Wichita State University Libraries
SOAR: Shocker Open Access Repository

Mehmet Bayram Yildirim

Industrial Engineering

A neuro-genetic approach to design and planning of a manufacturing cell

Tarik Cakar
Wichita State University

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

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

Recommended citation

Cakar, Tarik., Yildirim, Mehmet Bayram. and Mehmet Barut. 2005. A neuro-genetic approach to design and planning of a manufacturing cell. *Journal of Intelligent Manufacturing*, 16, 453–462

This paper is posted in Shocker Open Access Repository

<http://soar.wichita.edu/dspace/handle/10057/3447>

A neuro-genetic approach to design and planning of a manufacturing cell

TARIK CAKAR and MEHMET BAYRAM YILDIRIM*

Industrial and Manufacturing Engineering Department, Wichita State University, Wichita, KS 67220-0035, USA

E-mail: bayram.yildirim@wichita.edu

MEHMET BARUT

Finance Real Estate and Decision Sciences Department, Wichita State University, Wichita, KS 67220-0035, USA

Received October 2003 and accepted November 2004

In this paper, we propose a neuro-genetic decision support system coupled with simulation to design a job shop manufacturing system by achieving predetermined values of targeted performance measures such as flow time, number of tardy jobs, total tardiness and machine utilization at each work center. When a manufacturing system is designed, the management has to make decisions on the availability of resources or capacity, 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 desired. Four different priority rules are used as Earliest due date (EDD), Shortest Processing Time (SPT), Critical ratio (CR) and First Come First Serve (FCFS). In reaching the final decision, design alternatives obtained from the proposed system are evaluated in terms of performance measures. An illustrative example is provided to explain the procedure.

Keywords: Resource allocation, dispatching rule, neuro-genetic, hybrid scheduling

1. Introduction and motivation

In the current competitive global market, several external factors might effect the management of demand and capacity. Examples of seasonality in demand, a new competitor or shortage in sales due to an economic crisis in the market might force an under utilization of the capacity available. The recent efforts providing flexibility to the end users of products and/or services further forces managers to an efficient usage of resources. In such cases, management might consider a strategy of not to run all the resources made avail-

able at the strategic and aggregate planning. Much of the attention is usually given to the strategic resources such as machinery since the flexibility revealed from these resources is more critical compared to all other production related resources such as work force. Although for a given resource and work load structure it is straightforward to compute the performance of the system, to employ a what-if analysis requires runs of simulation. For managers whose goal is to maintain certain performance measures at predetermined levels to accommodate a certain level of unexpected demand, time consuming what-if approach may not be feasible at operational level, where managers would rather be more interested in short term decisions

*Author for correspondence

of determining the actual minimal amount of the available resources to be used in order to meet performance measure targets.

A legitimate forcing driver for managers to make above-mentioned decisions could be the need for simultaneous achievements of critical and conflicting objectives of delivery speed, reliability, higher customer satisfaction, and minimum cost. Recent trends, especially web-based information infrastructure, conceivably, may soon render most manufacturing companies to behave like make to order companies, which operates under job shop. In light of these practical managerial concerns, we define the need for a decision system to support managers by providing solutions for operational scheduling problems in a job shop environment, which are recognized as highly differential from shop to shop and NP-hard (Lee and Dagli, 1997).

In this study, a job shop, where a set of orders with known routings is ready for processing, is considered. The management wants to achieve the targeted values of certain performance measures listed as flow time, number of tardy jobs, total tardiness and machine utilizations at the work centres. 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 station to achieve performance values as close as to the targeted ones. On the other hand, the management also has to decide, which dispatching rule, among Earliest Due Date (EDD), Shortest Processing Time (SPT), First Come First Serve (FCFS), and Critical Ratio (CR), must be utilized on the shop floor. To solve this problem, a methodology that uses neural networks and genetic algorithms to determine the level of resources and the dispatching rule on the shop floor is proposed. Compared to a standalone simulation approach, this methodology finds a solution (which is as close to the planned performance values as possible) considerably faster, since it does not require a trial and error or complete enumeration approach.

