IMPROVING HOPFIELD NEURAL NETWORK PERFORMANCE AND PARAMETERS INVESTIGATION

Warattapop Chainate[a], Peeraya Thapatsuwan[a], Somyot Kaitwanidvilai[b], Paisarn Muneesawang[b] and Pupong Pongcharoen[*a]

[a] Department of Industrial Engineering, [b] Department of Electrical and Computer Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand 65000.

ABSTRACT

In this work, an appropriate setting of the Hopfield Network (HN) parameter was investigated and applied to the classical traveling salesman problem. The investigation on the requirement of raw data normalization was also carried out. Moreover, a modified training algorithm by embedded a heuristic called elitism for improving the performance of the conventional HN was additionally proposed. Computer experiments were implemented using various problem sizes. The results obtained from the experiments indicated that the appropriate setting of HN parameter should be specified with a low value. It was also found that the usage of raw data or normalized data did not influence on the performance of HN. Another experimental result suggested that the proposed hybrid HN did not only outperform conventional HN in terms of the quality of the results, but execution time was also faster.

KEYWORDS: Neural network, Meta-heuristics, Artificial intelligence, Traveling salesman, Combinatorial optimization.

1. INTRODUCTION

The traveling salesman problem (TSP) was proposed in Vienna in 1920s by the mathematician and economist Karl Menger [1]. A general concept of the TSP is the finding of the shortest tour through N cities. A salesman must visit only one time for each city without any revisiting except in returning to the started city. The TSP is a classic combinatorial optimization problem and it also is classified as NP-Hard problem [2] [3]. After that, the problem reappeared in mathematical circles of Princeton by Whitney and Flood. Ever since, the TSP has become one of the most favorite problems for researchers around the world. However, the TSP is applied to find the number of different tours possible over a specified configuration [4]. Many countries such as China or USA have used as studying cases of the TSP. Various methods such as Genetic Algorithms [5], Ant colonies [6] or Neural Networks [7] [8] [9] has been applied to solve the TSP.

Neural Networks (NN) was developed in 1943 by Warren S. McCulloch [10] and his student, Walter H. Pitts, whose suggested that computations can be performed by a network of simple binary neurons. NN are very powerful in scientific and engineering applications for prediction, classification and pattern recognition. But they have not been as successful when applied to optimization problems such as scheduling problem, container loading problem or traveling salesman problem [11]. The idea of using the NN to provide solutions to difficult optimization problems [12] [13] originated in 1985 when Hopfield and Tank developed the Hopfield Network (HN) aiming to solve the NP-Hard problem such as the TSP. Credited with bringing the NN from the dead [2], the HN are still not good enough to solve the optimization problems especially when compared with the performance obtained from using other types of the NN such as the Kohonen Network [14], the Boltzmann Machines [15], or the Elastic Net Method [16] [17].

The improvement of the HN performance by either applying a heuristic or investigation of the HN parameter setting is a challenge for researchers but studies are limited. Hedge et al. [18] studied the interaction between the parameters of the HN for the TSP and found that when the problem size was

* Corresponding author. Tel: + 66 55 261000 ext 4201 Fax: + 66 55 261062
E-mail: Pupongp@nu.ac.th and Pupongp@yahoo.com

increased, the appropriate parameters that provided the valid solution became narrower. Kamgar-Parsi and Kamgar-Parsi [19] have proposed a systematic method for selecting these parameters based on analyzing the dynamic stability of valid solutions. Talaván and Yáñez [9] introduced the parameter setting procedure for the continuous HN to solve 10 cities TSP which was adopted from Hopfield and Tank [20]. The equations were proposed the setting of the HN parameters that depend on the free parameter C and also recommended the setting with small value of C. In their paper, the range of parameter C from 10 to 0.0001 was considered and the appropriate setting of parameter C is 0.001. Wang et al. [21] proposed a learning method in the HN for solving combinatorial optimization problems such as the TSP. Their results showed that the validation and the performance of solutions can be improved by using the proposed learning method.

Two types of data (raw and reduced) may be used in the HN for solving the TSP [4] [8] [22]. Raw data is the usage of the real distance between cities, although raw data may also be reduced to normalized data [4]. However, the essential of choosing data types has not been practically suggested by previous research.

