dr hab. inż. Tadeusz Witkowski
The Faculty of Production Engineering,
Warsaw University of Technology, Poland
Paweł Antczak
The Faculty of Production Engineering,
Warsaw University of Technology, Poland
mgr inż. Grzegorz Strojny
The Faculty of Production Engineering,
Warsaw University of Technology, Poland

USE CONSTRAINT SATISFACTION ADAPTIVE NEURAL NETWORK FOR JOB- SHOP SCHEDULING

The paper presents the application of the Constraints Adaptive Neural Network to job-shop scheduling problem. The main idea of the CSANN method has been described. Especially the capacity of the net for adaptation to constraints of specific problem has been presented. The computer experiment has been proceeded to find the Johnson criterion (the minimal total time of the performance of all operations). The criterion has mainly been found as a function of the number of iterations of the computing process. Achieved results have been compared with the genetic algorithm AGHAR worked out for the solving of such type of problems.

1. INTRODUCTION

In general, scheduling problems have received a lot of interests from artificial neural network (ANN) researchers. As a result a number of neural networks have been developed to solve a wide range of scheduling problems. The existing studies can be classified according to the following network structures (or types):

1. Hopfield model and other optimising networks
2. competitive networks
3. back propagation networks

The most of the existing studies are based on Hopfield network. Hopfield network is a single layered and fully interconnected neural network model. It is an optimiser in the sense that the states of the neurons are updated in a random and asynchronous manner to minimize the energy of the network. In this case both the objective function (i.e., soft constraints) and hard constraints (i.e., constraints of the original problem) are coded into a single energy function with appropriate connection weights.

For example the Travelling Salesman Problem is mapped to a two dimensional neuron matrix using N neurons, where N is a number of cities. Foo and Takefuji [6] have utilized the Hopfield approach to map the $n|m$ job shop scheduling problem to an mn by $(mn+1)$ 2D neuron matrix. In this 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.

In competitive networks, the inhibitory links are established as a result of competition rather than being determined initially as in the Hopfield case. In designing such a network, one usually develops equations of motion for the elements of the problem and defines an appropriate energy function to show the convergence of the network. There are not many reported applications of competitive networks

Back propagation networks has been used more frequently than competitive networks. They are used especially because of their generalization property. It allows to find the relationship between problem data and optimal schedules and to determine the proper value of a look ahead parameter of a job priority rule, and to establish adequate weights an operational policy at the network centre level and the overall performance measure of a manufacturing system. They have also been used together with OR and AI tools in an integrated manner for real time scheduling systems.

All these studies have shown that scheduling problems can be successfully attacked by neural networks. At present, the neural network approach may not seem to be as good as conventional algorithms in terms of the quality of solutions but their inherent parallelism (parallel processing) offers some advantages. New models and methods based on neural network still have been created. The comparison between them and conventional methods have shown that neural network approach seems to be very perspective and its efficiency still grows.

The Constraints Satisfaction Adaptive Neural Network is such next method. It has been proposed and presented in [27].

2. STATEMENT OF THE JOB SHOP SCHEDULING PROBLEM

Consider a set of jobs $Z = \{Z_i\}$, $i \in$ I, where $I = \{\ 1, 2, ..., n\ \}$ is an admissible set of details (nodes), $U = \{u_q\}$, $q \in\ 1, m$, is a set of executors (machines, worksites). Each job Z_i is a group of details Π_i of equal partial task p_i of a certain range of production Operations of the technological processing of the i-th detail are denoted by $\{O_{ij}\}_{j=\xi}^{H_i}$

