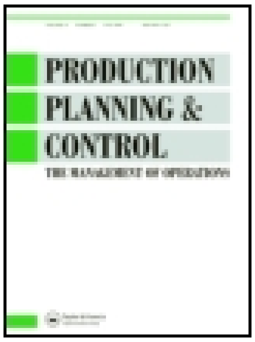


Production Planning & Control

ISSN: 0953-7287 (Print) 1366-5871 (Online) Journal homepage: https://www.tandfonline.com/loi/tppc20

Scheduling with neural networks: A review of the literature and new research directtions

Ihsan Sabuncuoglu

To cite this article: Ihsan Sabuncuoglu (1998) Scheduling with neural networks: A review of the literature and new research directtions, Production Planning & Control, 9:1, 2-12, DOI: 10.1080/095372898234460

To link to this article: https://doi.org/10.1080/095372898234460

Published online: 15 Nov 2010.

PRODUCTION PLANNING & CONTROL, 1998, VOL. 9, NO. 1, 2–12

Scheduling with neural networks: a review of the literature and new research directions

IHSAN SABUNCUOGLU

Keywords scheduling, neural networks

Abstract. Artificial neural networks (ANNs) attempt to emulate the massively parallel and distributed processing of the human brain. They are being examined for a variety of problems that have been difficult to solve with current serial type computing and processing; incuding a wide variety of combinatorial optimization problems. This paper deals with scheduling problems involving ANN applications. The objective is to review the entire literature of neural networks that are applied to various scheduling problems ranging from a single machine scheduling to satellite broadcasting scheduling. Both the theoretical developments and computational experiences are discussed. A classification framework is provided. Future research directions are also suggested.

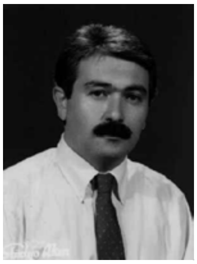
1. Introduction

Artificial Neural Networks (ANNs) are simplified mathematical models of theorized mind and brain. ANNs are referred to as artificial neural systems, connectionism, neural computation, parallel distributed processors. It is a computing architecture inspired from the structure and functions of the brain. Some of the characteristics associated with the brain are usually exhibited in these networks. These include the ability to learn from examples, generalize from situations and self organize the information.

ANNs applications can be found in a broad range of areas such as manufacturing (Rangwala and Dornfeld 1989), text to speech conversion (Sejnowski and Rosenberg 1987) and military systems (Hommertzheim *et al.* 1991). The ability to map and solve combinatorial optimization problems using neural networks has also motivated researchers since the beginning of ANN research. As a result, several neural networks have been developed for scheduling and other combinatorial optimization problems (Looi 1992, Masson and Wang 1990).

Over the last decade, the research interest in ANN has grown to such an extent that it is impossible to review the entire literature. There are already excellent survey papers on manufacturing applications (Udo and Gupta 1994), combinatorial optimization problems (Looi 1992), relationships of neural networks and operations research (Burke and Ignizo 1992), and neural network applications to production planning and control (Corsten and May 1996). In addition, Sharda (1994) has recently offered a comprehensive annotated bibliography of neural networks for MS/OR professionals. The papers cited above provide the reader with a general overview of neural networks and their applications.

The scope of this paper is rather narrower. We focus on scheduling problems and their solution methods using neural networks. Specifically, neural networks' models and their applications to scheduling problems are



Authors: Ihsan Sabuncuoglu, Dept of Industrial Engineering, Bilkent University, Ankara 06533, Turkey.

IHSAN SABUNCUOGLU is an Associate Professor of Industrial Engineering at Bilkent University. He received the BS and MS degrees in Industrial Engineering from the Middle East Technical University and the PhD degree in Industrial Engineering from the Wichita State University. Dr Sabuncuoglu teaches and conducts research in the areas of neural networks, simulation, scheduling and manufacturing systems. He has published papers in *IIE Transactions, International Journal of Production Research, International Journal of Flexible Manufacturing Systems, International Journal of Computer Integrated Manufacturing, Computers and Operational Research, Computers and Industrial Engineering, European Journal of Operational Research*, and *Journal of Operational Research Society*. He is on the Editorial Review Board of *International Journal of Operations and Quantitative Management*, and *Journal of Operations Management*. He is an associate member of Institute of Industrial Engineering and Institute for Operations Research and Management Science.

reviewed. To the best of our knowledge, this is the first survey paper that establishes a perspective on the status of neural network research relative to scheduling problems. Since it is also the first attempt towards a classification of the relevant literature, a wide variety of scheduling problems ranging from a single machine scheduling to more general problems (i.e. job shop scheduling and Hubble telescope scheduling) are considered in this paper. In that way, we hope that this wider perspective can provide important insights into applications of neural networks to scheduling and other optimization problems.

Although the aim is to review the recent research works as comprehensively as possible, our coverage is limited to the publications appearing in refereed journals and conference proceedings. A total of 59 papers are reviewed in this survey. These papers are included in the survey based on their appearances in the relevant literature. The strengths and weaknesses of each proposed model are summarized. The existing studies are also classified by identifying future research directions. Even though we attempt to bring the review more up-to-date, we may have left out some studies since the ANNs are still in the research stage and reports for their successful applications are forthcoming.

The rest of the paper is organized as follows. First, a historical development of neural networks and their applications to scheduling problems is presented. This is followed by a classification scheme and a survey of the ANN scheduling research. Finally, concluding remarks are made and future research directions are outlined.

2. Scheduling research and neural networks

Scheduling refers to the time-phased allocation of the system resources to various jobs, tasks or activities. It is one of the most frequently encountered problems in practice, ranging from day to day operation of manufacturing systems to scheduling of giant telescopes. Hence, the scheduling problems have received ample attention from both researchers and practitioners since the beginning of the 1950s.