The prime objectives of this paper were to investigate an appropriate setting of HN parameter to solve the Thailand Traveling Salesman Problem (TTSP). In addition, the determination of the essential of using the normalized data (reduced data) instead of the real data was investigated; and to improve the performance of the Hopfield Network (HN) by embedding the Elitism heuristic for providing short-term memory and also keeping elite solutions.

The outline of this paper begins with an overview of the Neural Networks and the Hopfield Network. Modified Hopfield Network by embedding with a memory of elitism is proposed in section 3. Section 4 presents the experimental design and analysis followed by conclusions in section 5.

2. NEURAL NETWORKS AND HOPFIELD NETWORKS

This section briefly reviews a basic concept and mathematical model of the Neural Networks (NN) and the Hopfield Neural Network (HN).

2.1 General basic of the Neural Networks

Neural Networks (NN) are mathematic models that are inspired by the human brain. They usually consist of a number of nodes or neurons (see Figure 1).

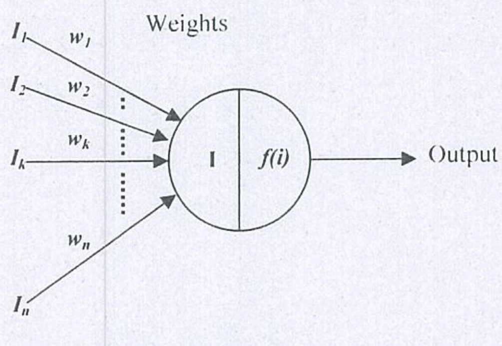

Figure 1.Simple model of neuron [3]

Each neuron is connected to another one via links with weights which represent the strengths of the connections. One neuron has one output and several inputs (could be its own output). When a neuron received input signals from another, a pre-processing operation (such as summation, product, minimize, etc.) is applied. The output of pre-processing operation is passed through a function called the activation function (such as linear function, sigmoid function, threshold function, etc.) to produce the final output of this neuron [3].

2.2 Hopfield Network for Traveling Salesman Problem

Considering the Hopfield Network (HN) for the Traveling Salesman Problem (TSP), a traveling tour is represented by a permutation matrix (see Figure 2). Given N cities, the tour can be represented by an N by N matrix with N^2 elements. Each row represents a city and each column represents the order of the cities to be visited.

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & a_{Xi} & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{matrix}$$

Figure 2. Permutation matrix of the HN for the TSP

Each neuron (a_{Xi}) has two states that $a_{Xi} = 0$ (OFF) and $a_{Xi} = 1$ (ON). $a_{Xi} = 1$ means that city X is the i^{th} city to be visited in the tour. The connection strength between two neurons is called weight $\left(W^{TSP}_{Xi,Yj}\right)$, means that the weight between neuron a_{Xi} and a_{Yj}.

$$W^{TSP}_{Xi,Yj} = -A\delta_{XY}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{XY}) - C - Dd_{XY}(\delta_{j,i+1} + \delta_{j,i-1}) \quad (1)$$

where A, B, C, D are constant parameters and $\delta_{ij} = \begin{cases} 1, & if\ i=j \\ 0, & otherwise \end{cases}$

Talaván and Yáñez [9] suggested the setting of the parameters that could solve any instance of the TSP as follows:

$$\begin{aligned} D &= \frac{1}{d_U} \\ B &= 3d_U + C \\ A &= B - Dd_L \\ N' &= N + \frac{3Dd_U}{C} \end{aligned} \quad (2)$$

Where, C is the only one free parameter to be specified with a constant value; d_U and d_L is respectively upper and lower bounds for the distance $d_{x,y}$; N is the number of cities; N' is developed to modify every threshold of the network. Talaván and Yáñez [9] also concluded that the small C gave a better solution for the TSP.