2. Literature review

The literature demonstrates that the design and control of scheduling problem in the job shop

can be solved with several methods including complete enumeration (branch and bound), mathematical modelling utilizing commercial optimization solvers, heuristics, dispatching rules, simulation and meta-heuristics involving Artificial Intelligence (AI) techniques. It is a common evident that a considerable amount of computing power is required when complete enumeration or mathematical programming models along with any commercial solvers such as CPLEX or OSL are used to find the optimal solution. For considerable large sized problems, where finding an optimal solution is not possible in an efficient manner, heuristics such as dispatching rules are used to find good solutions under practical computing power, and can significantly affect the design of the manufacturing systems (Askin and Standridge, 1993). The other widely recognized techniques, Expert Systems (ES) and Neural Network (NN), which require AI, have a wide variety of applications in scheduling problems. Rabelo and Alptekin (1989), using an integrated system consisted of both an ES and NNs, report better performance values (lower tardiness values) than those with six dispatching heuristics. The performance of NNs has been experimentally proven to exceed the expectations compared to other heuristics, as promising results for large scaled resource constrained scheduling problems from Vaithyanathan and Ignizio (1992)'s Hopfield NN approach. Cedimoglu (1993) also reports encouraging and supporting results that using a Back Propagation Neural Network (BPNN) outperforms several dispatching rules. Similarly, Pierreval (1993), by using more complex BPNN tested under different levels of operational factors, reports up to 94% of success rate. Kim *et al.* (1995) combine the BPNN with the apparent tardiness cost rule, and report a significant improvement of 8% on average. For a complete survey of neural network scheduling applications in manufacturing, we refer the readers to Sabuncuoglu (1998) and Zhang and Huang (1995). It is noteworthy that Philipoom and Rees (1997) conclude that neural networks far outperforms the traditional approaches for job shop scheduling and are statistically better. As stated by Sabuncuoglu and Gurgun (1996), although the studies about neural networks in the literature does not reveal a strong solid core competitiveness compared to

conventional operations research techniques, both in terms of quality and speed, they have inherent parallel processing capabilities that may offer advantages.

The AI techniques can enhance the capability of simulation processes for searching better solutions. The integration of simulation and AI techniques would enhance the speed capability of the simulation process. The use of simulation on its own provides a wide range of solutions. However, this is at a cost of considerable amount of valuable computation time. In addition, since this is a long trial and error process, there is no guarantee of near optimal solutions.

It has been frequently reported that metaheuristics such as genetic algorithms (GA) and NNs yield encouraging results, i.e., near optimum solution in a reasonable amount of time (Chrysosouris *et al.* 1990; Vujosevic, 1994; Cakar *et al.* 1996). The GA applications for job shop scheduling have been studied by Nakano and Yamada (1993), Lee *et al.* (1997), and Candido *et al.* (1998). One way to further improve the solution and the speed is to use hybrid systems consisting of at least one meta-heuristic, e.g., NN with optimization, or NN with GA. Specifically, Rabello *et al.* (1993) append a genetic solution to improve the results of Rabello and Alptekin (1989) and report a significant increase in success rate compared to that of 90% in Rabello and Alptekin (1989). Sim *et al.* (1994) propose a hybrid NN and ES, with a simulation component, to solve the dynamic job-shop problem by designing 16 BPNNs into the expert system. The results reveal better solutions compared to traditional priority rules.

Holter *et al.* (1995) present an intelligent single machine controller, which integrates planning and scheduling with monitoring and execution. The NNs trained with dispatching rules are used to predict the behaviour of different sequencing policies in the system, and then find initial good solutions which are then fed to a GA to find near optimal solutions. Dagli and Sittisathanchai (1995) utilize a hybrid genetic neuro-scheduler for generating job shop schedules based on a non-linear multi-objective function. The GA is used as a search technique for finding an optimal schedule from a pool of gene strings, which represents feasible solutions. Different objective functions are then evaluated by a NN mapping a complex set

of scheduling criteria (mean flow time, lead time, mean lateness, number of tardy jobs, and mean tardiness) with respect to appropriate values provided by experienced schedulers. Lee and Dagli (1997) use parallel genetic-neuro schedulers for job shop scheduling problems and show that this approach has a potential to find the global optimal solution more effectively. Yu and Liang (2001) consider a hybrid NN and GA approach to solve a job shop scheduling problem where the objective is to minimize the make-span and the number of tardy jobs simultaneously. The NNs are trained to detect whether the processing constraints are satisfied or not. The GA are utilized to find the best solution using a gradient search algorithm among the solutions provided by the NNs. Caskey (2001) utilizes methods like discrete event simulation, genetic search, and NNs to find good operating strategies to adapt rapidly changing conditions.

All of above-mentioned studies show that NNs are becoming an important approach in solving a wide range of combinatorial scheduling problems. One common focus of all studies in this area is that they assume a forward decision making process, working under the assumption of defined manufacturing design structure. The significance of our study is that we are shifting the focus toward backward decision making process assuming management has pre-determined the desired values of the performance measurements. This approach takes these values as inputs and determines the best design structure.