Earlier research works used the optimization approach for solving scheduling problems. The optimization approach basically involves formulating a given problem as a mathematical model and solving it using an appropriate (exact or heuristic) algorithm. Over the years this approach has been applied to a wide range of scheduling problems. As a result, numerous publications have emerged in the literature (Baker 1974, French 1982). However, this well established and rich body of scheduling theory has scarcely been used in practice. This may partly be due to a lack of understanding of the theoretical methods by practitioners and difficulty in applying the oversimplified models to the practical scheduling problems. Essentially, this difficulty lies in the modelling of the problem, because real life scheduling problems may not be formulated such that all the relevant factors (i.e. dynamic and stochastic nature of the sytsems, qualitative and quantitative aspects of the problems) are incorporated and the model is easily solved. Indeed, most of the static and deterministic problems studied in the scheduling theory are NP-hard. During the 1980s, however, the trend towards more flexible and automated systems (e.g. FMS and CIM) increased awareness of the gap between scheduling research and practice, which in turn leads to more realistic formulations, theoretical analysis of heuristics and implementation of real-time scheduling systems (Morton and Pentico 1993).

In the last two decades, the scientific community has shown considerable interest in the Artificial Intelligence (AI) discipline. Although the origin of AI goes back to the beginning of OR, the number of AI based scheduling applications has only started to increase during the 1980s when AI emerged from research labs and was put into practice. This may be attributed to successful implementations of Expert Systems (ES). The basic motivation behind these applications is that each scheduling system is unique, and therefore a wide variety of technical expertise, system specific knowledge and human judgement need to be considered for solving the scheduling problems. This trend has also resulted in numerous publications in recent years (Fox and Smith 1984, Stephen 1986, Kusiak and Cheng 1988). As noticed by researchers Hendry (1987) and Phelps (1986), there are a number of similarities as well as differences between AI and OR approaches: OR techniques are suitable for solving well structured and quantitative problems, whereas AI and ESs are more suitable for solving ill-structured and qualitative problems. At present, however, it appears that more benefits can be obtained from a synthesis of OR and AI methods in dealing with complex scheduling problems.

During the late 1980s, we have also seen widespread interest in Artificial Neural Networks (ANNs). Especially, advances in parallel processing technology and developments of new neural network architectures have resulted in a number of ANN applications to scheduling and other combinatorial optimization problems. Historically, AI and ANN have a common origin and a common goal of simulating human intelligence. However, they emerged as two distinct schools of thought along the lines of continuous and symbolic systems. As stated by Zahedi (1991), AI treats the brain as a black box and imitates the human reasoning process by using deductive reasoning. In contrast, ANN considers the brain as a white box, and imitates its structure and function by using inductive reasoning. Regardless of their differences, both of them play an important role by providing intelligence needed towards implementation of the concept of the 'factory of the future' or 'unmanned factory'. The relationships between operations research and neural networks are also studied in the literature (Hommertzheim *et al.* 1989, Burke and Ignizo 1992). It appears that ANN provides significant support to OR researchers in dealing with difficult optimization problems, at the same time OR helps ANN researchers in the development and application of this new technology. Hence, there are a number of joint research possibilities that should be explord by both OR and ANN communities.

3. A classification framework

As indicated in the previous section, scheduling problems have received a lot of interest from ANN researchers. Hence, a number of neural networks have been proposed to solve a wide variety of scheduling problems. These studies can be summarized according to various classification factors such as processing complexity (single versus multiple processors) and scheduling criteria (makespan versus flow-time). In this paper, however, we use some of the neural network types discussed in Dayoff (1990) as the basis of our classification. These network types are as follows:

(1) Hopfield network and other optimizing networks,
(2) Competitive networks,
(3) Multi-layer perceptrons (back-propagation networks).

Neural network applications to scheduling problems for each category are given next.

3.1. *Hopfield and other optimizing networks*

Researchers have first attempted to solve scheduling problems using Hopfield type networks due to their successful applications to the Travelling Salesman Problem (TSP). Essentially, the Hopfield network (figure 1) is a recurrent network consisting of a single layer of processing elements (PEs) with real valued connection weights. All elements are simply both input and output units. They are non-adaptive networks because interconnection weights (corresponding to synaptic efficiency in a real neuron) are not determined as a result of a learning process, but rather derived initially from the energy function. This energy function encodes the problem data in terms of the objective function (i.e. soft constraint) and the hard constraints (i.e. constraints of the original problem). The Hopfield network, which is partly inspired by the real neurons, is an optimizer. Each PE has an activity value or state. The entire network also has a state at each moment with an associated energy value corresponding to the objective function to be minimized by the network. The aim is to obtain a stable state where the energy is at a minimum. Hopfield (1982) showed that his network eventually reached a stable state by properly selecting connection weights and updating the states of the units. In the later work, Hopfield and Tank (1985) applied this network to the TSP by mapping the problem into a two-dimensional neuron matrix. Hopfield networks have shown that combinatorial problems can be attacked by neural networks, which in turn led to the scheduling applications described below.

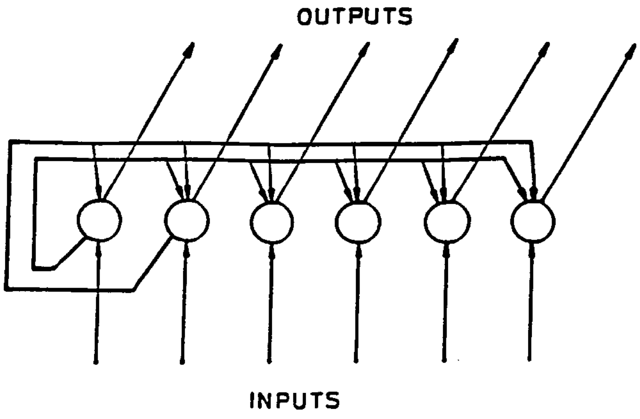


Figure 1. Hopfield network architecture.

Gulati and Iyengar (1987) developed a neural computing algorithm for the uniprocessor (or single machine) scheduling problem with hard deadlines and task priorities. The problem is mapped into a Hopfield Neural Network with $n\log(n_{\mathrm{p}})$ neurons, where n is the number of jobs and n_{p} indicates the total processing times of all jobs. In order to reduce complexity the time axis is scaled logarithmically. Their energy function consists of terms for tardiness, waiting time, overlapping and precedence relationships. The resulting neuron matrix gives the starting times of jobs in the sequence. During simulation experiments, a fast simulated annealing method was also used to obtain solutions for 20-job problems.