Hopfield Network has the energy function (E_{TSP}), which can be compared as a cost function in other optimizations methods [19].

$$E_{TSP} = \frac{A}{2}\sum_{X}\sum_{i}\sum_{j \neq i} a_{Xi}a_{Xj} + \frac{B}{2}\sum_{i}\sum_{X}\sum_{X \neq Y} a_{Xi}a_{Yi} + \frac{C}{2}\left(\sum_{X}\sum_{i} a_{Xi} - N\right)^2 + \frac{D}{2}\sum_{X}\sum_{Y \neq X}\sum_{i} d_{XY}a_{Xi}\left(a_{Y,i+1} + a_{Y,i-1}\right) \quad (3)$$

The first two terms are referred as "inhibitory connections" and the third term is "global inhibition". They will be activated (i.e., decreasing E_{TSP}) when the infraction of three constraints occurs. The last term in equation (3) is called "data term" [2].

Application of the HN to the TSP can be described by the following steps [2]:

1. Assign a positive value to parameter C and use equation (2) for setting all HN parameters.
2. Initialize the weight of all neurons with equation (1).
3. Random a subset of N neurons then assign value 1 to the selected N neurons.
4. Pick a neuron at random and calculate its output by:

Let U_{Xi} be the input of neuron a_{Xi}.

$$u_{Xi} = \left(\sum_{Y}\sum_{j}\left(W^{TSP}_{Xi,Yj} \times a_{Yj}\right)\right) + \left(C \times N'\right) \quad (4)$$

Then consider its output a_{Xi} with

$$a_{Xi} = \begin{cases} 1, & if\ u_{Xi} \geq 0 \\ 0, & otherwise \end{cases} \quad (5)$$

Repeat this step until all neurons are updated.

5. Calculate E_{TSP} using equation (3).
6. Repeat step 4 and 5 until the state of every neuron or E_{TSP} does not change.

3. MODIFIED HOPFIELD NETWORK

The concept of keeping the elitism used to improve the performance of Genetic Algorithms (GAs) by Murata and Ishibuchi [23] is to make sure that the best solution will survive into the next generation. In this work, the elitism heuristic was proposed to improve the training algorithm of the HN by using the short-term memory for keeping the first better solution. Subsequently, the initialization of the next run will random a set of neurons from the memory of elitism. The modified HN (mHN) can be divided into two parts: part of tuning (the memory) and part of running which are shown in Figure 3 and 4 respectively. Both part of tuning (step 1 to 7) and running (step 8 to 11) can be described as follows:

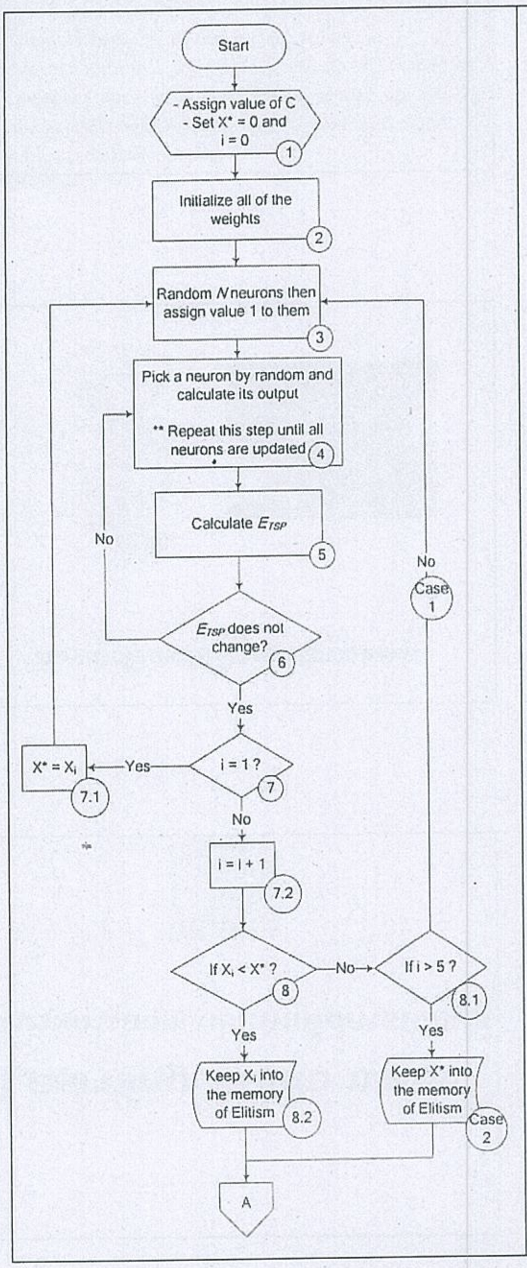

Figure 3. Part of memory tuning in the mHN

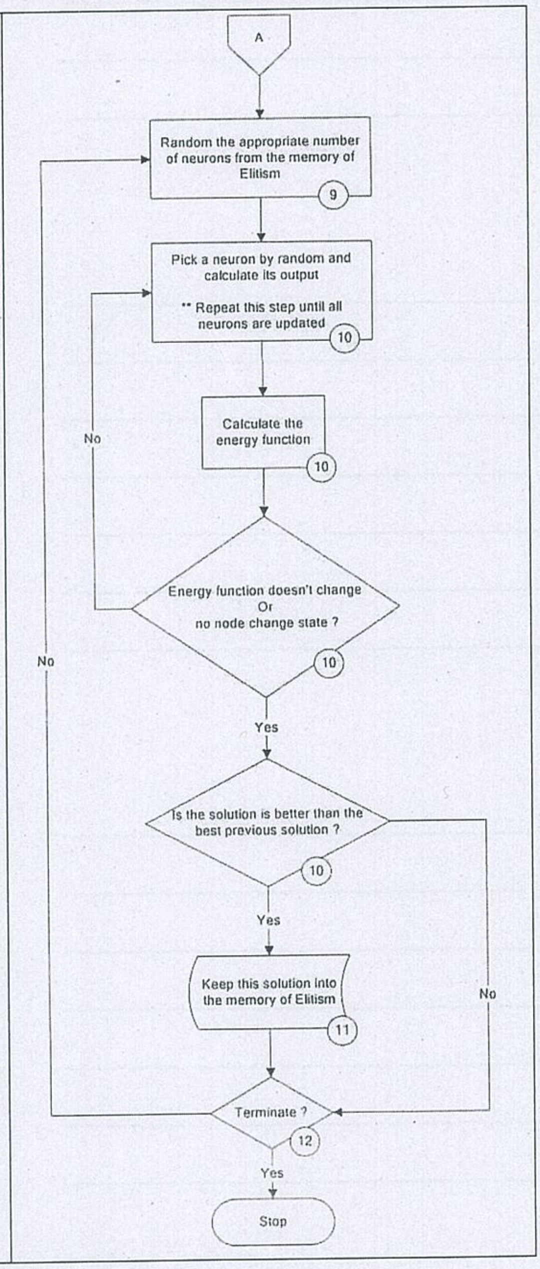

Figure 4. Part of running in the mHN

1. Assign a positive value to parameter C and then using equation (2) for determining others HN parameters. Set best solution (X^*) = 0 and number of feasible solution (i) = 0.

2. Initialize the weight of all neurons with equation (1).
3. Random a subset of N neurons then assign value 1 to the selected N neurons.
4. Pick a neuron at random and calculate its output by using equation (4) and (5)
5. Calculate the energy function (E_{TSP}) using equation (3).
6. Repeat step 4 and 5 until the state of every neuron or E_{TSP} does not change.
7. If it's the first solution ($i = 1$):
   7.1 Yes. update X_i and then keep it as the best solution ($X^* = X_i$). After that clear the state of all neurons to be 'OFF' (value '0') then go to step 3.
   7.2 No. increasing i ($i = i + 1$) then go to step 8.
8. Compare this solution (X_i) with the best solution (X^*)
   8.1 X^* is better than X_i: clear the state of all neurons to be 'OFF' (value '0'). This following procedure was proposed for eliminating a dead lock. Is the numbers of solution (i) > 5?
   Case (1) No. go to step 3
   Case (2) Yes. keep the best solution into the memory of Elitism.
   8.2 X_i is better than X^*: keep this solution (all neurons which have value is '1') into the memory of Elitism.
9. Random the appropriate numbers of neurons from the memory of Elitism then assign value '1' to the selected N neurons.
10. Update all neurons with the steps as the following steps: step 4. 5 and 6.
11. Compare this solution with the best solution (in the memory). If this solution is better then replace it into the memory. Clear the state of all neurons to be 'OFF' (value '0').
12. Repeat a part of running (step 9. 10 and 11) until terminating criteria is satisfied.

