capacity. In this case, the management might consider not to run all of the available resources such as machines and workers. However, a goal in this case might be keeping the level of certain performance measures at a certain level to accommodate a certain level of unexpected demand. In this study, we assume that the management has decided on the targeted values of the performance measures. We optimize the number of machines (resources) and the due dates of the incoming orders in order to achieve performance values as close to the targeted values as possible.

In this study the production environment that we consider is a flexible flow shop, that has workcenters in which there are identical machines (Kurz and Askin, 2003). Having identical machines in each work center helps to increase the overall capacity. Processing in this production environment occur in a linear fashion. We assume that there are four stages and at each stage, there are at most five identical machines. There is no machine breakdowns, i.e., all machines are continuously available. Each machine can process only one job at a time. Once an operation is started on a machine, it must be completed without preemption. There are infinite buffers between all stages, so there is no blocking of machines. Each stage has at most 5 parallel identical machines. The processing time for every job on every stage that it visits is known in advance and is constant. Furthermore, each job can be processed on only one machine at a time. However, the ready times for each job is not known. The interarrival times are exponential.

In this setting, the management wants to achieve certain goals (targeted performance values) such as flow time, number of tardy jobs, total tardiness and machine utilizations. In order to achieve these goals, on one hand, the management has to make decisions on the availability of resources, in our setting, the number of identical machines in each work stations to achieve performance values as close as to the targeted ones. On the other hand, the management also has to decide which dispatching rule must be utilized on the shop floor. The candidate priority scheduling/dispatching rules considered are Earliest Due Date (EDD), Shortest Processing Time (SPT), First Come First Serve (FCFS) and critical ratio (CR).

It is well-known that most of the scheduling problems are very hard problems, most of them being NP-hard. Thus, when complete enumeration or mathematical programming models along with commercial solvers such as CPLEX or OSL are used to find an optimal solution, usually a considerable amount of computing power is required. However, for large sized problems, neither of these approaches is found to be efficient. As a result, usually heuristics are used to find good solutions. For example, one can utilize dispatching (priority) rules, which usually takes a minimal amount of computation time. Several priority rules such as shortest processing time and longest processing time have been proposed for scheduling in Manufacturing Systems (MS). Priority rules are practical and easy to implement and they can affect the design of the manufacturing systems significantly (Askin and Standridge, 1993). However, no single dispatching rule clearly dominates for all criteria. Especially, in flexible flow shop environment there is not a significant difference between various dispatching rules at all (Ding and Kittichatphayak, 1994, Kurz and Askin, 2003 and Sriskandarajah and Sethi, 1989).
.
Utilizing metaheuristics such as Genetic Algorithms (GAs) and Artificial Neural Networks (ANNs) provide encouraging results, i.e., near optimum solutions, in a reasonable amount of time (Chryssolouris et. al. 1990, Vujosevic et. al., 1994, Çakar et al. 2002). Genetic algorithms has several applications to scheduling theory: for example, GA applications for job-shop scheduling have been studied by Nakano and Yamada (1993), Lee et al. (1997) and Candido et al. (1998).

It has been widely recognized that artificial intelligence techniques have a wide variety of applications in scheduling. For a complete survey of neural network applications in manufacturing, we refer the readers to Zhang and Huang (1995). Vaithyanathan and Ignizio (1992) propose a neural network approach for solving certain types of large-scale, resource-constrained scheduling problems. Rabelo and Alptekin (1989) provide lower tardiness values than those with six dispatching heuristics using ANNs. Cedimoglu (1993) used a Back Propagation Artificial Neural Network (BPANN) which outperform several dispatching rules. Kim et. al. (1995) combine the BPANN with the apparent tardiness cost rule.

Hybrid systems such as neural networks and optimization, or neural networks and simulation, or neural networks and genetic algorithms, are widely utilized to solve job-shop scheduling problems. For example, Sim et. al. (1994) propose a hybrid neural network and expert system simulation model to solve the dynamic job-shop problem and overcome this problem where sixteen BPANNs are embedded in the expert system. Jain and Meeran (1998) use a modified BPANN structure to solve job-shop scheduling problem using input-output mappings in which the model is trained on the optimum data rather than relying on the guesses of experts.