Then for Z_i , we can write $Z_i = (\Pi_i\ \{O_{ij}\}_{j=\xi}^{H_i})$ where $O_{ij} = (G_{ij},\ t_{ij(N)})$ is the j-th operation of processing the i-th group of details; ξ_i is the number of operation of the technological process at which one should start the processing the i-th group of details; H_i is the number of the last operation for a given group; G_{ij} is a group of interchangeable devices that is assigned to the operation O_{ij}; O_{ijq} is a operation O_{ij} proceeded on the machine u_q. $\overline{G_{ij}}$ is the i-th technological route being considered as a sequence of the groups of devices; G is a set of all groups of devices arose in the matrix $\|\{Z_i\}\|$; $T_{ijq}(N)$ is an elementary working time (duration in minutes) of the operation O_{ijq} with one detail δ that depends on the number of machine N in the group (on the specified operations); T'_{ijq} is the duration of tuning before the operation O_{ijq}; $\mathrm{M} = \max_i \mathrm{H}_i$ is the number of "generalized" operations; N_{gr} is the number of all groups of machines. It is required to construct a quasioptimal plan-schedule H^* that is given in the form of a matrix $\{S_{ijq}, \mu_{ij}, P_{ijq}\}$, $i \in\ 1, n$; $j \in\ 1, M$ (where S_{ijq} and P_{ijq} are instants of the beginning and the termination of the operation O_{ij}; $\mu_{ij} \in G_{ij}$ is the number of a specific machine assigned to the operation O_{ij}) and in the form of a vector of permutations ($\sigma_1, \sigma_2, \dots, \sigma_n$), where $\sigma_i = (\delta_{1i}, \delta_{2i}, \dots, \delta_{in})$, each of which assigns the order of starting of groups on the generalized operations. A certain function $F(H)$ serves as a numerical test for estimating the plan H. A plan H^* is called quasioptimal if $F(H^*) \le F(H)$ for all or almost all $H \in \Psi$, where Ψ is the set of admissible real plans.

We use the Johnson criterion (minimal total time of performance of operations). The desired plan H^* (ordering of production operations) should satisfy certain constraints.

3. DESCRIPTION OF CONSTRAINTS ADAPTIVE NEURAL NETWORK METHOD (CSANN)

3.1. Neural Units

To solve the problem mentioned above it can be mapped to Constraints Satisfaction Adaptive Neural Network (CSANN).

The CSANN, as each neural network consists of many interconnected parallel processing elements called neural units (simply – units). An i-th unit consists a linear summator and nonlinear activation function which are serialized. The summator receives signals $A_j (j=1,\dots,n)$ from connected units and sums them weighted with corresponding connection weights W_{ij} together with a bias B_i. The output of the summator, denoted here as N_i is passed through an activation function f. The output of function f is an input of next unit or units. So a unit can be defined as follows:

$$N_i = \sum_{j=1}^{n} \left(W_{ij} \times A_j\right) + B_i \qquad (1)$$

$$A_i = f\left(N_i\right)$$

where W_{ij} is the connecting weight from unit j to unit i.

The CSANN contains three kinds of unit, based on the general neural unit. The first kind of unit are called ST – units, representing the starting times of all operations. Each ST-unit represents one operation of the job shop scheduling problem with its activation corresponding to the starting time of the particular operation.

The second kind of unit, SC –units represent whether the sequence constraints are violated.

The third kind of unit, RC-units represent whether the resource constraints are violated.

The input of ST-unit, e.g., ST_i is calculated by

$$N_{ST_i}(t) = \sum_j \left(W_{ij} \times A_{SC_j}(t)\right) + \sum_k \left(W_{ik} \times A_{RC_k}(t)\right) + A_{ST_i}(t-1) \qquad (2)$$

where the net input of the unit ST_i is the sum of three terms. The first term represents the weighted activations of SC-units connecting with unit ST_i which implements feedback adjustments because of sequence violations. The second term represents the weighted activations of RC-units connected with the unit ST_i implementing feedback adjustments because of resource violations. The third term represents the previous activation, with the weight being +1, of the unit ST_i itself.

The activation function of ST-units is defined as follows:

$$A_{ST_i}(t)=\begin{cases} r_i, & N_{ST_i}(t)<r_i \\ N_{ST_i}(t), & r_i \le N_{ST_i}(t) \quad \le d_i - T_{ST_i} \\ d_i - T_{ST_i}, & N_{ST_i} > d_i - T_{ST_i} \end{cases} \tag{3}$$

where r_i and d_i are the release date and the due date, respectively of job i to which the operation corresponding to unit ST_i belongs. T_{STi} is the processing time of the operation corresponding to unit ST_i.

The SC-units receive the incoming weighted activations from the connected ST-units, representing operations of the same job. The RC- units receive the incoming weight activations from the connected ST-units, representing operations sharing the same machine. The net input of SC-unit or a RC-unit has the form as follows.

$$N_{C_i}(t)=\sum_j \left(W_{ij}\times A_{ST_j}(t)\right)+B_{C_i} \tag{4}$$

where C_i represents a SC-unit SC_i or a RC-unit RC_i and B_{Ci} is the bias of SC_i or RC_i. The bias B_{Ci} is added to the incoming weighted activations of the connected ST-units and equals to the processing time of a relative operation.