4. EXPERIMENTAL DESIGN AND ANALYSIS

Three sizes of TSP were investigated in the experiment. The first problem was considered as small and consists of 12 cities from the central region of Thailand. The medium size problem was considered in the northern and north-eastern regions of Thailand. which contains 36 cities. The last problem. which was relatively large. considered the traveling distance of all 76 cities of Thailand. The distance between cities was adopted from the Department of State Highway. Ministry of Transportation of Thailand.

Due to the objectives of this work. the experiments were divided into two parts: experiments A and B. Experiment A was aimed to determine the requirement of using normalization (reducing) of raw data and also to investigate the appropriate setting of parameter C. Experiment B was aimed to study the performance of the modified HN by comparing the results obtained from the proposed method with those obtained from conventional HN. A program was developed by using Microsoft visual basic 6.0 SP5. All of the experiments were simulated on personal computer with 1.4 GHz and 256 Mb of RAM.

4.1 Experiment A

In this experiment. the medium size (36 cities) of Thailand Traveling Salesman Problem (TTSP) was chosen to solve by the conventional HN. which aimed to determine the essential of using normalized data. This experiment was also aimed to investigate the appropriate setting of HN parameter (C). These two factors (types of data and value of parameter C) were therefore considered in this experiment. The values of these factors are shown in Table 1. The values of parameter C were adopted from Talaván and Yáñez [9].

Table 1. The factors of experiment A

Factors	Values
Type of data	Raw data. Norm data
Parameter C	3000. 2000. 1000. 500. 250. 100. 10. 1. 0.1. 0.01. 0.001

The experiment was carried out using full factorial experimental design [24]. The experimental program were executed 22 (2*11) runs per replication. The results obtained from 100 replications were analyzed and shown in Table 2.

Table 2. The results of the experiment A

C	HN with Raw data			HN with Norm data		
	Best (km.)	Average (km.)	Time (s)	Best (km.)	Average (km.)	Time (s)
3000	10017	13949.59	712	10661	15029.5	813
2000	9598	14006.75	706	10661	15029.5	813
1000	9901	13137.47	623	10661	15029.5	813
500	9501	12656.81	585	10661	15029.5	814
250	9501	12734.7	583	10661	14964.63	801
100	9501	12734.7	584	10017	13925.93	660
10	9501	12734.7	584	9501	12742.89	548
1	**9232**	**12545.98**	**593**	**9232**	**12579.98**	**548**
0.1	9232	12607.74	588	9232	12607.74	549
0.01	9232	12607.74	595	9232	12607.74	549
0.001	9232	12607.74	590	9232	12607.74	549

From Table 2, it can be seen that the best (shortest) and the average distance obtained from the conventional HN with raw data were better than the results using norm data except with large value of parameter C. This suggested that types of the data did not influence the performance of the conventional HN especially with the range of parameter C between 0.001-1 in which the best results were found. In this range, the runs with norm data were, however, taken less execution time than the runs with raw data. It should also be noted that the average shortest distance obtained using both type of data were found when parameter C was set to 1. This value is then used in the further experiment.

4.2 Experiment B

This experiment was aimed to study the performance of the modified HN (mHN) described in the previous section by comparing the results with those obtained by the conventional HN (cHN). The study was tested with three problem sizes. Types of the HN and problem sizes were therefore considered as factor in this experiment. The values of these factors are shown in Table 3.

Table 3. Factors of experiment B

Factors	Values
Structure	conventional HN (cHN), modified HN (mHN)
Problem size	Small, Medium, Large

Likewise the previous experiment, full factorial experimental design was used. The experimental program were therefore executed 6 (2*3) runs per one replication. The results obtained from 100 replications were analyzed and shown in Table 4.