The problem we consider in this paper is closely related to the work by Chrysosouris *et al.* (1990) who estimate the number of machines in a work station using BPNNs for a simpler configuration. They conclude and suggest that using a single BPNN is inadequate to obtain the desired solution no matter how good the parameters are chosen, which supports parallel processing of our proposed approach. Thus, we consider parallel BPNNs in conjunction with simulation to ensure good solutions for the problem on hand. Next, we feed these solutions to a GA as an initial population to obtain better solutions. This two phase approach provides near optimal solutions for a very complex non-deterministic non-linear multi-objective scheduling problem in a reasonable amount of time. Note that a similar two-

Fig. 1. The proposed decision support system structure.

phase approach has been utilized by Yu and Liang (2001).

In Section 3 we provide details of our proposed hybrid meta-heuristic model and provide an illustrative example. We conclude with a summary of the discussion of the findings.

3. Proposed neuro-genetic system

The proposed neuro-genetic system is developed as a user-friendly decision support software in Pascal integrated with the simulation language SIMAN. The end user enters the values of the targeted performance measures and system parameters via the user interface. As shown in Fig. 1, the system provides the best solution to the end user via the user interface without any involvement through the procedure.

The BPNN module of the system, consisting of parallel BPNNs trained in conjunction with simulation, obtains a set of good solutions. Functionally, this module plays a role of an inverse simulator function: the system designer sets the objectives or targeted values of performance measures; and then based on these objectives, the NN outputs a suitable MS design, where the overall system performance is solely determined by the proximity of the system's actual performance. Then, the set of solutions obtained from BPNN module are fed into the GA module to further improve the results. Based on the results from GA module, the decision support system recommends the level of resources and the dispatching rule on the shop floor. Compared to a standalone simulation approach, this methodology finds a solution (which is as close to the targeted per-

Table 1. Time and flow matrix for products to be processed

	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>
<i>Product ID</i>						
Demand for products	17	50	30	17	43	43
Inter-arrival times	20	10	15	18	10	12
<i>Processing Times</i>						
WC-1	25	30	20	—	30	—
WC-2	10	20	25	—	25	—
WC-3	25	10	—	15	20	40
WC-4	5	—	—	25	15	10

formance values as possible) considerably faster, since it does not require a complete enumeration. Note that, as stated by Udo (1992), expressing the desired performance in terms of some combination of measures cannot be simply optimized.

3.1. Operating environment

A job shop environment containing four workstations, each having at most five machines is considered. There are six different products arriving to the shop floor. The inter-arrival times for product orders follow an exponential distribution. Demand is deterministic. The demand for each product, interarrival times of the orders and processing times at each work center are given in Table 1. The total demand for all products is assumed to be 200 units. Due dates are calculated based on the Total Work Content method: the due date for each order is five times of the total processing time. Realize that the processing route is not fixed across the products. Thus, such

operation is considered as a job shop (Pinedo, 2002). For example, orders for products 1 and 5 are processed at all work centers (WC1 – WC4). Similarly, orders for products 4 and 6 are processed at WC3 and then at WC4.

3.2. BPNN module

Back Propagation algorithm, as an accurate mapping of the system, is used as a supervised learning algorithm to train NNs in order to obtain good results. In this study, the shop is simulated under different dispatching rules: EDD, SPT, FCFS, and CR. Four different sets of 50 example problems are used in training the parallel BPNNs (one for each dispatching rule). The reason for training four different BPNNs is due to the fact that each dispatching rule has its own characteristics and a conflicting objective with other rule(s). For example, while SPT rule reduces the mean flow time, EDD rule reduces the mean tardiness. Note that, the purpose of using these rules is different from that of classical purpose of minimization of the performance they are measuring. Instead, they are used for the purpose of finding the equilibrium states of the performance measures.

The inputs to the BPNNs are the mean flow time, mean tardiness, maximum completion time, machine utilization rate in each work center, and percentage of tardy parts. The outputs of the system are the number of machines in each work center, and the best dispatching rule to be utilized for scheduling decisions. The hidden layer contains 30 neurons. Each BPNN is trained and tested with different datasets to ensure that the risk of making an error while obtaining a set of near optimal solutions is minimized. Table 2 summarizes the training parameters and the test results across the networks. While the best results are achieved under a learning rate of 0.4 and a momentum rate of 0.7 for SPT, EDD, and FCFS networks, these rates are 0.44 and 0.66 for CR network, respectively. Each BPNN is then tested using 125 new examples (datasets), and the success of each network is obtained accordingly.