In this study, we propose a hierarchic neuro-genetic approach where a set of good solutions is obtained by the parallel BPANNs, and then these solutions are improved by utilizing a GA. We train BPANNs in conjunction with simulation to ensure good near optimal solutions for the problem being considered. The proposed BPANN plays the role of an inverse simulator function: the system designer sets the objectives or targeted values of performance measures. Based on these objectives the neural network outputs a suitable MS design, where the overall system performance solely determined by the proximity of the system's actual performance. Note that it is important to express the desired performance in terms of some combination of measures, which cannot be simply optimized (Udo et. al., 1992). The problem we consider in this paper is closely related to the work by Chryssolouris et. al. (1990) who estimate the number of machines in a work station using BPANNs for a simpler configuration.

2. The Neuro-genetic framework to determine the machine numbers

A short description of the neuro-genetic framework to achieve the management goals is described as follows:

1: Use simulation to obtain training set and test set
2: Train and test the parallel BPANNs
3: Input the performance measures to the trained BPANN
4: Obtain the solution alternatives
5: Simulate and obtain performance measures of the solution alternatives
6: Use genetic algorithms to improve the solution alternatives
7: Find the best solution alternative. Validate the results using simulation.

Now we will illustrate this framework on a sample flexible flow shop setting with stochastic arrivals and deterministic processing times and duedates.

3. The Flexible Flow Shop Environment

We consider a flexible flow shop environment containing four workstations, each having at most five machines. There are six different parts arriving to the shop floor. The inter-arrival times of parts follows an exponential distribution. The number of parts to be produced, interarrival times of the parts and processing times at each workcenter are given in Table 1. The total number of parts to be processed is assumed to be 200 parts. Due dates are calculated based on the Total Work Content method. The duedate for each part is five times of the total processing time. Parts in this job shop follow the work center order. For example, parts for job 1 and job 5 are processed at all workcenters (WC-1 through WC-4).

Table 1. Parts to be Processed

Jobs	J1	J2	J3	J4	J5	J6
Number of Parts	17	50	30	17	43	43
Inter-Arrival Times	20	10	15	18	10	12
Processing Times						
WC-1	25	30	20	16	30	28
WC-2	10	20	25	10	25	23
WC-3	20	10	5	15	20	40
WC-4	5	12	9	25	15	10

4. Finding a Good Solution Using Neural Networks

In this study, the shop is simulated under different dispatching rules, EDD, SPT, FCFS and CR. For each priority rule, the flow shop has been simulated on 50 randomly generated problems using ARENA/SIMAN simulation language. The simulation results has been used to train four Back Propagation Artificial Neural Networks (BPANNs) in parallel. The reason for training four different BPANNs is due to the fact that each dispatching rule has its own characteristics and conflicting objectives with other rule(s). The inputs to the BPANN are the mean flow time, mean tardiness, maximum completion time, machine utilization rate in each workstation and percentage of tardy parts. The outputs of the system are the number of machines in each workstation, and the best dispatching rule to be utilized for scheduling decisions.

Table 2. Training Parameters and test results for BPANNs

Trained Networks	**BPANN 1**	**BPANN 2**	**BPANN 3**	**BPANN 4**
Priority rule	SPT	EDD	CR	FCFS
Learning rate	0.4	0.4	0.4	0.4
Momentum coefficient	0.7	0.7	0.7	0.7
Activation function	Sigmoidal	sigmoidal	sigmoidal	sigmoidal
Iteration number	2.9 M	3 M	1.72 M	2.4 M
Sample size in training set	50	50	50	50
Learned sample in training set	50	50	50	50
Sample size in testing data set	100	100	100	100
Network achievement rate	%92	%99	%93	%97

Table 2 summarizes the training parameters and the test results across the networks. The best results are achieved under a learning rate of 0.4 and a momentum rate of 0.7 for SPT, EDD, CR and FCFS networks. When tested with the training set, the success rate for the BPANNs is 100%. To test the validity of these results, each BPANN is also tested using 100 new examples (data sets): the BPANN results are compared with the simulation results and the success of each network is obtained accordingly. Having run the BPANN for a maximum of 3 million iterations, the success rate for each BPANN is 92, 99, 93 and 97%, respectively. After obtaining the results from BPANNs, these solutions are fed to a genetic algorithm as an initial population.