Table 4. The results obtained from experiment B

Terms of results	Small-Size			Medium-Size			Large-Size		
	cHN	mHN	Improvement	cHN	mHN	Improvement	cHN	mHN	Improvement
Best solution (km.)	1020	753	26.18 %	9232	9172	0.65 %	27656	26017	5.93 %
Average solution (km.)	1167.28	1091.42	6.50 %	12545.98	10842.14	13.58 %	38432.07	29349.07	23.63 %
Execution time (s.)	14	14	0 %	593	479	19.22 %	8705	6155	29.29 %

From Table 4, it can be seen that the results obtained from the modified HN (mHN) were better than the results obtained from the conventional HN (cHN) for all problem sizes. The best solution was improved by 26.18% using proposed method to solve the small size problem whilst by 5.93% on the large size problem. The improvement on the average solution was increase when the problem size is enlarged. It was also found that the execution time of the runs that applied the mHN were less than those with the cHN especially when the problem size is increased.

An example run that used the mHN and the cHN to solve the large size problem was selected to demonstrate the progress of finding the best solution (shortest distance) during 100 runs as shown in Figure 5. It demonstrates the comparison of the (best so far) results which obtained from applying the cHN and the mHN to solve the large size problem. It can be seen that in the beginning of the series of runs, the mHN has a better start than the cHN. In the middle of the runs, the results obtained from the cHN were reduced quicker than another. However, the results obtained from the mHN were slowly decreased and achieved better result than the one with the cHN. This means that the elitism embedded within the mHN could play an important role for escaping the local optimums.

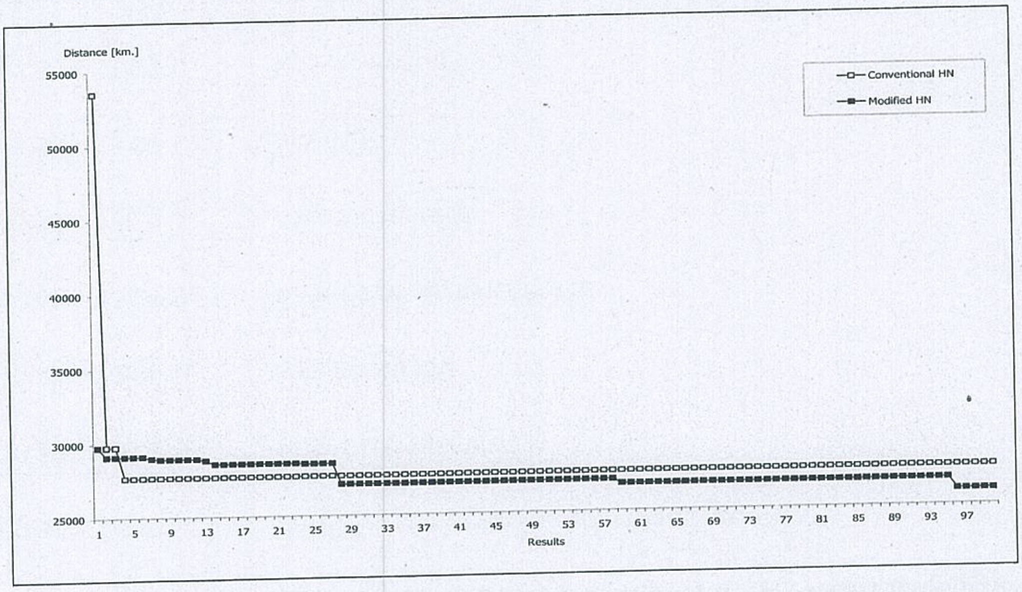

Figure 5. The results obtained from conventional and modified HN for the large size TTSP.

5. CONCLUSIONS

In this work, an appropriate setting of the Hopfield Network (HN) parameter, C, for the classical traveling salesman problem in the case of Thailand was investigated. Moreover, the investigation on the requirement of the reduced (normalized) data was also carried out. The experimental results suggested that the appropriate setting of HN parameter C that achieved the best solution with shortest distance was in the range of 0.001-1. However, the average distance from the experimental solutions suggested that the appropriate setting of C should be equal to 1. It was also found that the norm data has no influence on the quality of the results but the execution time of experimental runs using norm data were quicker than the runs with raw data.