Arizona *et al.* (1992) presented another neural network application for a single machine scheduling problem with the total flow time criterion under the just-in-time production environment. Their network architecture was an extension of the Hopfield model, so-called Gaussian machine model (i.e. one of the stochastic neural networks). In general, Hopfield based networks cannot guarantee optimal solutions. Sometimes these networks produce local optimum solutions depending on their initial states. In this study, the authors tried to avoid the problem of being trapped in a local optimum by using a stochastic network. The performance of the network was measured using randomly generated test problems. In simulation experiments, 50 problems for 10 jobs were constructed and solved by this network. The authors reported 90% success in finding optimal solutions.

The previous two studies discussed above assume scheduling of a single machine (or processor). The problems associated with more complicated systems (i.e. job shop) are also addressed in ANN literature. These are as follows.

The work by Lo and Bavarian (1991) is an extension of the two-dimensional Hopfield structure to a three-dimensional neural matrix, called Neuro-box. According to their formulation, the z axis represents the jobs, the x axis represents the machines and the y axis represents time. The performance of this network is not generally known. It has only been applied to two small size job-shop scheduling problems (e.g. three machines and 10 jobs) with a tardiness related cost function. However, the authors reported successes in finding feasible schedules in all simulation runs. One other drawback of their network is that it is not suitable for real-world scheduling problems, because the size of the network and the number of connections increase enormously for large size problems.

Foo and Takefuji (1988a, 1988b) have also utilized the Hopfield network by mapping the $n|m$ job-shop scheduling problem on an mn by $(mn + 1)$ 2D neuron matrix similar to those for solving the TSP. In their formulation, the energy function is composed of hard constraints (i.e. precedence and resource constraints) and the cost of total completion times of all jobs. A computation circuit computes the total completion times (costs) of all jobs, and the cost difference is added to the energy function of the stochastic neural network. Using a simulated annealing (SA) algorithm, the temperature of the system is slowly decreased until the energy of the system is at a local or global minimum. At the final stage of convergence, the solution to the job-shop problem is represented by a set of cost function trees encoded in the matrix of stable states. The method has been used successfully to solve several 4|3 (4-job, 3-machine) job-shop scheduling problems. SA is a stochastic search method that allows degenerate solutions within a certain probability and reduces the possibility of becoming stuck in a local minimum. In general, SA emerged as an alternative search technique for finding solutions for large optimization problems. It has its origin in statistical physics where low energy states (perfect lattice structures) of metals and other substances are obtained through slow cooling called annealing. Kirkpatrick *et al.* (1983) were the first to demonstrate its applications for combinatorial optimization problems.

In their later work, Foo and Takefuji (1988c) represented the same problem as an integer linear program with the cost function of the total completion times of all jobs. The authors coded the integer linear program into the Hopfield network and solved the same size problem above. This network is called integer linear programming neural network (ILPNN). With this new formulation, the total number of neurons in the network was reduced to $nm(nm + 1)/2$. The results of experiments indicated that ILPNN shows considerable computational advantages over the Hopfield-based networks.

Zhou *et al.* (1990, 1991) further improved the performance of ILPNN for the job-shop scheduling problem. Specifically, the authors unified the indices indicating operations and machines to obtain simpler integer programming representation of the problem. In their formulation, the number neural processor equals the number of operations (subjobs). Hence, the number interconnection grows linearly with the total number of operations. To assess the performance of the network, the authors solved several job-shop problems ranging from 2|3 to 20|20 size problems. For small size problems (i.e. 2|3 or 5|3), the results of the network were compared with known optimal solutions. For large size problems (i.e. 10|10 and 20|20), total completion time of the longest job was used as an indicator of near optimality. Simulation results indicated that the network produces good schedules with a reasonable computational effort.

In a recent study, Foo *et al.* (1995) compared their modified Tank and Hopfield neural network with that of Zhou *et al.* (1991). The authors showed that even though their network has more neurons and interconnections, it has a very simple activation function and hence it does not require extensive calculations.

Thawonmas *et al.* (1993) proposed a real-time scheduler using neural networks for scheduling independent and non-preemptable tasks with deadlines and resource requirements. The proposed network is a modified model of the Hopfield–Tank neural network which integrates TSP and LP types of models. A heuristic procedure is also embodied into the energy function of the proposed model to cope with deadlines. Simulation experiments with this network indicated that the neural network performs better than a conventional algorithm based on the earliest due deadline policy.

In another study, Satake *et al.* (1994) proposed a neural network based on the Hopfield model. In contrast to other Hopfield based applications, this model changes the threshold value at each transition of neuron to generate a non-delay schedule for job shop systems. The authors also used the Boltzmann machine to escape a local minimum. The proposed model was applied to several makespan problems ranging from simple problems (2 jobs–2 machines) to more difficult ones (i.e. 10

by 10 problems). The results indicated that 0 out of 15 problems are solved optimally, and the remaining five problems are solved near-optimally with reasonable computing times.

Van Hulle *et al.* (1991a, 1991b) attempted to solve the same job-shop problem using a goal programming network. According to their approach, the original MIP problem is first reformulated as a goal programming problem and is then mapped onto an analogue ANN. Indeed, this study is the first attempt to map the MIP problem directly onto an analogue network. The solution is obtained by repeatedly performing goal programming relaxation and binary adjustments until convergence is achieved. Even though this method does not guarantee an optimal solution for the original problem, the solution is always feasible with respect to the constraints of the job-shop problem. The proposed network was tested on a number of test problems. Their results indicated that the network finds good solutions even if the problem size increases.

Willems and Brandts (1995) proposed another Hopfield based neural network for the job-shop problem. The proposed model is different from the previous models in the sense that there are built-in rule of thumb optimization criteria in its architecture. The resulting network is implemented in an algorithmic format that allows the minimization of an energy function. This network is designed in such a way that the processing of the network elements (activation of units) can be in a fixed or random order. The proposed network was compared with rules rather than other networks. The results on the small job-shop problems indicated that it performs better than the traditional heuristic procedures.