5. Post Processing using Genetic Algorithms

Although the BPANNs are providing solutions very close to optimal for the test sets, there is a slight chance that they may provide a significant undesirable solution. To correct this, a genetic search procedure is appended to the solution system. This procedure allows the application of a specific BPANN under different priority rules to obtain a better solution.

The chromosomes for the genetic algorithms are defined by the number of machines in each work center and the priority rule that has been utilized in scheduling operations (i.e., they have five genes). The initial population is generated by the BPANNs and random ones. Using the genetic operators such as crossover and mutation, new offsprings are generated and those ones providing best solution, i.e., the minimum total absolute percentage deviation from the targeted performance measures) are included in the population pool. Preliminary analysis shows that 10 chromosomes are sufficient to obtain better solutions. Initially, four chromosomes are from the BPANNs and six more chromosomes are randomly generated. This genetic algorithm is run until the maximum number of allowable generations with a crossover and mutation rates of 1.00 and 0.04 respectively. The preliminary results shows that the trade-off between training a BPANN and using genetic search from a pure random initial population is significant with respect of time spent on training a BPANN and time spent in genetic search. We would like to note that giving the same data used for testing the BPANNs, neuro-genetic solution system has significantly improved the solution with an error close to zero under all networks trained for SPT, EDD, CR, and FCFS priority rules. Thus, the success rates went up to approximately to 100% for all problems in the testing data set.

6. An Illustrative Example

To illustrate the procedure above, suppose that the management goals for the flow shop described in section 2 are

Mean Flow Time (F)		: 1132
Mean Tardiness (T)		: 1410
Maximum Complete Time (C_{max})		: 3845
Machine utilization rate of work center-1	(MU-1)	: 0.99
Machine utilization rate of work center-2	(MU-2)	: 0.22
Machine utilization rate of work center-3	(MU-3)	: 0.24
Machine utilization rate of work center-4	(MU-4)	: 0.08
Percentage of Tardy Jobs (N_T)		: 0.57

Based on these inputs, the trained BPANNs produce the alternative solutions summarized in Table 3, along with percentage error analysis for each performance measure.

Table 3. Solutions obtained from each BPANN

Performance Measures & Target Values		BPANN(SPT)		BPANN (EDD)		BPANN (CR)		BPANN (FCFS)	
		1 5 2 4	% e	1 3 3 3	% e	1 4 2 3	% e	1 3 3 5	% e
$\overline{F}$	1132	1199	5.9	1217	7.5	1365	20.5	1268	12
$\overline{T}$	1410	1411	0.07	1383	1.9	1420	0.7	1406	0.28
C_{max}	3845	3845	0	3875	0.78	3840	0.01	3845	0
MU1	0.99	0.990	0	0.980	1	0.990	0	0.990	0
MU2	0.22	0.178	19.0	0.293	33.20	0.222	0	0.297	35.0
MU3	0.24	0.480	100	0.316	31.60	0.48	0	0.320	33.3
MU4	0.08	0.080	0	0.137	71.25	0.137	71.25	0.082	2.5
N_T	0.57	0.750	31.5	0.185	67.5	0.880	54.3	0.83	45.6
Mean Error (%)			19.55		26.84		18.35		16.08

As can be seen, after 3 million iterations, the total mean absolute deviation for each BPANN is between 16.08% and 26.84%. The best configuration utilizes the CR priority rule with 1, 3, 3, and 5 machines in WC#1 through WC#4, respectively. After post-processing with the genetic algorithm, the mean absolute deviation for this problem improves to 2.67%. This is achieved by utilizing the SPT as the dispatching rule and increasing the number of machines in WC#2 and WC#3 by one machine.

