Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 53 (2007) 95–122
computers & industrial engineering
www.elsevier.com/locate/dsw

A review on evolution of production scheduling with neural networks

Derya Eren Akyol *, G. Mirac Bayhan

Department of Industrial Engineering, Dokuz Eylul University, 35100 Bornova-Izmir, Turkey

Received 13 February 2006; received in revised form 6 December 2006; accepted 11 April 2007
Available online 13 April 2007

Abstract

The production scheduling problem allocates limited resources to tasks over time and determines the sequence of operations so that the constraints of the system are met and the performance criteria are optimized. One approach to this problem is the use of artificial neural networks (ANNs) stand alone or in conjunction with other methods. Artificial neural networks are computational structures that implement simplified models of biological processes, and are preferred for their robustness, massive parallelism, and learning ability. In this paper, we give a comprehensive overview on ANN approaches for solution of production scheduling problems, discuss both theoretical developments and practical experiences, and identify research trends. More than 50 major production and operations management journals published in years 1988–2005 have been reviewed. Existing approaches are classified into four groups, and additionally a historical progression in this field was emphasized. Finally, recommendations for future research are suggested in this paper.

Keywords: Artificial neural networks; Production scheduling; Review

1. Introduction

Scheduling is one of the most important functions in a production firm. It is the allocation of available production resources over time to meet some set of performance criteria. Typically the scheduling problem involves a set of jobs to be completed, where each job comprises a set of operations to be performed (Rodammer & White, 1989). It is known that the scheduling problem which belongs to a class of constraint optimization problems (COPs) is NP-hard. In the last decades, different solution methods such as mathematical programming, dispatching rules, expert systems, and neighborhood search have been proposed for modeling and solution of scheduling problems. Despite a vast amount of work existing in the literature, to find an efficient method to obtain optimal solutions in polynomial time motivated the researchers to apply neural networks to scheduling problems and to compare their performance with other techniques'.

* Corresponding author. Tel.: +90 232 3881047; fax: +90 232 3887864.
E-mail address: derya.eren@deu.edu.tr (D.E. Akyol).

0360-8352/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2007.04.006

Artificial neural networks (ANNs) can be put into local search based metaheuristics category which includes simulated annealing, noisy methods, guided local search methods, iterated local search, tabu search, threshold accepting, and variable neighborhood search (Osman, 2002). From a modeling viewpoint, they are mathematical representations of biological nervous systems that can carry out complex cognitive and computational tasks. They are composed of many nonlinear interconnected processing elements that are analogous to neurons, and connected via weights that are analogous to synapses. The modern age of neurocomputing started with the work of McCulloch and Pitts (1943) in which the first mathematical model of a single biological neuron was presented. Although McCulloch and Pitts' study showed that simple type of neural networks were able to learn arithmetic or logical functions, ANN algorithms have been successful enough for many applications in the mid 1980s (Potvin & Smith, 2003). ANNs attracted the attention of many researchers from different disciplines such as engineering, physics, mathematics, computer science, and medicine. In recent years, they have become popular in various real world applications including prediction and forecasting, function approximation, clustering, speech recognition and synthesis, pattern recognition and classification, and many others. Applications of ANNs to scheduling (for detailed survey see Sabuncuoglu, 1998) are in accordance with using ANNs as a highly parallel model for general-purpose computing and then applying them for different combinatorial optimization problems (for detailed survey see Looi, 1992; Smith, 1999).

In the literature, ANNs have attracted much attention because of their characteristics listed below:

- By exposing examples of the relationship to the network, ANNs learn and are used to capture the complex relationship between the input and output variables that are difficult or impossible to analytically relate such as the relationship between the performance measures and operational policy of a manufacturing system or between the job characteristics and the performance measure of a scheduling system. After learning the unknown correlation between the input and output data, they can generalize to predict or classify for the cases they were not exposed to.
- In some cases of designing manufacturing systems, ANNs are preferred to time consuming simulation approaches.
- As a schedule retrieval system, ANNs such as backpropagation networks (BPNs) produce a schedule for a given set of input parameters but unlike the Hopfield networks they do not generally perform optimization.
- BPNs are also used to select scheduling rules or a manufacturing strategy to achieve accurate estimations of parameters such as the values of the look ahead parameters of scheduling rules. They are used to estimate the system performance measures such as mean utilization, mean job tardiness, mean flow time, etc.
- In static scheduling environments, it is possible to obtain the optimal or near optimal schedules by mathematical modeling, dynamic programming, branch and bound or other advanced methods. But, since real manufacturing environments are dynamic, flexible scheduling methods are needed to react any change in the system that varies with time. Thus, in dynamic scheduling environments, ANNs are employed to reduce the need for rescheduling.
- While optimizing networks such as Hopfield network and its extensions are involved directly in the optimization by mapping the scheduling objective functions to be optimized and constraints of the problems on to these networks, competitive networks can detect regularities and correlations in input vectors and adapt future responses accordingly (Min, Yih, & Kim, 1998).

In recent years, besides their advantages of parallelism, learning, generalization capability, nonlinearity, and robustness, several limitations of ANNs such as settlement into local minima, trial and error parameter determination process, long learning time are perceived. To compensate its disadvantages, hybrid systems in which ANNs are combined with traditional heuristics or metaheuristics and/or evolutionary algorithms or different approaches, and evolutionary ANNs have been proposed.

The purpose of this paper is to give a comprehensive survey of recent research on ANN applications in production scheduling, and to identify some future research directions. The organization of the paper is as follows. Through Sections 2 and 3, we review the literature of scheduling with ANNs parallel to the gradual developments in ANNs. Some conclusions and future research directions are given in Section 4.

2. Scheduling with stand alone neural networks

2.1. Hopfield network and its extensions

The original Hopfield NNs, which consist of a fully connected network of neurons capable of performing computational tasks were introduced by Hopfield (1982). Using binary state neurons and a stochastic algorithm to update the neurons, this network serves as a content addressable memory that allows for the recall of data based on the degree of similarity between the input pattern and the patterns stored in the memory. This model is known as the discrete and stochastic Hopfield model.

In a later work, Hopfield (1984) proposed a deterministic model based on continuous neurons. The idea was inspired by the fact that the neurons of the original model were different than the real biological neurons and from the realistic functioning of electronic circuits. So by maintaining the important properties such as content-addressable memory of the original model, a new model is constructed. Massive parallelism and convenient hardware implementation of the network architecture are among the most important advantages of Hopfield networks. The architecture of a Hopfield network with three processing elements (neurons) is shown in Fig. 1. In this single layer network, each neuron is connected to other neurons but no neuron has a connection with itself.

The idea of using ANNs to provide solutions to NP-hard optimization problems was pioneered by Hopfield and Tank (1985) with the use of their network for solving the Traveling Salesman Problem (TSP). In their paper, Hopfield and Tank show that if an optimization problem can be represented by an energy function, then a Hopfield network that corresponds to this energy function can be used to minimize this function and to provide an optimal or near-optimal solution. Since then, because of the advantages of using Hopfield networks, extensive research has been carried out on the application of the Hopfield networks for solving different optimization problems. In this network, objective function and the problem constraints are encoded in terms of an appropriate *energy function*. The aim is to obtain a configuration minimizing the energy function. Translation of the optimization problem into an appropriate energy function is in general, a difficult task. It must be in a quadratic form to meet the form of the energy function of the Hopfield network. Applying the most common method, penalty function approach, the energy function of the network is set equivalent to the objective function of the problem, and the problem is reduced to an unconstrained form by including the constraints of the problem in the energy function as penalty terms (Potvin & Smith, 2003). By this way, the constraint violations are penalized. The next step is to compare the energy function of the problem with the energy function of the Hopfield network to derive the weights and external inputs. Then, by random initialization of the network and updating the neurons, the stable states are obtained.

The success in applying neural networks to the TSP motivated many scheduling researchers to employ Hopfield networks. Foo and Takefuji (1988a, 1988b) use a two-dimensional Hopfield TSP type matrix of neurons with $mn + 1$ rows and mn columns, where m and n are the number of machines and the number of jobs, respectively, to map their job shop scheduling problem on. To find the global minima of the energy function that represents the objective function of the problem, they applied simulated annealing (SA) which is a stochastic optimization technique and uses a stochastic hill-climbing algorithm with the added ability to escape from local minima in the state-space where conventional methods usually get trapped (Kirkpatrick, Gelatt, &

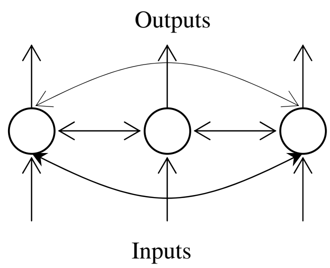

Fig. 1. The Hopfield network.

Vecchi, 1983). From the results obtained, it is seen that the proposed methodology gives near optimal solutions rather than an optimum solution. Therefore, to get better results and to reduce the number of neurons for the same problem, Foo and Takefuji (1988c) introduced integer linear programming networks as extensions of the original Hopfield network, and achieved better solutions. But, Van Hulle (1991) addresses that the network of Foo and Takefuji (1988c) generates constraint-violating solutions. To overcome this drawback, the original job shop scheduling problem was formulated again as a goal programming problem to be mapped onto a goal programming network. The simulation results showed that although the proposed approach yielded feasible solutions, it could not guarantee optimal solutions.

The limitations of the traditional Hopfield NNs based on the quadratic energy function triggered the authors Zhou, Cherkassy, Baldwin, and Olson (1991) to propose a neural network having a linear cost (energy) function rather than the quadratic energy function of the Hopfield network. Doing so, they aimed to improve the scaling properties of the Hopfield NNs. They compare their network with integer linear programming neural network of Foo and Takefuji (1988c) and TSP type Hopfield network method of Foo and Takefuji (1988a, 1988b) in terms of the number of neurons and interconnections required. The results obtained were very encouraging for both criteria.

Due to the problems of Hopfield NNs in solving optimization problems, various modifications were proposed to improve the convergence of the Hopfield network. While several authors modified the energy function of the Hopfield network to improve the convergence to valid solutions (Aiyer, Niranjan, & Fallside, 1990; Brandt, Wang, Laub, & Mitra, 1988; Van Den Bout & Miller, 1988), many others studied the same formulation with different penalty parameters (Hedge, Sweet, & Levy, 1988; Kamgar-Parsi & Kamgar-Parsi, 1992; Lai & Coghill, 1992). But although the modified versions of the Hopfield network could give valid solutions, they may not converge to good quality solutions. In the following years, poor solution quality of Hopfield networks was improved by integrating stochasticity into the Hopfield network. Boltzmann machine, Gaussian machine, Cauchy machine, and mean field annealing approaches were obtained by embedding stochastic properties into the Hopfield network.

A stochastic neural network for solving dynamic resource constrained scheduling problems was proposed by Vaithyanathan and Ignizio (1992). The authors represented their problem as a series of multidimensional knapsack problems, and used neural networks to solve these problems. The network included the combination of a Hopfield network and external neurons to give stochastic property. The experimental results showed that the network was able to avoid local minimum. As mentioned before, Gaussian machines developed by Akiyama, Yamashita, Kajiura, and Aiso (1989) as another alternative approach of escaping local minima were proposed for improving the efficiency and speed of the Boltzmann machine. Like continuous Hopfield networks, they have continuous outputs with a deterministic activation function. But in Boltzmann machines random noise is added to the external input of each neuron. In 1992, Arizono, Yamamoto, and Ohta proposed a Gaussian machine model for solving the single machine scheduling problem having the objective of total actual flow time minimization. Computational results show that in most of the problems the proposed network is successful in finding the optimal solutions.