Each of the expected performance measures is observed under each BPNN. As seen in Table 2, the success rates of BPNNs for SPT, EDD, CR, and FCFS are very high. When tested with the

training set, it is 100%. When tested with 125 different new datasets, the success rate for each BPNN is 0.968, 0.904, 0.976, and 0.980, respectively.

After obtaining the results from BPNNs, these solutions are fed to a GA as an initial population. This process is described in details in Section 3.3, and in Fig. 2.

3.3. Post processing with genetic module

Although the BPNNs are providing good solutions for the test sets, in the expense of some computational time, a GA is appended to further improve the solution against undesirable outcomes.

The GA use the analogy of genetic structure of a chromosome to obtain near optimal solutions to complex real life problems by modelling them into a gene vector and by imitating the process of natural selection via an evaluation function. In order to perform a genetic search, shop's design configuration and dispatching rules need to be modelled into chromosomes. The chromosome will consist of as many genes as the number of work stations plus one gene for the dispatching rule. In our study, for example, a chromosome consists of five genes: the first four genes represent the number of machines in each work-station, respectively, and the last gene represents the dispatching rule used. The evaluation function defined for each chromosome play the same role of that in natural selection. The reproduction role is defined and moderated by the following two operators: crossover and mutation. Pair-wise crossover is used to obtain better offsprings (solutions) from different chromosomes in the population pool. Once the index into the string of genes is identified, the data beyond that point is swapped between two chromosomes to reproduce the offsprings. In the following example assume that the random pointer suggests the crossover should start from the third gene. As a result of the cross-over operation on the current chromosomes, the following offsprings are obtained:

On the other hand, mutation operator randomly determines which genes in a randomly selected set of chromosomes are going to be changed in hope of obtaining a better solution. Thus, mutation rate determines the proportion of the genes that are going to alter. Hence, the population size,

Table 2. Training parameters and test results of BPNNs

<i>Trained networks</i>	<i>BPNN #1</i>	<i>BPNN #2</i>	<i>BPNN #3</i>	<i>BPNN #4</i>
Priority rules	SPT	EDD	CR	FCFS
Learning rate	0.40	0.40	0.44	0.40
Momentum coefficient	0.70	0.70	0.66	0.70
Activation function	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Iteration number	4.000.000	4.200.000	5.000.000	4.500.000
Sample size in training data set	50	50	50	50
Learned sample in training set	50	50	50	50
Sample size in testing data set	125	125	125	125
Network achievement rate(%)	96.8	90.4	97.6	98.0

Fig. 2. The neuro-genetic framework.

cross-over rate, mutation rate, and the number of generations are the control parameters.

In GAs, the procedure or the process is independent of any problem. It is, however, the fitness function that correlates the algorithm with the problem, making it problem specific. The powerfulness of the fitness functions in randomness of the variation, reproduction, and selection determines the strength of the link between the genetic process and the problem at hand. The fitness of the offsprings is practically calculated by comparison with a predetermined optimum (Lenski *et al.*, 2003). Since the objective of our problem is to determine the solution that will yield to the best design configuration for a given set of performance target values, the fitness function is defined as the mean absolute percentage deviations from the predetermined targets of performance measures and formulated as $\sum_{i=1}^n e_i/n$, where e_i is the percentage error of the i th performance measure. In our problem n is equal to eight: mean flow time (F), mean tardiness (T), completion time (C), machine utilization rates (Mu) in each workstation, and the percentage of tardy jobs (N_T). The percentage errors for each performance measure is calculated as follows:

$$e_i = \frac{|PM_O^i - PM_T^i|}{PM_T^i}, \quad i = F, T, C, Mu, N_T$$

where PM_O is the observed value, and PM_T is the target value of the corresponding performance measure.

Once the parameters are set, and the results are obtained from each BPNN, the initial population set is created. The initial population consists of the solutions from each BPNN, and randomly generated solutions. The genetic algorithm aims to improve the current solution by performing crossover and mutation operations on the population set (see Fig. 2). When a new offspring is generated using genetic operators, the MS design corresponding to this offspring is simulated to collect the values of the performance measures. These values are used in evaluation of the fitness function. The GA is run until the preset maximum number of generations is reached. The list of the chromosomes at that stage is sorted in descending order with respect to the fitness function, and the best solution is selected.