Table 4. Final Solution from Neuro-Genetic System

SHOP FLOOR PHYSICAL CONFIGURATION				
WC#1	**WC#2**	**WC#3**	**WC#4**	**Dispatching Rule**

1	4	4	5	SPT

Performance Measures	Target Values	Achieved Values	Error Rates
$\overline{F}$	1132	1128	0.4%
$\overline{T}$	1410	1380	2.1%
C_{max}	3845	3900	1.4%
MU1	0.99	0.99	0.0%
MU2	0.22	0.21	4.5%
MU3	0.24	0.25	4.2%
MU4	0.08	0.08	0.0%
N_T	0.57	0.62	8.8%

Conclusions

We have utilized a neuro-genetic approach to determine the number of machines in each worker center and the scheduling priority rule to minimize the mean absolute deviation from the targeted system performance measures. The proposed framework uses simulation to train the neural networks and validate the results, neural networks to model a highly nonlinear system and genetic algorithm to improve the results given by the neural networks. This framework can be utilized successfully not only when demand is lower than the capacity of the production system but also as a tool for multi-objective optimization of scheduling systems.

References

ASKIN, R. G. and STANDRIDGE, C.R., 1993, Modelling and Analysis of Manufacturing Systems, (John Willey & Sons, New York)

ÇAKAR, T. and TURKER, A.K., ÇAGIL, 1996, The Use of Neural Networks in Flexible Manufacturing System Design, in Proceedings of 7th International Machine Design and Production Conference, pp. 55-64, METU, Ankara, Turkey.

CANDIDO, M.A.B., KHATORE, S.K., and BARCIA, R.M., 1998, A Genetic Algorithm Based Procedure for More Realistic Job Shop Scheduling Problems, International Journal of Production Research, Vol.36, No.12, pp. 3437-3457.

CEDIMOGLU, I. H., 1993, Neural Network in Shop Floor Scheduling, Ph.D. Thesis, School of Industrial and Manufacturing Science, Cranfield University, UK.

CHRYSSOLOURIS, G., LEE, M., PIERCE, J., and DOMROESE, M., 1990, Use of Neural Networks for the Design of Manufacturing Systems, Manufacturing Review, 3, 187-194

DING, F.-Y., KITTICHATPHAYAK, D., 1994. Heuristics for scheduling .exible .ow lines. Computers and Industrial Engineering 26 (1), pp. 27- 34.

KIM, S.Y., LEE, Y.H., and AGNIORTI, D., 1995, A Hybrid Approach for Sequencing Jobs Using Heuristic Rules and Neural Networks, Production Planning and Control, 6, 445-454.

KURZ, M., E. AND ASKIN, R. G., Comparing scheduling rules for .exible .ow lines, Int. J. Production Economics, 85, pp. 371–388

LEE, C.-Y., PIRAMITHU, S., and TSAI, Y.-K., 1997, Job Shop Scheduling with a Genetic Algorithm and Machine Learning, International Journal of Production Research, Vol.35, No.4, pp.1171-1191.

NAKANO, R., and YAMADA, T., 1993, Conventional Genetic Algorithms for Job Shop Problems. in Proceedings of The Fourth International Conference on Genetic Algorithms, New York: Morgan Kaufmann, pp. 477-579.

RABELO, L.C., and ALPTEKIN, S., 1989, Using Hybrid Neural Networks/Expert Systems for Intelligent Scheduling in Flexible Manufacturing Systems, Proceedings of The IJCNN International Joint Conference on Neural Networks, Washington, June 18-22, 2, pp.608.

SIM, S.K., YEO, K.T., and LEE, W.H., 1994, An Expert Neural Network System, for Dynamic job-shop Scheduling, International Journal of Production Research, 32, 1759-1773.

SRISKANDARAJAH, C., SETHI, S.P., 1989. Scheduling algorithms for .flexible flowshops: Worst and average case performance. European Journal of Operational Research 43, 143–160.

UDO, G.J., 1992, Neural Networks Applications in Manufacturing Processes, Computers and Industrial Engineering, 1, 97-100.

VAITHYANATHAN, S., and IGNIZIO, J.P., 1992, A Stochastic Neural Network for Resource Constrained Scheduling, Computers and Operations Research, 19, 241-254.

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

ZHANG, H.C., and HUANG, S.H., 1995, Applications of Neural Networks in Manufacturing: a State of The Art Survey, International Journal of Production Research, 33, 705-728.