In another study, Vaithyanathan and Ignizo (1992) developed a stochastic neural network to solve resource constrained scheduling models. The resource constrained scheduling problem (i.e. assignments of tasks to resources) is first decomposed into a series of multi-dimensional knapsack problems. Then these small problems are used to establish equivalent neural network model representations. Their network consists of a combination of the Hopfield network plus external neurons that are used to provide stochasticity in the search process. A particularly unique feature of the method is the employment of transient feedback by which the solution is forced to be moved out of a local minimum to obtain an improved solution, if not the global optimal. Computational results obtained by the network have been very encouraging.

One of the most interesting applications of ANNs has been reported by Johnson and Adorf (1992) for scheduling astronomical observations of the Hubble Space Telescope (HST). In this study, the authors basically demonstrated how neural networks can be merged with other scientific techniques to solve such a difficult scheduling problem. According to their approach, the problem is first formulated as a 0–1 integer programming problem and then translated into a constraint graph, which forms the static structure of their neural network. This static network representation of the scheduling problem in the form of a constraint graph is equipped with a suitable stochastic network dynamics called the guarded discrete stochastic network (GDS) to obtain solutions for the problem. The GDS is different from both Hopfield dynamics and Boltzmann machine. It realizes a random automata network with a controlled stochastic neuron selection, but a deterministic neuron update rule. The resulting network is run on a Texas Instruments Explorer II workstation to generate six months of observation schedule. Computation time of the network is linear in the number of subintervals considered for activities to be scheduled. As indicated in the paper, the network is fast enough to permit the exploration of many alternative schedules before adapting one as a baseline. This system has been implemented in the workstation-based SPIKE scheduling system for long-range scheduling of HST observations.

Gislen *et al.* (1982, 1989) proposed a neural network for solving scheduling problems in the Swedish high school system (i.e. assignment of teachers to classes). Their network was an extension Potts neural network. However, it was improved by using a convenient encoding system and an efficient mean field algorithm. The authors tested the performance of the network using real problems from the education system. Computational experiments indicated that the network requires approximately 1 min of a CRAY XMP machine to solve the problem with 50 weekly hours, 60 classrooms, 45 classes and 90 teachers. A brief comparison with the linear programming approach is also given in this paper.

Finally, Liang and Hsu (1996) developed the Hopfield based neural network approach for short-term hydro scheduling. Their purpose was to determine the optimal amounts of generated powers for the hydro units in the system for 24 h of work in the future. The proposed approach is mainly a two-stage solution procedure. In the first stage, a Hopfield neural network is used to generate the schedule and later a heuristic rule based search algorithm is used to modify the schedule to satisfy all practical constraints. This method has been successfully implemented in scheduling the Taiwan power system.

3.2. *Competitive networks*

These networks differ from the Hopfield model with respect to inhibitory connections (figure 2). In a competitive network, the inhibitory links are established as a

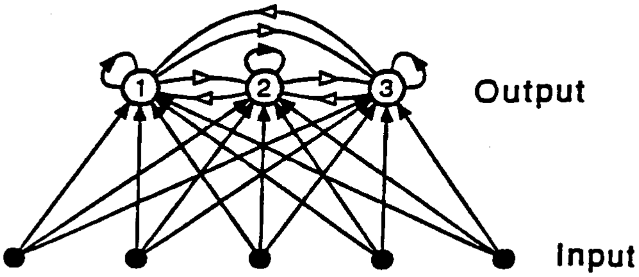


Figure 2. A simple competitive network. The connections with open arrows are inhibitory; the rest are excitatory (taken from Hertz *et al.* 1991).

result of competition rather than being determined initially as in the case of the Hopfield model. Neurons compete with each other to become active under certain conditions. Because of this competition property, their unsupervised learning paradigm is also called competitive learning. In designing such a network, one has to define equations of motion for the elements (i.e. constraints) of the problem and define an energy function to show convergence of the network. These equations are formulated as a local information transfer between the neurons and the network is simulated on digital computers. Then the energy function is constructed by studying convergence properties of the network.

As compared to other neural networks, the power of competitive networks has not been fully explored for scheduling problems. There are only a few reported applications in the literature. One such application is given by Fang and Li (1990) for the single machine total tardiness scheduling problem with unit processing times and different deadlines. In this study, the authors used an array of $n \times n$ neurons for this problem, where n represents the number of jobs. In their formulation, only one neuron was allowed to have the value 1 on each row and each column in a feasible solution. Two equations of motions were developed to check the row and column assignments. The convergence property was also given in the paper. The results of simulation experiments indicated good successes for problems with up to 20 jobs.

Sabuncuoglu and Gurgun (1996) proposed a parallized neural network for solving the single machine total tardiness problem and the minimum makespan job-shop scheduling problem. Even though this network looks like an extension of the Hopfield model, it is based on competition of a group of neurons as discussed in the previous network. The jobs compete with each other to be scheduled as early as possible. One noticeable difference between this network and others is that it has an additional external processor to control and monitor evolution of the network. With this processor, the network is forced to converge after a certain number of iterations. Their computational tests indicated that the network generates better solutions than one of the well-known heuristics for the single machine problem and finds the optimum solutions for most of the selected minimum makespan job-shop problems.

In another study, Pellerin and Herault (1994) developed a competitive neural network for a Timetable Construction (TC) problem. Their network is based on a Backward Lateral Inhabitation (BLI). The school timetable problem is a typical resource assignment problem. It is concerned with the assignment of courses and teachers to classrooms by avoiding the conflicts where courses taking place simultaneously involve common students, teachers or require the same classroom. This network was previously designed for signal and image processing. It is used in this study for its competition property that is established by using inhibition links in the network. Indeed, the inhibition property is well suited for coding of disjunctive constraints of the resource allocation problems. For example, if a resource is allocated to one user (active neuron), all other candidates should be inhibited (inactive neuron). The network with BLI has many similarities with the Hopfield model in terms of binary output and symmetrical connections. However, it is not completely interconnected and there is no learning algorithm. Also, iterations are carried out in a parallel way, rather than in a serial one. The network is represented by a two-dimensional table, column and rows corresponding to courses and classrooms, respectively. The proposed neural network is computationally very efficient. It takes about 1 sec on a personal computer to solve the problem with 800 courses, 50 classrooms, 20 student groups and 150 teachers.