Figure 2 shows the solution procedure of the proposed neuro-genetic system. This system has the potential to eventually improve the success rate to a 100%. Genetic search results in new chromosomes leading to zero or the smallest deviation possible from the target values of the performance measures and providing machine configuration and the priority rule to be followed. We would like to note that giving the same data used for testing the BPNNs, the neuro-genetic solution system empowered with simulation has significantly improved the solution with an error close to zero under all networks trained for SPT, EDD, CR, and FCFS priority rules. Note that the number of possible alternatives for the problem that is considered is extremely high. When there are m machines in each of the n workcenters and k possible dispatching rules, the number of feasible alternatives to be examined is km^n . In the following illustrative example, there are four work centers, at most five machines in each work center, and four priority rules. As a result, the total number of possible cases is equal to 2400. Our experiment reveals that the time spent to obtain a solution from proposed neuro-genetic approach is significantly less than that from either pure simulation, pure genetic or pure neural approach for the same quality solution.

4. An illustrative example

Here we would like to illustrate how the procedure is performed. For evaluation purpose, without loss of generality, we set the target values of each performance measures as follows:

Mean flow time (F)	603
Mean tardiness (T)	470
Maximum complete time (C_{max})	1990
Machine utilization rate of work center-1 (MU-1)	0.90
Machine utilization rate of work center-2 (MU-2)	0.40
Machine utilization rate of work center-3 (MU-3)	0.35
Machine utilization rate of work center-4 (MU-4)	0.85
Percentage of tardy jobs (N_T)	0.70

Table 3. Solutions obtained from each trained BPNN

Performance measures and target values		BPNN(SPT)		BPNN (EDD)		BPNN (CR)		BPNN (FCFS)	
		3 4 5 1	% e	3 4 5 1	% e	3 4 5 1	% e	3 3 5 1	% e
$\bar{F}$	603	419	30.5	467	22.5	601	0.3	601	0.3
$\bar{T}$	470	340	27.6	307	34.6	373	20.6	373	20.6
$C_{\max}$	1990	1625	18.34	1625	18.3	1625	18.3	1625	18.3
MU1	0.90	0.78	13.3	0.78	13.3	0.78	13.3	0.78	13.3
MU2	0.40	0.52	30.0	0.52	30.0	0.52	30	0.70	75
MU3	0.35	0.45	28.5	0.45	28.5	0.45	28.5	0.45	28.5
MU4	0.85	0.98	15.3	0.98	15.3	0.98	15.3	0.98	15.3
N_T	0.70	0.50	28.5	0.67	4.28	0.76	8.57	0.76	8.57
Mean Error (%)			24.0		20.8		16.8		36.0

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

These initial solutions suggest that the solution obtained by using the CR dispatching rule yields the lowest mean percentage deviation from the target values, followed by FCFS, EDD, and SPT. Our objective is to make sure that there is no other way to improve these results. For verification, we perform the next step, which is the genetic search. The four alternative solutions will be included in the initial population pool of genetic search algorithm. Preliminary analysis shows that 10 chromosomes are sufficient for the initial set of population pool. Thus, six more chromosomes are randomly generated. Cross-over rate and mutation rate are set as 1.00 and 0.04, respectively. Considering the definition of the overall performance measure, the best solution is obtained when the mean absolute deviation is minimal. Using these parameters, and the population pool described above, we obtain the following final solution, summarized in Table 4, as being the best.

Note that based on the results in Table 3, the best solution is obtained under the CR priority rule with 3-4-5-1 machine configuration with 16.80% average error. When genetic search is applied, the optimal solution is obtained as detailed in Table 4 under the EDD priority rule with 2-4-5-1 machine setting, and an average error rate of 1.70% is achieved for all measurements under consideration. Before the genetic search, the BPNN solution under EDD rule had an average

error rate of 20.8%. While the EDD rule was suggested to represent one of the worst case by BPNN process, the genetic search significantly improved the solution for this case by reducing the number of machines in the first work station by one.

5. Conclusions

This paper presents a two phase, neuro-genetic system to support managers in making efficient capacity design and scheduling decisions in a job shop environment. The framework presented in this paper can be used to obtain the optimal amount of resources in each workstation in conjunction with the right dispatching rule to schedule so as to minimize the deviations between the actual and predetermined target values of scheduling performance measures such as flow time, the number of tardy jobs, total tardiness, and machine utilization rates at each workstation. This managerial problem has significant implications: the proposed framework provides a tool for quality of service. It provides a balance among conflicting objectives for customer satisfaction, and efficiency/utilization at the manufacturing plant.