Lo and Bavarian (1993) extend the gradient approach of two-dimensional Hopfield network to a three-dimensional matrix, called neural box, in which the third dimension was the time. They use this network to solve the job shop scheduling and multiple traveling salesmen problem. Although the simulation results showed that the presented approach yields feasible schedules, too many numbers of neurons and interconnections are required for solving large sized problems.

Another extension of Hopfield network was proposed by Satake, Morikawa, and Nakamura (1994) for minimizing the makespan of the job shop scheduling problems. In the energy function, only one constraint is included, and the other constraints are reflected in the threshold values. In the simulation experiments, the presented network gave optimal or near optimal solutions The difference between the proposed network and the original Hopfield network is the revision of the threshold values of the network at each transition of neurons, and the inclusion of the Boltzmann machine (Hinton & Sejnowski, 1986) known as the integration of the dynamics of the discrete Hopfield model with the simulated annealing methodology. Following this work, Foo, Takefuji, and Szu (1995) propose a modified Hopfield and Tank network for job shop scheduling problems. The presented network, used for solving integer–linear programming problems, differs from the traditional Hopfield and Tank network with the addition of nonlinear step function h amplifiers and with the use of a linear energy function rather than the quadratic energy function of the original Hopfield and Tank network. They examine the proposed approach, and conclude that it requires more number of neurons and interconnections than those needed by the approach in Zhou et al. (1991) that includes a linear energy function, but it does not need extensive calculations as in Zhou et al. (1991).

In another study, Willems and Brandts (1995) map the sequencing and resource constraints of the integer linear programming representation of the job shop scheduling problem on an extension of Hopfield network that includes general rules of thumb as an optimization criterion. By comparing the proposed approach with heuristic rules, the authors obtained better solutions than the traditional heuristic approaches.

Besides its advantage of escaping from local minima, the Boltzmann machine requires large computational times as the size of the problem increases (Aarts & Korst, 1989). In order to reduce the excessive computation times of the Boltzmann machines, Peterson and Anderson (1987) propose mean field annealing by replacing the stochastic bipolar state neurons of the Boltzmann machine with deterministic and continuous neurons. The normalized mean field annealing (MFA) and the Hopfield neural network method (HNN) are applied to the n job m machines scheduling problem including resource and timing constraints in Huang and Chen (1999). To solve the problem, neural net optimization algorithm is used. In other words, states that both satisfy the constraints of the problem and minimize the energy function are found. In this work, rather than using linear programming or the k out of N rules to define the energy function, the objective function is formulated according to the constraints involved, step by step. Then the total energy with all constraints is obtained. The derived energy function is transformed into corresponding neural network for both algorithms HNN and MFA. According to the simulations results, the generated energy functions work successfully for multiprocessor problems.

Chen and Dong (1999) study a production scheduling problem in a major surface mount technology (SMT) factory in Western Canada to minimize the total setup cost in producing different products in one of the SMT assembly lines. A nonlinear mixed integer programming model is proposed to represent the problem with constraint equations. In order to solve the optimization problem, Hopfield–Tank neural network is used. The authors conclude that the computational times to reach optimal solutions using the network approach are comparable to those required by mathematical programming softwares, and significant reduction could be obtained in computational time if parallel computing were utilized.

Liansheng, Gang, and Shuchun (2000) develop an intelligent scheduling model by implementing a unified neural network algorithm. Their network is based on Hopfield neural network, and used to solve different schedule mode problems including job-shop scheduling, priority scheduling, dynamic scheduling, and JIT scheduling.

In a recent work, to deal with the earliness and tardiness multi machine scheduling problem including sequence dependent setup times, Akyol and Bayhan (2005) suggest a coupled gradient network approach which is the extension of Hopfield (1984) and Hopfield and Tank (1985). The aim of their study is to minimize the weighted sum of the earliness and tardiness penalties using a neural network approach rather than the traditional approaches in scheduling. Using the penalty function approach, the formulated problem is represented by an energy function. After six recurrent networks were designed, the dynamics are defined by gradient descent on the energy function. Although the authors explain the necessary steps to simulate their networks, to test the network was left to a further study.

Table 1 depicts the main characteristics of the reviewed approaches based on Hopfield-type networks.

Any optimization problem of scheduling that can be defined by a quadratic form can be tackled with Hopfield networks. Then, a Hopfield network whose energy function reaches its minima at the same points with the cost function that describes the scheduling problem must be designed. However, by performing gradient descent on the energy function, the Hopfield model gets easily trapped in local minimum states, and this causes decreasing efficiency especially in large sized problems. Additionally, determining the appropriate values of the penalty parameters, network parameters, and initial states are other critical issues associated with this model. Solving scheduling problems represented by many constraints will cause a tradeoff between the penalty terms to be minimized. Despite the promising results obtained by the proposed methods, some aspects still need further studying. There is no exact method that guarantees a global optimum solution. Even if it is achieved, the proposed models will suffer from extremely large computation times. Moreover, few studies are carried out for

Table 1
Characteristics of Hopfield type networks in production scheduling

Reference(s)	Approach	Objective(s)	Remarks	Application area
Foo and Takefuji (1988a, 1988b)	Hopfield network	Minimizing the sum of all the starting times of each job's last operation	Static scheduling A stochastic TSP-type network Infeasible for large size problems (number of neurons and interconnections increase as the problem size grows)	Job-shop scheduling problem
Foo and Takefuji (1988c)	An extension of Hopfield network	Minimizing the sum of all the starting times of each job's last operation	Static scheduling Integer linear programming network The number of neurons and interconnections required are reduced compared to the work of Foo and Takefuji (1988a, 1988b)	Job-shop scheduling problem
Zhou et al. (1991)	An extension of Hopfield network	Minimizing the sum of all the starting times of each job's last operation	Static scheduling Has a linear energy function Number of neurons and interconnections required are less than needed in the linear network of Foo and Takefuji (1988c)	Job-shop scheduling problem
Van Hulle (1991)	Hopfield network (Goal programming network)	Minimizing the total starting times of all operations	Static scheduling The original job-shop problem is reformulated as a goal programming problem to guarantee feasibility Is not suitable for large size problems	Job-shop scheduling problem
Vaithyanathan and Ignizio (1992)	An extension of Hopfield network	Scheduling as many number of jobs as possible without violating the resource constraints	Dynamic scheduling A combination of a Hopfield network and external neurons used to provide stochasticity	Resource constrained job shop scheduling problem
Arizono et al. (1992)	An extension of Hopfield network	Minimizing total actual flow time	Static scheduling A stochastic neural network (A Gaussian machine model)	Single machine scheduling
Lo and Bavarian (1993)	An extension of Hopfield network	Minimizing the total setup cost and the time required to complete all jobs	3D-network (includes time as the third dimension to provide a basis for dynamic scheduling) Not suitable for large size problems	Job-shop scheduling problem

Satake et al. (1994)	An extension of Hopfield network	Minimizing makespan	Static scheduling Incorporates a Boltzmann machine mechanism The energy function includes only one constraint, the other constraints and the objective function are reflected in the threshold values	Job-shop scheduling problem
Foo et al. (1995)	An extension of Hopfield network	Minimizing the sum of all the starting times of each job's last operation	Static scheduling Has a linear energy function Includes step-function amplifiers Better than the approach of Foo and Takefuji (1988c) Requires more neurons and interconnections than required in the work of Zhou et al. (1991) but requires less computations	Job-shop scheduling problem
Willems and Brandts (1995)	An extension of Hopfield network	Minimizing makespan	Static scheduling The network is designed using the rules of thumb	Job-shop scheduling problem
Huang and Chen (1999)	Hopfield network and the normalized mean field annealing method (obtained by embedding simulated annealing into the Hopfield network)	Solving the multiprocessor problem with timing and resource constraints	Static scheduling Is not suitable for large size problems	Job-shop scheduling problem
Chen and Dong (1999)	Hopfield network	Minimizing total setup cost in one of the surface mount technology assembly lines	Static scheduling The model is expressed by a nonlinear integer programming model	Job-shop scheduling problem
Liansheng et al. (2000)	An extension of Hopfield network	Minimizing total idle time of all machine tools	Static and dynamic scheduling The scheduling model is based on a unified NN algorithm	Different schedule mode problems including job-shop scheduling, priority scheduling, dynamic scheduling and JIT scheduling.
Akyol and Bayhan (2005)	An extension of Hopfield network	Minimizing the weighted sum of the earliness and tardiness penalties	Static scheduling A coupled network	Parallel machine scheduling problem

the comparison of the Hopfield networks and its extensions performance with the performance of best known heuristics or metaheuristics. So, we believe this issue will be given more importance in the near future.

2.2. Multilayer perceptrons

One of the important types of networks used in scheduling applications is a multilayer perceptron, a feed-forward network including a set of neurons connected by weighted links. It consists of an input layer, one or more hidden layers and an output layer. Backpropagation, which was first introduced by Werbos (1974), was later rediscovered independently by Parker (1985) and Rumelhart et al. (1986), and then modified in various manners by numerous researchers in order to overcome its deficiencies, is one of the most popular algorithms for training multilayer perceptrons. This learning rule is a kind of gradient descent technique with backward error propagation. Multilayered perceptrons trained with backpropagation learning algorithm are generally referred to as backpropagation networks. A typical backpropagation network is shown in Fig. 2. The weights of the network are randomly initialized before training starts. Then, a pair of patterns including the input patterns and the desired patterns is applied to the network. By propagating through the network layer by layer, a set of outputs is produced as the actual outputs of the network. At the output layer, the actual outputs are compared to the desired outputs, and an error signal is computed by subtracting the actual value from the desired value. This error signal is propagated backward through the network and the weight values are then adjusted by a magnitude proportional to the negative gradient of the error function, which is generally equal to the sum of squared errors. By this way, the difference between the actual and the desired outputs is minimized (Haykin, 1994).

Backpropagation networks have been successfully used in modeling, classification, forecasting, design, control, and pattern recognition. Their improved generalization capabilities over competing machine learning tools and their easy mechanism made them attractive to be utilized in production scheduling. A successful use of a backpropagation network for job shop scheduling environments can be found in Chryssolouris, Lee, and Domroese (1991) where they employ simulation and a backpropagation network to establish adequate weights between the operational policy of a work centre and performance measures such as the mean costs, mean flow time and mean tardiness. This study is among the first examples to the neural network based metamodels for the system design problems. A similar application can be found in Philipoom, Rees, and Wiegmann (1994), where a backpropagation network is proposed to determine due dates for job shops. In order to see whether neural networks are successful in assigning due dates, the assigned due dates are compared to regression based due date assignment rules. The proposed neural network outperforms the six linear rules and the nonlinear regression model with respect to mean absolute deviation and standard deviation of lateness criteria.

Because of their flexibility and adaptability properties, ANNs have been used not only in static scheduling environments but also in dynamically changing manufacturing environments where the values of the system attribute change continually (Arzi & Iaroslavitz, 1999; Chen, Huang, & Centeno, 1999; Chen & Muraki, 1997; Li, Chen, & Lin, 2003).