Bourret *et al.* (1990) also developed a competitive neural network for scheduling satellite broadcasting times. The problem is to assign a set of antennas to the satellites for given time slices. To perform this assignment, the authors used a three-layered network with a competition mechanism in the middle layer. This central layer consists of R nodes, each of which represents the assignment of a particular satellite to an antenna during a specified time slice. Once the network is stabilized, the nodes of R layer at or near equity represent the solution set. Since the authors have applied their method to only one example problem, the performance of the network is not generally known.

4. Multi-layer perceptrons and back propagation networks

Multi-layer perceptron (MLP) is an important class of neural networks. Typically, the network consists of a set of processing elements, called perceptrons, that are organ-

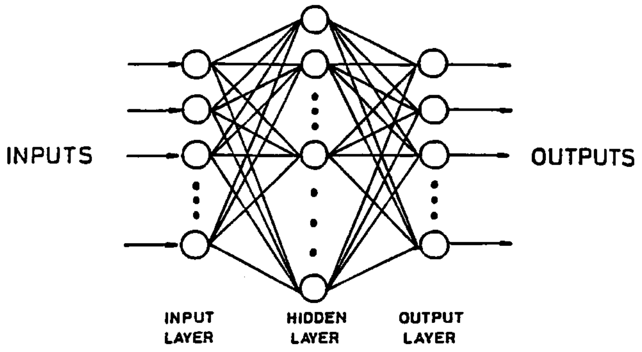


Figure 3. A back-propagation network.

ized in layers (i.e. input layer, hidden layer(s) and output layer). The input signal (or data) propagates through the network in a forward direction from one layer to another. The multi-layer network, which is an extension of a simple perceptron (single layer perceptron), has existed for a long time. However, it has been effectively used after the development of a particular learning algorithm, called the error back-propagation (BP) algorithm, or simply the back-propagation. Essentially, back-propagation is a systematic way of training a multi-layer artificial neural network (figure 3). The feed-forward network with this BP learning algorithm is called the back-propagation (BP) network.

From the current practice, the BP network seems to be one of the most widely used network architectures because of its ability to learn complex mappings and strong mathematical foundation. Hence, it has been applied to solve various problems (Dayhoff 1990). As compared to the Hopfield network, the BP paradigm has two phases: a training phase (feed-forward phase) and a recall (or use) phase. In the training phase, the network learns relationships between variables (or correlation between inputs and outputs) by means of examples. In the recall phase (generalization phase), it predicts outputs when exposed to unseen examples or new inputs. Even though the first phase of the BP paradigm consists of lengthy training sessions, the second phase (or test stage) is computationally very efficient. It usually takes a few seconds to get the results from a trained network. These characteristics of the BP network have been utilized in the following scheduling applications.

Sabuncuoglu and Hommertzheim (1992) have used BP networks to find the relationship between problem data (i.e. processing times, due-dates, etc.) and some of the properties of optimal solutions (or schedules). Their experiments with various single machine scheduling problems indicated that BP nets can learn to solve both the common due-date total tardiness problem and the flow time problem. In the case of *n* distinct due-dates that correspond to the original tardiness problem, the network occasionally produced sub-optimal solutions. During experiments, however, these optimally unsolved problems were added to the training set to enhance the performance of the network. It was also reported in this study that the performance of the network critically depends on the size of the problem and representatives of example problems in the training set. Especially, when the network was exposed to the larger size unseen problems, its performance deteriorated. This indicates that extrapolation capability of BP nets is not as good as their interpolation capability. A similar approach has been taken by Hayes and Sayegh (1982) who solved the two machine flow shop problem with the total flow time criterion.

In another study, Chryssolouris *et al.* (1991) used a BP network as a part of the operational model of a manufacturing system. Their network establishes adequate weights between the operational policy of a work centre and the overall performance measure of a manufacturing system. The results of simulation experiments indicated that the BP net is quite powerful when used to prescribe an operational policy for achieving a set of the goal performance measures (e.g. mean total costs, mean tardiness and mean flow-time).

Rabello and Alptekin (1989) developed an integrated (expert system and ANN) scheduling system in which a back-propagation network is used to rank a set of priority rules based on the current status of the system and job characteristics. In this system, outputs (relative weights) of the neural network are further analysed by an expert system to generate schedules for the mean tardiness criterion. In their simulation experiments, the network was able to predict the true results with 90% success. In a subsequent study, Rabello *et al.* (1993) proposed another hybrid system in which a BP network is used to select a subset of rules for the maximum tardiness and the mean flow time criteria. According to this approach, the schedules dictated by these rules are used as the elements of the initial population for a genetic search algorithm.

Kim and Lee (1993) combined a heuristic rule, the Apparent Tardiness Cost (ATC) rule, with an artificial neural network to solve single machine scheduling problems. In this hybrid approach, a back-propagation network is trained to determine proper values of the look-ahead parameter of the ATC rule. The results of computational tests based on 4500 randomly generated problems showed that this hybrid method improves due-date performance of the system significantly (e.g. 2–14% on average) as compared to the ATC rule with *a priori* fixed parameter.

In another study, Cho and Wysk (1994) developed an adaptive scheduling system for an intelligent workstation control. In this system, a back-propagation network is used to generate several part dispatching strategies based on workstation status and system characteristics (i.e. routing complexity, system congestion, job late factor, queue status, etc). This neural network was trained in the off-line mode by using example problems selected from the job-shop scheduling literature. Later, the network was implemented in the on-line mode for real-time scheduling decisions on the shop floor. Their extensive computational tests with this system indicated that BP is quite effective in identifying best dispatching strategies (i.e. scheduling rules). The authors also concluded that some characteristics of neural networks such as fast response time, potential answer to noisy (or incomplete) data make BP nets very suitable for real-time applications of manufacturing systems.