The activation function of a SC-unit or RC-unit is defined as follows:

$$A_{ST_i}(t)=\begin{cases} 0, & N_{C_i}(t)\ge 0 \\ -N_{C_i}(t), & N_{C_i}(t)<0 \end{cases} \tag{5}$$

Zero activation of a SC-unit or a RC – unit means that the corresponding sequence constraints or resource constraint is satisfied and there are no feedback adjustment from this SC-unit or RC-unit to connected ST-unit.

3.2. Adaptive Connections Weights and Unit Biases

In the proposed CSANN, the connection weights and biases of neural units are adaptively valued according to the actual activations of ST-units whilst the network is running, together with the sequence and resource constraints of the specific problem.

All units of CSANN, including ST-units, SC-units, RC-units are connected according to the two kinds of sequence and resource constraint of a specific job-shop scheduling problem, resulting in two blocks: SC-block, (sequence constraint block) and RC-block (resource constraint block). The SC-block consists of ST-units and SC-units. The RC-block consists of ST-units and RC-units. Each unit of a SC-block contains two ST--units, responding to two operations of a job, and one SC-unit, representing whether the sequence constraint between these two operations is violated (see Fig. 1). Each RC-block contains two ST-units, responding to two operations sharing the same machine, and one RC-unit representing whether the resource constraint between these two operations is violated (see Fig. 2).

Fig. 1 and Fig. 2 show how the adaptive weights are valuated. Fig. 1 illustrates an example of a SC-block unit, denoted by SC_{ikl}, and Fig. 2 illustrates an example of a RC-block unit, denoted by RC_{qikjl}. In Fig.1 and 2, I_{STikp} is the initial value set for the ST-unit

ST_{ikp}, responding to the initial starting time $S_{ikp}(O)$ of the operation O_{ikp}. In Fig.1 the two ST-units ST_{ikp} and ST_{ilq} represents the two operations O_{ikp} and O_{ilq} of job i. Their activations A_{STikp} and A_{STilq} represents the starting times S_{ikp} and S_{ilq} of O_{ikp} and O_{ilq}, respectively. The SC-unit SC_{ikl} represents whether the sequence constraint between O_{ikp} and O_{ilq} is violated, with B_{SCikl} being its bias. Then at time t during the processing of network, the connection weights W_1, W_2, the feedback connection weights W_3, W_4 and the bias B_{SCikl} of SC_{ikl} are adaptively valued as shown below:

when $S_{ilq} - S_{ikp} \geq T_{ikp}$ then $W_1 = -1,\ W_2 = 1,\ W_3 = -W,\ W_4 = W,\ B_{SCikl} = -T_{ikp}$ (6)

when $(S_{ilq} - S_{ikp} \geq T_{ikp}$ or $S_{ikp} - S_{ilq} \geq T_{ilq})$ and $S_{ikp}(t) \leq S_{ilq}(t)$ then

$$W_1 = -1,\ W_2 = 1,\ W_3 = -W,\ W_4 = W,\ B_{SCikl} = -T_{ikp} \tag{7}$$

when $(S_{ilq} - S_{ikp} \geq T_{ikp}$ or $S_{ikp} - S_{ilq} \geq T_{ilq})$ and $S_{ikp}(t) \geq S_{ilq}(t)$ then

$$W_1 = 1,\ W_2 = -1,\ W_3 = W,\ W_4 = -W,\ B_{SCikl} = -T_{ilq} \tag{8}$$

where W is a positive changeable parameter (e.g 0,5) used for feedback adjustment