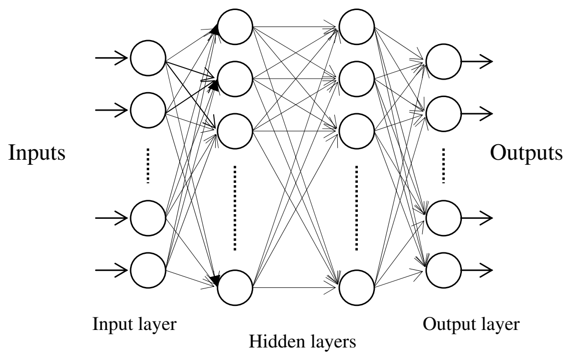

Fig. 2. A multilayer perceptron.

Geneste and Grabot (1997) show how to consider the information based on the workshop and the manufacturing orders structure and on the objectives of the workshop manager in order to select a relevant scheduling strategy. They propose parameterized scheduling heuristics and suggest two methods to tune the heuristic rule.

As pointed out by Jain and Meeran (1998) some of the main problems faced in the application of traditional backpropagation networks and in Hopfield networks are the lack of generalized learning capability to map inputs and outputs for NP hard problems, and the growing network size for large size problems, respectively. To overcome these shortcomings, the authors suggest a modified backpropagation model and use it for makespan minimization. The main difference between the proposed network and other backpropagation networks is that it performs optimization itself. The modified backpropagation system is compared with three priority dispatching rules; SPT, MWR, FCFS, and the Shifting Bottleneck Procedure of Adams, Balas, and Zawack (1988). The proposed system offers shorter makespans in considerable computational times than three dispatching rules.

For dealing with single machine sequencing problems, El-Bouri, Balakrishnan, and Popplewell (2000) develop a backpropagation neural network approach where they utilize a 11-9-1 three layered neural network in which each job is represented by its specific information and the output unit determines where the corresponding job lies in the sequence. The proposed network is evaluated for three performance criteria; mean flow time, mean weighted flow time, and maximum job tardiness. The network is successful in minimization of the mean flow time and the mean weighted flow time. The network also allows the jobs to be sequenced in order to minimize the maximum tardiness. For another performance criterion, minimization of the mean job tardiness, the network's capability is investigated and the results are compared with two sorting rules. Although the network's solutions are superior to those of the sorting rules, about 6–12% difference from optima motivated the authors to develop a Neural Job Classification and Sequencing System (NJCASS). The results showed that NJCASS has many advantages, for instance, it was flexible under different performance criteria. The approach proposed by Hamad, Sanugi, and Salleh (2003) bears some similarities to that of El-Bouri et al. (2000), although the former is applied to a single machine case. Hamad et al. (2003) deal with the non-identical parallel machines problem and propose a way of representing the problem to be fed into a backpropagation network, and try to minimize the sum of earliness and tardiness costs. In this study, the two-output representation is used instead of one-output unit (representing the target values) representation proposed in El-Bouri et al. (2000).

In their work, Park, Kim, and Lee (2000) present a neural network approach for solving identical parallel machine scheduling problems with sequence dependent set up times to minimize weighted tardiness. Their work is an extension of Kim, Lee, and Agnihotri's (1995) approach to parallel machine situation. The difference between them is the inclusion of an additional factor called set up time range factor. The presented approach is also an extension of Lee, Bhaskaran, and Pinedo's (1997) ATCS (Apparent Tardiness Cost with Setups) rule in which four factors are used to quantify the problem characteristics. The differences between them are that the proposed approach includes an additional factor, and also trains a backpropagation network to obtain the values of the look ahead parameters. The simulation experiments point out 4% improvement over the original ATCS rule.

Sabuncuoglu and Touhami (2002) use backpropagation networks as a simulation metamodel, and try to measure metamodel accuracy in estimating manufacturing system performances in the job shop scheduling environments. The numerical results show that metamodeling with neural networks can be used effectively to estimate the system performances. Another neural network based metamodel application can be found in the study of Fonseca and Navaresse (2002). In this work, ANNs are used as a valid alternative to the traditional job shop simulation approach. In order to generate the training and test sets, the simulation software package Arena is used and applied to a problem from Askin and Standridge (1993). It is seen that the average flow times obtained from three different simulation packages, i.e. Arena, SIMAN, and ProModel are almost identical to the simulation outputs of the developed neural network models.

In another study, Raaymakers and Weijters (2003) also used backpropagation networks to estimate the makespan of job sets in batch process industries. Because the amount of job interaction depends on the mix of the jobs and the resource sets, they use aggregate characteristics of the jobs and the resources to estimate the amount of interaction. The authors apply both neural networks and regression analysis to determine the relationship between the variables affecting the amount of interaction and the amount of interaction at the scheduling level. Two kinds of regression models are used in this study; the first one includes only main effects, and the other comprises main effects and also two way interactions. The computational results show that these regression models and neural networks give satisfactory solutions, but the neural network's estimation quality is significantly better than these models.

Cha and Jung (2003) address the schedule assessment problems with the complex and competing environment of manufacturing systems. In order to overcome this problem, they introduce a methodology to provide a consistent and dimensionless degree of satisfaction. They exploit fuzzy numbers to represent the final assessment result of a schedule.

Feng, Li, Cen, and Huang (2003) apply multilayered perceptron networks to design, develop and implement a production activity scheduling system to be used in a job shop environment. They present a different data encoding method to represent the processing time and processing sequence of the jobs to be processed, use backpropagation training algorithm to control local minimum solutions, and introduce a heuristic method for revising the initial output. The implementation of the developed scheduling system on a real life job shop problem helps to improve the production measures of the manufacturing plant.

Cakar and Cil (2004) employ backpropagation networks for the design of manufacturing systems. Performance measures such as mean flow time, mean tardiness, maximum completion time, machine utilization rate of each work center and percentage of late parts are fed as inputs into the neural network, and the number of machines in each work center is obtained as output from the system.

In addition to the studies above, Akyol (2004) exploits backpropagation networks to model six different heuristic scheduling algorithms applied to a makespan minimization problem of a flow shop. The author incorporates fuzzy representation into the preprocessing steps, and then trains the networks. Due to the comparison results between the proposed approach and the six heuristic algorithms, the proposed method is successful to predict the makespan of the n job m machine permutation flow shop environment.

When the articles reviewed above are considered, it can be said that backpropagation networks, except the study of Jain and Meeran (1998), are not directly involved in the optimization problem. That is, actual scheduling is not performed. Main characteristics of the reviewed multilayer perceptrons are presented in Table 2.

The successes of most of the studies are the result of good generalization capabilities of backpropagation networks which are used to capture the complex relationship between input and output variables of the scheduling problem under consideration. Additionally, as also pointed out by Sabuncuoglu and Touhami (2002), in recent years, for the design of manufacturing systems, the literature includes different neural network based metamodels in which the training data is provided by simulation. Despite the increase in training time, integration of simulation with neural networks will provide better results in less time compared to time consuming stand alone simulation approach. Although the popularity of backpropagation networks has grown significantly in the past few years, some problems still exist with the application of the backpropagation networks. That is, these networks are trained by a gradient based search technique which has the risk of getting stuck in local optimum and the starting point of connection weights becomes an important issue to reduce the possibility of being trapped in local optimum. Another difficulty with the construction of these types of networks is the necessity of generating a training set which is time consuming. Therefore, in recent years, the performance of these networks is tried to be enhanced by combining them with different heuristics or metaheuristics.

2.3. Competitive type networks

The works by Grossberg (1972), von der Malsburg (1973), Fukushima (1975), Willshaw and von der Malsburg (1976), and Grossberg (1976a, 1976b) are the first in the area of competitive learning. Unlike Hopfield networks, *the winner take all* strategy forms the base of the competitive networks. In this unsupervised network, there is a single layer of output neurons fully connected to the input neurons of the network. In the output layer known as the competitive layer, lateral inhibition occurs among the neurons, and each neuron tries to inhibit the neuron to which it is laterally connected. For an input pattern presented to the network, the neuron with the weight vector at the least distance from the input vector is called the winner and its output is set to one. A typical competitive network is illustrated in Fig. 3.

Table 2
Characteristics of Multilayer perceptrons in production scheduling

Reference(s)	Approach	Objective(s)	Remarks	Application area
Chryssolouris et al. (1991)	Multilayer perceptron	Determining operational policies for manufacturing systems to achieve a set of goal performance measures	Neural network based metamodel Suitable to complex applications	Job-shop scheduling problem
Philipoom et al. (1994)	Backpropagation network	Assignment of due dates for jobs based on system characteristics and system status	Neural network based metamodel Outperforms conventional regression based due date assignment rules	Job-shop scheduling problem
Chen and Muraki (1997)	Backpropagation network	To develop an action strategy framework to select the suitable modification mode for on-line scheduling and control	Dynamic scheduling On line rescheduling on the basis of preprocessed information about the plant status	Batch processes
Geneste and Grabot (1997)	Backpropagation network	To select a suitable scheduling strategy	A decision support system	Job-shop scheduling problem
Jain and Meeran (1998)	Modified backpropagation network	Minimizing makespan	Static scheduling Includes additional features such as a momentum parameter, jogging parameter, and a learning rate parameter to avoid local minima Is able to deal with large size problems	Job-shop scheduling problem
Chen et al. (1999)	Backpropagation network	To develop an intelligent scheduling and real time control system for rail-guided vehicle systems	Dynamic scheduling A neural network provides the material control strategy to be implemented and the network is trained by the data obtained by simulating various scenarios Superior to the static system with shorter flow time, higher system throughput and less WIP inventory	Scheduling of material handling system (FMS scheduling)
Arzi and Iaroslavitz (1999)	Backpropagation network	Choosing the most appropriate scheduling rule out of several predetermined ones	Dynamic scheduling ANN based production control system The objective criterion can be changed between different control periods	Flexible manufacturing cell
El-Bouri et al. (2000)	Backpropagation network	Minimizing mean flow time, mean weighted flow time, maximum job tardiness and mean job tardiness	Static scheduling A Neural Job Classification and Sequencing System (NJCASS) is developed to obtain better results	Single machine scheduling problem
Park et al. (2000)	Backpropagation network	Minimizing the sum of weighted tardiness	Static scheduling An extension of Kim et al.'s (1995) approach to parallel machine situation An extension of Lee et al.'s (1997) ATCS rule	Parallel machine scheduling problems with sequence dependent set up times
Sabuncuoglu and Touhami (2002)	Backpropagation network	Investigating the robustness of neural network based metamodels for estimating manufacturing system performances (long-term mean machine utilization, long-term mean job tardiness, short-term mean job tardiness)	Metamodel accuracy is affected by various factors Neural network based metamodel	Job-shop scheduling problem
Fonseca and Navaresse (2002)	Backpropagation network	Estimating average flow times	Dynamic scheduling Neural network based metamodel	Job-shop scheduling problem

(*continued on next page*)

Table 2 (*continued*)

Reference(s)	Approach	Objective(s)	Remarks	Application area
Raaymakers and Weijters (2003)	Backpropagation network	Estimating the makespan of a set of jobs	Static scheduling Uses aggregate information	Batch process industries
Feng et al. (2003)	Backpropagation network	Designing a scheduling software	A denotation bit is used to organize the sample data	Job-shop scheduling problem
Hamad et al. (2003)	Backpropagation network	Minimizing total weighted earliness and tardiness	Static scheduling	Unrelated parallel machine scheduling problems
Cha and Jung (2003)	Multilayer perceptron	To assess a production schedule	An aggregation methodology that uses fuzzy and crisp numbers	Job-shop scheduling problem
Li et al. (2003)	Backpropagation network	To find a method to reduce the need for rescheduling	Dynamic scheduling A functional virtual population concept is introduced	Flexible manufacturing systems
Cakar and Cil (2004)	Backpropagation network	To select the best design alternative according to the performance measures given	Neural network based metamodel	Manufacturing systems
Akyol (2004)	Backpropagation network	Minimizing makespan	Static scheduling An implementation alternative to scheduling heuristics	Flow shop scheduling problem

In order to apply competitive networks to solve optimization problems, the equations of motion for the problem constraints and an energy function that converges to stable states must be defined. For detailed information one can refer to Fang and Li (1990). Fang and Li (1990) obtain equations of motion for the 0–1 knapsack problem, the generalized assignment problem, and the single machine total tardiness scheduling problem including unit processing times and different deadlines. Although their study generated good results, the literature on the application of competitive networks to scheduling is sparse. More work has to be done in deriving the equations of motion to represent different constraints present in different types of scheduling problems.