Similarly, Pierreval (1993) developed a BP network to select the most suited heuristic by considering the configuration of the shop floor, characteristics of the manufacturing program and the performance criteria to be optimized. The author tested the performance of the network using simplified flowshop problems. The results indicated that the BP network finds the best scheduling rules in 94% of the cases. In the paper, the author also discusses various implementation issues for back propagation networks.

Sim *et al.* (1994) have also used a BP network for a dynamic job shop scheduling problem. Their network was incorporated into an expert system which activates the BP network to recognize the individual contributions of the dispatching rules according to prevailing shop conditions. The proposed approach was compared with traditional dispatching rules. The results indicated that the expert neural network provides better results.

Yih *et al.* (1993) proposed a hybrid OR/ANN approach for a real-time robot scheduling problem. In this study, a well-trained BP network was used together with a semi-Markov model to determine the optimal policies. Their experiments showed that the hybrid approach is more effective in improving scheduling performance than any human scheduler, both in terms of quality and productivity.

Liang *et al.* (1992) developed an integrated system for automating real time scheduling. Their system consists of three components: First, computer simulation is used to collect expert decisions. Then the data are optimized using a semi-Markov decision model to remove data redundancies and errors. Finally, the optimized data are used to build an ANN-based expert system. The ANN under consideration was a simple back-propagation network with appropriate definition of parameters. The resulting system was applied to a circuit board manufacturing process with a robot serving five tanks. The proposed system was also evaluated by comparing it with the human expert and using ANN alone. The results indicated that the integrated approach performs better than both the human expert and the ANN-based system.

Finally, Philipoom *et al.* (1994) presented another neural network application to assign due dates for job-shop scheduling. The proposed back-propagation approach was compared with the regression based models. The results indicated that the neural network outperformed other non-linear models for mean-absolute deviation and standard deviation of lateness criteria. In their later study, Philipoom *et al.* (1997) showed that neural networks are in fact statistically better than the traditional approaches for a wide variety of cost functions.

In most of the studies summarized above, discrete event simulation models of dynamic job-shop were employed to generate the required data sets for BP networks. As can also be noticed from these studies, the BP networks were mainly used for selecting appropriate scheduling rules and finding weights of scheduling rules or decision making criteria rather than generating schedules directly by the network.

5. Conclusions and suggestions for further research

A number of observations can be made from this review. First, the existing studies have demonstrated that scheduling problems can be attacked by neural networks. Even though neural networks do not seem to be as effective as the conventional tools (i.e. optimization based algorithms), their inherent parallelism can offer benefits to scheduling research in the future (e.g. increased speed of calculations due to parallel processing).

Second, the majority of ANNs proposed for scheduling problems are based on the Hopfield network (table 1). This is probably due to the fact that the Hopfield models have mostly been applied to combinatorial optimization problems. Although these applications yielded promising results, they still inherit the problems of Hopfield networks in the TSP applications (i.e. feasibility problems and excessive computation times for large size problems). Hence, more research work is needed to improve both the energy function and ways of setting the values of some of the key parameters.

As compared to Hopfield networks, there has been little attempt to develop competition-based networks for scheduling problems. Only a few studies have been reported in the literature (table 1). However, in a typical scheduling problem the jobs compete with each other for the first available position in a sequence, indicating the importance of competition-based networks for scheduling applications. This needs to be further investigated.

Table 1. Classification on the basis of neural network types.

Network type	Publications
Hopfield and other optimizing networks	Gislen *et al.* (1982, 1989), Gulati and Iyengar (1987), Foo and Takefuji (1988a, 1988b, 1988c), Zhou *et al.* (1990), Lo and Bavarian (1991), Van Hulle (1991), Zhou *et al.* (1991), Arizono *et al.* (1992), Johnson and Adorf (1992), Vaithyanathan and Ignizo (1992), Thawonmas *et al.* (1993), Satake *et al.* (1994), Foo *et al.* (1995), Willems and Brandts (1995), Liang and Hsu (1996)
Competitive networks	Bourret *et al.* (1990), Fang and Li (1990), Pellerin and Herault (1994), Sabuncuoglu and Gurgun (1996)
Multi-layer perceptrons	Hayes and Sayegh (1982), Rabello and Alptekin (1989), Chryssolouris *et al.* (1991), Sabuncuoglu and Hommertzheim (1992), Kim and Lee (1992), Liang *et al.* (1992), Rabello *et al.* (1993), Pierreval (1993), Yih *et al.* (1993), Cho and Wsyk (1994), Philipoom *et al.* (1994, 1997), Sim *et al.* (1994)

Table 2. Classification on the basis of application area.

Application areas	Publications
Single machine scheduling	Gulati and Iyengar (1987), Fang and Li (1990), Arizono *et al.* (1992), Sabuncuoglu and Hommertzheim (1992), Kim and Lee (1993), Sabuncuoglu and Gurgun (1996)
Flow shop scheduling	Hayes and Sayegh (1982), Pellerin and Herault (1994)
Job-shop scheduling	Foo and Takefuji (1988a, 1988b, 1988c), Rabello and Alptekin (1989), Zhou *et al.* (1990, 1991), Chryssolouris *et al.* (1991), Lo and Bavarian (1991), Van Hulle (1991), Kim and Lee (1993), Rabello *et al.* (1993), Sim *et al.* (1994), Thawonmas *et al.* (1993), Cho and Wsyk (1994), Philipoom *et al.* (1994, 1997), Satake *et al.* (1994), Foo *et al.* (1995), Willems and Brandts (1995), Sabuncuoglu and Gurgun (1996)
Robot scheduling	Liang *et al.* (1992), Yih *et al.* (1993)
Resource constrained scheduling	Vaithyanathan and Ignizo (1992)
Hydro scheduling	Liang and Hsu (1996)
Hubble telescope scheduling	Johnson and Adorf (1992)
Satellite broadcasting scheduling	Bourret *et al.* (1990)
Timetable scheduling	Gislen *et al.* (1982, 1989), Pellerin and Herault (1994)

In contrast, back-propagation networks have received considerable interest from ANN research. Most of them are standard BP applications such as selecting appropriate scheduling rules or finding parameters of the rules, rather than generating a schedule with some defined start and completion times of operations. Nevertheless, the BP approach provides an increased speed for the selection process that may be needed in real-time scheduling environments. Another point that may require further research is to investigate the generalization property of BP networks for solving large size problems, having learned to solve small size problems.