Mathematically, this multi-objective-highly non-linear strategic/operational problem is a very difficult problem because of the exponential number of feasible alternatives. The proposed neuro-genetic decision support system plays a role of an inverse simulator function: the inputs of the system are the target values of the performance measures. The neuro-genetic system outputs a suitable

Table 4. Final solution from neuro-genetic system

Performance measures	Target values	Achieved values	Error rates(%)
$\bar{F}$	603	603	0.00
$\bar{T}$	470	467	0.64
$C_{\max}$	1990	1980	0.51
MU1	0.90	0.93	3.23
MU2	0.40	0.40	0.00
MU3	0.35	0.37	5.41
MU4	0.85	0.83	2.41
N_T	0.70	0.71	1.41

Shop floor physical configuration; WC #1,2; WC #2,4; WC #3,5; WC #A, 1; Dispatching rule, EDD.

Manufacturing System design configuration, here, the number of machines, and the priority rule to schedule the jobs on the shop floor, given a set of desired performance measures more efficient than using solely simulation, artificial intelligence or GAs.

In this study, we have not considered the financial and environmental impacts of such strategic/operational decisions: for example, not running all of the machines might have energy savings, workforce savings and less pollution. Furthermore, testing the robustness of the system against conventional operations research techniques could be an addition and it is readily available as future opportunity.

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 Enterprise Corporation.

References

- Askin, R. G. and Standridge, C. R. (1993) *Modelling and Analysis of Manufacturing Systems*, Wiley, New York.
- Caskey, R. C. (2001) A manufacturing problem solving environment combining evaluation, search, and generalisation methods, *Computers in Industry*, **44**, 175–187.
- Cakar, T., Turker, A. and Cagil, G. (1996) The use of neural networks in flexible manufacturing system design. *Proceedings of 7th International Machine Design and Production Conference* METU, Ankara, Turkey, pp. 55–64.
- 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*, **36** (12), 3437–3457.
- Cedimoglu, I. H. (1993) *Neural Network in Shop Floor Scheduling*, Ph.D. Thesis, School of Industrial and Manufacturing Science, Cranfield University, UK.
- Chrysosolouris, G., 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. H. and Sittisathanchai, S. (1995) Genetic neuro-scheduler: a new approach for job shop Scheduling, *International Journal of Production Economics*, **41**, 135–145.
- Holter, T., Yao, X., Rabelo, L. S. Jones, A. and Yih, Y. (1995) Integration of neural networks and genetic algorithms for an intelligent manufacturing controller, *Computers and Industrial Engineering*, **29**, 211–215.
- 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.
- Lee, H. C. and Dagli, C. H. (1997) A Parallel Genetic-Neuro Scheduler for Job-Shop Scheduling Problems, *International Journal of Production Economics*, **51**, 115–122.
- 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*, **35**(4), 1171–1191.
- Lenski, R. E., Ofria, C., Pennock, R. T. Adami, C. (2003). The evolutionary origin of complex features, *Nature*, **423**, 139–144.

- Nakano, R. and Yamada, T. (1993) Conventional genetic algorithms for job shop problems. *Proceedings of the Fourth International Conference on Genetic Algorithms*, New York: Morgan Kaufmann, pp. 477–579.
- Philipoom, P. W. and Rees, L.P. (1997) Cost based due date assignment with the use of classical and neural network approaches, *Naval Research Logistics*, **44**, 421–446.
- Pierreval, H. (1993) Neural Network to select dynamic scheduling heuristic, *Reve des Systemes de Decision*, **2**(2) 173–190.
- Pinedo, M. (2002) *Scheduling: Theory, Algorithms and Systems*, 2nd Edition, Prentice Hall, NJ, 1995.
- 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*, June 18–22, Washington, pp. 2, 608.
- Rabello, L., Jones, A. and Tsai, J. (1993) Using hybrid systems for FMS scheduling. *2nd Industrial Engineering Research Conference Proceedings*, pp. 471–475.
- Sabuncuoglu, I. (1998) Scheduling with neural networks: a review of the literature and new research directions, *Production Planning and Control*, **9**(1), 2–12.
- Sabuncuoglu, I. and Gurgun, B. (1996) A neural network model for scheduling problems, *European Journal of Operational Research*, **93**, 288–299.
- 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.
- 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.
- Yu, H. and Liang, W. (2001) Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling. *Computers and Industrial Engineering*, **39**, 337–356.
- 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.