A neural network model including a three-dimensional structure as in the work of Lo and Bavarian (1993) was proposed by Sabuncuoglu and Gurgun (1996). It is very similar to the Hopfield network but includes an external processor for monitoring and controlling the network evolution. The difference between the Hopfield network and the proposed network is that the proposed network involves a competition property. In other words, the neurons (jobs) compete with each other to be in the first available position in the sequence. This network is employed for solving the single machine mean tardiness problem, and job shop scheduling with makespan minimization. The performance of the proposed network is compared with the Wilkerson and Irwin (WI) algorithm, in terms of mean tardiness and the computation time, and gives better solutions than WI. In addition, the proposed network finds optimal solutions in most of the test problems.

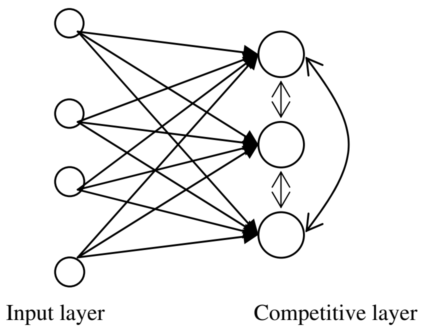

Fig. 3. A competitive network.

Chen and Huang (2001b) apply a competitive neural network in order to obtain solutions to the multiprocessor job scheduling problem with multiprocesses. The problem involves time and resource constraints, and is depicted by an energy function proved to be converging. This function is mapped onto the competitive Hopfield neural network (CHNN) known as a Hopfield neural network (HNN) with a *winner-take-all learning* mechanism. In other words, in competitive Hopfield neural network, instead of conventional deterministic learning rules, a competitive learning mechanism is used to update the neuron states so that the time required in obtaining coefficients is reduced and effective results are obtained.

Based on competitive learning, Kohonen (1982) proposed an unsupervised, clustering network known as self-organizing map in which only one neuron per group is on at a time. McMullen (2001) develops a neural network approach of the Kohonen self-organizing map (SOM) for solving a JIT production-sequencing problem with setups minimization and material usage stability. The experiments based on various test problems from the literature give near optimal solutions with respect to the objectives considered, and the SOM's overall performance is competitive with the search heuristics such as simulated annealing, tabu search and genetic algorithms (GAs). But the SOM needs more efforts to handle the CPU time problem.

In their later work, Min and Yih (2003) integrate simulation and a competitive neural network trained with the Kohonen learning rule, and develop a multi-objective scheduler to select dispatching rules for both machine and vehicle initiated dispatching decision variables, and to obtain the desired performance measures at the end of short production intervals. Extensive simulation experiments are conducted to collect the data including the relationships among the change of decision rule set and current system status and the performance measures of a semiconductor wafer fabrication system. A competitive network is used to group all instances of simulation outputs.

Main characteristics of the competitive type network approaches reviewed above are given in Table 3.

3. Scheduling with hybrid approaches

Several shortcomings of ANNs motivated the researchers to integrate neural networks with different computing techniques. As a result, there has been an explosive growth in the successful use of hybrid neural net-

Table 3
Characteristics of competition based networks in production scheduling

Reference(s)	Approach	Objective(s)	Remarks	Application area
Fang and Li (1990)	Competitive networks	Minimization of total tardiness	Static scheduling Satisfactory results and feasible solutions are obtained	Single machine scheduling
Sabuncuoglu and Gurgun (1996)	Competitive networks	Minimizing mean tardiness of single machine and minimizing makespan of job shops	Static scheduling Similar to the Hopfield network but includes an external processor	Single machine and job shop scheduling problems
Chen and Huang (2001b)	Competitive networks	Solving the multiprocessor problem with timing and resource constraints	Static scheduling The same problem considered in the work of Huang and Chen (1999) Is not suitable for large size problems	Job-shop scheduling problem
McMullen (2001)	Kohonen self-organizing map	Simultaneous optimization of setups and stability of material usage	Static scheduling Is worse than other heuristics in terms of CPU ratio	JIT production scheduling problem
Min and Yih (2003)	Competitive networks	Developing a multi-objective scheduler for the selection of dispatching rules to obtain the desired performance measures	Dynamic scheduling Considers both the machine initiated and vehicle initiated dispatching policies	Scheduling of semiconductor wafer fabrication system

works in scheduling. In this section, we review the scheduling studies exploiting the combinations of neural networks with different approaches.

Rabelo and Alptekin (1990) introduce an approach which hybridizes backpropagation neural networks with expert systems, and apply their hybrid system to find solutions for the FMS scheduling/rescheduling problem. To choose the best scheduling rules with respect to different criteria, ANNs are used to identify patterns in the tasks to be solved, and expert systems are used to monitor the performance of the system and to automate the learning process of the ANN.

One of the important shortcomings of ANNs is *trapping in local minima.* To handle this problem, ANNs are also combined with GAs which are first proposed and studied by Holland (1975). Works related with these combinations are summarized in Schaffer, Whitley, and Eshelman (1992).

Dagli and Sittisathanchai (1993) propose a hybrid approach combining GAs with neural networks. The approach finds the optimum solution in a few iterations for a problem from Foo and Takefuji (1988a, 1988b). Even the number of machines and jobs are increased, the results are also encouraging. Furthermore, the genetic neuro-scheduler proposed by these authors produces better solutions than the shortest processing time (SPT) rule for different sizes of problems. Another GA including hybrid approach system was developed by Rabelo, Yih, Jones, and Tsai (1993) for selecting candidate scheduling rules from a larger list of rules where backpropagation neural networks, parallel Monte Carlo simulation and inductive machine learning mechanism were integrated to minimize the maximum tardiness and mean flow time.

In recent years, the development of artificial intelligence techniques has provided a powerful way of dealing with dynamic scheduling problems. In the study of Sim, Yeo, and Lee (1994), the backpropagation neural network is integrated with an expert system for solving dynamic job shop scheduling problems, and by this way, the weakness of each stand alone method is tried to be overcome. The integrated method exploits the advantages of both techniques. That is, the expert system helps to reduce the training time of the neural network by training sub-networks separately, while the neural network learns about and handles the complex interactions of the scheduling considerations without the need for the long knowledge acquisition and development time of expert systems. The authors show that the proposed network has better performance than priority dispatching rules, and could tackle the adaptive scheduling problems.

One of the major drawbacks encountered with neural networks is their lack of explanation power. It is difficult to explain how the networks arrive at their solutions due to the complex nonlinear mapping of the input data by the networks. In many applications, to gain better understanding of the problems at hand, it is desirable to induce knowledge from trained neural networks. In the literature, applying machine learning techniques to extract dynamic scheduling knowledge has been a successful method. In their work, Li, Wu, and Torng (1997) combine an adaptive neural network classifier and a decision tree technique to obtain scheduling knowledge for flexible manufacturing systems. System performance data are fed into the adaptive resonance theory neural network model (Carpenter & Grossberg, 1987) as inputs, and classified according to the similarities between them. In order to find a definition for each class, a decision tree method is performed and then this is converted into a set of rules to be used as the real time scheduling knowledge.

In the same year, in order to overcome the problems of convergence, stability and sensitivity to the initial inputs belonging to Hopfield networks, Jeng and Chang (1997) presented a non-energy based neural network architecture that implements a heuristic rule, combination of most-valid operation first and shortest operation first rule. They used this network to solve job shop scheduling problems with makespan minimization, and obtained optimal or near optimal schedules.

Lee and Dagli (1997) design a parallel genetic-neuro scheduler including six different modules for solving large size job shop scheduling problems, and test it on different size of job shop scheduling problems. They obtain optimum solution in a few iterations and superior solutions to SPT, EDD, SLACK for minimizing the lead time.

Min et al. (1998) design a dynamic and real time FMS scheduler by combining the competitive neural network and search algorithm to meet the multiple objectives given by the FMS operator. Based on the current decision rules, a current system status and performance measure, the competitive network generates the next decision rules. The simulation results indicate that the FMS scheduler is able to satisfy multiple objectives given by the operator. Another multiple objective flexible manufacturing system (FMS) scheduler was developed by Kim, Min, and Yih (1998) with the same objective. Their approach is the integration of inductive learning, competitive neural network and simulation. According to the comparison results between the competitive network approach and the proposed integrated approach for different objectives, the use of inductive learning is effective to refine the rough scheduling knowledge.

Rather than the usual non-adaptive neural networks proposed in the literature, Yang and Wang (2000) propose a constraint adaptive neural network (CSANN) for the generalized job shop scheduling problem that is more complex than the traditional job shop scheduling problem. The problem is represented by the integer mathematical programming models, and then mapped onto a neural network that consists of two layers. In this study, three different heuristic algorithms are combined with the proposed CSANN. From the simulation experiments conducted, it is seen that the performance of CSANN is improved by combining CSANN with the proposed heuristics. Yang and Wang (2001) extended the work of Yang and Wang (2000) by combining a new heuristic based on obtaining a non-delay schedule and one of the heuristics in Yang and Wang (2000) used to increase the speed of the solving process of CSANN, with CSANN to form a new hybrid approach. The new hybrid approach is efficient in obtaining the minimum makespan, and is fast in making calculations. Another constraint neural network was introduced by Yu and Liang (2001) where they again try to solve the expanded job shop scheduling problem (EJSSP), which is more difficult to solve than the original job shop scheduling problem, by including additional constraints such as job delivery due dates and available time of the resources. They proposed a hybrid approach of neural networks and GAs. In order to describe the processing constraints and resolve the conflicts, three types of neurons were described. Then a constraint neural network (CNN) formed by these neurons was developed. To optimize the starting time of the EJSSP, a gradient CNN was constructed. This gradient CNN was combined with GA for optimizing the sequence of the scheduling problem. The results of the study showed that the hybrid approach was effective for complex scheduling problems.

To deal with fuzzy and random production disturbances faced commonly in manufacturing systems, Li, Li, Li, and Hu (2000) presented a production rescheduling expert simulation system based on Chinese manufacturing. It combines many different techniques and methods, including simulation, backpropagation neural network, expert knowledge, and dispatching rules. The simulation module provides training patterns for the network. Simulation results reveal that the production rescheduling expert system is practical and increases production efficiency.