Similarly in Fig. 2, RC_{qikjl} represents the resource constraint between O_{ikq} and O_{jlq} on machine q. At time t during the processing of the network, the adaptive weights and bias are valued as shown below:

when O_{ikq} and O_{jlq} have to occupied the same machine and $S_{ikq}(t) \leq S_{jlq}(t)$ then

$$W_5 = -1,\ W_6 = 1,\ W_7 = -W,\ W_8 = W,\ B_{RCqikjl} = -T_{ikq} \tag{9}$$

when O_{ikq} and O_{jlq} have to occupied the same machine and $S_{ikq}(t) \geq S_{jlq}(t)$ then

$$W_5 = 1,\ W_6 = -1,\ W_7 = W,\ W_8 = -W,\ B_{RCqikjl} = -T_{ilq} \tag{10}$$

To recapitulate, the architecture of the network used for CSANN method consists of two layers. The bottom layer consists only of ST-units, corresponding to the starting times of all operations. The top layer contains both SC-units and RC-units, which represents sequence and resource constraints, and provide feedback information to adjust ST-units for sequence and resource constraints satisfaction through SC-block and RC-block respectively.

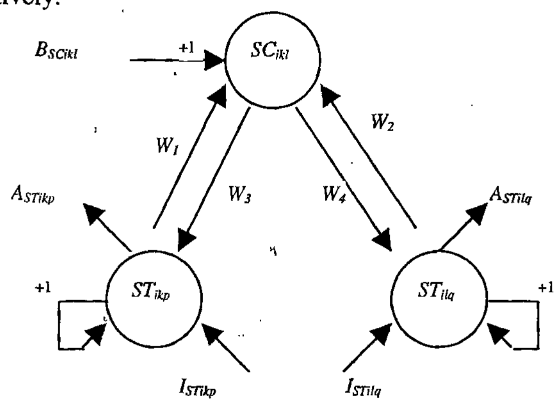

Fig. 1. SC-block unit representing sequence constraints (ref. [27])

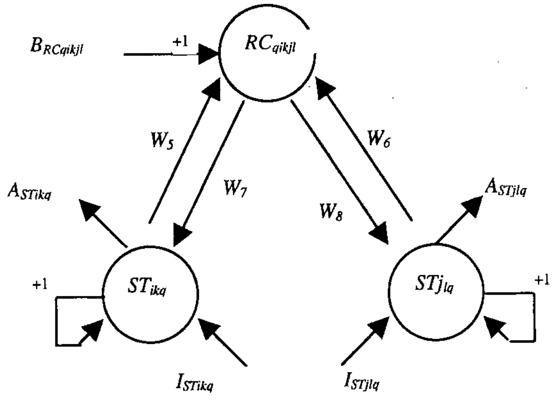

Fig. 2. RC-block unit representing resource constraints (ref. [27]).

4. RESULTS OF THE COMPUTER'S EXPERIMENT

The described method CSANN has been applied to the solving of the problem defined in Chapter 2. The computer's experiment has been proceeded and the structure of data described in [20] and [23] has been used. The Table 1 presents achieved results.

Results of the experiment.

The experiment has been proceeded for serial production case. Results of the experiment has been mainly compared with the genetic algorithm AGHAR [26]. The minimal Time for AGHAR has been 56293. The minimal time presented in Tab. 1 is 50242,02. So this result is 11% better than one achieved thanks to AGHAR.

Tab. 1

Problem	Number of operations	Number of iterations	TIME (Johnson criterion)
10/16/serial/*C min.*	160	5000	*50242,2*
10/16/ serial /*C min.*	160	1000	*50688,8*
10/16/ serial /*C min.*	160	500	*51105,1*
10/16/ serial /*C min.*	160	100	*52094*
10/16/ serial /*C min.*	160	50	*52230,4*
10/16/ serial /*C min.*	160	10	*53714,9*
10/16/ serial /*C min.*	160	5	*54275,6*
10/16/ serial /*C min.*	160	1	*58461,1*