Third, most of the neural networks developed for scheduling problems have been proposed for manufacturing systems. The majority of these applications are also centred around the job-shop scheduling problems (table 2). In these studies, hypothetical systems are usually considered. Moreover, simulated data are used in the experiments rather than actual data. Hence, there is a need for industrial applications to show successful implementations of ANNs.

In addition, there is no theoretical or experimental study which measures the relative performances, strengths, and weaknesses of these proposed neural networks. Hence, there is a need for further research to provide a fair comparison between these networks under various scheduling environments. From the existing studies, it seems that Hopfield and competitive neural networks are most suitable for generating schedules (i.e. finding an optimal schedule), whereas back-propagation networks are very effective in selecting scheduling rules and parametrization of the rules or scheduling policies. But these networks are not studied in terms of size and number of layers in the networks, and characteristics of scheduling problems (i.e. stochasticity, load variations—uniform versus bottleneck, size of the scheduling system, etc.). Sensitivity of the results of neural networks to noisy data for the scheduling problems is also worthwhile to investigate in future studies.

Moreover, most of these neural networks have been emulated on conventional computers using general purpose ANN packages or simulators (e.g. NeuralWare 1992). Hardware implementations should also be considered in future studies to improve the effectiveness of the ANN approach in practice.

Finally, as stated earlier, ANN and AI are very effective in dealing with ill-structured and qualitative problems in which judgement plays an important role in the outcome. Industrial scheduling problems have these characteristics in the sense that a human expert still develops schedules in practice. He/she solves the problem by using an appropriate algorithm and modifies it if necessary using the domain specific knowledge, his/her preference, commonsense and judgement. However, these capabilities were not well explored in the existing ANN applications. They have been applied mostly to well-structured and quantitative scheduling problems for which there exist numerous scheduling algorithms. Moreover, due to the lack of objective comparison between ANNs and OR-based scheduling algorithms, it is hard to judge their relative performances.

In light of the above observations, there is a need for further research to develop new neural network architectures, modify the existing ones, and integrate OR and ANN approaches in suitable hybrid architecture. The existing OR-based scheduling methods are purely algorithmic (i.e. procedural) and employ serial processing, whereas neural networks are essentially based on collective and parallel processing. Thus, overall processing capabilities of ANNs combined with efficiencies of traditional algorithmic approaches may result in better scheduling methods.

ANNs can also be integrated with the recently developed metastrategies, such as simulated annealing, tabu search, genetic algorithms, so that the resulting hybrid models can effectively search for the optimal schedule in the large solution space. In fact, all these combined strategies can be very useful for both engineers and scientists in pursuing solutions for various other combinatorial optimization problems.

References

Arizona, I., Yamamato, A., and Ohto, H., 1992, Scheduling for minimizing total actual flow time by neural networks. *International Journal of Production Research*, **30**(3), 503–511.

Baker, K. R., 1974, *Introduction to Sequencing and Scheduling* (New York: John Wiley).

Bourret, P., Remmy, F., and Goodall, S., 1990, A special purpose neural network for scheduling satellite broadcasting times. *Proceedings of the ICNN*, Vol. 3.

Burke, L. I., and Ignizo, J. P., 1992, Neural networks and operations research: an overview. *Computers and Operations Research*, **19**, 179–189.

Cho, H., and Wsky, A., 1994, A robust adaptive scheduler for an intelligent workstation controller. *International Journal of Production Research*, **31**(4), 771–789.

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

Corsten, H., and May, C., 1996, Artificial neural networks for supporting production planning and control. *Technovation*, **16**(2), 67–76.

Dayhoff, J. E., 1990, *Neural Network Architectures: An Introduction* (New York: Van Nostrand Reinhold).

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

Foo, Y. P. S., and Takefuji, Y., 1988c, Integer linear programming neural networks for job-shop scheduling. *Proceedings of the ICNN*, **2**, 341–348.

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

Foo, Y. P. S., and Takefuji, Y., 1988a, Stochastic neural networks for solving job-shop scheduling: Part 1. Problem presentation. *Proceedings of the ICNN*, **2**, 275–282.

Foo, Y. P. S., and Takefuji, Y., 1988b, Stochastic neural networks for solving job-shop scheduling: Part 2. Architecture and simulation. *Proceedings of the ICNN*, **2**, 283–290.

Fox, M. S., and Smith, S. F., 1984, ISI: A knowledge based system for factory scheduling. *Expert Systems Journal*, **1**(1), 25–49.

French, S., 1982, *Sequencing and Scheduling: An Introduction to Mathematics of the Job Shop* (New York: John Wiley).

Gislen, L., Peterson, C., and Solderberg, B., 1982, Complex scheduling with Potts neural networks. *Journal of Neural Computation*, **4**, 805–831.

Gislen, L., Peterson, C., and Soderberg, B., 1989, Teachers and classes with neural networks. *International Journal of Neural Systems*, **1**(2), 167–176.

Gulati, S., and Iyengar, S. S., 1987, Nonlinear networks for deterministic scheduling. *Proceedings of the ICNN*, **4**, 745–752.

Hayes, P. V., and Sayegh, S. I., 1982, A supervised neural network approach to optimization as applied to the *n*-job, *m*-machine job sequencing problem. *Proceedings of ANNIE'92*.

Hendry, L. C., 1987, The potential impact of artificial intelligence on the practice of O.R. *European Journal of Operational Research*, **28**, 218–225.