Another use of GA–neural network combination can be found in Lee and Shaw (2000) where they propose a two level neural network for a real time flow shop sequencing problem of a printed circuit board (PCB) manufacturing environment. Firstly, they construct a total of 10 problem sets including different number of machines and different number of jobs, and compare the performance of their pure neural network with two constructive heuristics: the deterministic greedy search and the NEH heuristic (Nawaz, Enscore, & Ham, 1983) on this problem set. They observe that the neural network approach is superior to constructive heuristics in terms of makespan and computational times. The performance of neural network approach is also compared with GAs, and it is seen that the neural network's performance is within 3.4% of those of GAs but the computational time needed by the neural network is only less than 0.2% of that of GAs. Furthermore, the neural network approach is combined with GAs, and the combined algorithm improves the solution quality and computational time of the GAs.

From optimization viewpoint, the Hopfield neural network and its extensions belong to the penalty method for solving the constrained real optimization into which a combinatorial optimization is converted (Li, 1996). The penalty function requires the weighting factors for the penalty terms to be sufficiently large in order to converge to a feasible solution. But as the penalty terms become stronger, the original objective function becomes weaker, and as they become larger and larger, the problem becomes ill conditioned. To deal with this problem, Li (1996) combined the augmented Lagrange multiplier method and the Hopfield network to obtain the augmented Lagrange Hopfield network. By this way, both the solution quality and the convergence properties of the Hopfield network were improved. The proposed approach helps to overcome the problems associated with the penalty method or the Lagrange multiplier method when used alone (Li, 1996). Following this work, Luh, Zhao, Wang, and Thakur (2000) proved the convergence of Lagrangian Relaxation Neural Networks (LRNN) for separable convex problems, and constructed LRNN for separable integer programming problems. They applied LRNN to separable job shop scheduling problems. By using Lagrange multipliers, the machine capacity constraints were relaxed, and the relaxed problem was decomposed into sub problems each of which was solved by dynamic programming. The performance of the method was much better than those of the existing neural network approaches.

The generically used expert scheduling system (GUESS) is an intelligent scheduling toolkit developed by Liebowitz et al. (1997). It includes a heuristic based approach, a hill-climbing algorithm and a GA approach to scheduling. Liebowitz, Rodens, Zeide, and Suen (2000) incorporated a Hopfield neural network approach into GUESS, compared its performance with the other approaches used by GUESS. The neural network approach produced good solutions for scheduling problems.

An altogether different approach was presented by Chen and Huang (2001a) for solving the multiprocessor scheduling problem involving non-preemptive multitasking with timing constraints. The proposed network known as a fuzzy Hopfield NN (FHNN) was different from the standard Hopfield network in the sense that *a fuzzy c-means clustering algorithm* was incorporated into it. In this method, each processor (job) was regarded as a data sample and every processor as a cluster. The objective function to be minimized was defined as the Euclidean distance between the data samples and the cluster sample, and the goal was to find the best set of clusters. In simulation experiments, the modified energy function of the network converged rapidly into a minimum value, and the penalty parameter determination problem, a major shortcoming of Hopfield NNs, was overcome.

Another neural network approach to adaptive scheduling can be found in the study of Shiue and Su (2002). In this approach, the aim is to develop a neural network based adaptive scheduling system to identify the important attributes of the system status and generate scheduling knowledge bases for an FMS system. The authors point out that by selecting important system attributes in manufacturing systems, better performance could be achieved in prediction. They develop an attribute selection algorithm based on the weights of backpropagation networks to measure the importance of system attributes in a neural network based adaptive scheduling (NNAS) system. Then, they combine their algorithm with the (NNAS) system and obtain an attribute selection neural network based adaptive scheduling (ASNNAS) system. Its performance is compared with the (NNAS) system's performance and with some dispatching rules for different criteria, and better solutions and less computational effort than the NNAS system are obtained for all the performance criteria.

Similar to their previous work, Shiue and Su (2003) develop an attribute selection decision tree (ASDT) based adaptive scheduling system by combining backpropagation networks with a decision tree learning (C4.5 algorithm) approach. This approach differs from Shiue and Su's (2002) approach in using the decision tree learning algorithm in constructing the scheduling system. The authors compare the classical DT-based approach with ASDT-based approach under different performance criteria, and conclude that using an attribute selection algorithm improves the generalization ability of knowledge bases, and causes less computational effort. In a similar work, Priore, Fuente, Pino, and Puente (2003) apply backpropagation networks and inductive learning (C4.5 algorithm) to acquire the scheduling knowledge by which the most appropriate dispatching rule in flexible manufacturing systems is determined. To improve the performance of the scheduling systems, they also propose a module used for generating new control attributes.

Wang, Jacob, and Roland (2003) address some limitations associated with traditional neural network models. Among these limitations are the requirement of excessive number of neurons, finding unfeasible solutions and the computational effort required for obtaining a solution. They propose a hybrid neural network approach to solve the flexible flow shop scheduling problem, which is a generalization of flow shop and parallel machine scheduling problems, with the objective of minimizing makespan. The authors exploit the structure of optimization problems and heuristic information, and compare their hybrid network with Ding and Kittichartphayak's (1994) heuristics with respect to the computational time and solution quality. The proposed hybrid approach outperforms all the heuristics on average and succeeds in dealing with the mentioned limitations.

A different application can be found in Agarwal, Pirkul, and Jacob (2003) where an Augmented Neural Network (AugNN) is proposed for solving the task scheduling problem. The proposed approach is a hybrid of the heuristic and the neural network approaches, and is used to minimize the makespan for scheduling n tasks on m identical machines. Unlike the traditional neural network approaches, the AugNN method proposes new input, output, and transfer functions such that the constraints are built into those functions and a heuristic can be embedded in them so that feasibility is always guaranteed. The heuristics used in this study are Highest Level First, Highest Level with Estimated Time First, Critical Path with most Immediate Successors First, Shortest Path Time, Longest Processing Time and Random. These six heuristics and AugNN (including these six heuristics and two learning rules) are compared based on three criteria: (a) reduction in gap between lower bound solution and heuristic solution, (b) number of cases with known optimum solutions, (c) number of cases where improvement in makespan occurs over heuristic. 570 problems of various sizes, ranging from 10 to 100 tasks, and from 2 to 5 machines are used for testing the performance of the AugNN over the six single pass heuristics. The suggested network outperforms the single pass heuristics with respect to all the three criteria.

Although, the gradient based search techniques such as the back-propagation are currently the most widely used optimization techniques for training neural networks, it has been shown that these gradient techniques are severely limited in their ability to find global solutions. Global search techniques have been identified as a potential solution to this problem. Glover (1986) proposed a meta heuristic approach, tabu search (TS), as a global search technique. Its popularity has grown significantly in the past few years (Sexton, Allidae, Dorsey, & Johnson, 1998). The work of Solimanpur, Vrat, and Shankar (2004) is a good example to this integration. The authors propose a neural network based TS method for solving the flow shop scheduling problems, and the initial permutation obtained from NEH algorithm is tried to be improved. This method is tested on 23 problems proposed by Taillard (1993) and compared with the BF–TS approach of Ben-Daya and Al-Fawzan (1998) in terms of makespan and computational time. It is seen that the proposed neuro-tabu search approach is effective over the BF–TS approach in terms of both criteria, and the tabu effect is reduced exponentially.

In Table 4 main characteristics of the reviewed hybrid approaches are summarized.

In recent years, the design of neural networks by evolutionary algorithms has been given great attention by researchers to develop adaptive systems that can change architectures and learning rules according to dynamic environments (Cho & Shimohara, 1998).

ANNs' performance is closely related with their architecture designs. Therefore, obtaining an optimal architecture design has been an important issue in the ANN field. But, since the basic principles governing the processing of information in neural networks is not well understood, optimal architecture design has been a very difficult task depending strongly on human experts having sufficient knowledge about ANNs and the problem to be solved. A trial and error method is used for the manual design that becomes more difficult and unmanageable as ANN complexity increases. Since the selection of the appropriate topology of a network, the best learning algorithm, and its parameters are problem dependent, in the literature there have been many attempts to automate the design of ANN architectures.

There has been a growing interest in using evolutionary search algorithms to eliminate the tedious trial and error work of manual design of ANNs. Evolutionary algorithms include evolution strategies (ESs) (Schwefel, 1981, 1995), evolutionary programming (EP) (Fogel, Owens, & Walsh, 1966), GAs (Goldberg, 1989; Holland, 1975; Jong, 1975), and a class of population-based stochastic search algorithms based on the ideas and principles of natural evolution. One important characteristic of these algorithms is that individuals in a population compete and exchange information with each other in order to perform certain tasks (Yao, 1999). Similar to ANNs, they have some advantages of robustness and parallelism. But they differ from ANNs in having global search capabilities that make them an applicable and an appealing approach. By maintaining diversity in the population, EAs can tackle large complex problems that generate many local optima. In contrast to gradient-based search algorithms, they do not use the gradient information. They are less likely to fall into local minima, and can be applied to problems for which little prior knowledge is available (Yao, 1997).

The ANNs designed by the evolutionary process are referred to as evolutionary ANNs (EANNs). However, every hybrid approach obtained by the combination of ANNs and EAs do not fall into the EANN category. In other words, EANNs belong to a special class of ANNs in which evolution is another essential form of adaptation in addition to learning. Using two forms of adaptation, EANNs can adapt to a dynamic environment efficiently and effectively (for more detailed information about evolution of ANNs see Yao, 1999).

Shugang, Zhiming, and Xiaohong (2005) propose a real-time scheduling algorithm to make a fuzzy classification for the operations of jobs in real time and then schedule them with the heuristic. To obtain the heuristic rule, a neuro-fuzzy network is trained with GAs. The proposed algorithm is highly efficient compared to the FIFO and the Lagrangian relaxation method.