REFERENCES

1. Baker K.. R, *Introduction to Sequence and Scheduling*. New York:Wiley, 1974.
2. Binato S., Hery W.J., Loewenstern D.M., Resende M.G.C., A greedy randomized adaptive search procedure for job shop scheduling, AT&T Labs Technical Report 2000, pp.1-19.
3. Conway R. W., Maxwell W. L, and Miller L. W., *Theory and Scheduling*. Reading, MA: Addison-Wesley, 1967.
4. Dubois D., Fargier H., and Prade H., "Fuzzy constraints in job-shop scheduling," *J. Intell. Manufacturing*, vol. 6, pp. 215-234, 1995.
5. Elzway S., Witkowski T., Methods construction of genetic algorithms to scheduling production, in: Modeling and optimization. Methods and application, Akademicka Oficyna Wydawnicza EXIT, Warsaw 2002,V33-V42.
6. Foo S. Y. and Takefuji Y., "Neural networks for solving job-shop scheduling: Part I. Problem representation," in *Proc. IEEE UCNH*, vol. II, San Diego, CA, 1988, pp. 275-282.
7. Hart E., Ross P., Nelson J., Producing robust schedules via an artificial immune system, IEEE World Congress on Computational Intelligence, International Conference on Evolutionary Computing, 1998.
8. Hentenryck P. V., *Constraint Satisfaction and Logic Programming*. Cambridge, MA: MIT Press, 1989.
9. Fox M. S. and Zweben M., *Knowledge-Based Scheduling*. San Manteo, CA: Morgan Kauftnann, 1993.
10. Hopfield J. J. and. Tank D. W, "'Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141-152, 1985.
11. Janiak A., Choose problems and algorithms of ranking of assignments and alocation of supplies, Akademicka Oficyna Wydawnicza PLJ, Warsaw 1999.
12. Kennedy M.P. and Chua L.O., "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 554-562, 1988.
13. Mańdziuk J. *Sieci Neuronowe typu Hopfielda. Teoria i przykłady zastosowań.* Akademicka Oficyna Wydawnicza EXIT, Warszawa 2000
14. Osowski S., *Sieci Neuronowe w ujęci algorytmicznym*, WNT, Warszawa 1996
15. Park J. H. and Jeong H., "Solving the TSP using an effective Hopfield network," in *Proc. IEEE Int. Conf. Neural Networks*, Paris, France, 1990.
16. Tadeusiewicz R. *Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami,* Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998
17. Tank D. W. and Hopfield J. J., "Simple neural optimization networks: An A/D converter, single decision circuit and a linear programming circuit," *IEEE Trans. Circuits Sysl.*, vol. 33, pp. 533-541, 1986.
18. Tagliarini G. A. and Page E. W., "Solving constraints satisfaction problems with neural networks," in *Proc. IEEE 1st IJCNN*, vol. III, 1987, pp. 741-747.
19. Wilson G. V. and Pawley G. S., "On the stability of the TSP algorithm of Hopfield and Tank," *Biol. Cybem.*, vol. 58, pp. 63-70, 1988.

20. Witkowski T., Scheduling tasks and scheduling production by of Szeza method optimization, Oficyna Wydawnicza Politechniki Warszawskiej, Warsaw 1998.
21. Witkowski T., Decisions in management enterprise, WNT, Warsaw 2000.
22. Witkowski T., Antczak A., Antczak P., Modele sztucznej inteligencji w systemach wspomagania decyzji przedsiębiorczych, Nowe możliwości i metody działań innowacyjnych małych i średnich przedsiębiorstw w układach wirtualnych, Wyd. WSE-I, Warsaw 2003, pp. 274-288.
23. Witkowski T., Elzway S., Antczak A., Projektowanie podstawowych operacji algorytmów genetycznych dla planowania produkcji, Probliemy uprawljenja i informatiki 6, 2003, pp. 129-147.
24. Witkowski T., Antczak A., Operation Research and Investigation of Variables and Parameters of the Genetic Algorithm for Optimization of Production Planning. Proceedings XXVI International Workshop Analysis on MDS: Analysis and Management of Ecological, Technogenic and Telecommunication Risk, Kiev 2003 pp. 69-77.
25. Witkowski T., Antczak P., Use of Artificial Neuron Networks for Task of Classification and Optimization of Production Planning, Proceedings XXVI International Workshop Analysis on MDS: Analysis and Management of Ecological, Technogenic and Telecommunication Risk, Kiev 2003, pp. 77-86.
26. Witkowski T., Antczak A., Witek T., The application of genetic algorithms and an immune systems approach for job-shop scheduling, Proceedings Conference Automation 2004, Warsaw 2004.
27. Yang S., Wang D., Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job-Shop Scheduling, IEEE Transactions Neural Networks, vol. 11, 2, 2000, pp. 474-486.
28. Żurada J., Barski M., Jędruch W., *Sztuczne sieci neuronowe. Podstawy teorii i zastosowania.* Wydawnictwo Naukowe PWN, Warszawa 1996.