Hertz, J., Krogh, A., and Palmer, R. G., 1991, *Introduction to the Theory of Neural Computations* (Addison-Wesley).

Hommertzheim, D. L., Huffman, J., and Sabuncuoglu, I., 1991, Training an artificial neural network: The pure pursuit maneuver. *Computers and Operations Research*, **18**(4), 343–353.

Hommertzheim, D. L., Sabuncuoglu, I., and Huffman, J., 1989, Connections between operations research and neural networks, the paper is presented at *TIMS/ORSA Joint National Meeting*.

Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science*, 2554–2558.

Hopfield, J. J., and Tank, D. W., 1985, Neural computation of decisions in optimization problems. *Biological Cybernetics*, **52**, 141–152.

Johnson, M. D., and Adorf, H. M., 1992, Scheduling with neural networks—The case space Hubble telescope. *Computers and Operations Research*, **19**(3/4), 209-275.

Kim, S., and Lee, Y., 1993, Enhancement of a job sequencing rule using an artificial neural network. *2nd Industrial Engineering Research Conference Proceedings*, 842–846.

Kirkpatrick, S., Gelatt, C. D., and Vechhi, M. P., 1983, Optimization by simulating annealing. *Science*, **220**, 671–680.

Kusiak, A., and Cheng, M., 1988, Expert systems for planning and scheduling manufacturing systems. *European Journal of Operational Research*, **34**(3), 745–752.

Liang, R. H., and Hsu, Y. Y., 1996, Short-term hydro scheduling using Hopfield neural network. *IEE Proc.-Gener. Transm. Distrib.*, **143**(3), 269–275.

Liang, T. P., Moskowitz, H., and Yih, Y., 1992, Integrating neural networks and semi-Markov process for automated knowledge acquisition: An application to real time scheduling. *Decision Sciences*, **23**(3), 1297–1313.

Lo, Z. P., and Bavarian, B., 1991, Scheduling with neural networks for flexible manufacturing systems. *Proceedings of the IEEE International Conference on Robotics and Automation*, 818–823.

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

Masson, E., and Wang, Y., 1990, Introduction to computation and learning in artificial neural networks. *EuropeanJ ournal of Operational Research*, **47**, 1–28.

Morton, T. E., and Pentico, D. W., 1993, *Heuristic Scheduling Systems with Applications to Production and Project Management* (New York: John Wiley).

NeuralWare Reference Guide, 1992, NeuralWare Inc., Pittsburgh, PA 15276.

Pellerin, D., and Herault, J., 1994, Scheduling with neural networks: Application to time table construction. *Neurocomputing*, **6**, 419–442.

Phelps, R. I., 1986, Artificial intelligence—An overview of similarities with O.R. *Journal of Operational Research Society*, **37**, 657–668.

Philipoom, P., Wiegmann, and Rees, L. P., 1994, Using neural networks to determine internally-set due-date assignments for job-shop scheduling. *Decision Sciences*, **25**, 42,825–42,851.

Philipoom, P., Wiegmann, and Rees, L. P., 1997, Cost-based due-date assignment with the use of classical and neural-network approaches. *Naval Research Logistics*, **44**, 421–446.

Pierreval, H., 1993, Neural network to select dynamic scheduling heuristic. *Revue des Sytemes de Decision*, **2**(2), 173–190.

Rabello, L., and Alptekin, S., 1989, Synergy of neural networks and expert systems for FMS scheduling. In K. Stecke and R. Suri (eds) *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operation Research Models and Applications*, 361–366.

Rabello, L., Jones, A., and Tsai, J., 1993, using hybrid systems for FMS scheduling. *2nd Industrial Engineering Research Conference Proceedings*, 471–475.

Rangwala, S., and Dornfeld, D. A., 1989, Learning and optimization of machining operations using computing abilities of neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **19**(2), 299–314.

Sabuncuoglu, I., and Gurgun, B., 1996, A neural network model for scheduling problems. *European Journal of Operational Research*, **93**(2), 288–299.

Sabuncuoglu, I., and Hommertzheim, D. L., 1992, Artificial neural networks: Investigations and developments of neural networks for scheduling problems. Presented at 1992 *TIMS/ORSA Joint National Meeting, Orlando, USA*.

Satake, T., Katsumi, M., and Nakamura, N., 1994, Neural network approach for minimizing the makespan of the general job-shop. *International Journal of Production Economics*, **33**, 67–74.

Sejnowski, T. J., and Rosenberg, C. M., 1987, Parallel networks that learn to pronounce English text. *Complex Systems*, **1**, 145–168.

Sharda, R., 1994, Neural networks for the MS/OR analyst: An application bibliography. *Interfaces*, **24**, 116–130.

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

Stephen, M. S., 1986, A survey of artificial intelligence-based scheduling system. *Proceedings of Fall Industrial Engineering Conference*, Boston, 395–405.

Thawonmas, R., Shiratori, N., and Noguchi, S., 1993, A real-time scheduler using neural networks for independent and nonpreemptable tasks with deadlines and resource requirements. *IEICE Trans. Inf. and Syst.*, **E76**(8), 947–955.

Udo, G. J., and Gupta, Y. P., 1994, Applications of neural networks in manufacturing management systems. *Production Planning & Control*. **5**, 258–270.

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

Van Hulle, M. M., 1991a, Goal programming network for linear programming. *Biological Cybernetics*, **65**, 243–252.

Van Hulle, M. M., 1991b, 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.

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

Yih, Y., Liang, T., and Moskowitz, H., 1993, Robot scheduling in a circuit board production line: A hybrid ORR/ANN approach. *IIE Transactions*, **5**(2), 26–33.

Zahedi, F., 1991, An introduction to neural networks and comparison with artificial intelligence and expert systems. *Interfaces*, **21**(2), 25–38.

Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Hong, D. W., 1990, Scaling neural network for a job-shop scheduling. *Proceedings of the ICNN*, Vol. 3.

Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Olson, D. E., 1991, A neural network approach to job-shop scheduling. *IEEE Transactions on Neural Networks*, **2**(1), 175–179.