Although the researchers deal with combining GAs, a branch of EAs, with ANNs, we could not identify much work on the use of EANNs in the area of scheduling. To the best of our knowledge, there is not any

Table 4
Characteristics of hybrid approaches in production scheduling

Reference(s)	Approach	Objective(s)	Remarks	Application area
Rabelo and Alptekin (1990)	A hybrid approach	Adaptive scheduling and control	Dynamic scheduling ANNs recognize the scheduling patterns and expert systems drive the inference strategy, monitor the performance of the system, automate the ANN learning process Hybrid of expert systems and backpropagation neural networks	FMS scheduling/ rescheduling problem
Dagli and Sittisathanchai (1993)	A hybrid approach	Minimizing the total completion time	Static scheduling Combination of GAs and backpropagation networks An ANN is applied to the evaluation module of a genetic algorithm	Job-shop scheduling problem
Rabelo et al. (1993)	A hybrid approach	Minimizing maximum tardiness and minimizing mean flow time	Dynamic scheduling Integration of backpropagation neural networks, parallel Monte Carlo simulation, genetic algorithms, and inductive machine learning mechanism An intelligent manufacturing controller is developed	FMS scheduling problem
Sim et al. (1994)	A hybrid approach	To select the best dispatching rule according to the prevailing workload condition and scheduling criteria	Dynamic scheduling (has the potential for adaptive and reactive scheduling) Backpropagation networks integrated with an expert system The use of expert systems reduces the training time of ANNs and the use of ANNs removes the need for the long knowledge-acquisition and development time of expert systems	Job-shop scheduling problem
Kim et al. (1995)	A hybrid approach	Minimizing total weighted tardiness	Static scheduling The backpropagation algorithm is combined with the apparent tardiness cost rule	Single machine scheduling
Li et al. (1997)	A hybrid approach	To obtain scheduling knowledge	Adaptive resonance theory (ART2) neural network combined with a decision tree technique ART2 is used to classify the performance data and the decision tree method is used to extract a definition for each class	FMS scheduling
Jeng and Chang (1997)	A hybrid approach	Minimizing makespan	Static scheduling Non-energy based neural network that implemented a heuristic rule A synchronous network	Job-shop scheduling problem
Lee and Dagli (1997)	A hybrid approach	Minimizing manufacturing lead time and maximizing production costs	Static scheduling Backpropagation neural network combined with GAs An ANN is applied to the evaluation module of a genetic algorithm	Job-shop scheduling problem

Min et al. (1998)	A hybrid approach	To develop an FMS scheduler to meet the multi-objectives desired by the operator	Dynamic scheduling Competitive networks combined with a search algorithm The competitive network classifies the input vectors according to their similarities and the search algorithm is used for real time processing and selecting the next decision rules	FMS scheduling
Kim et al. (1998)	A hybrid approach	To develop an FMS scheduler to meet the multi-objectives desired by the operator	Dynamic scheduling Inductive learning + competitive network + simulation The scheduling rules can be modified as the status of FMS changes	FMS scheduling
Yang and Wang (2000)	A hybrid approach	Minimizing makespan	Static scheduling Constraint adaptive neural network (CSANN) combined with three different heuristic algorithms Has a simpler architecture than the constraint satisfaction network of Willems and Brandts (1995)	Generalized job shop scheduling problem
Lee and Shaw (2000)	A hybrid approach	Minimizing makespan	Dynamic scheduling Combination of the neural network approach with GAs Guarantees feasible sequences	Flow shop scheduling
Li et al. (2000)	A hybrid approach	Aimed at satisfying different objectives such as minimizing job mean flow time, job queue length…, etc. but for the example problem, the objective is minimizing the sum of early and tardy times	Dynamic scheduling Integrates simulation, neural networks, expert knowledge, and dispatching rule	Production rescheduling problem
Liebowitz et al. (2000)	A hybrid approach	To incorporate an extension of Hopfield network into GUESS	Promising results are obtained	Different scheduling problems
Luh et al. (2000)	A hybrid approach	Minimizing total weighted earliness and tardiness	Static scheduling Combination of lagrangian relaxation with Hopfield network. A neural dynamic programming is developed to overcome the local minima and solution infeasibility of the traditional Hopfield network	Job-shop scheduling problem
Chen and Huang (2001a)	A hybrid approach	Optimization (minimizing the energy function of the multiprocessor problem expressed by multi-constraints)	Static scheduling Combination of Hopfield network with fuzzy c-means clustering algorithm One of the shortcomings of HNN, weightening factor determination process is avoided	Job-shop scheduling problem
Yang and Wang (2001)	A hybrid approach	Minimizing makespan	Static scheduling Constraint adaptive neural network (CSANN) combined with two different heuristic algorithms A new heuristic and one of the heuristics given in Yang and Wang (2000) are combined	Job-shop scheduling problem

(*continued on next page*)

Table 4 (*continued*)

Reference(s)	Approach	Objective(s)	Remarks	Application area
Yu and Liang (2001)	A hybrid approach	Minimizing makespan and minimizing the total penalty for tardy and early jobs	Static scheduling A hybrid approach of constraint NNs and GAs Better than the constraint NN alone	Expanded job-shop scheduling problem
Shiue and Su (2002)	A hybrid approach	To develop a neural network based adaptive scheduling system to identify the important attributes of the system status and generate scheduling knowledge bases for an FMS system	Dynamic scheduling (trained off-line but performs on-line scheduling control) Backpropagation network based adaptive scheduling system combined with an attribute selection algorithm	FMS scheduling
Shiue and Su (2003)	A hybrid approach	To develop a neural network based adaptive scheduling system to identify the important attributes of the system status and generate scheduling knowledge bases for an FMS system	Dynamic scheduling Backpropagation networks combined with a decision tree learning approach Includes an on-line scheduling and control mechanism	FMS scheduling
Wang et al. (2003)	A hybrid approach	Minimizing makespan	Static scheduling Artificial neural network combined with the structure of the optimization problem Guarantees feasible solutions	Flexible flow shop scheduling problem
Priore et al. (2003)	A hybrid approach	Minimizing mean tardiness and mean flow time	Dynamic scheduling Backpropagation networks and inductive learning To improve the system performance, a module is developed to generate new control attributes	FMS scheduling
Agarwal et al. (2003)	A hybrid approach	Minimizing makespan	Static scheduling Neural networks combined with heuristic approaches New input, output and transfer functions are proposed such that feasibility is always guaranteed Any heuristic with any learning strategy can be used with the proposed approach Performs well for large size problems	Task scheduling problem
Solimanpur et al. (2004)	A hybrid approach	Minimizing makespan	Static scheduling A neural network based tabu search method An improvement method	Flow shop scheduling
Shugang et al. (2005)	A hybrid approach	Minimizing the total weighted quadratic tardiness of all jobs	Dynamic scheduling (real time scheduling and rescheduling) EANN (the neuro-fuzzy network is trained by GAs)	Job-shop scheduling problem

scheduling application including the integration of other evolutionary algorithms with neural networks. It is doubtless that ANN researchers will benefit from the advantages of EAs by complementing and compensating each other's strengths and weaknesses to tackle the problems of scheduling.

In EANNs, evolution is employed at different levels to perform several tasks. At the lowest level, evolution can be employed to evolve weight training. In ANNs, weight training is usually formulated as minimization of an error function, such as the mean square error between target and actual outputs averaged over all examples. Connection weights are iteratively adjusted using training algorithms, such as BP and conjugate gradient algorithms based on gradient descent (Alvarez, 2002). Gradient descent based training algorithms have some disadvantages of getting stuck into a local minimum of the error function when the error function is multimodal and/or nondifferentiable. To overcome this drawback, evolution is introduced to find a near optimal set of connection weights without computing the gradient information.

At the next higher level, evolution can be employed to evolve the architecture of ANNs that strongly affects the information processing capabilities of ANNs. This helps to automate the design of ANNs which is a human experience dependent tedious trial and error work.

At the highest level, evolution can be employed to evolve ANN learning rule, which specifies how to adjust weights in weight training. Because the weight training has traditionally been regarded as a learning process, the evolution of learning rules can be considered as a process of learning to learn weights (Yao & Liu, 1998). For different types of architectures of ANNs under consideration, the ANN training algorithm may have different performance. When there is little prior knowledge about the architecture of ANNs, it becomes very difficult to design an optimal learning rule. By adapting a learning rule through evolution it is assumed that ANN's adaptivity will be enhanced in a dynamic environment. By this way, the relationship between learning and evolution will be modeled.

Since the evolutionary training method can deal with the global search problem of ANNs without computing the gradient information, it will be useful to employ them in solving production scheduling problems for which ANNs are incapable of finding a global minimum. Their application is not restricted to overcome the disadvantages of the backpropagation learning algorithm. EAs can also be used for optimizing recurrent neural networks such as Hopfield networks that possess the weakness of proving a local optimal solution to combinatorial optimization problems including scheduling. The applicability of the same evolutionary algorithm to train different types of networks reduces the human effort needed in developing different training algorithms. Besides having many advantages, EAs are not good at local fine-tuned search. In order to overcome this drawback, they are combined with local search algorithms such as simulated annealing, tabu search, backpropagation algorithm, etc. This kind of hybridization can improve the performance of EAs (Kido, Takagi, & Nakanishi, 1994; Mühlenbein, Schomisch, & Born, 1991; Yao, 1991).

4. Conclusions and future research

In this paper, we have tried to provide an extensive literature review on the applications of ANNs to different production scheduling problems. The reviewed articles were examined under four main categories – Hopfield type networks, multilayer perceptrons, competition based networks and hybrid approaches – according to the architectures they used. In order to see the gradual development in these works, the reviewed articles in each category were presented in a chronological order. In addition, we give advantages and disadvantages of the four main approaches in Table 5.

If we summarize briefly, Hopfield type networks are known to be optimizing models to solve various combinatorial optimization problems. In this single layer network, the objective function and the constraints of the problem are mapped in a suitable energy function and the state of the network is changed to minimize this energy function. The adjustment of system parameters does not depend on the difference between the desired and the actual output value of the system during the learning phase. The goal is to obtain feasible and good quality solutions. A multilayer feedforward network trained by backpropagation algorithm is known as a backpropagation network. Different than Hopfield type networks, backpropagation networks are not directly involved in optimization and need a training set to be used for the solution of particular problems. After training, application of the network involves only the computations of the feedforward phase (Fausett, 1994). These networks are generally preferred for their good generalization capabilities. A typical competitive

Table 5
Advantages and disadvantages of the main approaches

Approach	Advantages	Disadvantages
Hopfield-type networks	Massive parallelism Convenient hardware implementation of the network architecture Applicable to different kinds of combinatorial optimization problems in various disciplines	Gets easily trapped in local minimum states May not converge to good quality solutions Determining the appropriate values of the penalty parameters, network parameters and initial states is difficult and based on a trial-error process A tradeoff occurs between the penalty terms to be minimized The ways of incorporating constraints into the energy function affect the quality of the solution The termination criteria affect the quality of the results Translation of the problem into the energy function is difficult The network size grows with the problem size
Multilayer perceptrons	Universal approximators Better generalization capabilities over competing machine learning tools to capture the complex relationship between the input and output variables of the considered scheduling problem Easy mechanism to be utilized in production scheduling, even if the training is slow, a trained network can produce its output very rapidly	Gradient based training techniques have the risk of getting stuck in local minima The starting point of the connection weights becomes an important issue to reduce the possibility of being trapped in local optimum Generating a training set is time consuming Generalization ability depends on the adequacy of the training set Overlearning degrades the performance of the network Is not really indicated for combinatorial optimization Backpropagation algorithm's robustness and speed are sensitive to its control parameters

Competitive type networks	Is best applicable to optimization and classification problems Using competitive learning rule, the penalty terms are handled explicitly therefore the energy function is simplified and the time required in obtaining coefficients is reduced	Equations of motions need to be derived before solving the problem Cannot be applied to simplify the energy function of all scheduling problems Convergence should be analyzed carefully
Hybrid approaches	The problems of convergence, stability, penalty parameter determination and sensitivity to the initial inputs may be overcome The solution quality may be increased and computational time may be decreased Integrating global search techniques with neural networks can help to obtain global optimum solutions The advantages of each of the techniques can be combined to overcome the shortcomings They can compete effectively with other heuristics Adaptability to a dynamic environment is possible EANN eliminates the tedious trial and error work of manual design of ANNs By the evolution of connection weights, the shortcomings of gradient descent based training algorithms may be overcome ANNs' complexity can be decreased and its generalization can be increased by EAs, For some cases, evolutionary training can be faster and more reliable than BP algorithm Evolutionary learning can be applied to problems where gradient information is unavailable or costly	Disadvantages of the individual methodologies may be encountered Evolutionary training may be slow for some problems To employ EAs at any level of evolution is computationally expensive Two major problems in evolving ANNs; noisy fitness evaluation problem and the permutation problem

ANN consists of an input layer and a competition layer of processing nodes. All the nodes in the input layer are connected to every node in the competition layer. Input is fed to the nodes in the input layer processed through the connections by multiplication with the established connection weights. The sum of incoming value of a node in the competition layer competes with those of neighbor nodes (Kartam & Tongthong, 1998). The most extreme form of competition among neurons is called Winner Take All. After the competition, only one neuron in the competing group, the winner, will have a nonzero output signal. These networks are best applicable to optimization and classification problems and have been studied by a few researchers for solving scheduling problems. In recent years, NNs have been combined with other methods to form hybrid approaches to overcome some of the limitations of existing NNs. In hybrid approaches, one of the methods combined act as the main problem solver while the other assists it. EANNs can be considered as the combinations of ANNs and evolutionary search procedures. Different than other hybrid approaches, evolution is introduced into ANNs at three levels: connection weights, architectures, and learning rules. The two approaches combined function together in order to solve the problem.