A modified training algorithm with embedding a heuristic for improving the performance of HN was also proposed. Computer experiments were implemented using three problem sizes. The experimental results suggested that the proposed modified HN was not only outperforming the conventional HN in terms of the quality of the results, but the execution time was also faster. The elitism heuristics embedded within the modified HN played an important role for making sure that the best solutions found in the previous runs were carried on to the successive run. This mechanism helped the process of HN to escape the local optimum.

6. ACKNOWLEDGEMENTS

The work was supported by the Naresuan University Research Fund under the grant number NURF50-05-003.

REFERENCES

[1] Applegate, D., Bixby, R. and Cook, W. **1998**. On the solution of traveling salesman problems.

[2] Lu, Y. **1991**. Solving combinatorial optimization problems by simulated annealing, genetic algorithms, and neural networks. *Master thesis*, The University of Minnesota.

[3] Pham, D.T. and Karaboga, D. **2000**. *Intelligent optimization techniques*. London: Springer-Verlag.

[4] Flood, M.M. **1955**. The traveling salesman problem. *Operation Research*, *4*, 61-75.

[5] Yu, Y., Liu, Q. and Tan, L. **2001**. Solving TSP with distributed genetic algorithm and CORBA.

[6] Dorigo, M. and Gambardella, L.M. **1997**. Ant colonies for the traveling salesman problem. *Biosystems*, *43*(2), 73-81.

[7] Ghaziri, H. and Osman, I.H. **2003**. A neural network algorithm for the traveling salesman problem with backhauls. *Computers and Industrial Engineering*, *44*(2), 267-281.

[8] Modares, A., Somhom, S. and Enkawa, T. **1999**. A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research*, *6*(6), 591-606.

[9] Talaván, P.M. and Yáñez, J. **2002**. Parameter setting of the Hopfield network applied to TSP. *Neural Networks*, *15*(3), 363-373.

[10] McCulloch, W.S. and Pitts, W.H. **1943**. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

[11] Osman, I.H. and Laporte, G. **1996**. Metaheuristics: A bibliography. *Annals of Operations Research*, *63*, 513-623.

[12] Garey, M.R. and Johnson, D.S. **1979**. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: Freeman.

[13] Nemhauser, G.L. and Wolsey, L.A. **1988**. *Integer and Combinatorial Optimization*. John Wiley and Sons.

[14] Kohonen, T. **1982**. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59-69.

[15] Aarts, E.H.L. and Korst, J.H.M. **1989**. *Simulated Annealing and Boltzmann Machines*. John Wiley and Sons.

[16] Durbin, R. and Wilshaw, D. **1987**. An analogue approach to the traveling salesman problem using elastic net method. *Nature*, *326*, 689-691.

[17] Cochrane, E.M. and Beasley, J.E. **2003**. The co-adaptive neural network approach to the Euclidean Traveling Salesman Problem. *Neural Networks*, *16*(10), 1499-1525.

[18] Hedge, S., Sweet, J. and Levy, W. **1988**. Determination of Parameters in a Hopfield/Tank Computational Network. *Proceedings of 2nd IEEE International Conference on Neural Networks*.

[19] Kamgar-Parsi, B. and Kamgar-Parsi, B. **1992**. Dynamical stability and parameter selection in neural optimization. *Proceedings of 4th IEEE International Conference on Neural Networks*.

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

[21] Wang, R.L., Tang, Z. and Cao, Q.P. **2002**. A learning method in Hopfield neural network for combinatorial optimization problem. *Neurocomputing Letters*, *48*, 1021-1024.

[22] Dantzig, G., Fulkerson, R. and Johnson, S. **1954**. Solution of a large scale traveling salesman problem. *Operation Research*, *2*, 393-410.

[23] Murata, T., Ishibuchi, H. and Tanaka, H. **1996**. Multi-objective genetic algorithm and its applications to flow shop scheduling. *Computers and Industrial Engineering*, *30*(4), 957-968.

[24] Montgomery, D.C. **2001**. *Design and Analysis of Experiments*. 5th Edition, John Wiley and Sons.