Most of the approaches proposed in the reviewed articles are based on hybrid approaches, and a great emphasis has been given on the job shop scheduling problem, one of the hardest combinatorial optimization problems encountered in real scheduling environments. The literature presents many variants of traditional ANN approaches to improve their performance by trying to escape from the local minima, by reducing the computational effort required, by speeding convergence and by decreasing the number of neurons and interconnections.

Although widely preferred in the literature because of their highly parallel computational capabilities, one of the major problems in the application of Hopfield networks to optimization problems is the penalty parameter determination. Due to many constraints needed to express scheduling problems, the energy function will include too many penalty terms that result too many local minima. To satisfy all of the constraints while minimizing the objective function is very difficult, and a tradeoff exists between the constraint penalty terms and the objective function term. Thus, we believe that an important direction of future research is to search for the methods to overcome this tradeoff problem. In this regard, rather than using constant penalty parameters during simulations, employing time varying penalty parameters may be offered as a potential solution to this problem.

In the last years, ANNs have either been combined with artificial intelligence techniques – expert systems – with metaheuristics – GAs, tabu search, and simulated annealing – or with some heuristic procedures to form hybrid approaches providing superior solutions. As a global search technique, the combination of GAs with ANNs is widely used for obtaining optimal solutions, and considerable success is achieved by overcoming the slow convergence property of GAs and the local minima problem of ANNs. Future research should continue this trend by extending these works.

In the neural network design, setting of the parameters, initialization of the weights, configuration of the network are often problem specific, and the correct values of these parameters however are not known a priori. Therefore, for any given problem, a wide variety of parameters must be tried to generate confidence that a best solution has been found. Sensitivity of the ANNs to their initial configuration and inability of the gradient based search techniques to find global solutions motivated the researchers to employ EAs together with ANNs for the automatic adjustment of the parameters and the topology of the ANNs.

In recent years, following the need to solve real world dynamic scheduling problems, rather than non-adaptive neural networks whose connection weights and biases must be prescribed before the networks start to work, adaptive neural networks are developed and their performance is improved by combining them with several heuristic algorithms.

In the dynamic scheduling environments faced in real world manufacturing systems, scheduling and rescheduling problems can be handled by EANN's adaptation and learning properties. While several researchers develop new EAs for ANNS, some try to find remedies for these algorithms' shortcomings such as heavy computational loads, and time-consuming fitness evaluation (Hong, Lee, & Tahk, 2003; Palmes, Hayasaka, & Usui, 2003).

Together with its advantages, the hybrid approach of EAs and ANNs brings together unsolved problems from two complex areas. In addition, it is not clearly known at present how performance of EANNs in scheduling is. In order to provide a common platform for comparison, benchmark problems must be generated for different objective functions. The review on the EANN literature shows us that evolutionary optimization research area is not fully developed but is growing so fast.

The role of local search is not only important in the ANN field, it is also important in the field of EANNs. Combining EANNs with local search based metaheuristics which have an important feature of flexibility, will make them more effective and an important alternative to ANNs.

We believe that in the near future the researchers will benefit from the use of the recent advances in EAs, ANNs, metaheuristics, and their combinations. It can be concluded that, the future of ANNs lies in their use in conjunction with other advanced technologies.

References

Aarts, E., & Korst, J. (1989). *Simulated annealing and Boltzmann machines, a stochastic approach to combinatorial optimization and neural computing*. Chichester: John Wiley & Sons.

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science, 34*(3), 391–401.

Agarwal, A., Pirkul, H., & Jacob, V. S. (2003). Augmented neural networks for task scheduling. *European Journal of Operational Research, 151*(3), 481–502.

Aiyer, S. V. B., Niranjan, M., & Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. *IEEE Transactions on Neural Networks, 1*, 204–215.

Akiyama, Y., Yamashita, A., Kajiura, M., & Aiso, H. (1989). Combinatorial optimization with Gaussian machines. In *Proceedings of IEEE international joint conference on neural networks* (Vol. 1, pp. 533–540).

Akyol, D. E. (2004). Application of neural networks to heuristic scheduling algorithms. *Computers & Industrial Engineering, 46*, 679–696.

Akyol, D. E., & Bayhan, G. M. (2005). A coupled gradient network approach for the multi machine earliness and tardiness scheduling problem. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Lagana, H. P. Lee, & Y. Mun, et al. (Eds.). *Proceedings of ICCSA, Vol. 3483. Lecture notes in computer science* (pp. 596–605). Berlin: Springer.

Alvarez, A. (2002). A neural network with evolutionary neurons. *Neural Processing Letters, 16*, 43–52.

Arizono, I., Yamamoto, A., & Ohta, H. (1992). Scheduling for minimizing total actual flow time by neural networks. *International Journal of Production Research, 30*(3), 503–511.

Arzi, Y., & Iaroslavitz, L. (1999). Neural network-based adaptive production control system for a flexible manufacturing cell under a random environment. *IIE Transactions, 31*, 217–230.

Askin, R., & Standridge, C. (1993). *Modeling and analysis of manufacturing systems*. New York: John Wiley & Sons.

Ben-Daya, M., & Al-Fawzan, M. (1998). A tabu search approach for the flow shop scheduling problem. *European Journal of Operational Research, 109*, 88–95.

Brandt, R. D., Wang, Y., Laub, A. J., & Mitra, S. K. (1988). Alternative networks for solving the travelling salesman problem and the list-matching problem. *Proceedings of the International Conference on Neural Networks, 2*, 333–340.

Cakar, T., & Cil, I. (2004). Artificial neural networks for design of manufacturing systems and selection of priority rules. *International Journal of Computer Integrated Manufacturing, 17*(3), 195–211.

Carpenter, G. A., & Grossberg, S. (1987). ART: 2 self-organization of stable category recognition codes for analog input patterns. *Applied Optics, 26*, 4919–4930.

Cha, Y., & Jung, M. (2003). Satisfaction assessment of multi-objective schedules using neural fuzzy methodology. *International Journal of Production Research, 41*(8), 1831–1849.

Chen, M., & Dong, Y. (1999). Applications of neural networks to solving SMT scheduling problems – A case study. *International Journal of Production Research, 37*(17), 4007–4020.

Chen, R. M., & Huang, Y. M. (2001a). Multiprocessor task assignment with fuzzy Hopfield neural network clustering technique. *Neural Computing & Applications, 10*, 12–21.

Chen, R. M., & Huang, Y. M. (2001b). Competitive neural network to solve scheduling problems. *Neurocomputing, 37*, 177–196.

Chen, F. F., Huang, J., & Centeno, M. A. (1999). Intelligent scheduling and control of rail-guided vehicles and load/unload operations in a flexible manufacturing system. *Journal of Intelligent Manufacturing, 10*, 405–421.

Chen, W., & Muraki, M. (1997). An action strategy generation framework for an on-line scheduling and control system in batch processes with neural networks. *International Journal of Production Research, 35*(12), 3483–3507.

Cho, S. B., & Shimohara, K. (1998). Evolutionary learning of modular neural networks with genetic programming. *Applied Intelligence, 9*, 191–200.

Chryssolouris, G., Lee, M., & Domroese, M. (1991). The use of neural networks in determining operational policies for manufacturing systems. *Journal of Manufacturing Systems, 10*(2), 166–175.

Dagli, C., & Sittisathanchai, S. (1993). Genetic neuro-scheduler for job shop scheduling. *Computers and Industrial Engineering, 25*(1–4), 267–270.

Ding, F. Y., & Kittichartphayak, D. (1994). Heuristics for scheduling flexible flow lines. *Computers and Industrial Engineering, 26*, 27–34.

El-Bouri, A., Balakrishnan, S., & Popplewell, N. (2000). Sequencing jobs on a single machine: A neural network approach. *European Journal of Operational Research, 126*, 474–490.

Fang, L., & Li, T. (1990). Design of competition based neural networks for combinatorial optimization. *International Journal of Neural Systems, 1*(3), 221–235.

Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. New Jersey: Prentice-Hall.

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

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: John Wiley & Sons.

Fonseca, D. J., & Navaresse, D. (2002). Artificial neural networks for job shop simulation. *Advanced Engineering Informatics, 16*, 241–246.

Foo, Y. P. S., & Takefuji, Y. (1988a). Stochastic neural networks for solving job-shop scheduling: Part 1, problem presentation. In *Proceedings of joint international conference on neural networks* (Vol. 2, pp. 275–282).

Foo, Y. P. S., & Takefuji, Y. (1988b). Stochastic neural networks for solving job-shop scheduling: Part 2, architecture and simulations. In *Proceedings of joint international conference on neural networks* (Vol. 2, pp. 283–290).

Foo, Y. P. S., & Takefuji, Y. (1988c). Integer linear programming neural networks for job-shop scheduling. In *Proceedings of joint international conference on neural networks* (Vol. 2, pp. 341–348).

Foo, S. Y., Takefuji, Y., & Szu, H. (1995). Scaling properties of neural networks for job-shop scheduling. *Neurocomputing, 8*, 79–91.

Fukushima, K. (1975). Cognitrion: A self-organizing multilayered neural network. *Biological Cybernetics, 20*, 121–136.

Geneste, L., & Grabot, B. (1997). Implicit versus explicit knowledge representation in a job shop scheduling decision support system. *International Journal of Expert Systems, 10*(1), 37–52.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*, 533–549.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. MA: Addison-Wesley.

Grossberg, S. (1972). Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik, 10*, 49–57.

Grossberg, S. (1976a). Adaptive pattern classification and universal recording: I. Parallel development and coding of neural detectors. *Biological Cybernetics, 23*, 121–134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recording: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics, 23*, 187–202.

Hamad, A., Sanugi, B., & Salleh, S. (2003). A neural network model for the common due date job scheduling on unrelated parallel machines. *International Journal of Computer Mathematics, 80*(7), 845–851.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New Jersey: Prentice-Hall.

Hedge, S., Sweet, J., & Levy, W. (1988). Determination of parameters in a Hopfield/Tank computational network. In *Proceedings IEEE international conference on neural networks* (Vol. 2, pp. 291–298).

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in microstructure of cognition*. Cambridge: MIT Press.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. MI: The University of Michigan Press.

Hong, Y. S., Lee, H., & Tahk, M. J. (2003). Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks. *Engineering Optimization, 35*(1), 91–102.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Sciences, 79*, 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two state neurons. *Proceedings of National Academy of Sciences, 81*, 3088–3092.

Hopfield, J. J., & Tank, D. W. (1985). Neural computation of decision in optimization problems. *Biological Cybernetics, 52*, 141–152.

Huang, Y. M., & Chen, R. M. (1999). Scheduling multiprocessor job with resource and timing constraints using neural networks. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 29*(4), 490–502.

Jain, A. S., & Meeran, S. (1998). Job shop scheduling using neural networks. *International Journal of Production Research, 36*(5), 1249–1272.

Jeng, M. D., & Chang, C. Y. (1997). Non-energy based neural networks for job shop scheduling. *Electronics Letters, 33*(5), 399–400.

Jong, K. A. D. (1975). An analysis of the behavior of a class genetic adaptive systems. PhD thesis, Ann Arbor: University of Michigan.

Kamgar-Parsi, B., & Kamgar-Parsi, B. (1992). Dynamical stability and parameter selection in neural optimization. In *Proceedings of international joint conference on neural networks* (Vol. 4, pp. 566–571).

Kartam, N., & Tongthong, T. (1998). An artificial neural network for resource leveling problems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 12*, 273–287.

Kido, T., Takagi, K., & Nakanishi, M. (1994). Analysis and comparisons of genetic algorithm, simulated annealing, tabu search and evolutionary combination algorithm. *Informatica, 18*, 399–410.

Kim, C. O., Min, H. S., & Yih, Y. (1998). Integration of inductive learning and neural networks for multi-objective FMS scheduling. *International Journal of Production Research, 36*(9), 2497–2509.

Kim, S., Lee, Y. H., & Agnihotri, D. (1995). A hybrid approach to sequencing jobs using heuristic rules and neural network. *Production Planning and Control, 6*, 445–454.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59–69.

Lai, W. K., & Coghill, G. G. (1992). Genetic breeding of control parameters for the Hopfield/Tank Neural Net. In *Proceedings of the international joint conference on neural networks* (Vol. 4, pp. 618–623).

Lee, H. C., & Dagli, C. H. (1997). A parallel genetic neuro scheduler for job shop scheduling problems. *International Journal of Production Economics, 51*, 115–122.

Lee, I., & Shaw, M. J. (2000). A neural-net approach to real time flow-shop sequencing. *Computers and Industrial Engineering, 38*, 125–147.

Lee, Y. H., Bhaskaran, K., & Pinedo, M. (1997). A heuristic to minimize the total weighted tardiness with sequence dependent setups. *IIE Transactions, 29*, 45–52.

Li, Z. (1996). Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers. *IEEE Transactions on Neural Networks, 7*, 1507–1516.

Li, D. C., Chen, L. S., & Lin, Y. S. (2003). Using functional virtual population as assistance to learn scheduling knowledge in dynamic manufacturing environments. *International Journal of Production Research, 41*(17), 4011–4024.

Li, D. C., Wu, C., & Torng, K. Y. (1997). Using an unsupervised neural network and decision tree as knowledge acquisition tools for FMS scheduling. *International Journal of Systems Science, 28*(10), 977–985.

Li, H., Li, Z., Li, L. X., & Hu, B. (2000). A production rescheduling expert simulation system. *European Journal of Operational Research, 124*, 283–293.

Liansheng, G., Gang, S., & Shuchun, W. (2000). Intelligent scheduling model and algorithm for manufacturing. *Production Planning and Control, 11*(3), 234–243.

Liebowitz, J., Krishnamurthy, V., Rodens, I., Houston, C., Liebowitz, A., & Zeide, J. (1997). Intelligent scheduling with GUESS: Development and testing results. *Expert Systems, 14*, 119–128.

Liebowitz, J., Rodens, I., Zeide, J., & Suen, C. (2000). Developing a neural network approach for intelligent scheduling in GUESS. *Expert Systems, 17*(4), 185–190.

Lo, Z. P., & Bavarian, B. (1993). Multiple job scheduling with artificial neural networks. *Computers and Electrical Engineering, 19*(2), 87–101.

Looi, C. (1992). Neural network methods in combinatorial optimization. *Computers and Operations Research, 19*(3/4), 818–823.

Luh, P. B., Zhao, X., Wang, Y., & Thakur, L. S. (2000). Lagrangian relaxation neural networks for job shop scheduling. *IEEE Transactions on Robotics and Automation, 16*, 78–88.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115–133.

McMullen, P. R. (2001). A Kohonen self organizing map approach to addressing a multiple objective, mixed model JIT sequencing problem. *International Journal of Production Economics, 72*, 59–71.

Min, H. S., & Yih, Y. (2003). Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. *International Journal of Production Research, 41*(16), 3921–3941.

Min, H. S., Yih, Y., & Kim, C. O. (1998). A competitive neural network approach to multi-objective FMS scheduling. *International Journal of Production Research, 36*(7), 1749–1765.

Mühlenbein, H., Schomisch, M., & Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel Computing, 17*, 619–632.

Nawaz, M., Enscore, E., & Ham, I. (1983). A heuristic algorithm for the *n*-job, *m*-machine flowshop sequencing problem. *Omega, 11*, 91–95.

Osman, I. H. (2002). Preface, focused issue on applied meta-heuristics. *Computers and Industrial Engineering*, 205–207.

Palmes, P. P., Hayasaka, T., & Usui, S. (2003). Evolution and adaptation of neural networks. In *Proceedings of the international joint conference on neural networks* (Vol. 1, pp. 478–483).

Park, Y., Kim, S., & Lee, Y. H. (2000). Scheduling jobs on parallel machines applying neural network and heuristic rules. *Computers and Industrial Engineering, 38*, 189–202.

Parker, D. B. (1985). Learning logic: Casting the cortex of the human brain in silicon. Technical Report, TR-47. Center for Computational Research in Economics and Management Science, Cambridge, MA: MIT Press.

Peterson, C., & Anderson, J. R. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems, 1*, 995–1019.

Philipoom, P. R., Rees, L. R., & Wiegmann, L. (1994). Using neural networks to determine internally-set due date assignments for shop scheduling. *Decision Sciences, 25*(5/6), 825–851.

Potvin, J. Y., & Smith, K. A. (2003). Artificial neural networks for combinatorial optimization. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 429–455). Boston: Kluwer Academic Publishers.

Priore, P., Fuente, D., Pino, R., & Puente, J. (2003). Dynamic scheduling of flexible manufacturing systems using neural networks and inductive learning. *Integrated Manufacturing Systems, 14*(2), 160–168.

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

Rabelo, L., & Alptekin, S. (1990). Adaptive scheduling and control using artificial neural networks and expert systems for a hierarchical/distributed FMS architecture. In *Proceedings of the second international conference on computer integrated manufacturing* (pp. 538–545).

Rabelo, L., Yih, Y., Jones, A., & Tsai, J. S. (1993). Intelligent scheduling for flexible manufacturing systems. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 810–815).

Rodammer, F. A., & White, K. P. (1989). A recent survey of production scheduling. *IEEE Transactions on Systems, Man and Cybernetics, 18*, 841–851.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning international representations by error propogation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed: Explorations in the microstructure of cognition*. Cambridge: MIT Press.

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., & Gurgun, B. (1996). A neural network model for scheduling problems. *European Journal of Operational Research, 93*, 288–299.

Sabuncuoglu, I., & Touhami, S. (2002). Simulation metamodelling with neural networks: An experimental investigation. *International Journal of Production Research, 40*(11), 2483–2505.

Satake, T., Morikawa, K., & Nakamura, N. (1994). Neural network approach for minimizing the makespan of the general job-shop. *International Journal of Production Economics, 33*, 67–74.

Schaffer, J. D., Whitley, D., & Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: A survey of the state of the art. In *Proceedings of international workshop on combinations of genetic algorithms and neural networks* (pp. 1–37).

Schwefel, H. P. (1981). *Numerical optimization of computer models*. Chichester: John Wiley & Sons.

Schwefel, H. P. (1995). *Evolution and optimum seeking*. New York: John Wiley & Sons.

Sexton, R., Allidae, B., Dorsey, R., & Johnson, J. (1998). Global optimization for artificial neural networks: A tabu search application. *European Journal of Operational Research, 106*(2–3), 570–584.

Shiue, Y. R., & Su, C. T. (2002). Attribute selection for neural network based adaptive scheduling systems in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology, 20*, 532–544.

Shiue, Y. R., & Su, C. T. (2003). An enhanced knowledge representation for decision tree based learning adaptive scheduling. *International Journal of Computer Integrated Manufacturing, 16*(1), 48–60.

Shugang, L., Zhiming, W., & Xiaohong, P. (2005). Job shop scheduling in real-time cases. *International Journal of Advanced Manufacturing Technology, 26*(7–8), 870–875.

Sim, S. K., Yeo, K. T., & Lee, W. H. (1994). An expert neural network system for dynamic job shop scheduling. *International Journal of Production Research, 32*(8), 1759–1773.

Smith, K. (1999). Neural networks for combinatorial optimization: A review of more than a decade research. *Informs Journal on Computing, 11*, 15–34.

Solimanpur, M., Vrat, P., & Shankar, R. (2004). A neuro-tabu search heuristic for the flow shop scheduling problem. *Computers and Operations Research, 31*, 2151–2164.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research, 64*, 278–285.

Vaithyanathan, S., & Ignizio, J. P. (1992). A stochastic neural network for resource constrained scheduling. *Computers and Operations Research, 19*(3/4), 241–254.

Van Den Bout, D. E., & Miller, T. K. (1988). A traveling salesman objective function that works. In *Proceedings of IEEE international conference on neural networks* (Vol. 2, pp. 299–303).

Van Hulle, M. M. (1991). A goal programming network for mixed integer linear programming: A case study for the job shop scheduling problem. *International Journal of Neural Systems, 2*(3), 201–209.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik, 15*, 85–100.

Wang, H., Jacob, V., & Roland, E. (2003). Design of efficient hybrid neural networks for flexible flow shop scheduling. *Expert Systems, 20*(4), 208–231.

Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis. Cambridge, MA: Harvard University.

Willems, T. M., & Brandts, E. M. W. (1995). Implementing heuristics as an optimization criterion in neural networks for job-shop scheduling. *Journal of Intelligent Manufacturing, 6*, 377–387.

Willshaw, D. J., & von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London, B194*, 431–445.

Yang, S., & Wang, D. (2000). Constraint satisfaction adaptive neural network and heuristics combined approaches for generalized job shop scheduling. *IEEE Transactions on Neural Networks, 11*(2), 474–486.

Yang, S., & Wang, D. (2001). A new adaptive neural network and heuristics hybrid approach for job-shop scheduling. *Computers and Operations Research, 28*, 955–971.

Yao, X. (1991). Optimization by genetic annealing. In *Proceedings of second Australian conference on neural networks* (pp. 94–97).

Yao, X. (1997). Global optimization by evolutionary algorithms. In *Proceedings of the IEEE* (pp. 282–291).

Yao, X. (1999). Evolving artificial neural networks. In *Proceedings of the IEEE* (Vol. 87, pp. 1423–1445).

Yao, X., & Liu, Y. (1998). Towards designing artificial neural networks by evolution. *Applied Mathematics and Computation, 91*, 83–90.

Yu, H., & Liang, W. (2001). Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling. *Computers and Industrial Engineering, 39*, 337–356.

Zhou, D. N., Cherkassy, V., Baldwin, T. R., & Olson, D. E. (1991). A neural network approach to job-shop scheduling. *IEEE Transactions on Neural Networks, 2*, 175–179.